



# A new meta-heuristic algorithm based on chemical reactions for partitional clustering problems

Hakam Singh<sup>1</sup> · Yugal Kumar<sup>1</sup> · Sumit Kumar<sup>2</sup>

Received: 26 July 2018 / Revised: 29 December 2018 / Accepted: 10 March 2019 / Published online: 23 March 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

In the field of engineering, heuristic algorithms are widely adopted to solve variety of optimization problems. These algorithms have proven its efficacy over classical algorithms. It is seen that chemical reactions consist of an efficient computational procedure to design a new product. The formation of new product contains numbers of objects, states, events and well defined procedural steps. A meta-heuristic algorithm inspired through chemical reaction is developed, called artificial chemical reaction optimization (ACRO) algorithm. In this work, an ACRO algorithm is adopted to solve partitional clustering problems. But, this algorithm suffers with slow convergence rate and sometimes stuck in local optima. To handle these aforementioned problems, two operators are inculcated in ACRO algorithm. The performance of proposed algorithm is tested over well-known clustering datasets. The simulation results confirm that proposed ACRO algorithm is an effective and competitive algorithm to solve partitional clustering problems.

**Keywords** Artificial chemical reaction optimization · Clustering · Meta-heuristic algorithms · Chemical reaction

## 1 Introduction

Clustering is an unsupervised technique that can be widely adopted for data analysis. In clustering, there is no need of training the data. This technique adopts a distance measure to compute data objects in clusters. In present time, clustering can get wide attention from research community both of theoretical and practical point. This technique is widely adopted in diverse fields such as process monitoring [1], image processing [2], feature selection [3], gene expression and micro array data [4], e-learning [5] and healthcare [6] etc. Through literature, it is found that large numbers of meta-heuristic algorithms have been presented to obtain

optimal solution for clustering problems. Such algorithms are based on some natural phenomena, laws of physics, thermodynamics etc., behavior of swarms, insect, animals etc. The algorithms based on physics and thermodynamics laws are charged system search algorithm (CSS) [7, 8] and magnetic charged system search algorithm (MCSS) [9, 10]. The algorithms based on natural phenomena's are black hole (BH) [11] and galaxy based algorithm [12]. The algorithms based on the behavior's of bird, insect and animal are artificial bee colony algorithm (ABC), ant colony optimization (ACO), bat algorithm (BA) etc. In present time, these algorithms are more popular than classical algorithms and widely adopted by researchers to solve diverse optimization problems due to following reasons.

- Having different solution strategies.
- Less dependency on size of the problem, solution space, constraints and variables.
- Adapt itself according to problem domain.
- Effective mechanism to solve combinatorial and non linear problems
- Avoid local optima problem and less computation power.
- Search the solution space with different initial points.

✉ Hakam Singh  
hakamsingh011@gmail.com

Yugal Kumar  
yugalkumar.14@gmail.com

Sumit Kumar  
sumitkumarbsr19@gmail.com

<sup>1</sup> Department of Computer Science and Engineering,  
Jaypee University of Information Technology, Wanknaghat,  
Himachal Pradesh, India

<sup>2</sup> Department of Computer Science and Engineering, Amity  
School of Engineering, Amity University, Noida, India

Due to aforementioned advantages, these algorithms have been extensively applied in diverse fields such as image processing, biology, sciences, market research, biomedical process and engineering. Moreover, the improved versions of these algorithms have also been presented in literature [13].

Recently, an algorithm inspired through chemical reaction is reported in the literature to find the optimal solutions for global optimization problems [14, 15]. This algorithm is based on the chemical reactions like synthesis, decomposition, metathesis etc. The population of ACRO algorithm is denoted using reactants. The various encoding schemes like integer, character, real and float can be used to denote reactants and further, to solve the optimization problems. The different chemical reactions are applied to evaluate and generate new reactants during the execution of algorithm. The objective of this work is to evaluate the efficiency of ACRO algorithm to solve the partitioned clustering problems. Moreover, to improve the convergence rate of ACRO algorithm, some modifications are incorporated in it. The performance of proposed algorithm is tested on benchmark clustering datasets and experimental results are compared with state of art clustering algorithms reported in literature. The rest of paper is organized as. The ACRO algorithm is presented into sub Sect. 1.1. Section 2 presents the ACRO algorithm for partitioned clustering problems. Section 3 presents different clustering algorithms used for comparison. The simulation results of proposed algorithm are described in Sect. 4. Section 5 summarizes the entire work.

### 1.1 Artificial chemical reaction optimization (ACRO) algorithm

The motivation behind the ACRO algorithm is chemical reaction process. It is observed that atoms and molecules constantly move and collide in a viscous fluid. Atoms can be described as the elementary particle of a chemical reaction and having a type, charge, mass, position, radius, orientation, and velocity. Whereas, a molecule can be defined as a group of two or more atoms linked through bonds. During a chemical reaction several changes occur like make and break of bonds, change in orientation, substitution and replacement in molecules. It is considered that ACRO algorithm begins with the action of reactants in a vessel. Let us consider a fixed volume vessel filled with a uniform mixture of  $N$  chemical reactants which undergo different types of chemical reactions. Let  $R_i \in (1 \leq i \leq N)$  is the list of chemical types, and it is considered that these types can cooperate through  $M$  specified chemical reaction channels. In ACRO algorithm, the encoding scheme of the reactants is depending on the user choice and it can be binary, real, string, integer, etc. The new reactant can be produced through the interaction of one or two reactants. The ACRO algorithm starts with set of initial reactants in a solution. Further, reactants are

consumed and produced through different chemical reactions. The algorithm is stopped; either the termination condition is met or similar to a state where no more reaction can take place. For the chemical reactions, reactants are chosen based on their concentrations and potentials. Moreover, it is observed that consecutive and competing reactions are the two types of reactions that are mainly reported. Reactants are joined together serially in a consecutive reaction whereas in competing type reactions, different products occur based on the specific condition. It is also seen that sometimes output of one reaction can be a reactant of another reaction. There are many factors that can affect the execution of a reaction. However, ACRO algorithm is based on the simple concept of an equal probability of monomolecular or bimolecular reactions and their alternatives. The main steps of the ACRO algorithm are listed below.

Step 1: Initialize the problem and user defined parameters of algorithm.

Step 2: Evaluate the initial reactants and evaluation.

Step 3: Apply chemical reactions.

Step 4: Update reactants.

Step 5: Termination condition.

## 2 Proposed work: ACRO for clustering problems

This section presents ACRO algorithm to determine optimal cluster centers for partitioned based clustering problems. In partitioned based clustering problems, the main task is to find the optimum set of cluster centers with minimum intra cluster distance. Prior to apply the ACRO algorithm for partitioned clustering problems, two modifications are done in proposed algorithm. The aim of these modifications is to handle slow convergence rate and local optima problems. To achieve the same, two operators i.e. position based operator and neighborhood operator are incorporated in ACRO algorithm. The position based operator is incorporated in synthesis reaction step. Whereas, for inter reactant collision i.e. tradeoff between different reactants, a neighborhood operator is proposed. The main clustering steps of ACRO algorithm are mentioned in sub Sect. 3.2.

