Scientific Research and Essays Vol. 7(49), pp. 4176-4185, 17 December, 2012
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE11.1183
ISSN 1992-2248 ©2012 Academic Journals

Full Length Research Paper

Cat swarm optimization clustering (KSACSOC): A cat swarm optimization clustering algorithm

Yongguo Liu[1,2,3]*, Xindong Wu[3] and Yidong Shen[2]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P. R. China.
[2]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100191, P. R. China.
[3]Department of Computer Science, University of Vermont, Burlington, Vermont 05405, USA.

Accepted 15 November, 2012

Clustering is an unsupervised process that divides a given set of objects into groups so that objects within a cluster are highly similar with one another and dissimilar with the objects in other clusters. In this article, a new clustering method based on cat swarm optimization was proposed to find the proper clustering of data sets called *K*-means improvement and Simulated Annealing selection based cat swarm optimization clustering (KSACSOC). In the KSACSOC method, the seeking mode with *k*-means improvement was designed to enhance the clustering solution obtained in the process of iterations, and the tracing mode with simulated annealing selection was developed to explore the unvisited solution space. Experimental results on two artificial and six real life data sets are given to illustrate the superiority of the proposed algorithm over *k*-means algorithm, a simulated annealing clustering method, and a particle swarm optimization clustering method.

Key words: Clustering, cat swarm optimization, *k*-means, simulated annealing.

INTRODUCTION

Clustering is an important technique for discovering the inherent structure in any given pattern set without any prior knowledge. The clustering result should possess two properties: (1) homogeneity within the clusters, that is, the objects belonging to the same cluster should be as similar as possible, and (2) heterogeneity between the clusters, that is, the objects belonging to different clusters should be as different as possible. Clustering analysis has been applied in many fields such as machine learning, pattern recognition, and statistics (Pedrycz, 2005). Many clustering approaches have been reported which can be classified into two categories: hierarchical and partitional (Omran et al., 2007).

In this article, we focus our attention on partitional clustering. Some clustering techniques are available in the literature. Among them, *k*-means algorithm, a typical iterative hill-climbing method, is popular. However, the major drawbacks of the *k*-means algorithm are that it often gets stuck at local minima and its result is largely dependent on the choice of initial cluster centers (Selim and Ismail, 1984). *K*-harmonic means algorithm, another center-based clustering method, is proposed by Zhang et al. (1999) and modified by Hammerly and Elkan (2002) to solve the problem of initialization of the *k*-means algorithm. It is demonstrated that the *k*-harmonic means algorithm is essentially insensitive to the initialization of cluster centers. However, it tends to converge to local optima in some cases. In order to overcome the shortcomings of the *k*-means algorithm, researchers designed some improved clustering methods (Pedrycz, 2005; Omran et al., 2007). Recently, researchers employed metaheuristic techniques such as genetic algorithms, simulated annealing, particle swarm optimization, and tabu search to deal with the clustering problem so as to achieve the optimal or near-optimal solution within a specified number of iterations.

Cat swarm optimization (CSO), a recent metaheuristic technique firstly reported by Chu and Tsai (2007), models

*Corresponding author. E-mail: liuyg_cn@163.com.

the behavior of cats to solve the optimization problem. In this article, we employ cat swarm optimization to deal with the clustering problem, develop *k*-means improvement based seeking mode, and design simulated annealing selection based tracing mode. As a result, a new clustering method is proposed called *K*-means improvement and Simulated Annealing selection based cat swarm optimization clustering (KSACSOC). On one hand, *k*-means improvement fine-tunes the object distribution among different clusters so as to enhance the convergence of the KSACSOC algorithm, and on the other hand, simulated annealing selection accepts bad solutions probabilistically so as to strengthen the exploration of the unvisited solution space. In this paper, our aim is to introduce cat swarm optimization to deal with the clustering problem, explore its applicability to clustering analysis, and to hybridize cat swarm optimization with *k*-means algorithm and simulated annealing so as to combine the advantages of each one of them and evolve the proper clustering of data sets. To our best knowledge, this is the first reported study that reflects on the usage of the combination of cat swarm optimization, *k*-means algorithm, and simulated annealing in clustering analysis. Experimental results on two artificial and six real life data sets are given to illustrate that the KSACSOC algorithms can provide better objective function values and higher success rates than *k*-means algorithm, a simulated annealing clustering method, and a particle swarm optimization clustering method.

The remaining part of this article is organized as follows. The clustering problem under consideration and the related work are first reported. Then the KSACSOC algorithm and its components are described in detail. The impact of the elements of the KSACSOC algorithm is investigated. Performance comparison between the KSACSOC algorithm and some known clustering methods is then conducted for two artificial and six real life data sets. Finally, experimental results are analyzed and concluded.

