A hybrid approach for data clustering based on modified cohort intelligence and K-means

Ganesh Krishnasamy a,⇑, Anand J. Kulkarni b, Raveendran Paramesran a

a Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, Lembah Pantai, 50603 Kuala Lumpur, Malaysia
b Odette School of Business, University of Windsor, 401 Sunset Ave, Windsor, ON N9B 3P4, Canada

A R T I C L E   I N F O
Keywords:
Clustering
Cohort intelligence
Meta-heuristic algorithm

A B S T R A C T
Clustering is an important and popular technique in data mining. It partitions a set of objects in such a manner that objects in the same clusters are more similar to each other than objects in different clusters according to certain predefined criteria. K-means is simple yet an efficient method used in data clustering. However, K-means has a tendency to converge to local optima and depends on initial value of cluster centers. In the past, many heuristic algorithms have been introduced to overcome this local optima problem. Nevertheless, these algorithms too suffer several short-comings. In this paper, we present an efficient hybrid evolutionary data clustering algorithm referred to as K-MCI, whereby, we combine K-means with modified cohort intelligence. Our proposed algorithm is tested on several standard data sets from UCI Machine Learning Repository and its performance is compared with other well-known algorithms such as K-means, K-means++, cohort intelligence (CI), modified cohort intelligence (MCI), genetic algorithm (GA), simulated annealing (SA), tabu search (TS), ant colony optimization (ACO), honey bee mating optimization (HBMO) and particle swarm optimization (PSO). The simulation results are very promising in the terms of quality of solution and convergence speed of algorithm.

1. Introduction
Clustering is an unsupervised classification technique which partitions a set of objects in such a way that objects in the same clusters are more similar to one another than the objects in different clusters according to certain predefined criterion (Jain, Murty, & Flynn, 1999; Kaufman & Rousseeuw, 2005). The term unsupervised means that grouping is established based on the intrinsic structure of the data without any need to supply the process with training items.

Clustering has been applied across many applications, i.e., machine learning (Anaya & Boticario, 2011; Fan, Chen, & Lee, 2008), image processing (Das & Konar, 2009; Portela, Cavalcanti, & Ren, 2014; SiangTan & Matlsa, 2011; Zhao, Fan, & Liu, 2014), data mining (Carmona et al., 2012; Ci, Guizani, & Sharif, 2007), pattern recognition (Rassou & Kotropoulos, 2011; Yuan & Kuo, 2008), bioinformatics (Bhattacharya & De, 2010; Macintyre, Bailey, Gustafsson, Haviv, & Kowalczyk, 2010; Zheng, Yoon, & Lam, 2014), construction management (Cheng & Leu, 2009), marketing (Kim & Ahn, 2008; Kuo, An, Wang, & Chung, 2006), document clustering (Jun, Park, & Jang, 2014), intrusion detection (Jun et al., 2014), healthcare (Gunes, Polat, & Sebnem, 2010; Hung, Chen, Yang, & Deng, 2013) and information retrieval (Chan, 2008; Dhanapal, 2008).

Clustering algorithms can generally be divided into two categories; hierarchical clustering and partitioning clustering (Han, 2005). Hierarchical clustering groups objects into tree-like structure using bottom-up or top-down approaches. Our research however focuses on partition clustering, which decomposes the data set into a several disjoint clusters that are optimal in terms of some predefined criteria.

There many algorithms have been proposed in literature to solve the clustering problems. The K-means algorithm is the most popular and widely used algorithm in partitional clustering. Although, K-means is very fast and simple algorithm, it suffers from two major drawbacks. Firstly, the performance of K-means algorithm is highly dependent on the initial values of cluster centers. Secondly, the objective function of the K-means is non-convex and it may contain many local minima. Therefore, in the process of minimizing the objective function, the solution might easily converge to a local minimum rather than a global minimum (Selim & Ismail, 1984). K-means++ algorithm was proposed by Arthur and Vassilvitskii (2007), which introduces a cluster centers initialization procedure to tackle the initial centers sensitivity problem of
a standard K-means. However, it too suffers from a premature convergence to a local optimum.