### 2.1 Proposed modification in ACRO algorithm

To make the ACRO algorithm more effective and improve the quality of results, two modifications are proposed in ACRO algorithm. These modifications can be summarized as.

### 2.1.1 Position based operator

In this work, a position based operator is proposed in synthesis reaction step. The aim of this operator is to generate new reactant. In synthesis reaction, a new reactant is using Eq. 9. The working of position based operator is given as. Suppose,  $R_i$  and  $R_j$  are two reactants and generate a new reactant  $R_{i,new}$ . The elements in  $R_i$  and  $R_j$  have equal opportunity to select and pass to same position in new reactant  $R_i$ . Each element in  $R_i$  has fifty percent probability to select and passes to same position in reactant  $R_{i,new}$ . The rest of position  $R_{i,new}$  are filled through reactant  $R_j$ . For example,  $R_i = [7, 8, 3, 9, 1, 10, 2]$  and  $R_j = [11, 14, 2, 5, 8, 4, 1]$ , the new reactant  $R_{i,new}$  can be generated by combining the reactants  $R_i$  and  $R_j$ . Suppose, the elements 7, 3, 1, 2 are selected randomly from reactant  $R_i$  and rest of elements are selected from reactant  $R_j$ . The resultant reactant  $R_{i,new}$  can be expressed as  $R_{i,new} = [7, 14, 3, 5, 1, 4, 2]$ .

### 2.1.2 Neighborhood operator

For better inter reactant collision, i.e. better tradeoff between reactants, a concept of neighborhood operator is inculcated in ACRO algorithm. The aim of neighborhood operator is to overcome the local optima problem. To handle the local optima problem, two elements of current reactant are selected and inter change the position of these. In turn, a new reactant is generated which is different from previous reactant. For example, the position of current reactant can be given as  $R_i = [2, 3, 5, 6, 9, 4]$ . We can select two elements from the current reactant. The selected elements are five and eight. Now, the positions of the elements are interchanged each other. The newly generated reactant is [2–5, 7, 8, 16] and it different from previous reactant.

## 2.2 Steps of proposed ACRO clustering algorithm

Step 1: Initialization of user defined parameters and statement of the problem.

Clustering is a minimization problem and can be expressed as follows:

$$\text{Minimize } f(x) \text{ subject to } y_i \in C_k; \text{ where } i = 1, 2, 3, \dots, \\ D \text{ and } k = 1, 2, \dots, K$$

where  $f(x)$  is an objective function which is defined in terms of Euclidean distance;  $y_i$  is the set of data objects;  $D$  is the dimension of data object,  $C_k$  is the set of  $k^{\text{th}}$  cluster center and  $K$  describes number of clusters in a given data set. It is also noted that the computed cluster centers should be

in the range of  $y_i^{\min}$  and  $y_i^{\max}$  and also known as boundary constraints. Further, in this work, integer encoding scheme is used to obtain the desired results. The user defined parameters of ACRO algorithm and ReacNum i.e. maximum number of iteration is also defined in this step.

Step 2: Set initial reactants and evaluation.

In this step, the population of the ACRO algorithm is specified; the population is defined in terms of initial reactants. The initial reactants are uniformly identified from the feasible search space. For clustering problem, initially, two reactants i.e.  $R_1$  and  $R_2$  are identified from dataset such as  $R_1 = \{y_{i,1}, y_{i,2}, \dots, y_{i,d}\}$ ,  $R_2 = \{y_{j,1}, y_{j,2}, \dots, y_{j,d}\}$ , where,  $d$  denotes the length (dimension) of reactant. The number of reactants is similar to number of clusters ( $K$ ). Then, rests of reactants are derived from initially determined reactants  $R_1$  and  $R_2$  using following procedure. To compute  $R_1$  and  $R_2$ , firstly, a dividing factor ( $k$ ) is initialized. Suppose,  $k = 2$ ; then, two new extra reactants i.e.  $R_3$ ,  $R_4$  are generated from  $R_1$  and  $R_2$  using below mentioned procedure

$$R_3 = \left\{ \begin{array}{l} r * y_{i,1}, r^* y_{i,2}, \dots, r^* y_{i,\frac{d}{2}}; \\ r^* y_{j,\frac{d}{2}+1}, r^* y_{j,d-\frac{1}{2}+2}, \dots, r^* y_{j,1} \end{array} \right\} \quad (1)$$

$$R_4 = \left\{ \begin{array}{l} r^* y_{j,1}, r^* y_{j,2}, \dots, r^* y_{j,\frac{d}{2}}; \\ r^* y_{i,\frac{d}{2}+1}, r^* y_{i,d-\frac{1}{2}+2}, \dots, r^* y_{i,1} \end{array} \right\} \quad (2)$$

where,  $r$  denotes a random number in the range of  $0 < r < 1$ . Further, the more reactants are generated using the following procedure until, the number of reactants are not similar to the desired clusters ( $K$ ).

$$R_5 = \left\{ \begin{array}{l} r^* y_{i,1}, r^* y_{i,2}, \dots, r^* y_{i,\frac{2d}{3}}; \\ r^* y_{j,2d/3+1}, r^* y_{j,2(d-1)/3+2}, \dots, r^* y_{j,1} \end{array} \right\} \quad (3)$$

$$R_6 = \left\{ \begin{array}{l} r^* y_{i,1}, r^* y_{i,2}, \dots, r^* y_{i,\frac{d}{3}}; r^* y_{j,d/3+1}, r^* y_{j,2(d-1)/3+2}, \\ r^* y_{i,\frac{2d}{3}}, \dots, r^* y_{j,1} \end{array} \right\} \quad (4)$$

$$R_7 = \left\{ r^* y_{i,1}, r^* y_{i,2}, \dots, r^* y_{i,\frac{d}{3}}; r^* y_{j,d/3+1}, \dots, r^* y_{j,1} \right\} \quad (5)$$

$$R_8 = \left\{ r^* y_{j,1}, r^* y_{j,2}, \dots, r^* y_{j,\frac{d}{3}}; r^* y_{i,\frac{d}{3}}, \dots, r^* y_{i,1} \right\} \quad (6)$$

$$R_9 = \left\{ r^* y_{j,1}, r^* y_{j,2}, \dots, r^* y_{j,\frac{d}{3}}; r^* y_{i,\frac{d}{3}+1}, r^* y_{i,\frac{2(d-1)}{3}}, \dots, r^* y_{i,1} \right\} \quad (7)$$

$$R_{10} = \left\{ r^*y_{j,1}, r^*y_{j,2}, \dots, r^*y_{j,\frac{2d}{3}}; r^*y_{i,\frac{2(d-1)}{3}+1}, r^*y_{i,\frac{2d}{3}+2}, \dots, r^*y_{i,1} \right\} \quad (8)$$

Step 3: Applying chemical reactions.

