Expert Systems With Applications 96 (2018) 358–372

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A new quantum chaotic cuckoo search algorithm for data clustering

Saida Ishak Boushaki[a,*], Nadjet Kamel[b], Omar Bendjeghaba[c]

[a] LRIA (USTHB) and University of Boumerdes, Boumerdes, 35000, Algeria
[b] Ferhat Abbas Setif University 1, Setif, 19000 and LRIA-USTHB, Algeria
[c] LREEI, University of Boumerdes, Boumerdes, 35000, Algeria

ARTICLE INFO

Article history:
Received 2 August 2017
Revised 14 November 2017
Accepted 4 December 2017
Available online 5 December 2017

Keywords:
Meta-heuristic
Optimization
Data clustering
Quantum cuckoo search
Chaotic map

ABSTRACT

This paper presents a new quantum chaotic cuckoo search algorithm (QCCS) for data clustering. Recent researches show the superiority of cuckoo search (CS) over traditional meta-heuristic algorithms for clustering problems. Unfortunately, all the cuckoos have identical search behaviours that may lead the algorithm to converge to local optima. Also, the convergence rate is sensitive to initial centroids seeds that are randomly generated.
Therefore, the main contribution of this paper is to extend the CS capabilities using nonhomogeneous update inspired by the quantum theory in order to tackle the cuckoo search clustering problem in terms of global search ability. Also, the randomness at the beginning step is replaced by the chaotic map in order to make the search procedure more efficient and improve the convergence speed. In addition, an effective strategy is developed to well manage the boundaries.
The experimental results on six famous real-life datasets show the significant superiority of the proposed QCCS over eight recent well known algorithms including, genetic quantum cuckoo search, hybrid cuckoo search and differential evolution, hybrid K-means and improved cuckoo search, standard cuckoo search, quantum particle swarm optimization, differential evolution, hybrid K-means chaotic particle swarm optimization and genetic algorithm for all benchmark datasets in terms of internal and external clustering quality.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

The clustering is a famous technique used in several real applications and research fields including, image segmentation (Lihua, Liguo & Junhua, 2016), pattern-analysis, bioinformatics (Hruschka, Campello, & de Castro, 2006; Johnson & Purdom, 2017), decision-making and document clustering (Ishak Boushaki, Kamel, & Bendjeghaba, 2014b, 2015). It is the process of grouping a set of data into groups that have some meaning in the context of a particular problem. Similar objects must be grouped in the same cluster and dissimilar objects must be in different clusters (Barbakh, Wu, & Fyfe, 2009; Dubes, 1993; Gan, Ma, & Wu, 2007; Jain, Murthy, & Flynn, 1999; Xu & Wunsch II, 2005).

There are two main classes of clustering techniques: hierarchical clustering and partitioning clustering. The time complexity of the hierarchical clustering is quadratic, whereas it is almost linear in the partitioning approaches. This is why the partitioning approaches are widely used rather than hierarchical ones (Barbakh et al., 2009; Dubes, 1993; Gan et al., 2007; Jain et al., 1999; Xu & Wunsch II, 2005). The well known K-means is a partitioning clustering algorithm known by its simplicity (Dalhatu & Tie Hiang Sim, 2016). However, it suffers from the problem of random initialization which leads sometimes to local solutions (Jain, 2010). Recently, in order to overcome this drawback, meta-heuristics based algorithms have demonstrated their efficiency for NP-hard problems like the clustering (Handl, Corne, & Knowles, 2015; Hruschka, Campello, Freitas, & de Carvalho, 2009; Rana, Jasola, & Kumar, 2011; Swagatam, Ajith, & Amit, 2009). These algorithms are able to reach optimal or near-optimal solutions to such problems in reasonable time. In this purpose, clustering can be considered as a category of optimization problem which maximizes or minimizes an objective function called fitness function. Many research groups proved that the meta-heuristic algorithms are able to find the global solution for the clustering problem instead of local one and they provide successful results in linear time complexity (Hruschka et al., 2009).

One among the recent meta-heuristics is the cuckoo search (CS) optimization algorithm. It is based on the interesting breeding behaviour such as brood parasitism of certain species of cuckoos and typical characteristics of Lévy flights (Xing & Gao, 2014; Yang, 2014; Yang & Deb, 2009, 2010, 2013). A conceptual comparison and

* Corresponding author.
E-mail addresses: saida_ib@univ-boumerdes.dz (S. Ishak Boushaki), nkamel@univ-setif.dz (N. Kamel), bendjeghaba@univ-boumerdes.dz (O. Bendjeghaba).

https://doi.org/10.1016/j.eswa.2017.12.001
0957-4174/© 2017 Elsevier Ltd. All rights reserved.

comparative analysis of the CS with three widely used optimization algorithms prove that the CS is statistically more successful than Differential Evolution (DE), Particle swarm optimization (PSO) and Artificial Bee Colony (ABC) (Civicioglu & Besdok, 2013, 2014). Furthermore, recent researches showed the ability of CS for solving the clustering problem and it can produce best results compared to other meta-heuristics (Ishak Boushaki, Kamel, & Bendjeghaba, 2014a). However, the main drawback of this method is that all the cuckoos have homogeneous search behaviours that lead to poor results by trapping in local optima. Also, the number of iterations increases to find an optimal solution; this is due to the random initialization of centroids at the initial step.

On the other hand, two theories have attracted the attention of the scientists and achieved large successful applications in different research fields during the last decade: quantum theory and chaos theory. The first one is a recent science developed from physics. It is a mathematical framework or set of rules for the construction of physical theories. Its main object is to study and describe the fundamental phenomena in physical systems, particularly at the atomic and subatomic particles (Adesso, Bromley, & Cianciaruso, 2016; Jaeger, 2007; Nielsen & Chuang, 2000; Schrödinger, 1926). Quantum theory provides new concepts to evolutionary computation. More recently, numerous studies inspired by quantum theory have been established, to enhance both efficiency and speed capabilities of typical optimization algorithms. They are grouped in two different classes. The first class is characterized by the incorporation of the principles of genetic quantum computing operators like qubit representation, superposition of states, interference, measure and mutation presented in (Han & Kim, 2000, 2002, 2004; Yang, Wang, & Jiao, 2004; Zhang, 2010) into the different meta-heuristic algorithms for the resolution of combinatorial optimization problems (Draa, Meshoul, Talbi, & Batouche, 2010; Layeb, 2010, 2011, 2013; Layeb & Saidouni, 2008; Salim, Layeb, & Kartous, 2014). While the second class is characterized by employing Schrödinger equation that describes the behaviour of quantum system in order to enhance the search procedure of original meta-heuristic algorithms including quantum cuckoo search (QCS) (Cheung, Ding, & Shen, 2017) and quantum particle swarm optimization (QPSO) (Sun et al., 2012).The experimental results of QCS are promising and show that QCS is superior to the other five variants of the CS algorithm. This is the main motivation of using QCS.

The second one is a branch of study in mathematics. It has succeeded to model the chaotic behaviour of stationary regimes derived from a large research area including physics, engineering, economics, biology, and philosophy. A chaotic map is an excellent solution to model the dynamic behaviour of different kinds of systems in particular systems that are highly depending on initial conditions like meta-heuristics. Where chaotic variables are used instead of random ones, because chaos has a great performance, including the ability of the ergodicity and non-repetition (Caponetto, Fortuna, Fazzino, & GabriellaXibilia, 2003; Gilmore & Lefranc, 2011; Ott, 2002; Schuster & Just, 2005). Recently, chaotic maps have been integrated with several meta-heuristic algorithms, such as genetic algorithm (GA) (Ebrahimzadeh & Jampour, 2013), differential evolution (DE) (Guo et al., 2006), firefly algorithm (FA) (Gandomi, Yang, Talatahari, & Alavi, 2013a) and particle swarm optimization (PSO) (Gandomi, Yun, Yang, & Talatahari, 2013b).

The successful application of these two famous theories motivated us a lot to take advantage of their abilities in the present work.

The main contribution of this paper is to extend the CS capabilities using chaotic map and nonhomogeneous quantum update in order to tackle the cuckoo search clustering problems in terms of random initialization and global search ability. In addition, an effective strategy is developed to well manage the boundaries. The experimental results are promising and confirm the efficiency of the proposed approach.

The remaining of this paper is organized as follows: In Section 2, related works are presented. The master background theories related to our proposed algorithm, including chaotic map, original cuckoo search combined with Lévy flight and quantum cuckoo search are presented in Section 3. In Section 4, the proposed algorithm is presented and its main steps are explained. Numerical experimentation and results are provided in Section 5. Finally, conclusion and future work are drawn in Section 6.

2. Related works

Many studies have been done on the use of meta-heuristic algorithms for the clustering problem. In this section, we bring brief literature overview of meta-heuristic based clustering algorithms limited to the techniques the most related to the proposed algorithm. Initially, genetic algorithms (GAs) were investigated to improve the performance of classic clustering algorithms. For instance, Maulik and Bandyopadhyay (2000) have proposed a genetic algorithm-based clustering technique, called GA-clustering. The superiority of the GA-clustering algorithm over the K-means algorithm is extensively demonstrated by datasets experiments.

Van der Merwe and Engelhrecht (2003) have proposed two algorithms for the clustering problem: a standard PSO and a hybrid approach where the individuals of the swarm are seeded by the result of the K-means algorithm. The two PSO approaches were compared against K-means clustering, which showed the superiority of PSO approaches.

Sun, Xu, and Ye (2006) focused on exploring the applicability of the Quantum-behaved Particle Swarm Optimization to data clustering. The experiment results testified two conclusions. The first one is that QPSO has overall better performance than K-means and PSO clustering algorithms. The second one is that hybridization of QPSO and K-means approaches could improve QPSO clustering algorithm considerably. In the same year, Paterlini and Krink (2006) reported results of performance comparison between a GA, PSO and DE for a clustering problem. The results show that DE is consistently superior.

Later, Karaboga and Ozturk (2011) have used the Artificial Bee Colony (ABC) algorithm for data clustering on benchmark problems. The performance of ABC algorithm was compared to PSO algorithm and other classification techniques from the literature. The simulation results have indicated that ABC algorithm can efficiently be used for multivariate data clustering. In the same year, Moh'd Alia, Azmi Al-Betar, Mandava, and Tajudin Khader (2011) have presented a hybrid algorithm based on Harmony Search optimization (HS) and c-means to overcome cluster centers initialization problem in clustering algorithms. The experiments show that an HS is a good solution for this problem. Kwedlo (2011) has presented a new clustering method, called DE-KM, which combines differential evolution algorithm (DE) with the K-means. The experimental results show that the performance of the DE and K-means clustering method is better than the performance of differential evolution, K-means and genetic K-means algorithm.

Also in the same year, Chuang, Hsiao, and Yang (2011) proposed a hybrid algorithm based on chaotic PSO and K-means adapted for the clustering application. Results indicated robust performance of the proposed approach compared against six PSO, chaotic and K-means derived algorithms.

