Chapter 4
Modified Cohort Intelligence for Solving Machine Learning Problems

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 another than objects in the different cluster 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 chapter, we present an efficient hybrid evolutionary data clustering algorithm referred as to 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.

4.1 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 [1, 2]. The term unsupervised means that grouping is establish 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 [3, 4], image processing [5–8], data mining [9, 10], pattern recognition [11, 12],

bioinformatics [13–15], construction management [16], marketing [17, 18], document clustering [19], intrusion detection [19], healthcare [20, 21] and information retrieval [22, 23].

Clustering algorithms can generally be divided into two categories; hierarchical clustering and partitional clustering [24]. 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 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 [25]. K-means++ algorithm was proposed by Arthur and Vassilvitskii [26], 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, [27] 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 [28]. Genetic algorithm based clustering method was introduced by Maulik and Bandyopadhyay [29] to improve the global searching capability. Fathian et al. [30] proposed a honey-bee mating optimization approach for solving clustering problems. Shelokar et al. [31] proposed an ant colony optimization (ACO) for clustering problems. A particle swarm optimization based approach (PSO) for clustering was introduced by Chen and Ye [32] and Cura [33]. A hybrid technique for clustering called KNM-PSO, which combines the K-means, Nedler-Mead simplex and PSO was proposed by Kao et al. [34]. Zhang et al. [35] proposed an artificial bee colony approach for clustering. More recently, black hole (BH) optimization algorithm [36] 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 et al. [37]. 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 [37]. However, for clustering problems, as the number of clusters and dimensionality of data increase, CI might converge 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 [38, 39]. Stacey et al. [40] and Zhao et al. [39] 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 chapter is twofold:

1. Present a modified cohort intelligence (MCI).
2. Present a new hybrid K-MCI algorithm for data clustering.

4.2   The Clustering Problem and K-Means Algorithm

Let $R = [Y_1, Y_2, \ldots, Y_N]$, where $Y_i \in \Re^D$, be a set of N data objects to be clustered and $S = [X_1, X_2, \ldots, X_K]$ be a set of K clusters. 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 as 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:

$$F(X, Y) = \sum_{i=1}^{N} Min\left\{ \left\| Y_i - X_j \right\|^2 \right\}, \quad j = 1, 2, \ldots, K \qquad (4.1)$$

Also,

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

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:

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

Arthur and Vassilvitskii [26] introduced a specific way of choosing the initial centers for the K-means algorithm. The procedure of the K-means++ algorithm is outlined below:

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

4.3   Modified Cohort Intelligence

In this chapter, 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 ith iteration, a candidate in a particular cohort is represented by a set of K number of cluster centers, $S^c = [X_1^c, X_2^c, \ldots, X_j^c, \ldots, X_K^c]$, where $c = 1, 2, \ldots, C$ and X_j^c represents the cluster's center. For an example, Fig. 4.1 depicts a candidate solution of a problem with three clusters, $K = 3$ and all the data objects have four dimensions, $D = 4$. Thus, the candidate solution illustrated in Fig. 4.1 can be represented by $S^c = [x_1^c, x_2^c, \ldots, x_b^c]_{1 \times b}$, where $b = K \times D$. Then, each candidate S^c in that cohort will undergo mutation process to generate mutant candidate S_{mut}^c as following:

$$S_{mut}^c = S^{m1} + rand(.) * (S^{m2} - S^{m3}) \tag{4.2}$$
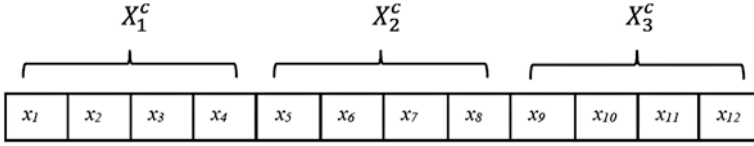
Fig. 4.1 Example of a candidate solution

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

$$S_{mut}^c = \left[x_{mut,1}^c, x_{mut,2}^c, \ldots, x_{mut,b}^c \right]_{1 \times b} \tag{4.3}$$

The selected candidate would be:

$$S_{trial}^c = \left[x_{trial,1}^c, x_{trial,2}^c, \ldots, x_{trial,b}^c \right]_{1 \times b} \tag{4.4}$$

$$x_{trial,z}^c = \begin{cases} x_{mut,z}^c \\ x_z^c \end{cases} \quad if \quad rand(.) < \gamma \tag{4.5}$$