Step 3.1: Bimolecular reactions.

Let us consider  $R_1 = \{y_{i,1}, y_{i,2}, \dots, y_{i,d}\}$  and  $R_2 = \{y_{j,1}, y_{j,2}, \dots, y_{j,d}\}$  are two reactants that can participate in bimolecular reaction. As a result of this, various bimolecular reactions are incorporated in ACRO algorithm such as synthesis reaction, displacement reaction, redox2 reaction, mono-molecular reaction. For all these operations, integer representation of population is considered rather than binary encoding. The detailed explanation of these reactions is described as below.

Step 3.2: Synthesis reaction: Using this reaction, a new reactant can be obtained using following equation

$$R = (r_1, r_2, r_3, \dots, r_i, r_j, r_k, \dots, r_n); \text{ where, } R_{i,\text{new}} = R_i + \lambda_i(R_j - R_i) \quad (9)$$

In the above equation,  $\lambda_i$  is a random value in the interval  $[0.25, 1.25]$ ;  $R_i$  and  $R_j$  is randomly selected reactant. Further, a position based operator is proposed in this step to generate new reactant.

Step 3.3: Displacement reaction.

The new reactants are obtained using below mentioned procedure. Suppose,  $R_k = (R_1, R_2, \dots, R_i, R_j, \dots, R_K)$  where,  $k = 1, 2, \dots, K$ .

$$R_{i,\text{new}} = R_i(1 - \lambda_{td}R_j) \quad (10)$$

$$R_{j,\text{new}} = \lambda_{td}R_j + (1 - \lambda_{td}R_i) \quad (11)$$

where  $\lambda_{td} \in \{0, 1\}$  and  $\lambda_{td+1} = 2.3(\lambda_{td})^{2\sin(\pi\lambda_{td})}$ ; where, in  $\lambda_{td}$  the suffix 'td' is increased by 1, when reaction is completed.

Step 3.4: Redox2 reaction.

In this reaction, if  $R_1$  is better reactant in terms of objective function, then in  $\lambda_{tr}$  the suffix 'tr' is updated by using one when the reaction is performed and it is computed using following equation.

$$R_{i,\text{new}} = R_i + \lambda_{tr}(R_i - R_j); \text{ where, } \lambda_{tr} \in \{0, 1\}. \quad (12)$$

Otherwise,  $R_2$  is better reactant, then in  $\lambda_{tr}$ , the suffix 'tr' is updated by using 1 when the reaction is performed and it is computed using following equation.

$$R_{j,\text{new}} = R_j + \lambda_{tr}(R_j - R_i); \text{ where, } \lambda_{tr} \in \{0, 1\} \quad (13)$$

The value of  $\lambda_{tr}$  is computed using following equation.

$$\lambda_{tr+1} = \begin{cases} 0 & \lambda_{tr} = 0 \\ \frac{0\lambda_{tr}}{\lambda_{tr,\text{mod}}(1)}\lambda_{tr} & \in (0, 1) \end{cases} \quad (14)$$

$$1/\lambda_{tr,\text{mod}}(1) = \frac{1}{\lambda_{tr}} - \frac{1}{\lambda_{tr}} \quad (15)$$

Step 3.4: Monomolecular reactions.

Step 3.4.1: Decomposition reaction.

In this reaction, suppose  $R = (R_1, R_2, \dots, R_i, R_j, \dots, R_n)$  be the reactant and  $C_k \in \{R_m, R_n\}$  be an atom that can take part in monomolecular reaction. The new atom of the molecule  $R_{i,\text{new}}$  is a random population or reactant from  $\{R_m, R_n\} \in C_k$ , but  $\{R_m, R_n\} \neq (R_i \text{ and } R_j)$ .

Step 3.4.2: Redox1 Reaction.

In this reaction, the new reactant is generated using following procedure.

$$R_{i,\text{new}} = R_m + \lambda_t(R_n - R_m) \quad (16)$$

where,  $\lambda_t \in \{0, 1\}$  such that initial  $\lambda_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$  and  $\lambda_{t+1} = 4\lambda_t(1 - \lambda_t)$ ;  $t$  is updated by using 1 when reaction is performed.

Step 4: Reactants Update.

In this step, a chemical equilibrium test is carried out. If the fitness function of the new generated reactants is better than other, the generated reactant includes into the chemical reactions process and put into reactant pool. The worst reactants are excluded reactant pool and chemical process.

Step 5: Neighborhood operator.

In this step, neighborhood operator is used to generate new reactant and overcome the local optima problem.

Step 6: Termination Condition.

If Reactnum is equal to the user defined maximum number of iterations, then algorithm stops its execution and produces the optimal cluster centers, otherwise steps 3 and 4 are repeated, until the desired results are not obtained or termination condition is not satisfied".

## 2.3 Pseudo code of the ACRO clustering algorithm

This subsection presents the main steps of proposed ACRO clustering algorithm. These steps are listed in Algorithm 1.

**Algorithm 1: Proposed ACRO Algorithm**

- Step 1: Set the initial reactants and user defined parameters of ACRO algorithm.
- Step 2: Set the initial reactants and evaluate the new reactants using equations 1-8.
- Step 3: Randomly select the reactants ( $R_i$ ) from population pool and compute the objective function value using following equation.

$$\text{minimum}(R_i, R_p) = \sum_{i=1}^N \min \|R_i - R_p\|^2, p = 1, 2, 3, \dots, K \quad (17)$$

- Step 4: Arrange the data objects into different clusters and compute the fitness function for each reactant using equation 18.

$$\text{Fitness}(R_p) = \frac{\sum_{j=1}^K \text{SSE}(R_p)}{\sum_{j=1}^K \text{SSE}(R_p)} \quad (18)$$

- Step 5: Apply the chemical reactions phase of ACRO algorithm.
- Step 6: Apply the monomolecular reaction phase of ACRO algorithm.
- Step 7: Update the reactants using chemical equilibrium test.
- Step 8: Apply neighborhood operator to generate new reactant
- Step 9: Repeat the steps 3-8, until termination condition is not.
- Step 10: Compute the optimal cluster centers

