A Hybrid CRO-K-Means Algorithm for Data Clustering

Sibarama Panigrahi, Balaram Rath and P. Santosh Kumar

Abstract Over the past few decades clustering algorithms have been used in diversified fields of engineering and science. Out of various methods, K-Means algorithm is one of the most popular clustering algorithms. However, K-Means algorithm has a major drawback of trapping to local optima. Motivated by this, this paper attempts to hybridize Chemical Reaction Optimization (CRO) algorithm with K-Means algorithm for data clustering. In this method K-Means algorithm is used as an on-wall ineffective collision reaction in the CRO algorithm, thereby enjoying the intensification property of K-Means algorithm and diversification of intermolecular reactions of CRO algorithm. The performance of the proposed methodology is evaluated by comparing the obtained results on four real world datasets with three other algorithms including K-Means algorithm, CRO-based and differential evolution (DE) based clustering algorithm. Experimental result shows that the performance of proposed clustering algorithm is better than K-Means, DE-based, CRO-based clustering algorithm on the datasets considered.

Keywords Data clustering · Chemical reaction optimization · K-Means algorithm

1 Introduction

Clustering is an unsupervised classification technique in which patterns or feature vectors or data items are organized in clusters/groups based on similarity, such that the data within the same cluster have high degree of similarity and patterns

S. Panigrahi (✉)
National Institute of Science and Technology, Palur Hills, Berhampur 761008, Odisha, India
e-mail: panigrahi.sibarama@gmail.com

B. Rath · P. Santosh Kumar
MIRC Lab, MITS Engineering College, Rayagada 765017, Odisha, India
e-mail: balaram.rath@gmail.com

P. Santosh Kumar
e-mail: santosh.mirc@gmail.com

L.C. Jain et al. (eds.), *Computational Intelligence in Data Mining - Volume 3*,
Smart Innovation, Systems and Technologies 33, DOI 10.1007/978-81-322-2202-6_57

belonging to different clusters have high degree of dissimilarity. Jain et al. [1] provides a survey on clustering algorithms and Jain and Gajbhiye [2] provides a comparative performance analysis on different clustering algorithms. However, the clustering techniques can be broadly classified into two types: hierarchical and partitional. Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters into smaller ones and generates a hierarchy of partitions in which each partition is nested within the partition at the next level in the hierarchy. On the other hand, partitional clustering attempts to directly decompose the data set into a set of disjoint clusters.

K-Means clustering algorithm [2, 3] is a well-known partitioning based technique which partitions n-patterns into k-clusters with each pattern belongs to the cluster with the nearest mean. For improving the performance and efficiency of K-Means clustering, various and numerous methods have been proposed. Dash et al. [4] proposed a clustering approach by hybridizing K-Means algorithm with principal component analysis (PCA) to reduce dimensionality of data set. Additionally employed a new method obtain the initial centroid values by finding the mean of all the data sets divided into k different sets in ascending order. This approach has the demerit of taking large amount of time and it may eliminate some of the features which are also important. However, Yedla et al. [5] proposed an enhanced K-Means algorithm with improved initial center using the distance from the origin to improve the performance of K-Means algorithm. Although, this approach seems to solve the initialization problem but does not give any guarantee better time complexity. For reducing the computational time Fahim et al. [6] proposed a clustering method by considering initial centroids randomly. However, this method does not provide unique result due to very sensitive to randomly selected initial center points. Zhang and Xia [7] proposed the initial centroids algorithm based on K-Means that have avoided alternate randomness of initial centroids. Despite of several years of research on improving the performance of K-Means algorithm, the major drawback is its chances to trapping to local optima because it is an iterative and hill climbing method. Therefore in order to achieve diversification property hybridization with some global optimization technique is required.

Obtaining the optimal centroids for the clusters can be considered as a multimodal search problem. Therefore, any global optimization like technique differential evolution (DE) [8], genetic algorithm (GA) [9], particle swarm optimization (PSO) [10], ant colony optimization (ACO) [11], a bee colony optimization (BCO) [12], an evolutionary strategy (ES) [13], quantum inspired algorithms (QEA) [14], chemical reaction optimization (CRO) [15–17] etc. can be used for clustering. Motivated by this few authors have applied global optimization techniques like genetic algorithm [18–20], differential evolution [21], Teaching-Learning-Based Optimization [22, 23] algorithms to clustering problems and have shown better performance. However, the major drawback of these approaches is that obtaining a trade-off between diversification and intensification. Motivated by this, a hybridization of K-Means algorithm (used to intensify some of the solutions) and CRO algorithm (diversifies the solution space) is used.