RELATED WORK

In this article, we focus on the clustering problem defined as follows

$$\min_{W,C} J(W,C) = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij} \parallel \mathbf{x}_i - \mathbf{c}_j \parallel^2 , \qquad (1)$$

subject to

$$\sum_{j=1}^{K} w_{ij} = 1 , \qquad (2)$$

where $\parallel \parallel^2$ denotes the squared Euclidean distance between object $\mathbf{x}_i$ and cluster center $\mathbf{c}_j$, N denotes the number of objects, K denotes the number of clusters, $C = \{C_1, \ldots, C_K\}$ denotes the set of K clusters, and $W = [w_{ij}]$ denotes the partition matrix. If object $\mathbf{x}_i$ is allocated to cluster C_j, then w_{ij} is equal to 1; otherwise w_{ij} is equal to 0. Cluster center $\mathbf{c}_j$ is defined as

$$\mathbf{c}_j = \frac{1}{n_j}\sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i , \qquad (3)$$

where n_j denotes the number of the objects that belong to cluster C_j. As Equation (1) is highly nonlinear and multimodal, the clustering solution often falls into local minima. It is known that the clustering problem is NP-hard (Brucker, 1978). If exhaustive enumeration is used to solve this problem, then one requires evaluating these partitions.

$$\frac{1}{K!}\sum_{j=1}^{K}(-1)^{K-j}\binom{K}{j} j^N \qquad (4)$$

It is seen that exhaustive enumeration cannot lead to the required solution for most problems within reasonable computation time (Spath, 1980).

As local iteration methods tend to converge to local optima, researchers adopted some metaheuristic techniques to solve the clustering problem. Laszlo and Mukherjee (2006) presented a genetic algorithm for evolving the cluster centers in the *k*-means algorithm. The set of the cluster centers is represented using a hyper-quadtree constructed on the data. Chang et al. (2009) reported a clustering method based on genetic algorithm with gene rearrangement (GAGR). Their method employs roulette wheel selection, path-based crossover, and adaptive mutation to deal with the clustering problem. Güngör and Ünler (2007) proposed a simulated annealing based *k*-harmonic means clustering algorithm. Their method is superior to the *k*-means algorithm and the *k*-harmonic means algorithm in most cases. Bandyopadhyay et al. (2001) implemented a simulated annealing clustering method called simulated annealing with *k*-means based clustering (SAKMC). In the SAKMC method, the *k*-means algorithm is used to modify the cluster centroids. By redistributing objects among clusters probabilistically, their approach obtains better results than the *k*-means algorithm. Al-sultan (1995) employed the string-of-group-numbers encoding and proposed a tabu search clustering algorithm (TSC). After a specified number of iterations, the best obtained solution is viewed as the clustering result. Sung and Jin (2000) presented a heuristic algorithm to partition data sets with nonoverlapping clusters by combining tabu

```
Begin
    initialize parameters and establish the initial population
    while (not termination-condition) do
    while $i \le P$ do
    if (cat $X_i$ is assigned to seeking mode $M_s$) then
    perform seeking mode $M_s$
    else
    perform tracing mode $M_t$
    end if
    end do
    reassign cats and update the best known cat $X_b$
    end do
    output cat $X_b$
    end
```

Figure 1. General description of the KSACSOC algorithm

search and two functional procedures, packing procedure and releasing procedure. Their method uses the generalized string property to group similar objects together and set up the initial solution. Then the releasing procedure separates packed elements from each other so as to promote the effectiveness of the solution search. Liu et al. (2008) designed a tabu search clustering method called TS-Clustering to solve the clustering problem. In the TS-Clustering method, three neighborhood modes are used to establish neighboring solutions. In addition, two indicators, average objective function value and diversity of neighboring solutions, are given to evaluate the neighborhood of the TS-Clustering method. Cohen and de Castro (2006) proposed a particle swarm optimization clustering algorithm which employs the concept of sociocognition and incorporates the self-organizing term. Computer simulations show that their algorithm can output better results than the *k*-means algorithm. Jarboui et al. (2007) reported a clustering approach based on the combinatorial particle swarm optimization algorithm (CPSO). In the CPSO method, each particle is represented as a string of length *n* (where *n* is the number of objects) and the *i*th element of the string denotes the group number assigned to object *i*. The CPSO algorithm obtains better results than a genetic clustering method in some cases. Shelokar et al. (2004) proposed an ant colony optimization method for grouping *N* objects into *K* clusters. The presented method employs distributed agents who mimic the way real ants find the shortest path from their nest to food source and back.

After reviewing the related work, we found that it is necessary to develop the cat swarm optimization clustering algorithm, explore the capability of cat swarm optimization to deal with the clustering problem, and to further improve the performance of the cat swarm optimization clustering method by combining cat swarm optimization with *k*-means algorithm and simulated annealing. Our motivation is to introduce cat swarm optimization to clustering analysis, demonstrate the feasibility and effectiveness of the KSACSOC algorithm for the clustering problem under consideration, and to provide a new way to handle the clustering problem.