**3 Other clustering algorithms****3.1 K-means algorithm**

In clustering field, K-means algorithm is most popular algorithm, developed by MacQueen [17]. K-means algorithm partitions  $n$  objects into  $k$  number of clusters that are optimal in terms of some criteria. In K-means algorithm, sum of Euclidean distance is taken as similarity measure and objects are assigned to clusters according to minimum Euclidean distance values. Several variants of K-means algorithm are also reported in literature. Zalik developed a variant of K-means algorithm that performs clustering without pre-assignment of number of clusters [18]. In this algorithm, the mean square error cost function is extended with new cost-function. To improve the quality of solution, an iterative K-mean approach is reported in [19]. In this work, the quality of solution is improved by removing one cluster (minus), dividing another one (plus) and applying re-clustering again, in each iteration. Author claimed that plus and minus can perform well with minimized sum of the squared Euclidean distance. Tzortzis and Likas applied MinMax method to overcome the poor initialization problem of k-means algorithm [20]. The MinMax method assigns weights to the clusters relative to their variance and optimize them. Further, MinMax k-means algorithm is improved using iterative procedure that enables the algorithm to automatic adopt the exponent of the datasets. Malinen et al.,

developed K-means\* algorithm based on inverse transformation [21]. Instead of fitting the clustering model to the data the K-means\* fit the data to an optimal clustering structure. Kang et al., hybridized the K-means algorithm with mussels wandering optimization algorithm to obtain new swarm intelligence-based method [22]. The proposed algorithm inherits the merits of both algorithm like local search ability of K-means and global optimization ability of MWO. Further the similarity-based clustering ensemble method and weighted-incorporated similarity-based clustering ensemble method are incorporated in the proposed algorithm.

**3.2 Ant colony optimization algorithm**

To optimize problem search space a meta-heuristic algorithm based on behavior of ants is proposed by Dorigo et al., in 1996, called ant colony optimization (ACO) [23]. The characteristics of ant system such as positive feedback for rapid discovery of good solution, distributed computation to avoid premature convergence and the use of a constructive greedy heuristic to accept solution in early stages of search process are adopted in this algorithm. Shelokar et al., adopted the ACO algorithm in clustering field to find optimal cluster centers [24]. In this work, a pheromone matrix is used to guide search towards optimal solution. Moreover, it is noticed that several variants of ACO algorithm are presented in literature for solving clustering problems. Wan et al., introduced chaotic ant swarm approach for partitional clustering [25]. The proposed algorithm is insensitive to clusters size

and density. Huang et al., hybridized continuous ant colony optimization algorithm with particle swarm optimization algorithm [26]. The hybrid ACO algorithm consists of four types of approaches such as sequence, parallel, sequence approach with an enlarged pheromone-particle table and global best exchange. The sequence approaches perform well with enlarged pheromone particle table as compared to other approaches. Menéndez et al., designed two medoids based ACO clustering algorithms [27]. Both algorithms employ the ant colony procedure to determine optimal cluster centers. The goal of Medoid seTACO algorithm is to choose the best medoids based on distance information, while the K-Adaptive Medoid seTACO algorithm is an extension of Medoid seTACO that automatically adjust the number of clusters.

### 3.3 Particle swarm optimization algorithm

PSO algorithm is based on the behaviors of birds flocking and fish schooling. Cura applied the PSO algorithm to optimize clustering search space problems [28]. The proposed algorithm is effective and applicable when the number of clusters is known or unknown. Chuang et al., designed an accelerated chaotic particle swarm optimization algorithm [29]. In this works, the chaotic mapping techniques (logistic maps) are combined with particle swarm optimization algorithm. To enhance the exploration ability of particle swarm optimization algorithm, it is hybridized with big bang big-crunch algorithm [30]. The aim of this hybridization is to overcome the premature convergence problem of PSO algorithm. Further, it is noted that to improve the performance of PSO algorithm, a cooperative co-evolution framework is reported for clustering [31]. The cooperative co-evolution method works in divide and conquers fashion. A hybrid PSO algorithm is presented for effective clustering in [32]. The PSO algorithm is combined with K-Harmonic means and improved cuckoo search algorithms. It is noticed that the proposed algorithm is quite insensitive to cluster center initialization and produces high quality clustering results. Moreover, Hatamlou and Hatamlou hybridized PSO with heuristic search algorithm to overcome the inadequacies of PSO algorithm [33]. In this work, the particle swarm optimization is used to produce an initial solution, while the heuristic search algorithm is used to improve quality of solution.

### 3.4 Cat swarm optimization algorithm

Cat swarm optimization algorithm is based on swarm intelligence of cats [34]. This algorithm works in two modes i.e. seeking mode and tracing mode. In seeking mode cat is in resting position but being alert, while in tracking mode cat is tracing some targets. Further, to improve the search ability of cat swarm optimization algorithm, Mohapatra et al., modified CSO algorithm using mutation operator [35]. The mutation

operator is used to obtain best cat position in search space. Further, it is noticed that Kumar and Sahoo combined the improved cat swarm optimization with k-harmonic means to solve clustering problems [36]. In this work, adaptive inertia weight factor and selection mechanism are introduced to enhance diversity and exploitation ability of CSO algorithm. In continuation of their work, Kumar and Singh designed an improved version of cat swarm optimization algorithm (ICSO) to solve global optimization problem [37].

### 3.5 Bat algorithm

A new meta-heuristic algorithm based on the behavior of bats, called Bat algorithm is presented in [38]. The microbats use echolocation to detect prey and avoid obstacles. When, microbats search for pray, they emit short pulses and listen for echo from surrounding to detect objects. Initially, Senthilnath et al., applied an approach-based on Bat algorithm to solve crop clustering problem [39]. Further, to cluster large application, a robust optimization technique based on bat behavior and k-medoids partitioning is reported in [40]. In this work, the swarm intelligence of bat algorithm used to guide in search space. Ashish et al., designed a parallel bat algorithm using map-reduce architecture to perform clustering on large volume data [41]. The developed algorithm works in three modules i.e. bat movement, fitness calculation and reduce module. Locations of bats are identified and updated using map function in bat movement module. The fitness of each bat is evaluated in fitness module. Further, reduce function is applied to obtain best bat position in reduce module. The performance of proposed algorithm is tested on five data sets and compared with particle swarm optimization. Authors claimed that proposed algorithm work efficiently on large data sets.

## 4 Experimental setup

The experimental setup of the proposed work is presented in this section. The efficiency of the proposed algorithm is evaluated on several benchmarks clustering datasets. These datasets are taken from UCI repository. In this study, two

**Table 1** Parameter settings of different algorithms

ACRO	
Population	$K \times d$
Dividing Factor (k)	2
$\lambda_{tr}$ in the range of	{0,1}
$\lambda_i$ in the range of	{0.25, 1.25}
$\lambda_{td}$ in the range of	{0,1}
Maximum number of iterations ( $T_{max}$ )	100

artificial datasets are also considered to test the performance of proposed algorithm. These datasets are generated in Matlab. The performance of proposed algorithm is evaluated using intra cluster distance and f-measure parameters. The intra cluster distance parameter determines the quality of clustering, whereas, f-measure denotes the precision. The experimental results of proposed algorithm are compared with K-means, PSO, ACO, CSO, BA and CRO algorithms [17, 24, 36, 42–45]. Tables 1 and 2 illustrate the parameters setting of the proposed algorithm and other algorithms used in this study.