Hatamlou (2013) has proposed a new clustering algorithm inspired by the black hole phenomenon. The experimental results show that the proposed black hole algorithm outperforms other traditional heuristic algorithms for several benchmark datasets.

In the next year, Ishak Boushaki et al. (2014a) have presented a new algorithm for data clustering based on the cuckoo search optimization. The performance of the proposed algorithm was assessed on four different datasets and compared with well known and recent algorithms: K-means, particle swarm optimization, gravitational search algorithm (GSA), the big bang–big crunch algorithm (BB-BC) and the black hole algorithm (BH). The experimental results proved the power of the new method to achieve the best values for almost all datasets.

Babrdel Bonab et al. (2015) have presented a new hybrid algorithm for data clustering, which is based on cluster center initialization algorithm (CCIA), bees algorithm, and differential evolution, known as CCIA-BADE-K. The evaluation results of the proposed algorithm and its comparison with other alternative algorithms in the literature confirm its superior performance. In the same year, Tvrdík and Krivy (2015) proposed an hybrid clustering algorithm by combining DE with K-means. The experimental results showed that hybrid variants with K-means algorithm are essentially more efficient than the non-hybrid ones. Also in the same year, Bouyer, Ghafarzadeh, and Tarkhaneh (2015) have proposed a hybrid algorithm for data clustering (HCSDE) based on DE and CS. The results have shown that the proposed algorithm outperforms the CS, DE, PSO, GSA, BB-BC and BH.

More recently, Pandey et al. (2016) have proposed a novel algorithm for data clustering based on K-means and improved cuckoo search where two parameters of CS are adjusted for fine-tuning of solution vectors and extend the capabilities of CS. The effectiveness of the proposed method is tested on the three microarray datasets and experimental results show that the proposed method outperforms the existing methods.

Niu, Duan, Liu, Tan, and Liu (2017) have presented population-based clustering technique based on PSO and Lloyd's K-means algorithm. Comparative experiments have shown that the proposed technique can obtain better and more stable solutions than five individual based counterparts in most cases.

For all standard well known meta-heuristic optimization algorithms, the number of iterations increases to find an optimal solution; this is due to the random initialization at the initial step. Furthermore, there are two main disadvantages associated with almost all these meta-heuristics. First, they involve numerous parameters that are difficult to tune (Personal Learning Coefficient, mutation probability, etc.). Second, due to the large search space, they often require a large number of iterations which renders them computationally prohibitive. Therefore, most of these approaches extend the capabilities of the original algorithm by hybridizing in most cases with K-means algorithm or with other alternative ones. However, hybrid algorithms have high complexity and need more computation. In contrast to promoting a certain meta-heuristic, other approaches employ new concepts inspired by recent theories like chaos and quantum theory to enhance its own performance in a simple way.

In this paper, the second solution is chosen to extend the capabilities of the standard CS, focused on quantum theory and the chaotic maps.

3. Chaotic map and quantum cuckoo search algorithm

In this section the master background related to chaotic map, original cuckoo search combined with Lévy flight and quantum cuckoo search are presented.

3.1. Chaotic map

Most of meta-heuristic algorithms depend hardly on initial seeds. So, if the initial solutions are not well selected, they lead to local minimum and poor results. However, using chaotic map at the beginning step instead of random ones enhance the stochastic searches.

A chaotic map is a mathematical function that exhibits some sort of chaotic behaviour over time. The different values of chaotic function constitute chaotic sequence. In chaotic sequence, the chaotic value (c_{n+1}) at time $n+1$ depends only on the chaotic value at time n (c_n). There are different chaotic maps. In this study the famous logistic map (Hilborn, 2004) is chosen based on preliminary experiments. It is given by the following equation:

$$c_{n+1} = ac_n(1 - c_n), \quad a = 4 \tag{1}$$

We note that, in this experiments, $c_n \in (0, 1)$ and $a = 4$, in this experiments as suggested in most research works.

3.2. Cuckoo search and Lévy flight

Cuckoo search is one of the latest nature-inspired meta-heuristic algorithms, proposed by Yang and Deb (Yang & Deb, 2009, 2010, 2013). It is inspired from two special behaviors of birds: the parasitic behavior of cuckoo and the Lévy flight behavior of fruit. It is focused on three idealized rules:

1. Each cuckoo lays one egg at a time, and dumps its egg in randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generations;
3. The number of available host nests is fixed and the egg laid by a cuckoo is discovered by the host bird with a probability $pa \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest, and build a completely new nest. For simplicity, this last assumption can be simulated by the fraction (pa) of the population size of worse nests that are replaced by new random nests.

Cuckoo moves from the current nest to the new one using a random step length which is drawn from a Lévy distribution (Yang & Deb, 2009, 2013; Xing & Gao, 2014; Yang, 2014). A random walk is a Markov chain whose next location depends only on the current location and the transition probability. For the standard cuckoo search, the new nest (x_{i+1}) is determined in terms of the current nest (x_i) and global best nest (x_b) as follows:

$$x_{i+1} = x_i + \alpha * S(x_i - x_b) \tag{2}$$

Here $\alpha > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. S is a random walk based on Levy flight.

The step length S is calculated using Mantegna's algorithm as follows:

$$S = \mu / |\upsilon|^{1/\beta} \tag{3}$$

where, β is a fixed parameter in [1, 2] and considered to be 3/2 .μ and υ are drawn from a normal distribution as:

$$\mu \sim N\left(0, \delta_\mu^2\right), \qquad \nu \sim N\left(0, \delta_\nu^2\right) \tag{4}$$

Note that δ_μ and δ_ν present in Eq. (4) are the variance of distributions given by the following equation:

$$\delta_\mu = \left[\Gamma(1+\beta)\sin(\pi\beta/2)/\Gamma\left([(1+\beta)/2].\beta.\ 2^{(\beta-1)/2}\right)\right]^{1/\beta}, \quad \delta_\nu = 1 \tag{5}$$

where, Γ is the standard Gamma function.

The basic steps of the Cuckoo Search (CS) can be summarized by the pseudo code shown in Fig. 1.

```
Cuckoo Search via Lévy Flights
begin
-Objective function f(x), x = (x1, ..., xd)^T
-Generate initial population of n host nests xi (i = 1, 2, ..., n)
while (t <MaxGeneration) or (stop criterion)
   -Get a cuckoo randomly by Lévy flights
   -Evaluate its quality/fitness Fi
   -Choose a nest among n (say, j) randomly
  if (Fi > Fj),
    Replace j by the new solution;
  end
   -A fraction (pa) of worse nests are abandoned and new ones are built;
   -Keep the best solutions (or nests with quality solutions);
    -Rank the solutions and find the current global best;
end while
-Post process results and visualization;
End
```

Fig. 1. Pseudo code of the standard CS (Ishak Boushaki et al., 2014a).

3.3. Quantum cuckoo search

In standard cuckoo search (CS) all the cuckoos have identical search behaviours. So, all the individuals of the population have the same mechanism to calculate the new update. It is based on the global best solution and the current one. Therefore, the algorithm may trap in local optima leading to premature convergence.

However in quantum cuckoo search (QCS) nonhomogeneous update is performed by selecting randomly one update of three ones with the same probability of appearance. The first one is the update given in standard CS. While, the two other ones are inspired from the quantum theory.

The quantum theory has introduced new concepts to classical physics. But, the most fundamental equation of the quantum theory is the Schrödinger equation, developed in 1926 by the Austrian physicist Erwin Schrödinger. It has important applications in physics and other fields. The equation is a type of differential equation known as a wave-equation, which serves as a mathematical model of the movement of waves. Its ultimate correctness rests on its ability to explain and describe experimental results (Thornton And Rex, 2013).

The time-dependent Schrödinger wave equation for a particle determines how the wave functions evolve through time and space and it is illustrated by the following equation:

$$j\hbar\frac{\partial}{\partial t}\psi(r,t) = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(r,t)\right]\psi(r,t) \tag{6}$$

where, j is imaginary unit, $\hbar$ is the reduced Planck constant, $\psi(r, t)$ is the wave function of the quantum system at position r and time t, m is particle's mass, $V(r, t)$ is the potential and ∇^2 is the Laplacian operator.

In QCS, each cuckoo is treated as a particle moving in an N-dimensional Hilbert space with a given energy, and thus its state is characterized by a wave function which only depends on the position of the particle. Also, the delta potential energy of the particle well centered at p is assumed to be the Dirac delta function, given by the following formula:

$$V(r) = -\gamma\delta(r-p) = -\gamma\delta(X) \tag{7}$$

where, $\delta(X)$is the Dirac delta function and γ is intensity of Dirac delta function.

Then, the particle's state is subject to the following time independent stationary Schrodinger equation:

$$\frac{d^2\psi}{dX^2} + \frac{2m}{\hbar^2}[E+\gamma\delta(X)]\psi = 0 \tag{8}$$

where, E is the energy of the particle.

Consequently, from (7) and (8), the solution of the normalized wave function of the particle in the bound state with the bound condition: $\psi \to 0$, as $|X| \to +\infty$ is given by:

$$\psi(X) = \frac{1}{\sqrt{L}} exp^{-\frac{|X|}{L}} \tag{9}$$

where, $L = \frac{\hbar^2}{mX}$ and exp is the exponential function.

Knowing that $|\psi^2|$ represents the probability density function of X and it is calculated by the following formula:

$$|\psi(X)|^2 = \frac{1}{L}\exp\left(-\frac{2|X|}{L}\right) \tag{10}$$

Then, the corresponding probability distribution function is given by:

$$F(X) = 1 - \exp^{-\frac{2|X|}{L}} \tag{11}$$

Finally, the position of the particle is measured using Monte Carlo inverse transformation. The process of measuring the particle's position in quantum mechanics is achieved by collapsing the quantum state to the classical state. The solution of $|X|$ is given by:

$$|X| = \frac{L}{2}\ln\left(\frac{1}{\eta}\right) \tag{12}$$

where, η is a random number uniformly distributed on (0, 1) and ln is the natural logarithm function.

By putting

$$L = 2\delta|\bar{x} - x_i|\ln\left(\frac{1}{\eta}\right) \tag{13}$$

where, x_i is the current position and $\bar{x}$ is the mean position of the population of size n is calculated by:

$$\bar{x} = \sum_{j=1}^{n} x_j \tag{14}$$

Based on Eqs. (12) and (13), in QCS two new updates are defined in addition to the classical update illustrated in Eq. (2), with the same probability of appearance.

The first new update is given by the following equation:

$$x_{i+1} = \bar{x} + \delta*(\bar{x} - x_i)*\ln(1/r) \tag{15}$$

where, δ is a control parameter and r is a random number drawn from a uniform distribution on the interval [0, 1].

Given that $\delta * \ln(\frac{1}{r}) \to +\infty$ if $r \to 0$, in order to guarantee that there is no large fluctuation around the global best nest (x_b), the second new update is given by the following formula:

$$x_{i+1} = x_i + \delta*(x_b - x_i)*\exp(r) \tag{16}$$

So, in QCS nonhomogeneous update is performed based on Eqs. (2), (15) and (16). In Section 4.2 we will present how these updates are adapted for the clustering problem.