The rest of this paper is organized as follows. Section 2 gives a brief overview of K-Means, DE-based and CRO-based clustering algorithms. Section 3 describes the proposed Hybrid CRO-K-Means Clustering algorithm, Sect. 4 gives the experimental set up and comparison between the proposed algorithm and some of the existing algorithms, and Sect. 5 concludes the paper.

2 Clustering Algorithms

2.1 K-Means Clustering Algorithm

K-Means is one of the simplest centroid based unsupervised learning algorithms that use K number of clusters (fixed a priori) to discriminate the data. Consider a data matrix X with n-data point patterns. The main goal is to find optimal positions of K-centroids (one for each cluster and represented by C_i) so as to partition n-data points into K-disjoint subsets by minimizing the sum of squared error. The sum squared error (E) is calculated as in Eq. 1.

$$E = \sum_{i=1}^{K} \sum_{j=1}^{n} \left\| X_j - C_i \right\|^2 \tag{1}$$

where ||.|| is the Euclidean distance.

Algorithm:

1. Randomly select K centers or centroids.
2. Calculate the distance between each data point and cluster centers and assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
3. For each cluster, recompute the cluster centers.
4. Repeat step 2 and 3, until no cluster center changes.

2.2 Differential Evolution Based Clustering Algorithm

The differential evolution (DE) algorithm is a simple and efficient stochastic direct search method for global optimization. It was introduced several years ago (1997) [8]. It has several advantages such as: ability to find the global minimum of a non-differentiable, nonlinear and multimodal function irrespective of initial values of its parameters, parallel implementabilty, ease of use and good convergence properties. Therefore, a wide variants of it has been developed [18]. The major difference between the DE variants lies in the mutation and crossover strategies. The conventions used above is DE/a/b/c, where 'DE' stands for 'differential evolution', 'a' represents the base vector to be perturbed, 'b' represents number of difference vectors used for perturbation of 'a' and 'c' represents the type of crossover used (bin: binary, exp: exponential). Interested reader may go through [8, 18] to have a detail description regarding DE algorithm and its variants.

DE Clustering Algorithm:

1. Initialize each individual solution to contain K randomly chosen points from the dataset.
2. Calculate fitness function for each individual.
3. Perform Crossover and Mutation upon solutions based on the scheme DE/a/b/c.
4. Replace the existing solutions with the new ones if and only if the new solutions have better fitness value.
5. Repeat step 3 and 4, until some termination criteria is satisfied.

Select the best chromosome in the population which will act as the resultant clusters.

2.3 Chemical Reaction Optimization Based Clustering Algorithm

Chemical reaction optimization (CRO) algorithm is a recently established population based metaheuristics optimization technique [15, 16] which is inspired by the nature of chemical reactions. It does not attempt to capture every detail of chemical reaction rather loosely couples chemical reaction with optimization. A chemical reaction starts with some initial species/reactants/molecules in the unstable states and undergoes a sequence of collisions and become the final products in stable states [16]. During the chemical reaction the intra-molecular structure of a reactant changes. Most of the reactions are reversible in nature i.e. a reaction may be turned back due to instability. It enjoys the advantages of both Simulated Annealing (SA) and GA [16]. The major difference between CRO and other evolutionary techniques is that, the population size (that is the number of reactants) may vary from one iteration to the other where as in evolutionary techniques the population size remains fixed. But few authors [25] have proposed fixed population sized CRO algorithms and shown that fixed population sized CRO not only performs better but also easier to implement. The basic unit in CRO is a molecule (chromosome) which is considered as a single possible solution in the population. Each molecule has potential energy (PE) which refers to fitness value, to characterize the molecule. A chemical change of a molecule is triggered by a collision which may be uni-molecular collision (one molecule taking part in chemical reaction) or inter-molecular collision (two or more molecule collide with each other). The corresponding reaction change is termed as Elementary Reaction.

The chemical reaction optimization based Clustering algorithm consists of following steps:

1. Initialize each individual solution to contain K randomly chosen points from the dataset.
2. Evaluation of Potential Energy (PE) of each Reactant representing a set of cluster centers.
3. Applying Chemical reactions to generate new reactants.
4. Reactants update for better potential energy.
5. Check termination criteria. if satisfied go to step-6 otherwise go to step-3
6. Use the reactant having best Potential Energy as the cluster center.