MATERIALS AND METHODS

Cat swarm optimization clustering (KSACSOC) algorithm

The KSACSOC algorithm observes the architecture of cat swarm optimization, integrates one-step *k*-means algorithm in seeking mode to improve the convergence of the clustering method, and hybridizes simulated annealing in tracing mode to avoid being trapped in local minima. Figure 1 gives the general description of the KSACSOC algorithm. In Figure 1, X_i denotes the *i*th solution, X_b denotes the best known solution, M_s denotes the seeking mode, M_t denotes the tracing mode, and P denotes the population size. Here, cat i is denoted by its position X_i representing the *i*th clustering solution. In this study, the clustering solution is made up of real numbers representing the coordinates of cluster centers. Then the length of the solution is $K \times m$, where K is the number of clusters and m is the number of object attributes. The first m elements denote the m dimensions of the first cluster center, the next m elements represent those of the second cluster center, and so on. For instance, let $m = 2$ and $K = 3$, then the solution (3.7 4.8 6.5 2.9 2.5 4.7) represents the coordinates of three cluster centers {(3.7 4.8) (6.5 2.9) (2.5 4.7)}. For initializing solution X_i, we randomly choose K distinct objects from the data set and view them as the initial cluster centers. Subsequently, this study will report the design approaches in detail.

Assignment of cats

In initialization stage, cats are randomly assigned between seeking mode and tracing mode. That is, we randomly select P_s cats into the seeking mode and set P_t cats into the tracing mode. Here, P_s and P_t denote the number of cats belonging to the seeking mode and the tracing mode, respectively. They are defined as

$$\begin{cases} P_t = \lceil R_{mr} \times P \rceil \\ P_s = P - P_t \end{cases}, \qquad (5)$$

where R_{mr} denotes the mixture ratio used to tune the number of cats in two modes. It is known that cats often spend most of their time resting and observing their environment. If they decide to move while resting, the movement is done carefully and slowly. This behavior is represented by the seeking mode. The tracing mode models the chasing of cats for a target. Cats spend very little time chasing things as this leads to over use of energy resources. Hence to guarantee that cats spend most of their time resting and observing, that is, most of the time is spent in the seeking mode, R_{mr} is allocated a small value.

Seeking mode

This mode is used to model the cat during a period of resting but being alert, looking around its environment for next move. In cat swarm optimization, four factors are given: seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self position consideration (SPC). Given solution X_l , $l = 1, \ldots, P_s$, its neighboring solution X_r^l is created by randomly plus or minus SRD percents CDC dimensions of solution X_l , where $r = 1, \ldots, N_{SM}$ and N_{SM} denotes the size of SMP, that is, N_{SM} denotes the number of neighboring solutions. Then one neighboring solution will be selected to renew solution X_l (Chu and Tsai, 2007). In order to combine cat swarm optimization with the clustering problem under consideration, incorporate the domain knowledge in the clustering procedure, and to further improve the performance of the cat swarm optimization clustering method, we first establish neighboring solutions of solution X_l , then design *k*-means improvement to fine-tune these neighboring solutions, finally employ proportional selection to choose a neighboring solution to update solution X_l . The seeking mode is stated as follows:

Step 1: Creation of neighboring solutions. Given solution X_l , $l = 1, \ldots, P_s$, and the number of neighboring solutions N_{SM} , cluster C_j to be modified is first randomly chosen, then object $\mathbf{x}_i$ belonging to cluster C_j is selected in random as the new cluster center of cluster C_j , finally all objects are reassigned to their respective nearest clusters. In this way, neighboring solution X_r^l is created. We continue this process until producing all neighboring solutions of solution X_l .

Step 2: *K*-means improvement. After establishing neighboring solutions of solution X_l , we adopt *k*-means improvement implemented by one-step *k*-means algorithm to tune the distribution of objects among different clusters and improve the performance of neighboring solutions. This method is stated as: Given neighboring solution X_r^l , reassign object $\mathbf{x}_i$ to cluster C_j if and only if the following condition holds:

$$\| \mathbf{x}_i - \mathbf{c}_j \|^2 < \| \mathbf{x}_i - \mathbf{c}_k \|^2 \quad , \qquad (6)$$

where $i = 1, \ldots, N$, $j, k = 1, \ldots, K$, and $j \neq k$. After all objects are reassigned, the new cluster centers $\mathbf{c}'_1, \ldots, \mathbf{c}'_K$ will be

$$\mathbf{c}'_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i \quad , \qquad (7)$$

where n_j denotes the number of the objects belonging to cluster C_j . After *k*-means improvement, the modified solution is viewed as neighboring solution X_r^l. This process continues until each neighboring solution of solution X_l is modified.