In order to alleviate the local minima problem, many heuristic clustering approaches have been proposed over the years. For instance, Selim and Al-Sultan (1991) proposed a simulated annealing approach for solving clustering problems. A tabu search method which combines new procedures called packing and releasing was employed to avoid local optima in clustering problems (Sung & Jin, 2000). Genetic algorithm based clustering method was introduced by Maulik and Bandypadhyay (2000) to improve the global searching capability. In Fathian, Amiri, and Maroosi (2007), a honey-bee mating optimization was applied for solving clustering problems. Shelokar, Jayaraman, and Kulkarni (2004) proposed an ant colony optimization (ACO) for clustering problems. A particle swarm optimization based approach (PSO) for clustering was introduced by Chen and Ye (2004) and Cura (2012). A hybrid technique for clustering called K-NM-PSO, which combines the K-means, Nedler–Mead simplex and PSO was proposed by Kao, Zahara, and Kao (2008). Zhang, Ouyang, and Ning (2010) proposed an artificial bee colony approach for clustering. More recently, black hole (BH) optimization algorithm (Hatamlou, 2013) was introduced to solve clustering problems. Although these heuristic algorithms address the flaws of K-means but they still suffer several drawbacks. For example, most of these heuristic algorithms are typically very slow to find optimum solution. Furthermore, these algorithms are computationally expensive for large problems.

Cohort intelligence (CI) is a novel optimization algorithm proposed recently by Kulkarni, Durugkar, and Kumar (2013). This algorithm was inspired from natural and society tendency of cohort individuals/candidates of learning from one another. The learning refers to a cohort candidate’s effort to self-supervise its behavior and further adapt to the behavior of other candidate which it tends to follow. This makes every candidate to improve/ evolve its own and eventually the entire cohort behavior. CI was tested with several standard problems and compared with other optimization algorithms such as sequential quadratic programming (SQP), chaos PSO (CPSO), robust hybrid PSO (RHPSO) and linearly decreasing weight PSO (LDWPSO). CI has been proven to be computationally comparable and even better performed in terms of quality of solution and computational efficiency when compared with these algorithms. These comparisons can be found in the seminal paper on CI (Kulkarni et al., 2013). However, for clustering problems, as the number of clusters and dimensionality of data increase, CI might converge very slowly and trapped in local optima. Recently, many researchers have incorporated mutation operator into their algorithm to solve combinatorial optimizing problems. Several new variants of ACO algorithms have been proposed by introducing mutation to the traditional ACO algorithms and achieve much better performance (Lee, Su, Chuang, & Liu, 2008; Zhao, Wu, Zhao, & Quan, 2010). Stacey, Jancic, and Grundy (2003) and Zhao et al. (2010) also have integrated mutation into the standard PSO scheme, or modifications of it. In order to mitigate the short-comings of CI, we present a modified cohort intelligence (MCI) by incorporating mutation operator into CI to enlarge the searching range and avoid early convergence. Finally, to utilize the benefits of both K-means and MCI, we propose a new hybrid K-MCI algorithm for clustering. In this algorithm, K-means is applied to improve the candidates’ behavior that generated by MCI at each iteration before going through the mutation process of MCI. The new proposed hybrid K-MCI is not only able to produce a better quality solutions but it also converges more quickly than other heuristic algorithms including CI and MCI. In summary, our contribution in this paper is twofold:

1. Present a modified cohort intelligence (MCI).

This paper is organized as follows: Section 2 contains the description of the clustering problem and K-means algorithm. In Sections 3 and 4, the details of cohort intelligence and the modified cohort intelligence are explained. In Section 5, we discussed the hybrid K-MCI algorithm and its application to clustering problems. Section 6 presents the experimental results that prove our proposed method outperforms other methods. Finally, we conclude and summarize the paper in Section 7.