4. The proposed quantum chaotic cuckoo search clustering algorithm

In this section, we first present the global overview of our proposed QCCS algorithm. Then the different procedures of QCCS adapted for the clustering problem are illustrated; namely: the chaotic initialization, the quantum update and bound handling strategy.

4.1. The global overview of QCCS

The main goal of the clustering process is to group the most similar objects in the same cluster or group. Each object is defined by a set of attributes or measurements.

To determine the similar objects, the measure of similarity between them is used. Several similarity measures are defined in the literature. But the Euclidean distance is the most popular and it is given by the following formula:

$$\text{distance}\left(o_i, o_j\right) = \left(\sum_{p=1}^{m} \left|o_{ip} - o_{jp}\right|^{\frac{1}{2}}\right)^2 \quad (17)$$

where, m is the number of attributes and o_{ip} is the value of the attribute number p of the object number $i\,(o_i)$.

Determine the best partition between a huge numbers of possible ones lead to consider the clustering as an optimization problem that optimizes an objective function. In our proposed algorithm, the sum intra cluster (SSE) measured by the Euclidean distance is used. This function minimizes the distance between each object and the centroid of the cluster that is assigned to in order to generate compact groups. The SSE is defined by the following formula:

$$SSE = \sum_{i=1}^{k}\sum_{j=1}^{n} W_{ij}{*}\sqrt{\sum_{p=1}^{m}\left(o_{jp} - c_{ip}\right)^2} \quad (18)$$

where, $W_{ij} = 1$ if the object j (o_j) is in the cluster i and 0 otherwise. k is the number of clusters, n is the number of objects, m is the number of attributes and c_{ip} is the value of the attribute number p of the centroid of the cluster number i and o_{jp} is the value of the attribute number p of the object number j.

Given the chosen fitness function, the clustering becomes a typical minimization problem.

The general overview of our proposed QCCS is like almost all optimization algorithms. So, first the population is initialized. Then, a set of instructions are repeated in loop until some criteria will be satisfied. Finally the global best nest is displayed. In this proposed algorithm the first population is generated by employing the chaotic sequence instead of random one. While the new update is performed by the nonhomogeneous quantum update presented in QCS. In the following section the detail of how the main procedures of QCCS are adapted for the clustering problem including the chaotic initialization, the quantum update and bound handling strategy are illustrated. The pseudo code of QCCS clustering algorithm is shown in Fig. 3.

4.2. Chaotic initialization

An efficient technique is chosen at the initial step by using the chaotic map to generate the first population of solution rather than random one. For the clustering problem the final solution is sensitive to the first nest. By using the chaotic map, the property of the non repetition and ergodicity accelerate the search speed by exploring all search space efficiently. The clustering is a multidimentional problem where a data is represented in m dimensional space. Finding k clusters means finding k vectors of m dimension where each vector is a centroid of a cluster. Consequently, a nest solution is represented by a matrix of k rows and m columns. To assure the chaotic initialization of the population, a vector of m dimensions is generated by the chaotic map for each centroid as follows:

$$x_{ij} = Lb + (Ub - Lb){*}c \quad (19)$$

where: x_{ij} is the jth centroid of the nest number i.

Lb is the lower bound vector, Ub is the upper bound vector and c is a vector of m dimensions generated by the logistic map illustrated in Eq. (1).

4.3. Quantum update

In the original CS, the homogeneous update presented in Eq. (2) is performed. While, in QCS two other kinds of updates are established in order to ensure their nonhomogeneity. They are focused on making a small perturbation in the current solution in combination with the mean solution and the best one in order to overcome the problem of being trapped in local optima.

We note that, in the clustering problem each nest is represented by k centroids, where the centroids order is important, so each one represents a class in the real dataset. Therefore, in each iteration centroids rearrangement is required for all nests of the population in order to ensure this purpose. So each population is replaced by new one with centroids rearranged. This later is used in all the arithmetic operations between nests in order to ensure a significant calculus.

Given a population with centroids to be rearranged, first centroids order of the real classes is chosen as it is given in a random nest of the population. This later is used as reference nest for the other ones. Then, all nests of the population follow the reference nest centroids order. Let a given nest of k centroids to be rearranged. Then, its ith centroid is replaced by its closest centroid to the ith centroid of the reference nest.

The new population with centroids rearranged is used first to determine the real mean by using the formula (14) and then to calculate the new update.

In order to illustrate the benefit of rearrangement of centroids into the clustering problem, an example is given in the calculus of the mean nest with and without rearrangement of centroids.

Example of calculus of the mean nest with and without rearrangement of centroids

Given a small dataset generated by two dimensions Gaussian distribution, composed of three different classes C_1, C_2 and C_3, with number of instances equal three, three and two respectively. The classes are defined as follows:

$$C_1 = \begin{bmatrix} 554370 & 280085 \\ 548888 & 292555 \\ 551269 & 313208 \end{bmatrix}; \; C_2 = \begin{bmatrix} 437025 & 406583 \\ 437285 & 409028 \\ 446310 & 388359 \end{bmatrix};$$

$$C_3 = \begin{bmatrix} 451408 & 304435 \\ 449331 & 302362 \end{bmatrix}$$

For simplicity, the population is composed only of two nests. Let x_1 and x_2 be the two nests generated at a given iterations number:

$$x_1 = \begin{bmatrix} 553862.18658 & 290440.34142 \\ 449695.59631 & 299669.68882 \\ 442651.94469 & 394570.80044 \end{bmatrix};$$

$$x_2 = \begin{bmatrix} 449607.73357 & 299707.10429 \\ 442650.34222 & 394581.38336 \\ 553862.26250 & 290445.00027 \end{bmatrix}$$

The nest mean of the population ($\bar{x}$) calculated using the formula (14) is given by:

$$\bar{x} = \begin{bmatrix} 501734.96007 & 295073.72285 \\ 446172.96926 & 347125.53608 \\ 498257.10359 & 342507.90035 \end{bmatrix}$$

In order to calculate the real mean with rearrangement of centroids ($\bar{x}_R$), the rearrangement of centroids of the nest of the population is performed as follows:

- The nest x_1 is chosen as reference nest and the centroids order is established as it is given in the reference nest x_1;
- The centroids order of the nest x_2 is rearranged by using the Euclidean distance given in Eq. (17). Let be Distance($x_1(i,:\,)$, $x_2(j,:\,)$) is the Euclidean distance between the ith centroid of the nest x_1 and the jth centroid of the nest x_2.

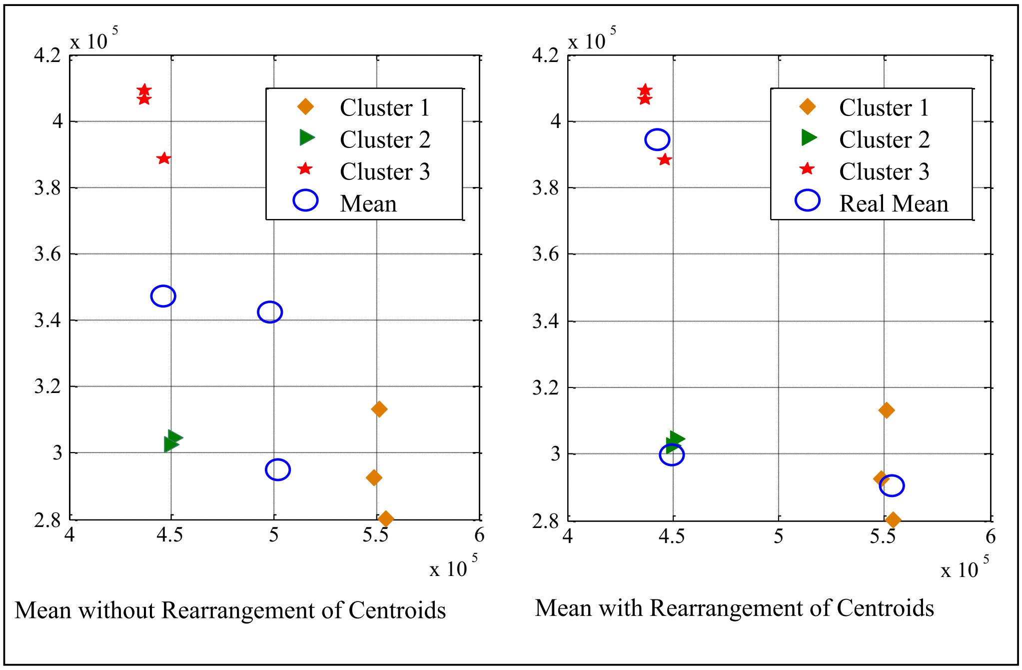

Fig. 2. Mean with and without rearrangement of centroids.

Distance(x_1(1,:), x_2(1,:)) = 104,665.48555
Distance(x_1(1,:), x_2(2,:)) = 152,359.54496
Distance(x_1(1,:), x_2(3,:)) = 4.65946

Consequently, the third centroid of x_2 is the closest one to the first centroid of x_1. Then, a permutation between the first centroid of x_2 and the third one of x_2 is performed and x_2 will be:

$$x_2 = \begin{bmatrix} 553862.26250 & 290445.00027 \\ 442650.34222 & 394581.38336 \\ 449607.73357 & 299707.10429 \end{bmatrix}$$

In the same way, the other centroids are rearranged:
Distance(x_1(2,:), x_2(2,:)) = 95,172.81841
Distance(x_1(2,:), x_2(3,:)) = 95.49753

Then, a permutation between the second centroid of x_2 and the third one of x_2 is performed and x_2 will be:

$$x_2 = \begin{bmatrix} 553862.26250 & 290445.00027 \\ 449607.73357 & 299707.10429 \\ 442650.34222 & 394581.38336 \end{bmatrix}$$

x_2 is rearranged and the real mean ($\bar{x}_R$) is then calculated using Eq. (14) and it is given by:

$$\bar{x}_R = \begin{bmatrix} 553862.22454 & 290442.67084 \\ 449651.66494 & 299688.39655 \\ 442651.14345 & 394576.09190 \end{bmatrix}$$

This example is illustrated in Fig. 2. This figure shows the significant influence of centroids rearrangement in the computation of the mean. So, in the case of the calculus of the mean without centroids rearrangement, the mean has no significance and does not represent the population. While, in the case of the calculus of the mean with centroids rearrangement, the mean has great significance and represents effectively the population.

In QCS only one update of three ones is selected by using a random number (l) with the same probability of appearance. The new nonhomogeneous update in the quantum cuckoo search clustering algorithm is given by the following equations:

$$x_{i+1} = \begin{cases} x_{iR} + \alpha.u|v|^{1/\beta} * (x_{iR} - x_{bR}) & l \in \left]\frac{2}{3}, 1\right] \\ \bar{x}_R + \delta * (\bar{x}_R - x_{iR}) * \ln(1/r) & l \in \left]\frac{1}{3}, \frac{2}{3}\right] \\ x_{iR} + \delta * (x_{bR} - x_{iR}) * \exp(r) & l \in \left[0, \frac{1}{3}\right] \end{cases} \quad (20)$$

where, u and v are matrices of the clustering problem dimension (k^*m), drawn from normal distribution as presented in Eqs. (4) and (5). l is a random value drawn from a uniform distribution on the interval [0, 1]. δ is a control parameter. $\bar{x}_R$ is the real mean obtained with rearrangement of centriods performed before the calculus of the mean using Eq. (14). x_{iR} and x_{bR} are the ith nest and the global best nest respectively obtained by the new population with rearrangements of centroids.