### 4.1 Experimental results

Tables 3 and 4 present the results of the proposed ACRO algorithm and other clustering using artificial dataset1 and

artificial dataset2. It is seen that the proposed algorithm gives better results in comparison to all other algorithms being compared. It is also observed that K-means algorithm obtains worst results among all other algorithm using intra cluster distance parameter both of artificial dataset1 and artificial dataset2 using intra cluster distance parameter.

Tables 5 and 6 illustrate the results of the proposed algorithm and other algorithm using iris and cancer datasets. It is reported that the proposed algorithm obtains better quality results than other algorithms. Further, it is noted that the K-mean algorithm obtains maximum intra cluster distance among all algorithms using both of datasets. It is also noticed that the performance of the K-means, PSO, ACO and CSO algorithm is similar in case of f-measure parameter using iris dataset. But, the significant difference is occurred in terms

**Table 2** Parameter settings of different algorithms

CSO		PSO	
Population	$K \times d$	Number of swarms	$10 \times K \times d$
Seeking Memory Poo (SMP)	10	$c_1 = c_2$	2
Mixture Ratio (MR)	0.5	$\omega$ min	0.5
C	2	$\omega$ max	1
Maximum number of iterations ( $T_{max}$ )	100	Maximum number of iterations ( $T_{max}$ )	100
BA		ACO	
Population	$K \times d$	Number of ants	50
A0	0.9	Probability threshold for maximum trail	0.98
R	0.1	Local search probability	0.01
$f_{min}$	0	Evaporation rate	0.01
$f_{ma}$	2	Maximum number of iterations ( $T_{max}$ )	100
$\alpha = \gamma$	0.9		
Maximum number of iterations ( $T_{max}$ )	100		

**Table 3** Simulation results of proposed ACRO, K-Means, PSO, ACO, CSO BA and CRO algorithms using artificial dataset1

Parameters	Algorithms						
	K-means	PSO	ACO	CSO	BA	CRO	ACRO
Best case	161.32	153.45	154.29	159.73	151.81	147.91	143.36
Avg. case	166.12	161.24	158.52	163.37	154.98	151.23	149.56
Worst case	174.64	169.39	165.42	168.13	159.08	155.42	154.21
SD	7.625	6.437	4.712	5.418	4.564	6.78	5.243
F-measure	0.94	0.96	0.99	0.95	0.97	1	1

**Table 4** Simulation results of proposed ACRO, K-Means, PSO, ACO, CSO BA and CRO algorithms using artificial dataset2

Parameters	Algorithms						
	K-means	PSO	ACO	CSO	BA	CRO	ACRO
Best case	761.45	753.24	759.87	752.34	756.09	753.41	746.53
Avg. case	768.38	759.82	766.15	756.21	761.44	759.24	752.26
Worst case	776.49	769.58	773.41	764.39	169.17	764.08	758.13
SD	6.837	7.614	6.845	4.936	5.432	6.213	4.214
F-measure	0.89	0.91	0.93	0.937	0.946	0.958	0.976

**Table 5** Simulation results of proposed ACRO, K-Means, PSO, ACO, CSO BA and CRO algorithms sing iris dataset

Parameters	Algorithms						
	K-means	PSO	ACO	CSO	BA	CRO	ACRO
Best Case	97.52	97.05	97.21	96.98	96.84	97.81	95.56
Avg. Case	113.56	98.73	98.36	97.64	97.53	97.59	96.73
Worst Case	125.23	99.89	99.59	98.78	98.09	98.02	97.48
SD	15.326	0.467	0.426	0.392	0.263	0.292	0.196
F-Measure	0.781	0.78	0.778	0.781	0.782	0.782	0.785

**Table 6** Simulation results of proposed ACRO, K-Means, PSO, ACO, CSO BA and CRO algorithms using cancer dataset

Parameters	Algorithms						
	K-means	PSO	ACO	CSO	Bat	CRO	ACRO
Best Case	2989.46	2978.68	2983.49	2985.16	2972.36	2854.74	2912.66
Avg. Case	3248.25	3116.64	3178.09	3124.15	3098.93	3148.45	3063.34
Worst Case	3566.94	3358.43	3292.41	3443.56	3282.75	3064.58	3179.25
SD	256.58	107.14	93.45	128.46	56.24	81.6257	71.22
F-Measure	0.832	0.826	0.829	0.831	0.833	0.836	0.835

**Table 7** Simulation results of proposed ACRO, K-Means, PSO, ACO, CSO BA and CRO algorithms using CMC dataset

Parameters	Algorithms						
	K-means	PSO	ACO	CSO	BA	CRO	ACRO
Best Case	5834.21	5792.48	5756.42	5712.78	5689.16	5739.66	5681.56
Avg. Case	5912.46	5846.63	5831.25	5804.52	5778.14	5847.83	5746.32
Worst Case	5983.06	5936.14	5929.36	5921.28	5914.25	5904.32	5894.63
SD	54.24	48.86	44.34	43.29	39.54	46.21	36.41
F-Measure	0.337	0.333	0.332	0.334	0.336	0.335	0.339

**Table 8** Simulation results of proposed ACRO, K-Means, PSO, ACO, CSO BA and CRO algorithms using Wine dataset

Parameters	Algorithms						
	K-means	PSO	ACO	CSO	BA	CRO	ACRO
Best Case	16775.32	16491.26	16456.81	16429.54	16372.02	16589.42	16248.23
Avg. Case	18059.91	16424.52	16526.12	16386.21	16557.89	16643.29	16334.85
Worst Case	18783.23	16589.13	16621.44	16595.45	16556.76	16782.16	16396.56
SD	784.62	91.33	53.67	59.04	41.78	55.65	34.59
F-Measure	0.52	0.517	0.521	0.522	0.523	0.52	0.526

**Table 9** Simulation results of proposed ACRO, K-Means, PSO, ACO, CSO BA and CRO algorithms using Glass dataset

Parameters	Algorithms						
	K-means	PSO	ACO	CSO	BA	CRO	ACRO
Best case	222.43	264.56	273.22	256.53	256.47	274.11	261.47
Avg. case	246.51	278.71	281.46	264.44	269.61	280.9	266.23
Worst case	258.38	283.52	286.08	282.27	278.24	293.65	274.14
SD	18.32	8.59	6.58	15.43	7.09	9.56	8.11
F-Measure	0.426	0.412	0.402	0.416	0.43	0.41	0.428