2. The clustering problem and K-means algorithm

Let \( R = \{Y_1, Y_2, \ldots, Y_N\} \) where \( Y_i \in \mathbb{R}^D \) be a set of \( N \) data objects to be clustered and \( S = \{X_1, X_2, \ldots, X_K\} \) be a set of \( K \) centers. In clustering, each data in set \( R \) will be allocated in one of the \( K \) clusters in such a way that it will minimize the objective function. The objective function, intra-cluster variance is defined by the sum of squared Euclidean distance between each object \( Y_i \) and the center of the cluster \( X_j \) which it belongs. This objective function is given by:

\[
F(X, Y) = \sum_{i=1}^{N} \min \left\{ \|Y_i - X_j\|^2 \right\}, \quad j = 1, 2, \ldots, K
\]  

(1)

Also,

- \( X_j \neq \emptyset, \forall j \in \{1, 2, \ldots, K\} \)
- \( X_i \cap X_j = \emptyset, \forall i \neq j \) and \( i, j \in \{1, 2, \ldots, K\} \)
- \( \bigcup_{j=1}^{K} X_j = R \)

In partitional clustering, the main goal of K-means algorithm is to determine centers of \( K \) clusters. In this research, we assume that the number of clusters \( K \) is known prior to solving the clustering problem. The following are the main steps of K-means algorithm:

1. Randomly choose \( K \) cluster centers of \( X_1, X_2, \ldots, X_K \) from data set \( R = \{Y_1, Y_2, \ldots, Y_N\} \) as the initial centers.
2. Assign each object in set \( R \) to the closest centers.
3. When all objects have been assigned, recalculate the positions of the \( K \) centers.
4. Repeat Steps 2 and 3 until a termination criterion is met (the maximum number of iterations reached or the means are fixed).

Arthur and Vassilvitskii (2007) introduced a specific way of choosing the initial centers for K-means algorithm. The procedure of K-means++ algorithm is outlined below:

1. Choose one center \( X_1 \), uniformly at random from \( R \).
2. For each data point \( Y_i \), compute \( D(Y_i) \), the distance between \( Y_i \) and the nearest center that has already been chosen.
3. Take new center \( X_j \), choosing \( Y \in R \) with probability \( \frac{D(Y)^2}{\sum_{i \in R} D(Y_i)^2} \).
4. Repeat Steps 2 and 3 until \( K \) centers have been chosen.
5. Now that the initial centers have been chosen, proceed using standard K-means clustering.

3. Cohort intelligence

Cohort intelligence (CI) is a new emerging optimization algorithm, which is inspired from natural and society tendency of cohort candidates of learning from one another. The term cohort refers to a group of candidates competing and interacting with one another to achieve some individual goal which is inherently common to all the candidates. Each candidate tries to improve its own behavior by observing every other candidates in a cohort. Every candidate might follow certain behavior in the cohort which
according to itself may result improvement in its own behavior. This allows every candidate to learn one another and improves cohort’s behavior. If the candidates behavior does not improve considerably after a number of iterations, the cohort behavior is considered saturated.

For instance, a general unconstrained problem (in minimization sense) is given by:

\[
\min f(x) = f(x_1, x_2, \ldots, x_n),
\]

The sampling interval, \(\Psi_t\) is given by \(x_{\text{min}}^i \leq x_i \leq x_{\text{max}}^i, i = 1, 2, \ldots, N\).

Assume the objective function \(f(x)\) acts as the behavior of an individual candidate in a cohort, whereby the individual will naturally tries to enrich itself by modifying its qualities/features, \(X = [x_1, x_2, \ldots, x_n]\). In a cohort with \(C\) number of candidates, every individual candidate \(c = 1, 2, \ldots, C\) has its own set features \(X^c = [x^c_1, x^c_2, \ldots, x^c_n]\) which makes the behavior of \(f(X^c)\). The individual behavior of each candidate \(c\) will be observed by every other candidate as well as by itself within that cohort. To be more specific, a candidate has a natural tendency to follow \(f^t(X^c)\) if it is better than \(f^t(X^c)\), i.e. \(f^t(X^c) < f^t(X^c)\). Since \(f^t(X^c)\) is better than \(f^t(X^c)\), the candidate will tend to follow the features of \(f^t(X^c)\), which is given by \(X^c = [x^c_1, x^c_2, \ldots, x^c_n]\) with certain variations \(t\).