3 Proposed Methodology

In this paper the hybridized CRO-K-Means algorithm uses four kinds of elementary reactions (two uni-molecular and two bi-molecular): on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. K-Means algorithm is hybridized as an on-wall ineffective collision of the CRO algorithm. K-Means algorithm can able to find the local minima and the use of other three reactions may assist in trapping from local optima. Therefore chances of obtaining near optimal solution in lesser number of iterations can be guaranteed.

The pseudo-code of clustering using CRO is shown in Algorithm 1 where *ReacNum* is the size of the population ‘P’ best_mole represents the molecule having the highest potential energy (PE) in current iteration, and the elementary reactions are shown in subsequent algorithms.

Algorithm 1 (CRO Based Clustering Algorithm)

```
Set the iteration-counter i=0
Load the dataset in X.
while Halting criteria not satisfied do
//Initialize the ReacNum of Reactants/Molecules randomly from the dataset: $P^i=\{R_1^i, R_2^i, R_3^i…., R_{ReacNum}^i\}$, with $R_j^i=\{W_{j,1}^i,…….,W_{j,D}^i\}$ for j=1,2,3..... ReacNum, with W representing one dimension of the attribute of cluster center, D=length of each Reactant (Number of clusters □ Dimension of each cluster centroid), $P^i$ = Population of ith iteration.
//Calculate the fitness (PE) of each set of cluster centers (reactant)
for j=1 to ReacNum
       Calculate the PE of jth reactant in the population.
end of for
//Apply Chemical Reactions to obtain new cluster centers
While (termination criteria is not satisfied) do
    begin
    for j=1 to ReacNum
       // perform all reaction over the reactants of $P^i$
       Get $rand_1$ randomly in an interval [0, 1]
       if $rand_1 \leq 0.5$
           Get $rand_2$ randomly in an interval [0, 1]
           if $rand_2 \leq 0.5$
                Decomposition ($R_j$);
           else
                On-wall ineffective collision($R_j$)
           end of if
       else
           Get $rand_3$ randomly in an interval [0, 1]
           if  $rand_3 \leq 0.5$
                Select the best reactant $R_k(R_k \neq R_j)$
                Synthesis ($R_j$, $R_k$)
           else
                Select another reactant $R_k(R_k \neq R_j)$
                randomly inter-molecular ineffective
        collision($R_j$, $R_k$)
           end of if
       end of if
       Apply strictly greedy Reversible Reaction for
       increased PE to update reactants
    end of for
   Set the iteration counter i=i+1
end of while
best_mole=The Reactant having best PE in the final iteration.
The resultant clusters centroids will be stored in best_mole.
```

3.1 On-wall Ineffective Collision Reaction

In this reaction K-Means algorithm is applied on the reactant considering the values of the reactants as the initial cluster centers. The pseudo-code of the decomposition reaction is described in Algorithm 2.

```
Algorithm 2 (On-wall ineffective collision($R_j$))
Input: A reactant $R_j$
Apply the K-Means algorithm (considering the $R_j$ values as the initial cluster center position) to obtain the new cluster centers.
Output: A new reactant $R_1$
```

3.2 Decomposition Reaction

In this reaction the value of a randomly selected atom of the reactant is changed in order to perform more local search. Consider a reactant $R_j = \{W_{j,1}, W_{j,2} \ldots, W_{j,D}\}$ with $W_{j,x}$ ($x \in [1, n]$) be an atom of the reactant-j. The rate of change of the atom of the reactant depends on the rate of reaction (λ). The rate of reaction is a random number generated from a Cauchy distribution because it diversifies the solution more as compared to traditional uniform distribution. The pseudo-code is described in Algorithm 3.

```
Algorithm 3 (Decomposition($R_j$))
Input: A reactant $R_j$
Duplicate $R_j$ to produce $R_1$
Select an atom k ($k \in [1:n]$) randomly.
$W_{1,k} = L + \lambda \times (U - L)$
Where $\lambda$= cauchyrnd(0.6,0.1), is a random number generated from Cauchy distribution with a location parameter 0.6 and scale parameter  0.1. It is regenerated if the random number falls out of the range [0, 1].
Output: A new reactant $R_1$
```

Here two reactants $R_j = \{W_{j,1}, \ldots, W_{j,D}\}$ and $R_k = \{W_{k,1}, W_{k,2} \ldots, W_{k,D}\}$ will take part in the reaction. These reactions help in diversification of the solution by generating a new solution that is significantly different from the current solution. These reactions occur with a probability of 50 %. Following types of bimolecular reactions are used.