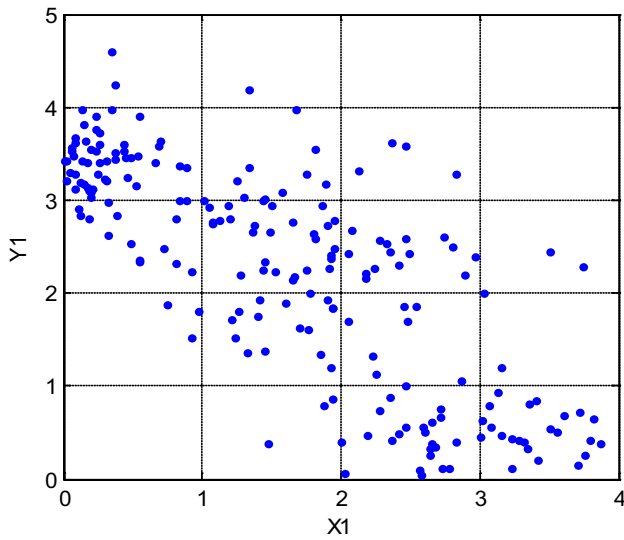


Fig. 1 Dispersion of data objects in artificial dataset1

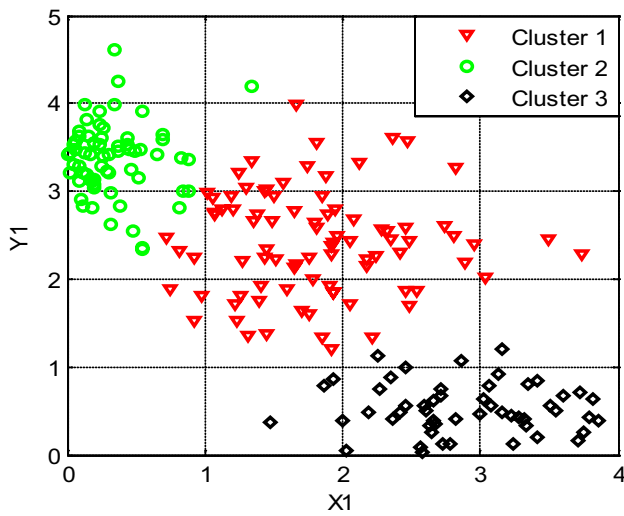


Fig. 2 Dispersion of data objects in artificial dataset2

of intra cluster distance parameter. Further, it observed that PSO algorithm have worst performance using f-measure parameter among rest of algorithm for cancer dataset.

Tables 7, 8 and 9 demonstrate the results of proposed ACRO and other algorithms using CMC, wine and glass dataset. It is revealed that the proposed algorithm provides enhanced results in comparison to other algorithms for CMC and wine datasets. Moreover, it is noticed that performance of BA algorithm is better for glass dataset in comparison to all other algorithms. Further, it is reported that K-means algorithm obtains maximum intra cluster distance for CMC and wine datasets, whereas, ACO algorithm obtains maximum intra cluster distance for glass dataset. For wine and glass datasets, K-means and ACO algorithms exhibit worst

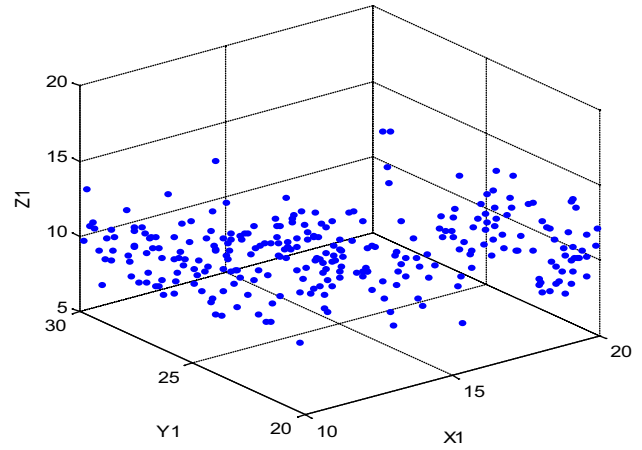


Fig. 3 Clustering results of proposed ACRO algorithm on artificial dataset1

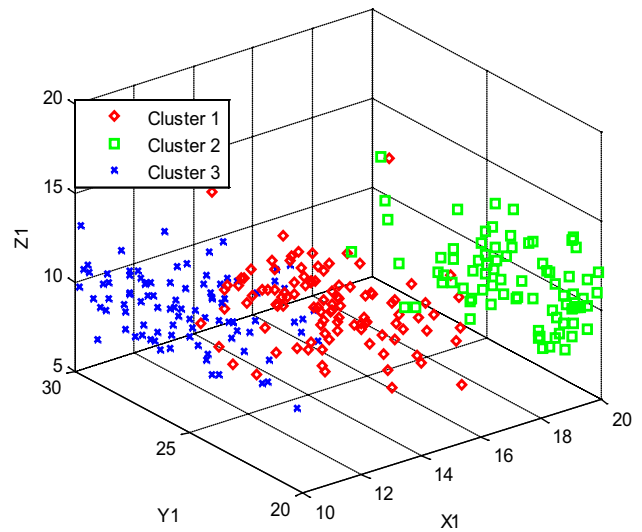


Fig. 4 Clustering results of proposed ACRO algorithm on artificial dataset2

f-measure results. In case of CMC dataset, PSO, ACO and CSO provide similar f-measure results.

Figures 1 and 2 illustrate the dispersion of the data objects in artificial dataset 1 and dataset 2 respectively. These artificial datasets are generated in Matlab. Artificial dataset 1 is a two dimensional dataset, whereas artificial dataset 2 is three dimensional dataset. Figures 3 and 4 show the clustering of the data objects into different clusters using proposed algorithm.

Figures 5 and 6 show the clustering of the iris dataset using proposed artificial chemical reaction optimization algorithm. The data objects of iris dataset are divided into three clusters such as setosa, versicolour and virginica. It is seen that data objects of setosa cluster is linearly separable

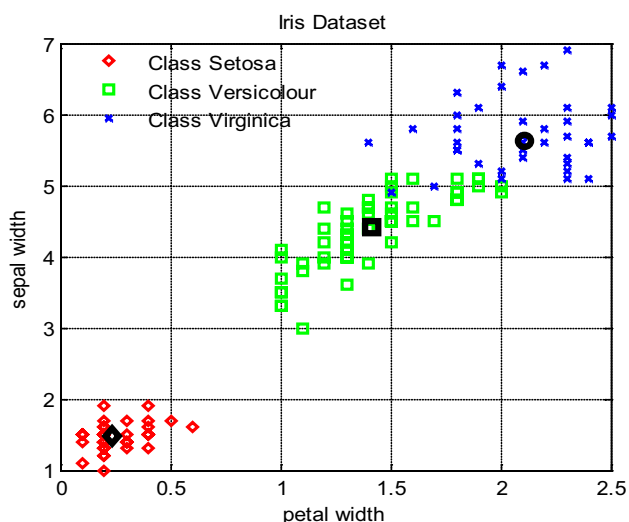


Fig. 5 Clustering results of proposed ACRO algorithm on Iris Dataset (2D View)