The following describes the implementation of CI:

Step 1: Initialize the number of candidates \(C\), sampling interval for each quality, \(\Psi\), sampling interval reduction factor \(r \in [0, 1]\), convergence parameter \(\epsilon\), number of iterations \(n\) and number of variations \(t\).

Step 2: The probability of selecting the behavior \(f^t(X^c)\) of every associated candidate \(c\) is calculated using:

\[
P^t = \frac{1}{f^t(X^c)}
\]

Step 3: Every candidate generates a random number \(\text{rand} \in [0,1]\) and using the roulette wheel approach decides to follow corresponding behavior \(f^t(X^c)\) and its features \(X^c = [x^c_1, x^c_2, \ldots, x^c_n]\). The superscript indicates that the behavior is selected by the candidate and not known in advance. The roulette wheel approach could be most appropriate as it provides chance to every behavior in the cohort to get selected based purely on its quality. In addition, it also may increase the chances of any candidate to select the better behavior as the associated probability \(P^t\) presented in Eq. (2) is directly proportional to the quality of the behavior \(f^t(X^c)\). In other words, the better the quality, the higher the probability of being followed by the candidates in the cohort.

Step 4: Every candidate shrinks the sampling interval \(\Psi^c\) for its every features \(x^c_i\) to its local neighborhood. This is performed as follows:

\[
\Psi^c_i \in [x^c_i - (1/2) ||\Psi_i|| , x^c_i + (1/2) ||\Psi_i||]
\]

where \(\Psi_i = (||\Psi_i||) \times r\).

Step 5: Each candidate samples \(C\) qualities from within the updated sampling interval \(\Psi^c\) for its every features \(x^c_i\) and computes a set of associated \(t\) behaviors, i.e. \(F^t = f(X^c), f(X^c), \ldots, f(X^c)\) and selects the best behavior \(f^t(X^c)\) from \(F^t\). This makes cohort with \(C\) candidates updates its behavior and can be expressed as \(F = f(X^c), f(X^c), \ldots, f(X^c)\).

Step 6: If there is no significant improvement in the behavior \(f^t(X^c)\) of every candidate in the cohort, the cohort behavior could be considered saturated. The difference between the individual behaviors is not very significant for successive considerable number of iterations, if:

\[
\max(f^t(X^c^0) - \max(f^t(X^c)^{n-1}) \leq \epsilon, \quad \text{and}
\]

\[
\min(f^t(X^c^0) - \min(f^t(X^c)^{n-1}) \leq \epsilon, \quad \text{and}
\]

\[
\max(f^t(X^c^0) - \min(f^t(X^c)^{n}) \leq \epsilon
\]

Step 7: Accept any of the behaviors from current set of behaviors in the cohort as the final objective function value \(f(S)\) and stop if either of the two criteria listed below is valid or else continue to Step 2:

1. If maximum number of iterations exceeded.
2. If cohort saturates to the same behavior by satisfying the conditions of Eqs. (4)--(6).

4. Modified cohort intelligence

In this paper, we present a modified cohort intelligence (MCI) to improve the accuracy and the convergence speed of CI. Premature convergence may arise when the cohort converges to a local optimum or the searching process of algorithm is very slow. Therefore, we introduced a mutation mechanism to CI in order to enlarge the searching range, expand the diversity of solutions and avoid early convergence.

Assume for \(i\)th iteration, a candidate in a particular cohort is represented by a set of \(K\) number of cluster centers,

\[
S = [X_1, X_2, \ldots, X_5, X_6]^b, \quad (b = K \times D)
\]

Each data object \(X\) is assigned one of the \(K\) clusters \(S\). Then, each candidate \(S\) in that cohort will undergo mutation process to generate candidate \(S_{\text{mut}}\) as following:

\[
S_{\text{mut}} = S^{m1} + \text{rand}() \ast (S^{m2} - S^{m3})
\]

Variables \(m1\), \(m2\) and \(m3\) are three candidates which are selected randomly from \(C\) candidates in such a way that \(m1 \neq m2 \neq m3 \neq c\).