3.3 Synthesis Reaction

In this reaction one reactant is produced due to reaction between a reactant $R_j = \{W_{j,1}, \ldots, W_{j,D}\}$ and another randomly reactant and $R_{rand} = \{W_{k,1}, W_{k,2} \ldots, W_{k,D}\}$; of the iteration. Here, instead of traditional normal or uniform distribution; the rate of reaction (λ) is generated from a Cauchy distribution with a location parameter 0.7 and scale parameter 0.1. The pseudo-code of this reaction is described in Algorithm 4.

```
Algorithm 4 (Synthesis ($R_j$,$R_k$))
Input: Two reactants $R_j$, $R_{rand}$
$R_1=R_J+ \lambda\times (R_{rand}-R_J)$
Where $\lambda$=cauchyrnd(0.7,0.1), is a random number generated randomly from Cauchy distribution with location parameter 0.7 and scale parameter 0.1. It is regenerated if the random number falls put of the range [0.5, 1].
Output: A new reactant $R_1$
```

3.4 Inter-molecular Ineffective Collision Reaction

In this reaction two solutions R_1 and R_2 are obtained from reaction between two reactants R_j and R_k. The pseudo-code of this reaction is described in Algorithm 5.

```
Algorithm 5(inter-molecular ineffective collision ($R_j$,$R_k$))
Input: Two reactants $R_j$, $R_k$
$R_1=\lambda_t\times R_j+\lambda_t \times (1- R_k)$
$R_2=\lambda_t\times R_k+\lambda_t \times (1-R_j)$
Where $\lambda_t$ is a random number generated uniformly between[0,1] for $t^{th}$ iteration.
Output: Two reactants $R_1$and $R_2$
```

3.5 Reactant Update

Algorithm 6 (Reversible Reaction())

```
For Reactions producing single products
     If $R_j$ under goes reaction to produce $R_1$
      If PE ($R_1$) < PE($R_j$)
           Replace $R_j$ by $R_1$
      end of if
     end of if

For Reactions producing two products
     If $R_j$ under goes reaction to produce $R_1$ and $R_2$
        If PE($R_1$)<PE($R_j$)
            Replace $R_j$ by $R_1$
        end of if
        If PE($R_2$)<PE($R_j$)
            Replace $R_j$ by $R_2$
        end of if
   end of if
```

Algorithm 6 describes the pseudo-code of the reversible reaction. Every reaction is followed by a reversible reaction to update the reactants (cluster centers) for next iteration i.e. reactants having better fitness (PE) survive for the next iteration. In this reaction, a reaction producing a single product replaces the replaces the used reactant for better PE; and for reactions producing two products, best product replace the used reactant for better PE. Thus the number of reactants of the population remains same throughout the reaction process.

4 Experimental Set up and Simulation Results

For comparative performance analysis of proposed hybrid CRO-K-Means algorithm with K-Means, DE/rand/1/bin, and CRO [26] based clustering algorithms, four real world datasets such as: Lung cancer, Iris, Breast cancer Wisconsin and Haberman's survival from UCI machine learning data repository are considered. *Lung cancer* dataset has 32 instances with predictive attributes which are nominal, and takes integer values from 0 to 3. *Iris* dataset has 150 patterns which are random samples of three species of the iris flower such as: setosa, versicolor, and virginica. *Breast cancer Wisconsin data*set consists of 699 patterns which were collected periodically as Dr. Wolberg reports his clinical cases. *Haberman's survival dataset* contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had

Table 1 Properties of the data sets considered

Data set	Number of data points/ instances	Number of attributes	Number of clusters
Lung cancer	32	56	3
Iris	150	4	3
Breast cancer	214	10	3
Haberman's survival	306	3	3

undergone surgery for breast cancer. Number of data points, number of attributes and number of clusters of all four datasets are listed in Table 1.

4.1 Performance Measure

In order to evaluate the effectiveness of proposed methodology with the other algorithms considered following three performance measures are considered.

Clustering metric: The clustering metric 'M' for K clusters C_1, C_2,…, C_K is calculated as in Eq. 2.

$$M(C_1, C_2, \ldots, C_K) = \sum_{i=1}^{K} \sum_{x_j \in C_i} ||\mathrm{x}_j - C_i|| \tag{2}$$

The objective is to minimize 'M'.