Also, in order to enhance research procedure, the previous global best nest is retained and reverted back in the worst nest.

4.4. Bound handling technique

The aim of boundary handling procedure is to keep the cuckoos inside the available search space. A simple method based on classical strategy (Cheng, Shi, & Qin, 2011) is used in the standard cuckoo search to handle boundary values. It is based to set the cuckoo at boundary when it exceeds the boundary. So, if the cuckoo moves outside the search bound for any dimension d, then it is replaced by the bound value. Empirically, the upper and low bound for dimension d are fixed as the maximum and minimum of dataset on this dimension. Although, this method solves the problem of boundary but it is not efficient in the search procedure.

In this paper, an efficient technique is used to handle the boundary problem in order to improve the search mechanism. So, when the cuckoo flies outside the maximum for any dimension, then it is adjusted by the upper bound and the corresponding value of the global best nest on this dimension. In the same way when the cuckoo flies outside the minimum for any dimension, then it is adjusted by the lower bound and the corresponding value of the global best nest on this dimension. The main idea behind

Quantum Chaotic Cuckoo Search Clustering Algorithm

Set the initial parameters

- *pa* (the probability of worse nests)
- *nb_nest* (the number of host nest and it is the population size)
- k (number of clusters)
- Max_Iter (the maximum number of iterations)

1. Initialize the population by the chaotic map using equation (19);
2. Calculate the fitness of the population and find the global best one;

While (t < Max_Iter) or (stop criterion)

3. Replace the population by a new one with centroids rearranged and calculate the new nonhomogeneous update using equation (20);
4. Adjust boundary using bound handling technique;
5. If the new nest is better than the old nest, update the old nest by the new one;
6. Generate a fraction (*pa*) of new nests to replace the worse nests;
7. Adjust boundary using bound handling technique;
8. If the new nest is better than the old one, replace the old nest by the new one;
9. retain the previous global best nest in the worst nest;
10. Find the global best nest;

End While;

11. Print the global best nest and fitness

Fig. 3. The pseudo code of QCCS.

Table 1
Summary of the main characteristics of the used datasets.

Datasets	Number of instances	Number of classes	Number of features	size of classes
Iris	150	3	4	50, 50, 50
Wine	178	3	13	59, 71, 48
Cmc	1473	3	9	629, 334, 510
Cancer	683	2	9	444,239
Seeds	209	3	7	70, 70, 69
Blood	748	2	4	570, 178

this strategy is to force the convergence of the solution to the global best nest in order to improve the search procedure.

5. Experiments and results

In this section, we illustrate the summary description of the main characteristics of the used datasets at first. Second, we discuss the experiments related parameters. Then, we have investigated the CS enhancement using different chaotic map and the boundary handling strategy. Finally, to emphasize the advantages of the proposed approach compared to the existing approaches in the literature, we conducted different experiments and the obtained results are thoroughly discussed.

5.1. Datasets

In order to test the performance of the proposed approach, the experiments have been conducted on six different well known benchmark datasets obtained from the famous UCI Machine Learning Repository (Blake, & Merz, 1998). The description summary of the main characteristics of the used datasets is given in Table 1.

5.1.1. Iris dataset

The Iris dataset contains 150 random samples of flowers from the iris with four attributes. They are unscrewed into 3 classes of 50 instances, where each class represents a type of iris plant (Setosa, Versicolor and Virginica).

5.1.2. Wine dataset

The Wine dataset describes the quality of wine from physicochemical properties, grown in the same region in Italy but derived from three different cultivars. There are 178 instances with 13 numeric features represent the quantities of 13 constituents found in each of the three types of wines.

5.1.3. CMC dataset

Contraceptive method choice (CMC) dataset is created by Tjen-Sien Lim. It is a subset of the 1987 national Indonesia contraceptive prevalence survey. The samples are married women who were either not pregnant or do not know if they were at the time of the interview. The problem is to predict the current contraceptive method choice (no use, long-term method, or short-term methods) of a woman based on her demographic and socioeconomic characteristics.

5.1.4. Cancer dataset

Cancer dataset represents the Wisconsin breast cancer databases. The dataset contains 683 instances with 9 features: Clump Thickness, Cell Size Uniformity, Cell Shape Uniformity, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nuclei, and Mitoses. Each instance has one of two possible classes: benign or malignant.

5.1.5. Seeds dataset

Seeds dataset contains 209 instances represented by the measurements of geometrical properties of kernels belonging to three different varieties of wheat (Kama, Rosa and Canadian). A soft X-ray technique and GRAINS package were used to construct all seven, real-valued attributes. Studies were conducted using combine harvested wheat grain originating from experimental fields, explored at the Institute of Agro physics of the Polish Academy of Sciences in Lublin.

5.1.6. Blood dataset

Blood dataset is taken from the Blood Transfusion Service Center in Hsin-Chu City in Taiwan. It consists of 748 donors selected

Table 2
GA, KCPSO, DE and KICS parameters setting.

Algorithm	Parameter	value
KCPSO	ω (inertia weight)	0.5 + (rand/2)
	c1 (Personal Learning Coefficient)	2
	c2 (Global Learning Coefficient)	2
GA	Pc (Crossover percentage)	0.8
	Pm (Mutation percentage)	0.3
	γ	0.2
	μ (Mutation rate)	0.02
	β (Selection pressure)	8
DE	BMin (Lower bound of scaling factor)	0.2
	BMax (Upper bound of scaling factor)	0.8
	PCr (Crossover probability)	0.2
KICS	Pa (Probability of worse nests)	[0.05,0.5]
	α (Step scaling factor)	[0.02,0.5]

at random from the donor database. Each instance is represented by 4 numerical features. These instances are grouped into 2 classes representing whether he/she donated blood in March 2007.

5.2. Related parameters

In this experiment, each algorithm was run for 50 individual times using the Matlab software with Windows 7 operating system (2.67 GHz Intel Core i5-560 M CPU and 4.00 GB RAM). The performance of the QCCS algorithm is compared against eight well known and recent algorithms reported in the literature, including genetic quantum cuckoo search (GQCS) presented in (Salim et al., 2014), hybrid cuckoo search and differential evolution algorithm (HCSDE) (Bouyer et al., 2015), hybrid K-means and improved cuckoo search algorithm (KICS) (Pandey et al., 2016), standard cuckoo search algorithm (CS) (Ishak Boushaki et al., 2014a), quantum particle swarm optimization algorithm (QPSO) presented by Sun et al. (2012), standard differential evolution algorithm (DE), hybrid K-means chaotic particle swarm optimization algorithm (KCPSO) (Chuang et al., 2011) and genetic algorithm with roulette wheel selection (GA).

All optimization algorithms are tested using the same population size and iterations number. The population size is set to 15 as suggested in (Yang, 2010). While the iterations number was fixed to 100 iterations for all programs. In order to assure good comparisons, the chosen parameters for the different clustering algorithms were set in the best performance following the suggestions of the original papers and/or the most commonly used configurations. Detailed parameter settings for the GA, KCPSO and DE are shown in Table 2.

The probability of worse nests was set to 0.25 in QCCS and CS, while it was set to 0.7 for HCSDE as recommended in (Bouyer et al., 2015). In QCCS, δ was set to 1.6. For all tables the best value obtained among all the techniques is reported in bold.

5.3. Chaotic initialization and boundary handling CS enhancement

In these experiments, the following two performance issues are studied. Firstly, how Chaos initialization enhanced Cuckoo Search clustering (CCS). Secondly, how Boundary Handling strategy enhanced Cuckoo Search search (BHCS) clustering algorithm.

To verify the effectiveness of chaotic initialization using Eq. (19) on standard cuckoo search, the CCS was compared against the standard CS. The best optimization results and the mean optimization results are presented in Tables 3 and 4 respectively.

As illustrated in Table 3, the ergodicity and spread-spectrum characteristics of chaotic map improve the CS. So, all the space solution is scanned efficiently. The CCS achieves the best optimization for all benchmark tests. As illustrated in Table 4, CCS achieves the best mean for all datasets.

In order to investigate the benefits of boundary handling technique on the CS, the enhanced boundary handling CS (BHCS) is compared against a standard CS. The best optimization results and the mean optimization results are presented in Table 5 and Table 6 respectively.

From Tables 5 and 6, it is clear that the boundary handling strategy improves the clustering results for all datasets. The convergence behaviours of CS, chaotic CS (CCS) and BHCS on different datasets are illustrated on Fig. 4.

Fig. 4 illustrates convergence behaviours of CS, chaotic CS (CCS) and BHCS on the six different datasets. It is clear from Fig. 4 that both CCS and BHCS enhance standard CS for all benchmark datasets.

5.4. QCCS comparison

In order to test the validity and the efficiency of the proposed approach, we have evaluated and compared it with GQCS, HCSDE, KICS, CS, QPSO, DE, KCPSO and GA in term of two metrics: an internal and an external quality measure. For the internal quality measure, we considered the sum of intra-cluster distances (SSE); it is represented by the formula (18). Whereas, the external quality measure is calculated by the error rate (ER), which represents the percentage of misplaced objects as given by formula (21).

$$ER = (number\ of\ misplaced\ objects/total\ of\ objects\ whithin\ dataset) * 100 \tag{21}$$

The detail of the intra-cluster distances, measured in terms of different Criteria (C): the Best (B), the Mean (M), the Worst (W) and the standard deviation (Std) obtained by the different clustering algorithms is shown in Table 7.

From Table 7 it is obvious that the proposed QCCS is significantly superior to all of the algorithms. It reaches the best SSE in term of best value, mean value, worst value with the less standard deviation for all the six datasets.

Fig. 5 shows a visualization of clustering results on two dimensions by using six different clustering algorithms (QCCS, GQCS, KICS, KCPSO, HCSDE and QPSO) on Iris dataset. It can be seen that QCCS achieves a good clustering results.

As presented in Table 7, in Iris dataset QCCS achieves the best fitness. Also it has efficient standard deviation compared to the other algorithms. The KCPSO is the second best and the difference between GQCS, KCPSO and KICS is not important. However, the convergence speed of HCSDE is better than CS. GA is a little better than the HCSDE and the DE algorithm has the worst convergence speed. The convergence behaviours of different clustering algorithms on Iris dataset are illustrated in Fig. 6.

In Wine dataset, as seen from the same table, QCCS is significantly superior to other clustering algorithms and has better intra-cluster distance value. The KCPSO is the second best and all KCPSO and GQCS has obtained almost the same results. While, the convergence speed of KICS is better than all CS, QPSO, HCSDE and

Table 3
The best optimization results of CCS against CS.