The selected candidate would be:

\[
S_{\text{rand}} = [X_{\text{mut}, 1}, X_{\text{mut}, 2}, \ldots, X_{\text{mut}, b}]_{1:b}
\]

where \(z = 1, 2, \ldots, b, \text{rand}()\) is a random number between 0 and 1, \(g\) is a random number less than 1 and \(D\) is the dimensionality of data objects. Thus, the new features for candidate \(c\) in the \(i\)th iteration are selected based on its objective function:

\[
S_{\text{new}} = \begin{cases} S_{\text{rand}} & \text{if } f(S) \leq f(S_{\text{rand}}) \\ S_{\text{new}} & \text{otherwise} \end{cases}
\]

This mutation process is performed to other remaining candidates in cohort.

5. Hybrid K-MCI and its application for clustering

In this paper, we propose a novel algorithm referred to as the hybrid K-means modified cohort intelligence (K-MCI) for data clustering. In this algorithm, K-means is utilized to improve the candidates’ behavior generated by MCI. After a series run of K-means, then each candidate will go through the mutation process as described in Section 4. The new proposed algorithm benefits from
the advantages of both K-means and MCI. This combination allows the proposed algorithm to converge more quickly and achieve a more accurate solutions without getting trapped to a local optimum. The application of the hybrid K-MCI on the data clustering is presented in this section. In order to solve the clustering problem using the new proposed algorithm, following steps should be applied and repeated:

**Step 1**: Generate the initial candidates. The initial $C$ candidates are randomly generated as described below:

$$
\text{Candidates} = \begin{bmatrix}
S^1 \\
S^2 \\
\vdots \\
S^c \\
\vdots \\
S^C 
\end{bmatrix}
$$

(12)

$$S^i = [X^i_1, X^i_2, \ldots, X^i_K]$$

(13)

$$X^i_j = [x^i_1, x^i_2, \ldots, x^i_j]$$

(14)

where $c = 1, 2, \ldots, C, K$ is the number of clusters, $j = 1, 2, \ldots, K$ and $D$ is the dimensionality of cluster center $X^i_j$.

Thus,

$$S^i = [X^i_1, X^i_2, \ldots, X^i_K]_{1 \leq i \leq b}, \text{ where } b = K \times D$$

(15)

The sampling interval $\Psi_l$ is given by $x^i_{min} \leq x^i \leq x^i_{max}$ ($i = 1, 2, \ldots, b$), where $x^i_{min}$ and $x^i_{max}$ (each feature of center) are minimum and maximum value of each point belonging to the cluster $X^i_j$.

**Step 2**: Perform K-means algorithm for each candidate as described in Section 2.

**Step 3**: Perform mutation operation for each candidate as described in Section 4.

**Step 4**: The objective function $f(S^i)$ for each candidate is calculated using Eq. (1).

**Step 5**: The probability of selecting the behavior $f(S^i)$ of every candidate is calculated using Eq. (2).

**Step 6**: Every candidate generates a random number $rand [0,1]$ and by using the roulette wheel approach decides to follow corresponding behavior $f^i(S^i[l])$ and its features $S[l] = [x^i[l], x^i_{2[l]}, \ldots, x^i_{K[l]}]$. For example, candidate $c[1]$ may decide to follow behavior of candidate $f^i(S^i[1])$ and its features $S[1] = [x^i[1], x^i_{2[1]}, \ldots, x^i_{K[1]}]$.

**Step 7**: Every candidate shrinks the sampling interval $\Psi^i[l]$ for its every features $x^i[l]$ to its local neighborhood according to Eq. (3)

**Step 8**: Each candidate samples $t$ qualities from within the updated sampling interval of its selected features $x^i[l]$. Then, each candidate computes the objective function for these $t$ behaviors and selects the best behavior $f^i(S^i)$ from this set. For instance with $t = 15$, candidate $c[1]$ decides to follow the behavior of candidate $f^i(S^i[2])$ and its features $S[2] = [x^i_{1[2]}, x^i_{2[2]}, \ldots, x^i_{K[2]}]$. Then, candidate $c[1]$ will sample 15 qualities from its updated sampling interval features of $X^i_{2[c]}$. Next, candidate $c[1]$ will compute the objective function of its behaviors according to Eq. (1), i.e. $F^i[1] = f(S^i[1]^1, f(S^i[1]^2, \ldots, f(S^i[1]^{15})$ and select the best behavior $f(S^i[1])$ from within this set.