Note that for the evolutionary algorithms DE, CRO and Proposed methodology clustering metric is used as the fitness function or PE.

Intra-cluster Distance: The intra-cluster distance is mean of maximum distance between two data vectors within a set of clusters and is calculated as in Eq. 3.

$$\frac{1}{K} \sum_{i=1}^{K} \max_{ZpZq \in Ci} d(Zp, Zq) \tag{3}$$

The objective is to minimize the intra cluster distance.

Inter-cluster distance: It is the minimum distance between the centroids of the clusters.

The objective is to maximize the inter-cluster distance.

4.2 Simulation Results

For all the evolutionary algorithms the population size or Reactant number is fixed to 10 and run up to a maximum of 100 iterations or 90 % of chromosomes or reactants converge to same point. As the evolutionary methods are stochastic in nature, ten independent simulations were performed for each data set and for each algorithm. The mean and standard deviation of ten iterations for each algorithm for Lung cancer, Iris, Breast cancer Wisconsin, and Haberman's survival datasets are shown in Tables 2, 3, 4 and 5 respectively. Note that for a particular problem and

Table 2 Comparison of the performance of different clustering algorithms for lung cancer dataset

Clustering algorithms	Clustering metric Mean ± St. Dev.	Intra-cluster distance Mean ± St. Dev.	Inter-cluster distance Mean ± St. Dev.
K-Means	129.48 ± 1.59	7.51 ± 0.29	3.18 ± 0.38
DE-Clustering	127.44 ± 0.67	7.76 ± 0.28	3.27 ± 0.13
CRO-Clustering	126.98 ± 0.22	7.49 ± 0.12	3.38 ± 0.03
Proposed method	126.10 ± 0.15	7.38 ± 0.11	3.46 ± 0.03

Table 3 Comparison of the performance of different clustering algorithms for iris dataset

Clustering algorithms	Clustering metric Mean ± St. Dev.	Intra-cluster distance Mean ± St. Dev.	Inter-cluster distance Mean ± St. Dev.
K-Means	102.42 ± 10.44	2.59 ± 0.17	1.59 ± 0.41
DE-Clustering	97.39 ± 0.17	2.56 ± 0.05	1.79 ± 0.01
CRO-Clustering	97.22 ± 2.92E-14	2.51 ± 4.56E-16	1.80 ± 4.56E-16
Proposed method	97.14 ± 2.90E-14	2.46 ± 4.55E-16	1.84 ± 4.22E-16

Table 4 Comparison of the performance of different clustering algorithms for breast cancer Wisconsin

Clustering algorithms	Clustering metric Mean ± St. Dev.	Intra-cluster distance Mean ± St. Dev.	Inter-cluster distance Mean ± St. Dev.
K-Means	2.88E + 03 ± 18.12	17.24 ± 0.42	3.39 ± 2.14
DE-Clustering	2.86E + 03 ± 4.31	16.64 ± 0.58	4.38 ± 1.84
CRO-Clustering	2.86E + 03 ± 1.56	16.61 ± 0.33	4.37 ± 1.79
Proposed method	2.79E + 03 ± 1.59	15.92 ± 0.23	4.47 ± 1.24

Table 5 Comparison of the performance of different clustering algorithms for Haberman's survival dataset

Clustering algorithms	Clustering metric Mean ± St. Dev.	Intra-cluster distance Mean ± St. Dev.	Inter-cluster distance Mean ± St. Dev.
K-Means	2.29E + 03 ± 12.98	41.78 ± 3.25	13.12 ± 1.24
DE-Clustering	2.28E + 03 ± 18.16	41.76 ± 3.02	12.88 ± 0.47
CRO-Clustering	2.26E + 03 ± 4.12	42.42 ± 3.28	13.22 ± 1.88
Proposed method	2.16E + 03 ± 3.98	40.20 ± 2.08	13.45 ± 1.73

for a particular run the initial value of cluster centers for the three evolutionary methods are kept same. For Differential evolution DE/rand/1/bin strategy with F = 0.5 and Cr = 0.6 is used.

It can be observed from the simulation results that the proposed hybrid clustering method performs better than K-Means, DE-Clustering and CRO-Clustering algorithms for the Lung cancer, Iris, Breast cancer Wisconsin, and Haberman's survival dataset considering Clustering Metric, Intra-Cluster Distance and Inter-Cluster Distance as performance measure.