Step 3: Update of solution X_l . In order to renew solution X_l , we employ proportional selection, a genetic operator in genetic algorithms, to determine the candidate solution in this article. The probability of choosing neighboring solution X_r^l is defined as

$$p_r^l = \sum_{u=1}^{N_{SM}} f(X_u^l) \Big/ f(X_r^l) \quad . \qquad (8)$$

That is, the lower the objective function value of neighboring solution X_r^l , the more likely it is selected as the candidate solution, and vice versa. After neighboring solution X_r^l is chosen as the candidate one, we replace solution X_l with neighboring solution X_r^l and return the updated solution X_l .

Tracing mode

Tracing mode is designed to model the case of the cat in tracing targets. Once a cat goes into the tracing mode, it moves according to its velocities for each dimension. In cat swarm optimization, the main aim of the tracing mode is to update positions and velocities of cats. In this study, to maintain diversified population and promote the exploration of the unvisited space, we integrate simulated annealing into the tracing mode as the selection criterion. By accepting some bad solutions in the current population to next population according to the simulated annealing selection criterion, the tracing mode helps to avoid the solution search trapping in local minima. The tracing mode is described as follows:

Step 1: Update of velocity. Given cat X_i , $i = 1, \ldots, P_t$, velocity V'_i of its child X'_i is updated as

$$v'_{ij} = v_{ij} + r_1 \times c_1 \times (x_{bj} - x_{ij}) \quad , \qquad (9)$$

where $j = 1, \ldots, K \times m$, r_1 is a random value in the range of $[0, 1]$, c_1 is a constant for extending the velocity of the cat to move in the solution space , and x_{bj} and x_{ij} denote the *j*th elements of the best known solution X_b and solution X_i , respectively. Here, c_1 is set to be 2.

Step 2: Renewal of position. Update child X'_i of solution X_i as

$$x'_{ij} = x_{ij} + v'_{ij} \quad . \qquad (10)$$

Step 3: Simulated annealing selection. Given solution X_i and its child X'_i , the simulated annealing selection is performed as

$$\begin{cases} p_i^s = 1 & \text{if } f(X'_i) \leq f(X_i) \\ p_i^s = e^{-\frac{f(X'_i) - f(X_i)}{T}} & \text{if } f(X'_i) > f(X_i) \end{cases} \quad , \qquad (11)$$

where p_i^s denotes the survival probability of child X'_i , $f(X_i)$ denotes the objective function value of solution X_i , $f(X'_i)$

Table 1. Clustering results of different mixture ratios.

Mixture ratio	Avg	SD	Min	SR (%)	Time (s)
0.02	488.0213	0	488.0213	100	49.69
0.04	488.0213	0	488.0213	100	45.49
0.06	488.0213	0	488.0213	100	39.56
0.08	488.0213	0	488.0213	100	34.24
0.1	488.0213	0	488.0213	100	32.30
0.2	488.0231	0.0082	488.0213	95	40.41
0.3	488.0296	0.0169	488.0213	80	43.49
0.4	488.0296	0.0169	488.0213	80	48.18
0.5	488.0363	0.0264	488.0213	75	56.37

denotes the objective function value of child X_i', and T denotes the annealing temperature.

Reassignment of cats

After seeking mode and tracing mode are performed, cats are reassigned between these two modes. Here, we randomly select some cats into the tracing mode according to mixture ratio R_{mr}, then set the others into the seeking mode. The reassignment of cats is described as follows.

Step 1: Given the population after seeking mode and tracing mode, set $i=1$.

Step 2: Cat X_i is randomly assigned into seeking mode or tracing mode according to mixture ratio R_{mr}.

Step 3: If $f(X_i)<f(X_b)$, then $X_b=X_i$ and $f(X_b)=f(X_i)$. Set $i=i+1$. If $i\le P$, then go to Step 2, otherwise return the reassigned population, the best known cat X_b, and its objective function value $f(X_b)$.

RESULTS

In this paper, computer simulations were conducted in Matlab on an Intel Core 2 Duo processor running at 3 GHz with 4 GB real memory. We first evaluate the impact of the elements of the KSACSOC algorithm, and then conduct performance comparison between the proposed method and some known clustering methods for two artificial and six real life data sets. Each experiment includes 20 independent trials.

Performance evaluation

Here, we discuss the choice of different parameters and operations in order to explore the good performance of the KSACSOC algorithm. An artificial data set, Data-52 with five overlapping clusters, is adopted to illustrate the experimental results. We first consider the mixture ratio R_{mr} which was used to tune the number of cats in seeking mode and tracing mode. Here, the *k*-means improvement and the simulated annealing selection are used. Different mixture ratios are compared as shown in Table 1. Five indicators were used to evaluate these mixture ratios. The first three indicators are the average (Avg), standard deviation (SD), and minimum (Min) values of the clustering results. In this experiment, each mixture ratio can attain the minimum objective function value 488.0213. Among all mixture ratios, the ones less than 0.2 provide the minimum value in each trial. To explore the ability of mixture ratios to achieve the minimum value, we adopt another indicator, success rate (SR), which is defined as the number of trials where the best result was obtained divided by the number of the total trials. In addition, the average run time when the best result is firstly attained was employed to show the convergence speed of different mixture ratios. We found that too large or too small mixture ratios led to the increase of the run time. In addition, too large mixture ratios result in the decrease of the success rate. When $R_{mr}=0.1$, it requires less run time than the others to find the minimum value. Therefore, the value of the mixture ratio is chosen to be 0.1.