**Step 9**: Accept any of the $c$ behaviors from current set of behaviors in the cohort as the final objective function value $f(S)$ and its features $S^i = [x^i_1, x^i_2, \ldots, x^i_K]$ and stop if either of the two criteria listed below is valid or else continue to Step 2:

1. If maximum number of iterations exceeded.
2. If cohort saturates to the same behavior by satisfying the conditions given by Eqs. (4)–(6).

The flow chart of the hybrid K-MCI is illustrated in Fig. 2

6. Experiment results

Six real data sets are used to validate our proposed algorithm. Each data set from UCI Machine Learning Repository has a different number of clusters, data objects and features as described below (Bache & Lichman, 2013):

- **Iris data set** ($N = 150, D = 4, K = 3$): which consists of three different species of Iris flowers: Iris Setosa, Iris Versicolour and Iris Virginica. For each species, 50 samples with four features (sepal length, sepal width, petal length, and petal width) were collected.

- **Wine data set** ($N = 178, D = 13, K = 3$): This data set are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivators: class 1 (59 instances), class 2 (71 instances), and class 3 (48 instances). The analysis determined the quantities of 13 features found in each of the three types of wines. These 13 features are alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non-flavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline.

- **Glass data set** ($N = 214, D = 9, K = 6$): which consists of six different types of glass: building windows float processed (70 objects), building windows non-float processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), tableware (9 objects), and headlamps (29 objects). Each type of glass has nine features, which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron.

- **Breast Cancer Wisconsin data set** ($N = 683, D = 9, K = 2$): This data set contains 683 objects. There are two categories: malignant (444 objects) and benign (239 objects). Each type of class consists of nine features, which includes clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses.

- **Vowel data set** ($N = 871, D = 3, K = 6$): which consist of 871 Indian Telugu vowels sounds. There are six-overlapping vowel classes: a (72 instances), e (89 instances), i (172 instances), u (151 instances), A (207 instances) and o (180 instances). Each class has three input features corresponding to the first, second, and third vowel frequencies.
Contraceptive Method Choice data set \( (N = 1473, D = 9, K = 3) \): This data set is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who either were not pregnant or did not know if they were at the time of interview. The problem is to predict the choice of current contraceptive method (no use has 629 objects, long-term methods have 334 objects, and short-term methods have 510 objects) of a woman based on her demographic and socioeconomic characteristics.

The performance of our proposed algorithm on these selected data set is compared with several typical stochastic algorithms.
such as the CI, MCI, ACO (Shelokar et al., 2004), PSO (Kao et al., 2008), SA (Niknam & Amir, 2010; Selim & Alsultan, 1991), GA (Maulik & Bandyopadhyay, 2000), TS (Niknam & Amir, 2010), HBMO (Fathian & Amir, 2008), K-means and K-means++. We have utilized two criteria to evaluate the performance of these algorithms: (i) the intra-cluster distances, as defined in Eq. (1) and (ii) the number of fitness function evaluation (NFE). For the first criteria, numerically smaller the value of the intra-cluster distances indicates higher the quality of the clustering. As for the second criteria, NFE represents the number of times that the clustering algorithm has calculated the objective function equation (1) to reach the optimal solution. The smaller NFE value indicates the high convergence speed of the considered algorithm.

The required parameters for the implementation of hybrid K-MCI, MCI and CI for clustering are shown in Table 1. The algorithms are implemented with Matlab 8.0 on a Windows platform using Intel Core i7-3770, 3.4 GHz and 8 GB RAM computer. Table 2 shows the summary of the intra-cluster distances obtained by the clustering algorithms on the selected data sets. The results are best, average, worst and the standard deviation of solutions over 20 independent runs. The NFE criteria in Table 2 indicates convergence speed of the respective algorithms.