5 Conclusion

In this paper, a hybrid of chemical reaction optimization and K-Means algorithm for data clustering is proposed. In this hybrid method K-Means algorithm is used as a chemical reaction of the CRO algorithm. The K-Means algorithm intensifies the solution up to local optima meanwhile other chemical reactions assist in overcoming the local optima by exploring the solution space; thereby a trade between intensification and diversification is implicitly maintained and chances of reaching a global optima increases. In order to evaluate the effectiveness of the proposed method four real world datasets have been considered. Results revealed that the proposed method performs better than the K-Means, DE based Clustering and CRO based clustering algorithms for the aforementioned datasets. However, the major drawback of the proposed method is that it takes more amount of time than other algorithms since it uses K-Means algorithm as one of the reactions.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**, 264–323 (1999)
2. Jain, S., Gajbhiye, S.: A Comparative performance analysis of clustering algorithms. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **2**, 441–445 (2012)
3. Abbas, O.A.: Comparisons between data clustering algorithms. Int. Arab J. Inf. Technol. **5**, 320–325 (2008)
4. Dash, R., Mishra, D., Rath, A.K., Acharya, M.: A hybridized K-means clustering algorithm for high dimensional dataset. Int. J. Comput. Sci. Technol. **2**, 59–66 (2010)
5. Yedla, M., Pathakota, S.R., Srinivasa, T.M.: Enhancing K means algorithm with improved initial center. Int. J. Comput. Sci. Inf. Technol. **1**, 121–125 (2010)
6. Fahim, A.M., Salem, A.M., Torkey, F.A., Ramadan, M.A., Saake, G.: An efficient K-means with good initial starting points. Georgian Electron. Sci. J. Comput. Sci. Telecommun. **2**, 47–57 (2009)
7. Zhang, C., Xia, S.: K-means clustering algorithm with improved initial center. In: Second International Workshop on Knowledge Discovery and Data Mining, WKDD, pp. 790–792 (2009)
8. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)

9. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading (1989)
10. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
11. Socha, K., Doringo, M.: Ant colony optimization for continuous domains. Eur. J. Oper. Res. **185**, 1155–1173 (2008)
12. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm—a novel tool for complex optimization problems. In: Proceedings of 2nd International Virtual Conference on Intelligent Production Machines and Systems, pp. 454–459 (2006)
13. Beyer, H.G., Schwefel, H.P.: Evolutionary strategies: a comprehensive introduction. Nat. Comput. **1**, 3–52 (2002)
14. Han, K.H., Kim, J.H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evol. Comput. **6**, 580–593 (2002)
15. Lam, A.Y.S., Li, V.O.K.: Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans. Evol. Comput. **14**, 381–399 (2010)
16. Lam, A.Y.S.: Real-coded chemical reaction optimization. IEEE Trans. Evol. Comput. **16**, 339–353 (2012)
17. Alatas, B.: ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst. Appl. **38**, 13170–13180 (2011)
18. Al-Shboul, B., Myaeng, S.: Initializing K-means clustering algorithm by using genetic algorithm. World Acad. Sci. Eng. Technol. **54**, 114–118 (2009)
19. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recogn. **33**, 1455–1465 (2000)
20. Bandyopadhyay, S., Maulik, U.: An evolutionary technique based on K-means algorithm for optimal clustering in R^N. Inf. Sci. **146**, 221–237 (2002)
21. Paterlini, S., Krink, T.: High performance clustering with differential evolution. evolutionary computation. In: CEC2004, vol. 2, pp. 2004–2011 (2004)
22. Satapathy, S.C., Naik, A.: Data clustering based on teaching-learning-based optimization. In: Swarm, Evolutionary and Memetic Computing Conference Part II, LNCS, vol. 7077, pp. 148–156 (2011)
23. Naik, A., Satapathy, S.C., Parvathi, K.: Improvement of initial cluster center of c-means using teaching learning based optimization. Procedia Technol. **6**, 428–435 (2012)
24. Das, S., Suganthanam, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**, 4–31 (2011)
25. Sahu, K.K., Panigrahi, S., Behera, H.S.: A novel chemical reaction optimization algorithm for higher order neural network training. J. Theoret. Appl. Inf. Technol. **53**, 402–409 (2013)
26. Baral, A., Behera, H.S.: A novel chemical reaction-based clustering and its performance analysis. Int. J. Bus. Intell. Data Min. **8**, 184–198 (2013)