Datasets	Iris	Wine	Cmc	Cancer	Seeds	Blood
CS	97.98364	16,363.12921	5778.45388	3155.31409	329.96712	407,942.00547
CCS	**97.12873**	**16,316.18096**	**5604.82403**	**3138.89078**	**317.11765**	**407,721.09736**

Table 4
The mean optimization results of CCS against CS.

Datasets	Iris	Wine	Cmc	Cancer	Seeds	Blood
CS	102.51332	16,420.81062	5962.09604	3261.96733	347.19781	408,838.23528
CCS	**100.62080**	**16,409.22651**	**5824.16496**	**3244.72214**	**338.65254**	**408,303.06945**

Table 5
The best optimization results of BHCS against CS.

Datasets	Iris	Wine	Cmc	Cancer	Seeds	Blood
CS	97.98364	16,363.12921	5778.45388	3155.31409	329.96712	407,942.00547
BHCS	**97.42134**	**16,305.12949**	**5651.04113**	**3079.44737**	**318.19129**	**407,729.98447**

Table 6
The mean optimization results of BHCS against CS.

Datasets	Iris	Wine	Cmc	Cancer	Seeds	Blood
CS	102.51332	16,420.81062	5962.09604	3261.96733	347.19781	408,838.23528
BHCS	**100.40263**	**16,377.57822**	**5805.11196**	**3258.33611**	**337.36875**	**408,343.62008**

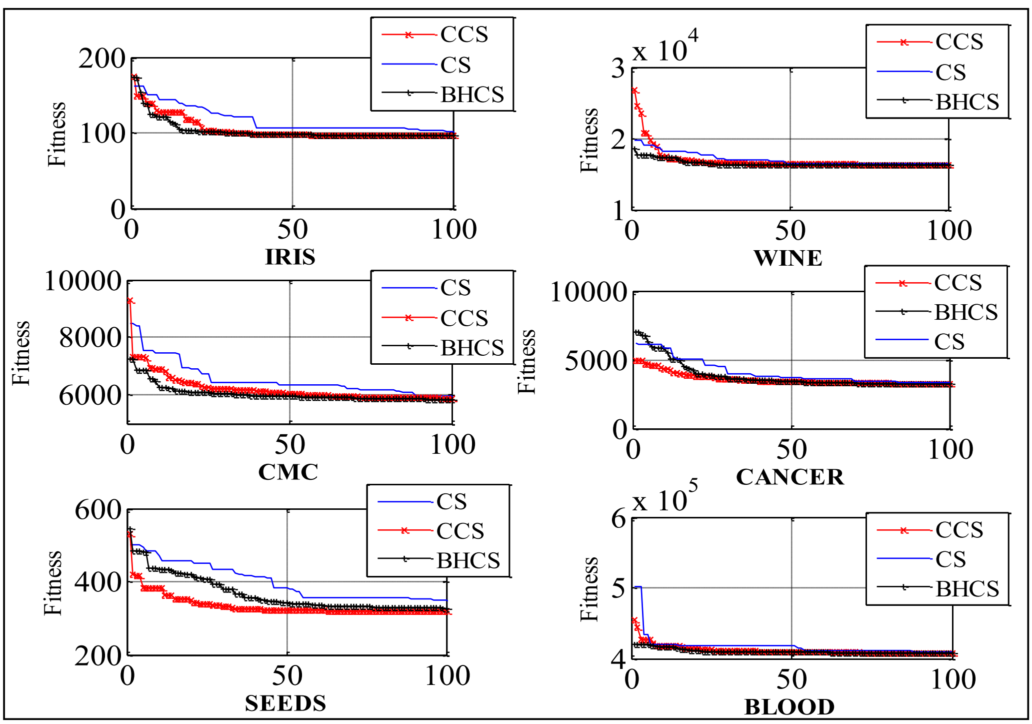

Fig. 4. Convergence behaviours of CS, CCS and BHCS on different datasets.

DE. However, the GA algorithm has the worst convergence speed. Fig. 7 shows the convergence behaviours of different clustering algorithms on Wine dataset.

As illustrated in Table 7, in Cmc dataset, QCCS can produce the best SSE in comparison with the other clustering algorithms. KCPSO is the second best followed by GQCS and the difference between GQCS and KICS is not significant. However all DE and GA have achieved almost a close optimization and it is superior to all HCSDE, QPSO and DE. However CS algorithm gives the worst optimization results on this dataset. The convergence behaviours of different clustering algorithms on Cmc dataset are shown in Fig. 8.

From the same table, it is clear that in Cancer dataset, QCCS achieved superior performance over the other clustering techniques in terms of convergence speed and standard deviation and the difference is significant. GQCS is the second best and it is close to KCPSO. They are followed by KICS. The GQCS is hybridization between the CS and genetic quantum computing. This is why GQCS has a good performance. So, the qubit representation offers a good diversity to the CS algorithm, while the interference and mutation operations help to intensify the search around the global best solutions. However, GA is better than all DE, QPSO and CS. HCSDE algorithm has achieved the worst SSE in comparison with the other algorithms. The convergence behaviours of different clustering algorithms on Cancer dataset are shown in Fig. 9.

As presented in Table 7, in Seeds dataset the proposed algorithm outperforms the other algorithms. KCPSO achieved almost close results to QCCS but with greater standard deviation than QCCS. It is followed by GQCS and the difference between GQCS and KICS is not significant. However, GA is better than all DE, QPSO and HCSDE and CS algorithm has obtained the worst results in comparison with the other algorithms. The convergence behaviours of different clustering algorithms on Seeds dataset are shown in Fig. 10.

From the same table, in Blood dataset, it is clear that QCCS is significantly superior to other clustering algorithms. GQCS is the second best followed by KICS then, KCPSO algorithm. However,

Table 7
Comparison of QCCS, GQCS, HCSDE, CS, QPSO, DE, KICS, KCPSO and GA clustering algorithms on the six datasets.

Algorithm	C	Iris	Wine	Cmc	Cancer	Seeds	Blood
QCCS	B	**96.65548**	**16,292.26452**	**5532.22476**	**2964.38951**	**310.74077**	**407,714.23116**
	M	**96.65623**	**16,293.26836**	**5532.71992**	**2964.41463**	**310.74671**	**407,714.23137**
	W	**96.66771**	**16,294.34770**	**5535.29050**	**2964.49945**	**310.77383**	**407,714.23329**
	std	**0.00266**	**0.71534**	**0.63436**	**0.02761**	**0.00690**	**0.00057**
CS	B	97.98364	16,363.12921	5778.45388	3089.77652	329.96712	407,942.00547
	M	102.51332	16,420.81062	5962.09604	3200.79638	347.19781	408,838.23528
	W	106.76087	16,525.72806	6205.93042	3476.06894	363.04983	410,019.61510
	std	2.18224	45.54086	115.23954	102.96476	8.52919	536.15978
GQCS	B	96.72782	16,294.58073	5537.17113	2965.82528	311.19737	407,714.44667
	M	96.85578	16,298.99439	5539.01262	2968.31705	311.60089	407,716.71566
	W	96.99663	16,305.37124	5540.27414	2970.80948	311.90521	407,724.58269
	std	0.07361	2.94997	0.78717	1.54070	0.19768	2.24878
HCSDE	B	98.25029	16,311.41199	5594.17775	3519.97434	318.88114	408,228.56713
	M	101.90691	16,428.54262	5680.31312	4009.04435	336.96191	411,305.32702
	W	107.53549	16,931.04478	5844.79814	4390.92378	363.37773	416,367.36060
	std	2.72158	163.67147	65.82339	241.51374	11.83082	2307.6760
KICS	B	96.73490	16,298.62741	5537.53403	2967.21670	311.40804	407,722.17812
	M	96.95257	16,341.46391	5540.65242	2973.38783	312.01572	407,807.43970
	W	97.19012	16,437.38458	5542.18213	2982.06865	312.41720	407,986.31323
	std	0.14438	37.75100	1.61266	4.18593	0.30436	71.23283
DE	B	98.87911	16,315.16231	5582.26753	2973.22562	321.28358	407,717.39514
	M	107.47667	16,378.14590	5694.75891	3087.96153	345.43344	408,478.04310
	W	116.95855	16,571.43269	5771.42764	3249.49690	370.56843	410,161.03017
	std	5.03306	56.74034	47.51672	82.03888	13.63513	788.00224
KCPSO	B	96.69197	16,293.89248	5534.50738	2966.59922	310.84751	407,765.22388
	M	96.72374	16,298.01763	5535.85046	2969.95736	310.91544	407,976.22760
	W	96.75015	16,304.46569	5537.29598	2972.06914	310.98571	408,273.71832
	std	0.01599	2.67438	0.82178	1.45602	0.03444	127.04569
QPSO	B	97.90195	16,314.80249	5688.65636	3031.71359	323.33507	407,750.10913
	M	106.60821	16,426.23249	5912.79089	3275.89841	347.01846	408,379.54947
	W	130.41299	16,818.29344	6161.09146	3845.67339	365.98251	409,550.28500
	std	10.07272	106.90106	146.80549	233.57723	11.93185	532.17490
GA	B	96.72273	16,302.80240	5577.62452	2974.07239	312.94765	407,723.78559
	M	99.11813	16,557.08215	5691.10960	3050.87498	327.62516	409,976.80458
	W	120.97865	18,378.61465	5950.58455	3149.41743	382.55278	415,255.02699
	std	5.08461	508.12590	84.08198	47.90260	18.71269	2899.78136

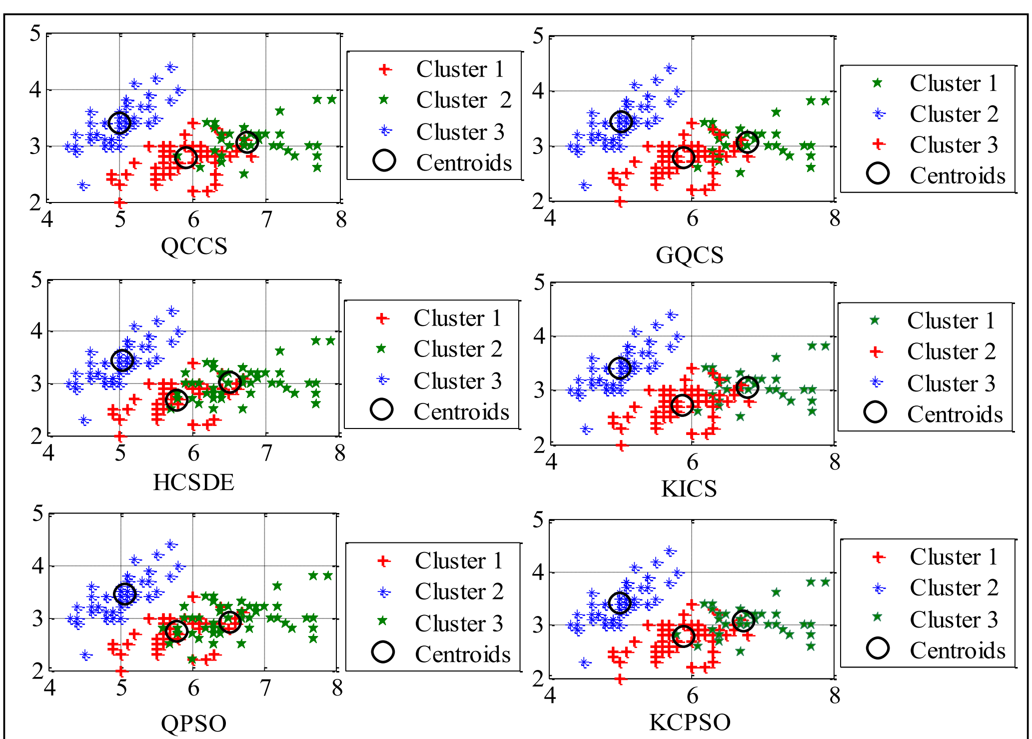

Fig. 5. Visualization of clustering results.