The simulations results given in Table 2, shows that our proposed method performs much better than other methods for all test data sets. Our proposed method is able to achieve the best optimal value with a smaller standard deviation compared to other methods. In short, the results highlighted the precision and robustness of the proposed K-MCI compared to other algorithms including...
CI and MCI. For Iris data set, K-MCI and MCI algorithm are able to converge to global optimum of 96.5554 for each run, while the best solutions for CI, K-Means, K-means++, GA, SA, TS, ACO, HBMO and PSO are 96.6557, 97.3259, 97.3259, 113.9865, 97.4573, 97.3659, 97.1007, 96.752 and 96.8942. The standard deviation for K-MCI is zero, which is much less than other methods. K-MCI is also able to achieve the best global result and has a better average and worst result for the Wine data set compared to other methods. As for CMC data set, K-MCI has the best solution of 5693.73, while the best solutions for CI, MCI, K-Means, K-means++, GA, SA, TS, ACO, HBMO and PSO are 5695.33, 5694.28, 5703.20, 5703.20, 5705.63, 5849.03, 5885.06, 5701.92, 5699.26 and 5700.98. Furthermore, K-MCI has a much smaller standard deviation than the other methods for CMC data set. For vowel data set, our proposed method also manages to achieve best, average, worst solution and standard deviation of 148967.24, 148987.55, 149048.58 and 36.086. These obtained values are much smaller than other methods.

We notice the effect of applying mutation operator to CI by comparing the results between MCI and CI from Table 2. For instance, MCI has achieved a best, average, worst solutions of 16295.16, 16296.51 and 16297.98 with a standard deviation of 0.907 for Wine data set while CI has obtained best, average, worst solutions of 16298.01, 16300.98 and 16305.60 with a standard deviation of 2.118. Thus, by applying mutation operator, MCI is able to produce a better quality solution compared to the original CI.

The simulation results from Table 2 for K-MCI, MCI and CI points out the advantages of hybridizing K-means into MCI. The best global solution of K-MCI, MCI and CI for the Wine data set are 16292.44, 16295.16 and 16298.01. These results prove that the K-MCI provides a higher clustering quality than the standalone MCI and CI. Besides improving the clustering quality, the combination of K-means with MCI, will further enhance the convergence characteristics. CI and MCI need 17,500 and 16,500 function evaluations, respectively to obtain the best solution for Wine data set. On the other hand, K-MCI only takes 6250 function evaluations to achieve the best optimal solution for the same data set. Hence, K-MCI converges to optimal solution very quickly. Although stand-alone K-means and K-means++ algorithms converge much faster than other algorithms including K-MCI, they have a tendency to prematurely converge to a local optimum. For instance, K-means++ algorithm only needs 261 function evaluations to obtain the best solution for Wine data set but these solution results are suboptimal.

In summary, the simulation results from Table 2 validates that our proposed method is able to attain a better global solution with a smaller standard deviation and fewer numbers of function evaluations for clustering. Finally, we have included Tables 3–5 to illustrate the best centers found by K-MCI in the test data.