In this paper, *k*-means improvement was designed to fine-tune the distribution of objects, improve the similarity between objects and their cluster centroids, and to enhance the speed of convergence of the clustering algorithm. In order to explore the ability of the *k*-means improvement to promote the performance of the clustering algorithm, we define no improvement which means there was no modification on the neighboring solutions and they were directly used for the candidate solution selection. Here, the simulated annealing selection was used. Figure 2 shows that the no improvement was much inferior to the *k*-means improvement in terms of the convergence speed. The clustering results of these two improvement methods are shown in Table 2. Please note that in terms of the run time, although the *k*-means improvement

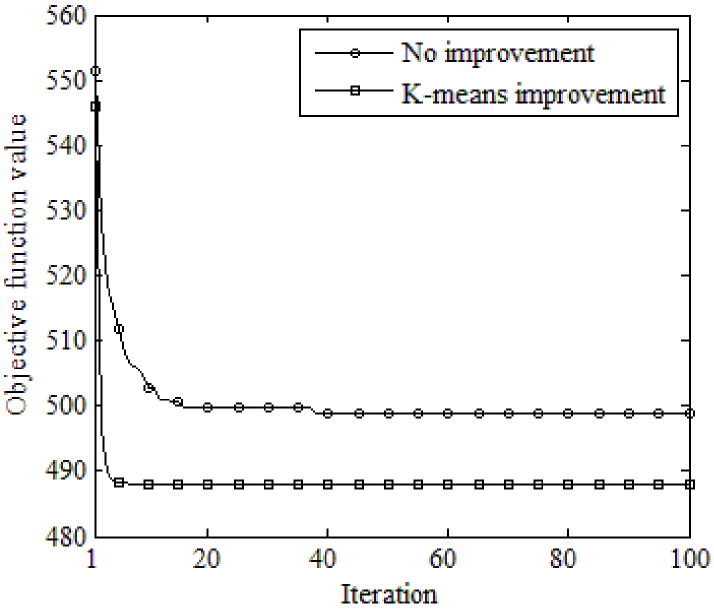

Figure 2. Comparison of two improvement methods

Table 2. Clustering results of two improvement methods.

Improvement mode	Avg	SD	Min	SR (%)	Time (s)
No improvement	498.9382	14.0178	488.0213	15	5.47
K-means improvement	488.0213	0	488.0213	100	32.30

requires more time than the no improvement to find the minimum value, it finds the best result in each trial while the latter only outputs this value in 3 of 20 runs. As a result, equipped with the *k*-means improvement in the seeking mode, the performance the KSACSOC algorithm can be greatly improved in terms of the success rate.

Finally, we consider the population renewal in the tracing mode. Here, our aim is to keep diversified population and promote the exploration of the unvisited space by integrating simulated annealing into the tracing mode as the selection criterion. Like no improvement in the seeking mode, we define no selection which means there is no operation on the parent population. In this way, the parent population is only used to establish the child population in the tracing mode as cat swarm optimization does in the field of function optimization. Here, the *k*-means improvement is used. Two selection methods are compared as shown in Figure 3. With the cooperation of the *k*-means improvement, the simulated annealing selection further helps the clustering algorithm to find the best result in each trial as shown in Table 3. Moreover, we find the phenomenon that most contributions result from the *k*-means improvement while the simulated annealing selection makes fewer contributions than the former. How to increase contributions from the simulated annealing selection and keep a good balance between these two operations in terms of the contribution rate will be our focus in the future research.

Performance comparison

Here, the KSACSOC algorithm was applied to two artificial and six real life data sets and compared with *k*-means algorithm, SAKMC algorithm, and CPSO algorithm. All real life data sets are available at http://ftp.ics.uci.edu/pub/machine-learning-databases/. Experimental data sets are described as follows:

(i) Data-52 data set ($N=250$, $m=2$, $K=5$), which consists of 250 overlapping objects where the number of clusters is five (Bandyopadhyay et al., 2001).
(ii) Data-62 data set ($N=300$, $m=2$, $K=6$), which consists of 300 nonoverlapping objects where the number of clusters is six (Bandyopadhyay et al., 2001).
(iii) Crude oil data set ($N=56$, $m=5$, $K=3$), which consists of 56 objects characterized by five features: vanadium, iron, beryllium, saturated hydrocarbons, and aromatic hydrocarbons. There are three crude-oil samples from three zones of sandstone: wilhelm, sub-mulnia,