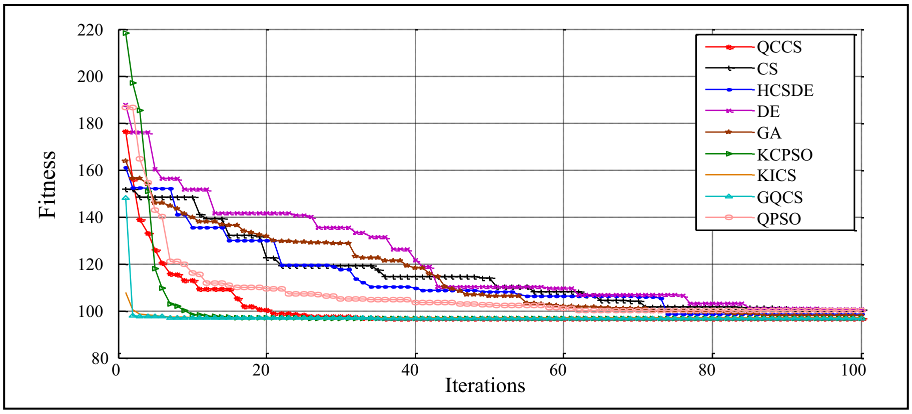

Fig. 6. The convergence behaviours of different clustering algorithms (QCCS, GQCS, KICS, HCSDE, CS, QPSO, DE, KCPSO and GA) on Iris dataset.

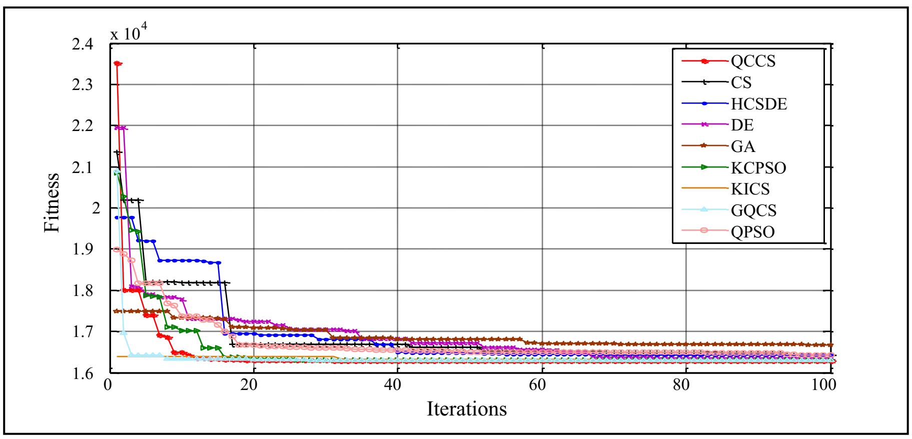

Fig. 7. The convergence behaviours of different clustering algorithms (QCCS, GQCS, KICS, HCSDE, CS, QPSO, DE, KCPSO and GA) on Wine Dataset.

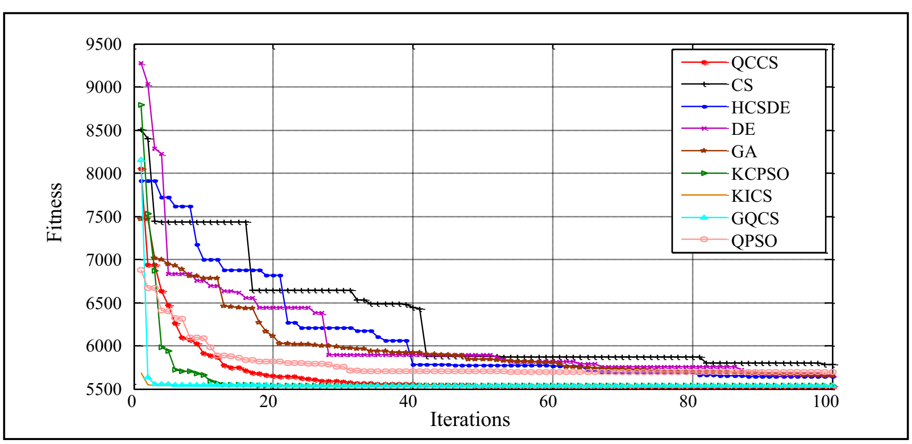

Fig. 8. The convergence behaviours of different clustering algorithms (QCCS, KICS, HCSDE, CS, DE, KCPSO and GA) on Cmc Dataset.

QPSO is better than all DE, CS and GA. HCSDE algorithm has obtained the worst results in comparison with the other algorithms. The convergence behaviours of different clustering algorithms on blood dataset are shown in Fig. 11.

From most figures, we can state that all KCPSO and KICS converge more rapidly than QCCS at the beginning steps. This is due that all KCPSO and KICS are hybridized with the famous K-means and the hybrid algorithms regroup the advantages of all the participated ones. In addition to the fast convergence of K-means, all these algorithms are improved version of the original algorithms. So, in the case of KICS the first population is generated by the K-means. Where, each nest of the population is initialized by the output of the execution of K-means until the convergence. Also, in the improved CS (ICS), the parameters of the algorithm are dynamically changed with the number of iterations. While in the chaotic PSO (CPSO) the random parameters of PSO are generated by the logistic map in order to use the proprieties of chaotic map and therefore improve the global convergence.

Although KICS and KCPSO have faster convergence than QCCS at the beginning steps, they may be trapped in local optima like K-means. Also, their individuals have a homogeneous search procedure that may lead to premature convergence and poor results.

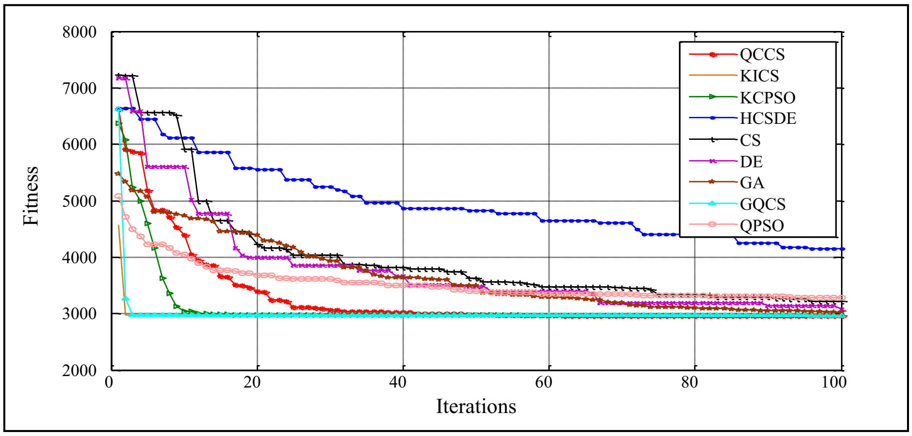

Fig. 9. The convergence behaviours of different clustering algorithms (QCCS, GQCS, KICS, HCSDE, CS,QPSO, DE, KCPSO and GA) on Cancer Dataset.

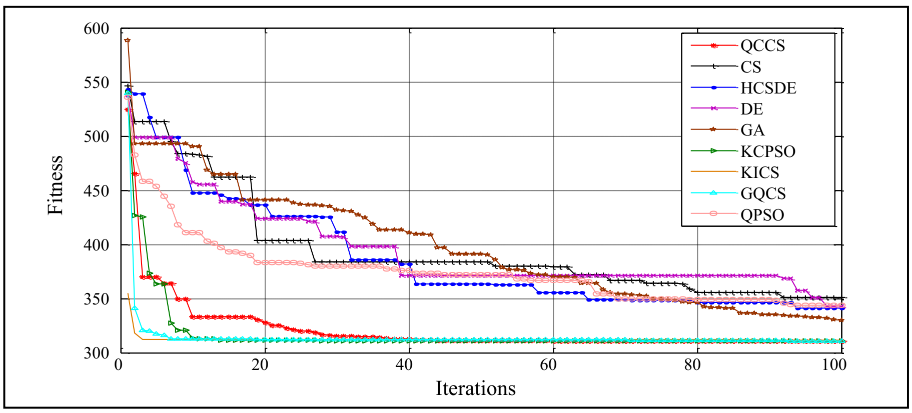

Fig. 10. The convergence behaviours of different clustering algorithms (QCCS, KICS, HCSDE, CS, DE, KCPSO and GA) on Seeds Dataset.

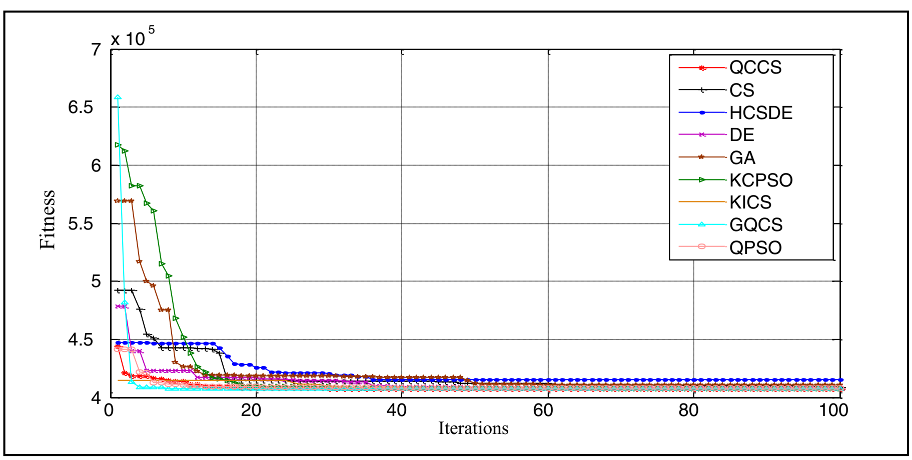

Fig. 11. The convergence behaviours of different clustering algorithms (QCCS, GQCS, KICS, HCSDE, CS, QPSO, DE, KCPSO and GA) on Blood Dataset.