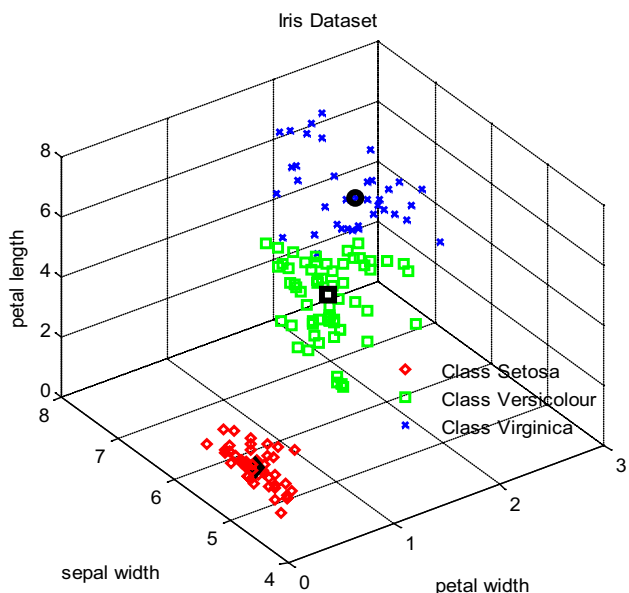


Fig. 6 Clustering results of proposed ACRO algorithm on iris dataset (3D View)

form versicolour and virginica. While, the data objects of versicolour and virginica clusters are linearly inseparable that can also affect the performance of the algorithm. But, it is observed that proposed algorithm gives better results for all three clusters. Figure 5 shows the illustration of

Table 11 Friedman statistical test results using intra cluster distance

Method	Statistical value	p value	Hypothesis	Critical value
Friedman Test	21.183673	0.0017	Rejected	12.591587

the data objects of iris dataset using sepal width and petal width attributes, whereas Fig. 6 shows the illustration of data objects using petal length, sepal width and petal width.

### 4.2 Statistical results

This subsection demonstrates the statistical tests on the proposed work. The aim of statistical tests is to provide quantitative decisions about the proposed work. The statistical tests determine whether there is enough evidence to reject the null hypothesis or not. If, null hypothesis is rejected, then the proposed algorithm is statistically better than compared algorithms. Otherwise, the performance of proposed algorithm is similar to the existing algorithm. Hence, it is necessary that an algorithm can perform better both in terms of experimental and statistical. In this study, Friedman statistical test is applied to determine the significant difference between the performances of proposed algorithm and other algorithms being compared. Further, two hypothesis is designed.  $H_0$  hypothesis stands for populations belongs to same sample and  $H_1$  stands for the populations belongs to different sample

- Intra cluster distance: The results of the Friedman statistical test for intra cluster distance parameter are illustrated in Tables 10 and 11. Table 10 depicts the average ranking of each algorithm using Friedman test. It is seen that the proposed ACRO algorithm obtains the first rank among all other algorithms i.e. 1.5. It is also observed that ACO algorithm obtains sixth rank i.e. 5.75. The statistical results of the test are reported in Table 1. The degree of freedom is 6. The critical value for the test is 12.591587. Hence, it is noticed that null hypothesis is rejected at the level of 0.10. After rejection of null hypothesis, a posthoc test is performed between proposed algorithm i.e. ACRO and rest of algorithms. In this work, Nemenyi posthoc test is applied for the same. The results of the posthoc test are reported in Table 12. Table 12 illustrates the unadjusted p values obtained using Nemenyi posthoc test. Hence, it is observed that a significant difference

Table 10 Rank of each algorithm using Friedman test (intra cluster distance)

	K-Means	PSO	ACO	CSO	BA	CRO	ACRO
Avg. ranking	5.5	5	5.75	3.75	2.75	3.75	1.5
Overall rank	5	4	6	3	2	3	1

**Table 12** Results of Nemenyi posthoc test (p-values without adjustment) using intra cluster distance

	K-Means	PSO	ACO	CSO	BA	CRO
PSO	0.948498					
ACO	0.999459	0.997234				
CSO	0.33914	0.928962	0.627226			
BA	0.383365	0.948498	0.67574	1		
CRO	0.999812	0.994704	1	0.57761	0.627226	
ACRO	0.021791	0.297637	0.079418	0.928962	0.905096	0.065133

**Table 13** Rank of each algorithm using Friedman test (F-measure)

K-Means	PSO	ACO	CSO	BA	CRO	ACRO
3.8	6	6.2	4.5	2.3	3.8	1.4

**Table 14** Friedman statistical test results using F-measure

Method	Statistical value	p value	Hypothesis	Critical value
Friedman Test	20.164286	0.002589	Rejected	12.591587

is occurred between the proposed algorithm and rest of algorithm.

- F-measure: Tables 13 and 14 demonstrate the results of the Friedman test using f-measure parameter. It is seen that proposed algorithm obtains first rank among all algorithms being compared i.e. 1.4. The statistical results of the Friedman test are reported in Table 14. The critical value for the Friedman test is 12.591587 and the p-value obtain for statistical test is 0.002589. Hence, it is stated that null hypothesis is rejected at the level of 0.10. After rejection of null hypothesis, Nemenyi post hoc test is taken between proposed algorithm i.e. ACRO and other algorithms. The results of the posthoc test are reported in Table 15. Hence, it is stated that substantial difference is occurred between the performance of the proposed algorithm and all other algorithms.

**Table 15** Results of Nemenyi posthoc test (p-values without adjustment) using F-measure

	K-Means	PSO	ACO	CSO	BA	CRO
PSO	0.57761					
ACO	0.57761	1				
CSO	0.999812	0.807113	0.807113			
BA	0.928962	0.065133	0.065133	0.766379		
CRO	1	0.57761	0.57761	0.999812	0.928962	
ACRO	0.57761	0.008059	0.008059	0.33914	0.994704	0.57761

## 5 Conclusion

In this work, an artificial chemical reaction optimization algorithm is presented for solving partitioned clustering problems. The proposed algorithm is inspired from the chemical reaction process. In this algorithm, reactants are used to search the optimal solution and these reactants are uniformly determined from the search space. Further, optimal solution for the problems can be represented using the reactants. The main work of the ACRO algorithm is to measure optimal cluster centroid for partitioned clustering problems. The reactants represent the initial cluster centers which are determined uniformly from the dataset. The proposed algorithm is applied to optimize the value of initially chosen reactants through its various steps. The performance of the proposed algorithm is tested on several real life clustering problems and compared with state of art clustering algorithms. From simulation results, it is noticed that the proposed algorithm achieves better clustering results in comparison to other clustering algorithms. Finally, it is stated that proposed ACRO algorithm is one of the efficient and effective algorithm for solving partitioned clustering problems.