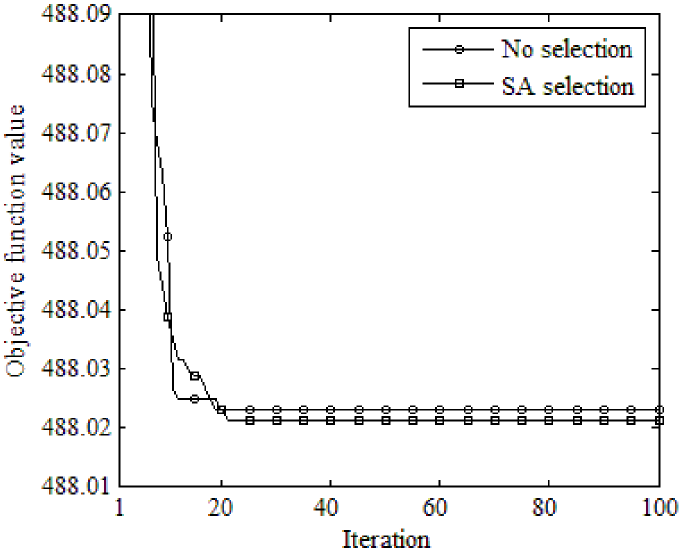

Figure 3. Comparison of two selection methods.

Table 3. Clustering results of two selection methods

Selection mode	Avg	SD	Min	SR (%)	Time (s)
No selection	488.0231	0.0082	488.0213	95	22.44
SA selection	488.0213	0	488.0213	100	32.30

and upper.

(iv) Fisher's iris data set ($N = 150$, $m = 4$, $K = 3$), which consists of three different species of iris flower: iris setosa, iris virginica, and iris versicolour. For each species, 50 samples with four features each (sepal length, sepal width, petal length, and petal width) are collected.

(v) Wine data set ($N = 178$, $m = 13$, $K = 3$), which consists of 178 objects characterized by 13 such features as alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and praline, are the results of a chemical analysis of wines brewed in the same region in Italy but derived from three different cultivars.

(vi) Ripley's glass data set ($N = 214$, $m = 9$, $K = 6$), which consists of six different types of glass: building windows float processed, building windows non-float processed, vehicle windows float processed, containers, tableware, and headlamps, each with 9 features, which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron.

(vii) Wisconsin breast cancer ($N = 683$, $m = 9$, $K = 2$), which consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data: malignant and benign.

(viii) Vowel data set ($N = 871$, $m = 3$, $K = 6$), which consists of 871 Indian Telugu vowel sounds. The data set has three features corresponding to the first, second, and third vowel frequencies and six overlapping classes.

The settings of parameters are described as follows: In the SAKMC algorithm, the initial temperature was set to be 100, the terminal temperature was set to be 0.01, and the temperature multiplier μ was set to be 0.05. In the CPSO algorithm, the population size is equal to 50, the number of generations is equal to 100, the inertia weight w is equal to 0.85, the parameter α for fitting intensification and diversification is equal to 0.35 while the acceleration constants c_1 and c_2 are equal to 0.3 and 1.2, respectively. The settings of above-mentioned parameters are recommended by their corresponding references. The detail descriptions of these parameters can be found in their corresponding references. In the KSACSOC algorithm, for a fair performance comparison, the initial annealing temperature and the temperature multiplier μ are the same as those in the SAKMC algorithm, the population size and the numbers of

Table 4. Clustering results of different experimental methods

Data set	Avg SD Min			
	K-means	**SAKMC**	**CPSO**	**KSACSOC**
Data-52	522.6443	488.0213	498.5200	488.0213
	146.5945	0	17.7208	0
	488.0912	488.0213	488.0912	488.0213
Data-62	1510.2811	1022.6890	656.4185	543.1716
	1311.1291	391.5944	248.2955	0
	543.1716	543.1716	543.1716	543.1716
Crude oil	1649.3574	1647.1893	1654.3407	1647.1893
	8.3428	0	13.7490	0
	1647.4434	1647.1893	1647.1893	1647.1893
Iris	88.5895	78.9408	79.3596	78.9408
	22.9646	0	0.8195	0
	78.9408	78.9408	78.9408	78.9408
Wine	2410119.5336	2370689.6868	2372361.5411	2370689.6868
	93861.8097	0	5036.8336	0
	2370689.6868	2370689.6868	2370689.6868	2370689.6868
Glass	379.5242	338.7449	386.9310	336.0705
	47.3381	0	44.8582	0.0436
	336.2686	338.7449	336.0605	336.0605
Breast cancer	19323.1816	19323.1738	19332.6619	19323.1738
	0.0135	0	22.1749	0
	19323.1738	19323.1738	19323.1738	19323.1738
Vowel	32317325.6990	32047019.4454	32018731.5261	30698026.8160
	1219278.0975	1429309.4308	1473238.1457	6603.5251
	30750872.9479	30724074.8212	30701698.2741	30688529.1096

generations are the same as those in the CPSO algorithm. In addition, with the increase of the number of neighboring solutions N_{SM}, the KSACSOC algorithm has more choices to select from but has to spend more computational effort in providing the correct result. Therefore, deciding the proper value for parameter N_{SM} is the process of exploring a balance between quality and cost in this article, the case that can reach our goal.