It is the case of blood dataset where the behaviour of the proposed QCCS is significantly better than KICS and KCPSO. However, in QCCS different improvements are incorporated into standard CS. The chaotic map is used to generate the initial centroids of the population in order to explore efficiently almost all the global search space by using ergodicity and spread-spectrum proprieties of chaotic map. Also, in the standard CS all the cuckoos have identical search behaviours that may make the algorithm trap in local optima and then leading to premature convergence. However, in QCCS this problem is resolved by using nonhomogeneous update inspired from quantum theory. So, solutions are diversified and the search procedure is explored perfectly by discovering new regions leading to a promising direction approaching to the global best. Furthermore, the boundaries are well managed using an effective strategy in order to improve the search mechanism. This is why QCCS succeeds to achieve the global best for all datasets.

The mean error rate obtained by the different clustering algorithms on the employed datasets is shown in Table 8. As shown in this table, for all datasets, the proposed algorithm has minimum error rate. In Iris dataset QPSO can produce an error rate close to the proposed algorithm. However, in Wine dataset, all GA and

Table 8
Mean error rate of QCCS, GQCS, KICS, HCSDE, CS, QPSO, DE, KCPSO and GA clustering algorithms on the six datasets.

Datasets	QCCS	CS	GQCS	HCSDE	KICS	DE	KCPSO	QPSO	GA
Iris	**09.43**	09.80	10.00	09.90	10.66	10.76	10.66	09.46	10.00
Wine	**28.70**	29.10	28.78	29.04	28.98	28.84	28.90	28.76	28.76
Cmc	**57.11**	57.18	57.29	57.52	57.64	57.28	57.12	57.15	57.68
Cancer	**03.51**	04.94	**03.51**	04.74	**03.51**	03.82	03.55	03.80	03.87
Seeds	**10.52**	11.26	**10.52**	11.02	**10.52**	11.62	**10.52**	11.62	13.42
Blood	**34.89**	**34.89**	**34.89**	**34.89**	**34.89**	**34.89**	**34.89**	**34.89**	**34.89**

Table 9
Comparative results of the paired student's t-test for mean fitness for the six datasets.

Datasets	Method	t	95% of Confidance Interval	Two- tailed P	Significance
Iris	QCCS	**158,363.95391**	**96.65496–96.65751**	**4.08827E−088**	Extremely significant
	CS	204.76403	101.46546–103.56117	3.08610E−033	Extremely significant
	GQCS	5734.92845	96.82043–96.89113	9.83957E−061	Extremely significant
	HCSDE	163.21415	100.60008–103.21374	2.28986E−031	Extremely significant
	KICS	2926.88172	96.88324–97.02190	3.49384E−055	Extremely significant
	DE	93.08047	105.05993–109.89341	9.72949E−027	Extremely significant
	KCPSO	26,366.37118	96.71606–96.73142	2.54128E−073	Extremely significant
	QPSO	46.13394	101.77156–111.44485	5.67148E−021	Extremely significant
	GA	84.97121	96.67663–101.55962	5.47670E−026	Extremely significant
Wine	QCCS	**91,826.36428**	**16,292.89698–16,293.63974**	**1.28419E−083**	Extremely significant
	CS	1571.70183	16,398.94314–16,442.67809	4.72005E−050	Extremely significant
	GQCS	24,083.50525	16,297.57790–16,300.41089	1.42015E−072	Extremely significant
	HCSDE	437.52498	16,349.95205–16,507.13318	1.68099E−039	Extremely significant
	KICS	1886.85808	16,323.33691–16,359.59091	1.465350E−051	Extremely significant
	DE	1258.19964	16,350.90074–16,405.39106	3.23386E−048	Extremely significant
	KCPSO	26,563.61110	16,296.73346–16,299.30179	2.20574E−073	Extremely significant
	QPSO	669.78083	16,374.90152–16,477.56346	5.15374E−043	Extremely significant
	GA	142.03300	16,313.09399–16,801.07032	3.20564E−030	Extremely significant
Cancer	QCCS	**467,985.73241**	**2964.40137–2964.42789**	**4.68372E−097**	Extremely significant
	CS	135.50215	3151.35552–3250.23724	7.83372E−030	Extremely significant
	GQCS	8397.83099	2967.57725–2969.05686	7.01187E−064	Extremely significant
	HCSDE	72.35621	3893.07606–4125.01265	1.15010E−024	Extremely significant
	KICS	3096.25202	2971.37786–2975.39780	1.19981E−055	Extremely significant
	DE	164.06990	3048.56870–3127.35435	2.07342E−031	Extremely significant
	KCPSO	8891.13133	2969.25821–2970.65650	2.37042E−064	Extremely significant
	QPSO	61.13314	3163.74101–3388.05581	2.79143E−023	Extremely significant
	GA	277.61445	3027.87346–3073.87650	9.52012E−036	Extremely significant
Cmc	QCCS	**38,016.76606**	**5532.41531–5533.02452**	**2.42930E−076**	Extremely significant
	CS	225.51436	5906.76116–6017.43092	4.93438E−034	Extremely significant
	GQCS	30,671.79898	5538.63464–5539.39060	1.43537E−074	Extremely significant
	HCSDE	376.15670	5648.70652–5711.91971	2.96808E−038	Extremely significant
	KICS	14,975.89009	5539.87806–5541.42678	1.18181E−068	Extremely significant
	DE	522.40301	5671.94268–5717.57514	5.78934E−041	Extremely significant
	KCPSO	29,363.07503	5535.45586–5536.24506	3.28686E−074	Extremely significant
	QPSO	175.56058	5842.29891–5983.28288	5.73424E−032	Extremely significant
	GA	295.03314	5650.73573–5731.48347	2.99645E−036	Extremely significant
Seeds	QCCS	**196,072.03868**	**310.74339–310.75003**	**7.06501E−090**	Extremely significant
	CS	177.43772	343.10232–351.29329	4.68561E−032	Extremely significant
	GQCS	6870.69926	311.50596–311.69581	3.17681E−062	Extremely significant
	HCSDE	124.14878	331.28107–342.64275	4.12346E−029	Extremely significant
	KICS	4468.40901	311.86957–312.16187	1.12746E−058	Extremely significant
	DE	110.42865	338.88622–351.98066	3.80486E−028	Extremely significant
	KCPSO	39,340.25232	310.89889–310.93198	1.26795E−076	Extremely significant
	QPSO	126.77141	341.28911–352.74781	2.77307E−029	Extremely significant
	GA	76.31638	318.63983–336.61048	4.192520E−025	Extremely significant
Blood	QCCS	**3,103,101,540.04540**	**407,714.23110–407,714.23165**	**1.15050E−169**	Extremely significant
	CS	3323.79378	408,580.78602–409,095.68454	3.11857E−056	Extremely significant
	GQCS	790,290.15831	407,715.63586–407,717.79547	2.22364E−101	Extremely significant
	HCSDE	776.90209	410,197.24411–412,413.40994	3.07558E−044	Extremely significant
	KICS	24,954.66398	407,773.23564–407,841.64376	7.22982E−073	Extremely significant
	DE	2259.52974	408,099.66597–408,856.42022	4.77147E−053	Extremely significant
	KCPSO	13,997.53935	407,915.22373–408,037.23147	4.26607E−068	Extremely significant
	QPSO	3.344.92506	408,124.01364–408,635.08531	2.76479E−056	Extremely significant
	GA	616.26972	408,584.40888–411,369.20028	2.50708E−042	Extremely significant

QPSO have achieved an error rate close to QCCS. In Cmc dataset, all of the algorithms obtain an error rate close to each other and the difference is not important. As seen from Table 8, it is clear that in Cancer dataset the best minimum error rate is obtained by all QCCS, GQCS and KICS. In Seeds dataset, the best minimum error rate is obtained by all QCCS, KICS, GQCS and KCPSO. However in Blood dataset all the clustering algorithms have achieved the same error rate. Despite there is no much correlation between the intra-cluster distance and the error rate, QCCS succeeds to achieve the minimum error rate for all benchmark datasets.

In order to validate the experimental results, the statistical computation is also being performed for the considered datasets using student's t-test (Owen, 1965).

Student's t-test is applied to a confidence level of 95% with the null hypothesis that there is no significant difference for 50 runs using different clustering algorithms. The output of the above experiment is demonstrated in Table 9. From this table, it is visualized that in all cases these are significant differences i.e. null hypothesis is rejected.

6. Conclusion

In this paper, a new quantum chaotic cuckoo search algorithm (QCCS) is proposed to solve efficiently the data clustering problem. The proposed algorithm is based on the cuckoo search optimization enhanced by inspired techniques from quantum theory and chaotic map.

In QCCS the randomness of the initial population is resolved by using the performance proprieties of chaotic map including the ergodicity and non-repetition. In addition to that, the search procedure is improved by nonhomogeneous quantum update technique where the solutions are diversified and the new regions of search space are better exploited. Furthermore, an efficient strategy is presented to handle the boundary values in order to enhance the search procedure.

The performance of the proposed approach was assessed on six different datasets from the famous UCI Machine Learning Repository and compared to eight well known and recent algorithms. The conducted comparisons in terms of sum intra cluster, error rate and Student's t-test show the superiority and the efficiency of the proposed approach to achieve a best clustering quality for all benchmark datasets. Finally, as a future work first we plane to apply the proposed approach on specific datasets. Then, we suggest applying the quantum theory to other recent meta-heuristic algorithms.

Conflicts of interest

None.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. We would like to thank the anonymous reviewers for their helpful comments and advice in improving this work.

References

Adesso, G., Bromley, T. R., & Cianciaruso, M. (2016). Measures and applications of quantum correlations. *Journal of Physics A: Mathematical and Theoretical, 49*, 473001. doi:10.1088/1751-8113/49/47/473001.

Babrdel Bonab, M. (2015). An effective hybrid of bees algorithm and differential evolution algorithm in data clustering. *Mathematical Problems in Engineering*.

Barbakh, W., Wu, Y., & Fyfe, C. (2009). Review of clustering algorithms. In *Non-standard parameter adaptation for exploratory data analysis* (pp. 7–28). Berlin/Heidelberg: Springer.

Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases*. Available from *http://archive.ics.uci.edu/ml/datasets.html*.

Bouyer, A., Ghafarzadeh, H., & Tarkhaneh, O. (2015). An efficient hybrid algorithm using cuckoo search and differential evolution for data clustering. *Indian Journal of Science and Technology, 8*, 60146.

Caponetto, R., Fortuna, L., Fazzino, S., & GabriellaXibilia, M. (2003). Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Transactions on Evolutionary Computing, 7*, 289–304.

Cheng, S., Shi, Y., & Qin, Q. (2011). Experimental study on boundary constraints handling in particle swarm optimization: From population diversity perspective. *International Journal of Swarm Intelligence Research, 2*(3), 43–69.

Cheung, N. J., Ding, X. M., & Shen, H. B. (2017). A nonhomogeneous cuckoo search algorithm based on quantum mechanism for real parameter optimization. *IEEE Transactions on Cybernetics, 47*(2), 391–402. doi:10.1109/TCYB.2016.2517140.