## References

1. Teppola P, Mujunen SP, Minkkinen P (1999) Adaptive fuzzy C-means clustering in process monitoring. *Chemom Intell Lab Syst* 45(1):23–38
2. Zhou H, Liu Y (2008) Accurate integration of multi-view range images using k-means clustering. *Pattern Recogn* 41(1):152–175
3. Webb A (2002) *Statistical pattern recognition*. Wiley., New Jersey, pp 361–406
4. Dunn III WJ, Greenberg MJ, Callejas SS (1976) Use of cluster analysis in the development of structure-activity relations for anti-tumor triazines. *J Med Chem* 19(11):1299–1301
5. Anaya AR, Boticario JG (2011) Application of machine learning techniques to analyses student interactions and improve the collaboration process. *Expert Syst Appl* 38(2):1171–1181
6. Hung YS, Chen KLB, Yang CT, Deng GF (2013) Web usage mining for analyzing elder self-care behavior patterns. *Expert Syst Appl* 40(2):775–783
7. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. *Acta Mech* 213(3):267–289
8. Kumar Y, Sahoo G (2014) A charged system search approach for data clustering. *Progress Artif Intell* 2(2–3):153–166
9. Kaveh A, Share MAM, Moslehi M (2013) Magnetic charged system search: a new meta-heuristic algorithm for optimization. *Acta Mech* 224(1):85–107
10. Kumar Y, Sahoo G (2015) Hybridization of magnetic charge system search and particle swarm optimization for efficient data clustering using neighborhood search strategy. *Soft Comput* 19(12):3621–3645
11. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. *Inform Sci* 222:175–184
12. Shah-Hosseini H (2011) Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *Int J Comput Sci Eng* 6(1–2):132–140
13. Baykasoğlu A, Özbakır L, Tapkan P (2007) Artificial bee colony algorithm and its application to generalized assignment problem. In: *Swarm intelligence, focus on ant and particle swarm optimization*. InTech, UK
14. Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. *Expert Syst Appl* 38(10):13170–13180
15. Alatas B (2012) A novel chemistry based metaheuristic optimization method for mining of classification rules. *Expert Syst Appl* 39(12):11080–11088
16. He Y, Pan W, Lin J (2006) Cluster analysis using multivariate normal mixture models to detect differential gene expression with microarray data. *Comput Stat Data Anal* 51(2):641–658
17. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: *Fifth Berkeley symposium on mathematics, statistics and probability*. University of California Press, pp 281–297
18. Žalik KR (2008) An efficient k'-means clustering algorithm. *Pattern Recogn Lett* 29(9):1385–1391
19. Ismkhan H (2018) Ik-means+: an iterative clustering algorithm based on an enhanced version of the k-means. *Pattern Recogn* 79:402–413
20. Tzortzis G, Likas A (2014) The MinMax k-Means clustering algorithm. *Pattern Recogn* 47(7):2505–2516
21. Malinen MI, Mariescu-Istodor R, Fränti P (2014) K-means\*: Clustering by gradual data transformation. *Pattern Recogn* 47(10):3376–3386
22. Kang Q, Liu S, Zhou M, Li S (2016) A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence. *Knowl-Based Syst* 104:156–164
23. Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B (Cybern)* 26(1):29–41
24. Shelokar PS, Jayaraman VK, Kulkarni BD (2004) An ant colony approach for clustering. *Anal Chim Acta* 509(2):187–195
25. Wan M, Wang C, Li L, Yang Y (2012) Chaotic ant swarm approach for data clustering. *Appl Soft Comput* 12(8):2387–2393
26. Huang CL, Huang WC, Chang HY, Yeh YC, Tsai CY (2013) Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering. *Appl Soft Comput* 13(9):3864–3872
27. Menéndez HD, Otero FE, Camacho D (2016) Medoid-based clustering using ant colony optimization. *Swarm Intell* 10(2):123–145
28. Cura T (2012) A particle swarm optimization approach to clustering. *Expert Syst Appl* 39(1):1582–1588
29. Chuang LY, Hsiao CJ, Yang CH (2011) Chaotic particle swarm optimization for data clustering. *Expert systems with Appl* 38(12):14555–14563
30. Hatamlou A (2017) A hybrid bio-inspired algorithm and its application. *Appl Intell* 47(4):1059–1067
31. Jiang B, Wang N (2014) Cooperative bare-bone particle swarm optimization for data clustering. *Soft Comput* 18(6):1079–1091
32. Bouyer A, Hatamlou A (2018) An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. *Appl Soft Comput* 67:172–182
33. Hatamlou A, Hatamlou M (2013) PSOHS: an efficient two-stage approach for data clustering. *Memetic Comput* 5(2):155–161
34. Chu SC, Tsai PW, Pan JS (2006, August) Cat swarm optimization. In *Pacific rim international conference on artificial intelligence*. Springer, Berlin, Heidelberg, pp 854–858
35. Mohapatra P, Chakravarty S, Dash PK (2016) Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. *Swarm Evolutionary Comput* 28:144–160
36. Kumar Y, Sahoo G (2015) A hybrid data clustering approach based on improved cat swarm optimization and K-harmonic mean algorithm. *Ai Commun* 28(4):751–764
37. Kumar Y, Singh PK (2018) Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering. *Appl Intell* 48(9):2681–2697
38. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) *Nature inspired cooperative strategies for optimization (NICSO 2010)*, vol 284. Springer, Berlin, Heidelberg, pp 65–74
39. Senthilnath J, Kulkarni S, Benediktsson JA, Yang XS (2016) A novel approach for multispectral satellite image classification based on the bat algorithm. *IEEE Geosci Remote Sens Lett* 13(4):599–603
40. Aboubi Y, Drias H, Kamel N (2016) BAT-CLARA: BAT-inspired algorithm for clustering LARge applications. *IFAC-PapersOnLine* 49(12):243–248
41. Ashish T, Kapil S, Manju B (2018) Parallel bat algorithm-based clustering using mapreduce. In: *Networking communication and data knowledge engineering*. Springer, Singapore, pp 73–82
42. Zhan ZH, Zhang J, Li Y, Chung SH (2009) Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern B Cybern* 39:1362–1381
43. Gandomi AH, Yang XS, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. *Neural Comput Appl* 22(6):1239–1255
44. Kumar Y, Sahoo G (2016) A hybridize approach for data clustering based on cat swarm optimization. *Int J Inf Commun Technol* 9(1):117–141
45. Baral A, Behera HS (2013) A novel chemical reaction-based clustering and its performance analysis. *Int J Bus Intell Data Min* 8(2):184–198

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.