The average (Avg), standard deviation (SD), and minimum (Min) values of the objective function are shown in Table 4. Among experimental methods, the *k*-means method fails to find the minimum values for Data-52, Crude oil, Glass, and Vowel in each trial, which shows that it tends to fall into local minima. The SAKMC algorithm attains the minimum values for Data-52, Crude oil, Iris, Wine, and Breast cancer in all runs. In addition, its average values for experimental data sets are less than those provided by the *k*-means algorithm. The CPSO algorithm obtains the best results for Data-62, Crude oil, Iris, Wine, Glass, and Breast cancer in some trials but does not report the minimum values for Data-52 and Vowel within the specified number of generations. The KSACSOC algorithm provides the minimum values in each trial in face of all data sets except Glass and Vowel. Considering Glass, we found that its average and minimum values are very close to each other. In face of Vowel, the proposed method outputs the standard deviation value far less than those of the other three methods.

The success rates of experimental methods are compared as given in Table 5 to show their capability to achieve the best results of different sets of data. Among experimental methods, the *k*-means algorithm provides

Table 5. Success rates of different experimental methods

Data set	SR (%)			
	K-means	SAKMC	CPSO	KSACSOC
Data-52	0	100	0	100
Data-62	15	40	80	100
Crude oil	0	100	55	100
Iris	50	100	25	100
Wine	85	100	85	100
Glass	0	0	10	95
Breast cancer	75	100	55	100
Vowel	0	0	0	5

Table 6. Run time of different experimental methods.

Data set	Time (s)			
	K-means	SAKMC	CPSO	KSACSOC
Data-52	-	10.59	-	87.19
Data-62	0.04	8.35	1.12	5.58
Crude oil	-	0.34	0.51	1.05
Iris	0.02	5.10	0.64	2.22
Wine	0.04	0.03	1.54	5.77
Glass	-	-	15.49	130.25
Breast cancer	0.08	8.07	3.45	7.58
Vowel	-	-	-	98.39

the lowest success rates in most cases. Considering the other three clustering methods based on metaheuristics, the CPSO algorithm is the worst, the SAKMC algorithm is the second, and the KSACSOC algorithm is the best. In all runs, the KSACSOC algorithm attains the ideal success rates except Glass and Vowel. It successfully finds the minimum value for Glass in 19 of 20 runs. But in face of Vowel, its success rate was only 5%.

The average run time when the minimum values are firstly attained by different methods is recorded as shown in Table 6. The symbol “-” denotes that the item does not exist. For example, in face of Data-52, the *k*-means algorithm fails to find the minimum value in all trials. Then this item is labeled as “-”. In all experiments, the *k*-means algorithm terminates much faster than the other three methods, but it falls into local minima in most cases. In face of Data-52, Crude oil, and Wine, the SAKMC algorithm outputs their minimum values sooner than the KSACSOC algorithm. But it cannot provide meaningful results for Glass and Vowel within the specified number of iterations. The KSACSOC algorithm finds the minimum values for Data-62, Iris, and Breast cancer faster than the SAKMC algorithm. In order to understand the performance of three metaheuristic clustering methods better, we use Glass to show the iteration process as shown in Figure 4. It is seen that the convergence trend of the SAKMC algorithm is the slowest but it can further attain better results than the CPSO algorithm with the increase of the number of iterations, the CPSO algorithm converges to its results faster than the SAKMC algorithm but traps in local minima in many cases, and the KSACSOC algorithm outputs the correct result within fewer iterations than the SAKMC algorithm and the CPSO algorithm.

DISCUSSION

In this study, we found that there are two important aspects that deserve particular attention in the future. One is to further accelerate the convergence speed of the KSACSOC algorithm under the condition that poor solutions may be accepted according to the simulated annealing selection criterion while the other is to further increase the contribution rate of the simulated annealing selection and achieve harmony between *k*-means improvement and simulated annealing selection. In addition, we will extend the principle of the KSACSOC algorithm to the case where the number of clusters is not known *a priori*. In this case, clustering indices such as Davis Bouldin index and Dunn’s index may be considered to evolve the number of clusters automatically.