Chuang, L.-Y., Hsiao, C.-J., & Yang, C.-H. (2011). Chaotic particle swarm optimization for data clustering. *Expert Systems with Applications, 38*(12), 14555–14563.

Civicioglu, P., & Besdok, E. (2013). A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review, 39*(4), 315–346 Springer Netherlands.

Civicioglu, P., & Besdok, E. (2014). Comparative analysis of the cuckoo search algorithm. In X.-S. Yang (Ed.). In *Cuckoo search and firefly algorithm, theory and applications, studies in computational intelligence: 516* (pp. 85–113). Switzerland: Springer International Publishing.

Dalhatu, K., & Tie Hiang Sim, A. (2016). Density base k-mean's cluster centroid initialization algorithm. *Internationa Journal of Computer Application, 137*(11), 0975–8887.

Draa, A., Meshoul, S., Talbi, H., & Batouche, M. (2010). A quantum inspired differential evolution algorithm for solving the N-queens problem. *The International Arab Journal of Information Technology, 7*(1), 21–27.

Dubes, R. (1993). Cluster analysis and related issue. In C. Chen, L. Pau, & P. Wang (Eds.), *Handbook of pattern recognition and computer vision* (pp. 3–32). Singapore: World Scientific.

Ebrahimzadeh, R., & Jampour, M. (2013). Chaotic genetic algorithm based on lorenz chaotic system for optimization problems. *International Journal of Intelligent Systems and Applications, 5*(5), 19–24.

Gan, G., Ma, C., & Wu, J. (2007). Data clustering: theory, algorithms, and applications. *Data clustering: theory, algorithms, and applications*. Philadelphia, USA: Society for Industrial and Applied Mathematics.

Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013a). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation, 18*(1), 89–98.

Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013b). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation, 18*(1), 327–340.

Gilmore, R., & Lefranc, M. (2011). *The topology of chaos* (2nd ed.). WILEY-VCH Verlag GmbH & Co. KGaA.

Guo, Z., Bo, C., Min, Y., & Cao, B. (2006). Self-adaptive chaos differential evolution. In L. Jiao (Ed.), *Advances in natural computation: second international conference ICNC, Xi'an, China. Proceedings, part I*. In *Lecture Notes in Computer Science: vol. 4221* (pp. 972–975). Berlin, Germany: Springer. et al.

Han, K. H., & Kim, J. H. (2000). Genetic quantum algorithm and its application to combinatorial optimization problem. In *Proc. IEEE congr. on evolutionary computation, California, CA, USA, july* (pp. 1354–1360).

Han, K. H., & Kim, J. H. (2002). Quantum inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation, 6*(6), 580–593.

Han, K. H., & Kim, J. H. (2004). Quantum-inspired evolutionary algorithms with a new termination criterion, Hε gate, and two-phase scheme. *IEEE Transactions on Evolutionary Computation, 8*(2), 156–169.

Handl, J., Corne, D., & Knowles, J. (2015)., Evolutionary clustering. in Encyclopedia of Machine Learning (2nd ed.).

Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information Sciences, 222*, 175–184.

Hilborn, R. C. (2004). *Chaos and nonlinear dynamics: An introduction for scientists and engineers* (2nd ed.). New York, NY, USA: Oxford University Press.

Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & de Carvalho, A. C. P. L. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems Man and Cybernetics Part C: Applications and Reviews, 39*(2), 133–155.

Hruschka, E. R., Campello, R. J. G. B., & de Castro, L. N. (2006). Evolving clusters in gene-expression data. *Information Sciences, 176*, 1898–1927.

Ishak Boushaki, S., Kamel, N., & Bendjeghaba, O. (2014a). A new algorithm for data clustering based on cuckoo search optimization, in genetic and evolutionary computing. In V. Snášel, J. S. Pan, & P Krömer (Eds.), *Proceedings of the seventh international conference on genetic and evolutionary computing, - Prague, Czech Republic*. In *Advances in Intelligent Systems and Computing: 238* (pp. 55–64). Springer International Publishing.

Ishak Boushaki, S., Kamel, N., & Bendjeghaba, O. (2014b). A new hybrid algorithm for document clustering based on cuckoo search and k-means. In T. Herawan, R. Ghazali, & M. Mat Deris (Eds.), *Recent advances on soft computing and data mining, proceedings of the first international conference on soft computing and data mining, UniversitiTun Hussein Onn Malaysia, Johor, Malaysia June 16th-18th, 2014*. In *Advances in Intelligence Systems and computing: 287* (pp. 59–68). Switzerland: Springer International Publishing. 2014. doi:10.1007/978-3-319-07692-8_6.

Ishak Boushaki, S., Kamel, N., & Bendjeghaba, O. (2015). Improved cuckoo search algorithm for document clustering. In A. Amine, L. Bellatreche, Z. Elberrichi, E. Neuhold, & R. Wrembel (Eds.), *In IFIP Advances in information and communication technology, computer science and its applications, proceedings of 5th IFIP TC 5 international conference: 456* (pp. 217–228). Springer International Publishing. 2015. doi:10.1007/978-3-319-19578-0_18.

Jaeger, G. (2007). *Quantum information* (pp. 81–89). New York, NY, USA: Springer.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 3*, 651–666.

Jain, K., Murthy, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.

Johnson, M., & Purdom, E. (2017). Clustering of mRNA-seq data based on alternative splicing patterns. *Biostatics, 18-2*, 295–307. https://doi.org/10.1093/biostatistics/kxw044.

Karaboga, D., & Ozturk, C. (2011). A novel clustering approach. Artificial Bee Colony (ABC) algorithm. *Applications of Soft Computing, 11*, 652–657.

Kwedlo, W. (2011). A clustering method combining differential evolution with the k-means algorithm. *Pattern Recognition Letters, 32*(12), 1613–1621.

Layeb, A. (2010). A quantum inspired particle swarm algorithm for solving the maximum satisfiability problem. *International Journal of Combinational Optimization Problems and Informatics, 1*(1), 13–23.

Layeb, A. (2011). A novel quantum inspired cuckoo search for knapsack problems. *International Journal. Bio-Inspired Compution, 3*(5), 297–305.

Layeb, A. (2013). A hybrid quantum inspired harmony search algorithm for 0-1 optimization problems. *Journal of Computational and Applied Mathematics, 253*, 14–25.

Layeb, A., & Saidouni, D.-E. (2008). A new quantum evolutionary local search algorithm for MAX 3-SAT problem. In *Hybrid artificial intelligence systems* (pp. 172–179). Heidelberg, Germany: Springer.

Lihua, T., Liguo, H., & Junhua, Y. (2016). Research on image segmentation based on clustering algorithm. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 9*(2), 1–12 http://dx.doi.org/10.14257/ijsip. 2016.9.2.01.

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition, 33*, 1455–1465.

Moh'd Alia, O., Azmi Al-Betar, M., Mandava, R., & Tajudin Khader, A. (2011). Data clustering using harmony search algorithm. In *SEMCCO: vol. 2* (pp. 79–88).

Nielsen, M., & Chuang, I. L. (2000). *Quantum computation and quantum information*. Cambridge university press.

Niu, B., Duan, Q., Liu, J., Tan, L., & Liu, Y. (2017). A population-based clustering technique using particle swarm optimization and k-means. *Natural Computing, 16*(1), 45–59.

Ott, E. (2002). *Chaos in dynamical systems,*. Cambridge, UK: Cambridge University Press.

Owen, D. (1965). The power of student's t-test. *Journal of the American Statistical Association, 60*, 320–333.

Pandey, A. C., Rajpoot, D. S., & Saraswat, M. (2016). Data clustering using hybrid improved cuckoo search method. *Contemporary computing (IC3), ninth international conference on, IEEE* 10.1109/IC3.2016.7880195.

Paterlini, S., & Krink, T. (2006). Differential evolution and particle swarm optimisation in partitional clustering. *Computational Statistics & Data Analysis, 50*(5), 1220–1247.

Rana, S., Jasola, S., & Kumar, R. (2011). A review on particle swarm optimization algorithms and their applications to data clustering. *Artificial Intelligence Review, 35*, 211–222.

Salim, C., Layeb, A., & Kartous, W. (2014). A new quantum cuckoo search algorithm for multiple sequence alignment. *Journal of Intelligent Systems, 23*(3), 261–275.

Schrödinger, E. (1926). An undulatory theory of the mechanics of atoms and molecules. *Physical Review, 28*, 1049–1070.

Schuster, H. G., & Just, W. (2005). *Deterministic chaos: An introduction*. Weinheim: Wiley-VCH Verlag GmbH.

Sun, J., Fang, W., Wu, X., Palade, V., & Xu, W. (2012). Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection. *Evolutionary Computation, 20*(3), 349–393.

Sun, J., Xu, W., & Ye, B. (2006). Quantum-Behaved particle swarm optimization clustering algorithm. In *Proceedings of the second international conference on advanced data mining and applications, August 14-16* 10.1007/11811305_37.

Swagatam, D., Ajith, A., & Amit, K. (2009). *Metaheuristic clustering* (p. 252). Springer Science & Business Media. 24 mars.

Thornton, S. T., & And Rex, A. (2013). Modern physics for scientists and engineers. (4th ed.). United States of America: Cole, Cengage Learning, (Chapter 6).

Tvrdík, J., & Krivy, I. (2015). Hybrid differential evolution algorithm for optimal clustering. *Applications of Soft Computing, 35*, 502–512.

Van der Merwe, D. W., & Engelhrecht, A. P. (2003). Data clustering using particle swarm optimization. In *The 2003 congress on evolutionary computation* (pp. 215–220).

Xing, B., & Gao, W-J. (2014). Cuckoo inspired algorithms. In *Innovative computational intelligence: A rough guide to 134 clever algorithms*. In *Intelligent systems reference library: vol. 62* (pp. 105–121). Switzerland: Springer International Publishing. Part II, Chapter 7.

Xu, R., & Wunsch II, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks and Learning Systems, 16*(3), 645–678.

Yang, S. Y., Wang, M., & Jiao, L. C. (2004). A novel quantum evolutionary algorithm and its application. In *Proc. IEEE congr. on evolutionary computation, Portland, USA, june 19-23* (pp. 820–826).

Yang, X.-S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research, 40*, 1616–1624.

Yang, X-S. (2010). *Engineering optimization: An introduction with metaheuristic applications* (1st ed.). Wiley Publishing.

Yang, X-S. (2014). In X-S. Yang (Ed.), *Cuckoo search and firefly algorithm: Theory and applications*. Switzerland: Springer International Publishing 10.1007/978-3-319-02141-6.

Yang, X-S., & Deb, S. (2009). Cuckoo search via levy flights. In *NaBIC 2009: Proceedings of world congress on nature & biologically inspired computing, world congress* (pp. 210–214). Coimbatore: IEEE Publications.

Yang, X-S., & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation, 1*(4), 330–343.

Zhang, G. (2010). Quantum-inspired evolutionary algorithms: A survey and empirical study. *Journal of Heuristics, 17*(3), 303–351.