### Table 4
The achieved best centers on the glass and vowel data set.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Center 1</th>
<th>Center 2</th>
<th>Center 3</th>
<th>Center 4</th>
<th>Center 5</th>
<th>Center 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>1.52434</td>
<td>1.51956</td>
<td>1.51362</td>
<td>1.52132</td>
<td>1.51933</td>
<td>1.51567</td>
</tr>
<tr>
<td></td>
<td>0.01215</td>
<td>0.45229</td>
<td>0.65548</td>
<td>3.51952</td>
<td>3.52765</td>
<td>0.06326</td>
</tr>
<tr>
<td></td>
<td>1.12869</td>
<td>1.53305</td>
<td>3.13123</td>
<td>1.01524</td>
<td>1.36555</td>
<td>2.21016</td>
</tr>
<tr>
<td></td>
<td>71.98256</td>
<td>73.01401</td>
<td>70.50441</td>
<td>71.89317</td>
<td>72.85826</td>
<td>73.25524</td>
</tr>
<tr>
<td></td>
<td>0.19252</td>
<td>0.38472</td>
<td>5.33024</td>
<td>0.21094</td>
<td>0.57913</td>
<td>0.02744</td>
</tr>
<tr>
<td></td>
<td>0.23039</td>
<td>0.00433</td>
<td>0.67322</td>
<td>0.03588</td>
<td>0.00837</td>
<td>1.02698</td>
</tr>
<tr>
<td></td>
<td>0.15156</td>
<td>0.00599</td>
<td>0.01490</td>
<td>0.04680</td>
<td>0.06182</td>
<td>0.00382</td>
</tr>
<tr>
<td>Vowel</td>
<td>506.98650</td>
<td>623.71854</td>
<td>407.89515</td>
<td>439.24323</td>
<td>357.26154</td>
<td>375.45357</td>
</tr>
<tr>
<td></td>
<td>1839.66652</td>
<td>1309.59677</td>
<td>1018.05210</td>
<td>987.68488</td>
<td>2291.44000</td>
<td>2149.40364</td>
</tr>
<tr>
<td></td>
<td>2556.20000</td>
<td>2333.45721</td>
<td>2317.82688</td>
<td>2665.47618</td>
<td>2977.39697</td>
<td>2678.44208</td>
</tr>
</tbody>
</table>

### Table 5
The achieved best centers on the cancer data set.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Center 1</th>
<th>Center 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer</td>
<td>7.11701</td>
<td>2.88942</td>
</tr>
<tr>
<td></td>
<td>6.64106</td>
<td>1.12774</td>
</tr>
<tr>
<td></td>
<td>6.62548</td>
<td>1.20072</td>
</tr>
<tr>
<td></td>
<td>5.61469</td>
<td>1.16400</td>
</tr>
<tr>
<td></td>
<td>5.24061</td>
<td>1.99334</td>
</tr>
<tr>
<td></td>
<td>8.10094</td>
<td>1.12116</td>
</tr>
<tr>
<td></td>
<td>6.07818</td>
<td>2.00537</td>
</tr>
<tr>
<td></td>
<td>6.02147</td>
<td>1.10133</td>
</tr>
<tr>
<td></td>
<td>2.32582</td>
<td>1.03162</td>
</tr>
</tbody>
</table>

### 7. Conclusion
CI is a new emerging optimization method, which has a great potential to solve many optimization problems including data clustering. However, CI may converge very slowly and prematurely converge to local optima when the dimensionality of data and number of clusters increase. With the purpose of assuaging these drawbacks, we proposed modified CI (MCI) by implementing mutation operator into CI. It outperforms CI in terms of both quality of solution and the convergence speed. Finally in this paper, we proposed a novel hybrid K-MCI algorithm for data clustering. This new algorithm tries to exploit the merits of two algorithms simultaneously, where K-means is utilized to improve the candidates’ behavior at each iteration before these candidates are given back again to MCI for optimization. This combination of K-means and MCI allows our proposed algorithm to convergence more quickly and prevents it from falling to local optima. We tested our proposed method using the standard data sets from UCI Machine Learning Repository and compared our results with six state-of-art clustering methods. The experimental results indicate that our algorithm can produce a higher quality clusters with a smaller standard deviation than other algorithms including K-MCI, K-means++, GA, SA, TS, ACO, HBMO and PSO. Furthermore, CI and MCI need 17,500 and 16,500 function evaluations, respectively to obtain the best solution for Wine data set. On the other hand, K-MCI only takes 6250 function evaluations.
number of clusters. We may combine MCI with other heuristic algorithms to solve clustering problems, which can be seen as another research direction. Finally, our proposed algorithm may be applied to solve other practically important problems such as image segmentation (Bhandari, Singh, Kumar, & Singh, 2014), travelling salesman problem (Albayrak & Allahverdi, 2011), process planning and scheduling (Seker, Erol, & Botsali, 2013) and load dispatch of power system (Zhisheng, 2010).

Acknowledgment

This work was supported by the HIR-MOHE Grant No. UM/C/HIR/MOHE/ENG/42.

References