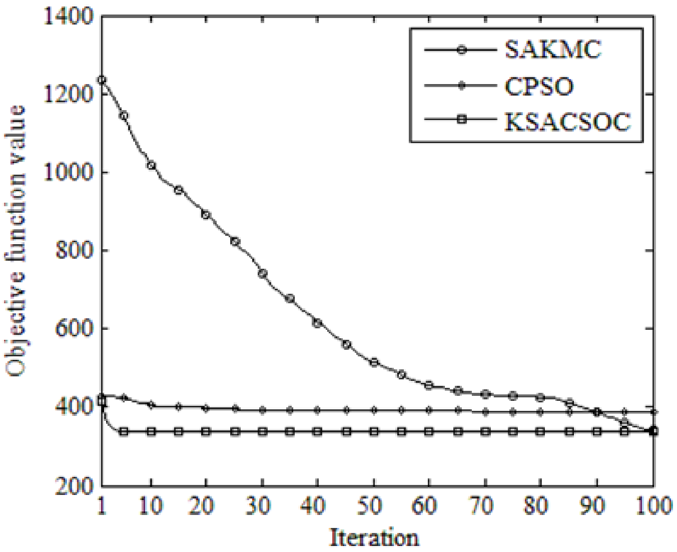

Figure 4. Comparison of three clustering methods for Glass.

Conclusions

As a fundamental problem and technique for data analysis, clustering has become increasingly important in many research fields. In this paper, a cat swarm optimization clustering method called KSACSOC was proposed to deal with the clustering problem. In the KSACSOC algorithm, *k*-means improvement based seeking mode was designed to improve the performance of neighboring solutions by tuning the object distribution among different clusters, and simulated annealing selection based tracing mode was developed to explore the unvisited solution space by accepting poor solutions probabilistically. As a result, the KSACSOC algorithm can output the best results for experimental data sets under the criterion of minimum sum of squares clustering and can also provide higher success rates than the *k*-means algorithm, the SAKMC algorithm, and the CPSO algorithm.

ACKNOWLEDGMENTS

This research was supported in part by the National Natural Science Foundation of China (NSFC) under grants 60903074 and 60828005, the National High Technology Research and Development Program of China (863 Program) under grant 2008AA01Z119, the National Basic Research Program of China (973 Program) under grant 2009CB326203, and the US National Science Foundation (NSF) under grant CCF-0905337.

REFERENCES

Al-sultan KS (1995). A tabu search approach to the clustering problem. Patt. Recognit. 28(9):1443-1451.

Bandyopadhyay S, Maulik U, Pakhira MK (2001). Clustering using simulated annealing with probabilistic redistribution. Int. J. Patt. Recognit. Artif. Intell. 15(2):269-285.

Brucker P (1978). On the complexity of clustering problems. Lect. Notes Econ. Math. Syst. 157:45–54.

Chang DX, Zhang XD, Zheng CW (2009). A genetic algorithm with gene rearrangement for k-means clustering. Patt. Recognit. 42(7):1210-1222.

Chu SC, Tsai PW (2007). Computational intelligence based on the behavior of cats. Int. J. Innov. Comp. Inf. Control. 3(1):163-173.

Cohen SCM, de Castro LN (2006). Data clustering with particle swarms. In: Proceedings of IEEE Congress on Evolutionary Computation, BC, Canada pp. 1792-1798.

Güngör Z, Ünler A (2007). *K*-harmonic means data clustering with simulated annealing heuristic. Appl. Math. Comput. 184(2):199-209.

Hammerly G, Elkan C (2002). Alternatives to the k-means algorithm that find better clusterings. In: Proceedings of International Conference on Information and Knowledge Management, Virginia, USA pp. 600-607.

Jarboui B, Cheikh M, Siarry P, Rebai A (2007). Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. Appl. Math. Comput. 192(5):337-345.

Laszlo M, Mukherjee S (2006). A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering. IEEE Trans. Patt. Anal. Mach. Intell. 28(4):533-543.

Liu YG, Yi Z, Wu H, Ye M, Chen KF (2008). A tabu search approach for the minimum sum-of-squares clustering problem. Inf. Sci. 178(12):2680-2704.

Omran MGH, Engelbrecht AP, Salman A (2007). An overview of clustering methods. Intell. Data Anal. 11(6):583-605.

Pedrycz W (2005). Knowledge-based clustering. Wiley, New Jersey.

Selim SZ, Ismail MA (1984). K-means-type algorithm: generalized convergence theorem and characterization of local optimality. IEEE Trans. Patt. Anal. Mach. Intell. 6(1):81-87.

Shelokar PS, Jayaraman VK, Kulkarni BD (2004). An ant colony approach for clustering. Anal. Chim. Acta 509(2):187-195.

Spath H (1980). Cluster analysis algorithms. Wiley, Chichester.

Sung CS, Jin HW (2000). A tabu-search-based heuristic for clustering. Patt. Recognit. 33(5):849-858.

Zhang B, Hsu M, Dayal U (1999). K-harmonic means - A data clustering algorithm. Technical Report HPL-1999-124, Hewlett-Packard Lab.