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

$$S_{new}^c = \begin{cases} S^c & if \quad f(S^c) \leq f(S_{trial}^c) \\ S_{trial}^c & otherwise \end{cases} \tag{4.6}$$

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

4.4 Hybrid K-MCI and Its Application for Clustering

In this chapter, 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 Sect. 4.3. 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:

$$Candidates = \begin{bmatrix} S^1 \\ S^2 \\ \vdots \\ S^c \\ \vdots \\ S^C \end{bmatrix} \qquad (4.7)$$

$$S^c = [X_1^c, X_2^c, \ldots, X_K^c] \qquad (4.8)$$

$$X_j^c = [x_1^c, x_2^c, \ldots, x_D^c] \qquad (4.9)$$

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_j^c.
Thus,

$$S^c = \left[x_1^c, x_2^c, \ldots, x_i^c, \ldots, x_b^c \right]_{1 \times b}, \quad where \ \ b = K \times D \qquad (4.10)$$

The sampling interval ψ_i is given by $x_i^{c,\min} \le x_i \le x_i \le x_i^{c,\max}$, where, $x_i^{c,\min}$ and $x_i^{c,\max}$ (each feature of center) are minimum and maximum value of each point belonging to the cluster X_j^c.

Step 2. Perform K-means algorithm for each candidate as described in Sect. 4.2.

Step 3. Perform mutation operation for each candidate as described in Sect. 4.3.

Step 4. The objective function $f(S^c)$ for each candidate is calculated.

Step 5. The probability of selecting the behavior $f^*(S^c)$ of every candidate is calculated.

Step 6. Every candidate generates a random number *rand* [0, 1] and by using the roulette wheel approach decides to follow corresponding behavior $f^*(S^{c[?]})$ and its features $S^{c[?]} = [x_1^{c[?]}, x_2^{c[?]}, \ldots, x_b^{c[?]}]$. For example, candidate c [1] may decide to follow behavior of candidate $f^*(S^{c[2]})$ and its features $S^{c[2]} = [x_1^{c[2]}, x_2^{c[2]}, \ldots, x_b^{c[2]}]$.

Step 7. Every candidate shrinks the sampling interval $\psi^{c[?]}$ for its every features $x_i^{c[?]}$ to its local neighborhood.

Step 8. Each candidate samples t qualities from within the updated sampling interval of its selected features, $x_i^{c[?]}$. Then, each candidate computes the

objective function for these t behaviors and selects the best behavior $f^*(S^c)$ from this set. For instance with $t = 15$, candidate c [1] decides to follow the behavior of candidate $f^*(S^{c[2]})$ and its features $S^{c[2]} = [x_1^{c[2]}, x_2^{c[2]}, \ldots, x_b^{c[2]}]$. Then, candidate c [1] will sample 15 qualities from its updated sampling interval features of $x_i^{c[2]}$. Next, candidate c [1] will compute the objective function of its behaviors according, i.e. $F^{c[1]} = [f(S^{c[1]})^1, f(S^{c[1]})^2, \ldots, f(S^{c[1]})^{15}]$ and selects the best behavior $f^*(S^{c[1]})$ 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 $S^c = [x_1^c, x_2^c, \ldots, x_b^c]$ 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 convergence condition.

The flow chart of the hybrid K-MCI is illustrated in Fig. 4.2.

4.5   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 [41]:

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, nonflavanoid 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

```
┌─────────────────────────────────────────────────────────┐
│   Initialize number of candidates C in the cohort,       │
│   quality variations t and set-up interval reduction     │
│   factor r                                               │
└─────────────────────────────────────────────────────────┘
                            │
        ┌───────────────────┼─────────────────────────────┐
        │      ┌─────────────────────────────────┐         │
        │      │     Generate initial candidates  │         │
        │      └─────────────────────────────────┘         │
        │                   │                               │
        │      ┌─────────────────────────────────┐         │
        │      │ Perform K-means algorithm for    │         │
        │      │        each candidate            │         │
        │      └─────────────────────────────────┘         │
        │                   │                               │
        │      ┌─────────────────────────────────┐         │
        │      │ Perform mutation operation for   │         │
        │      │        each candidate            │         │
        │      └─────────────────────────────────┘         │
        │                   │                               │
        │      ┌─────────────────────────────────┐         │
        │      │ Calculate objection function for │         │
        │      │        every candidate           │         │
        │      └─────────────────────────────────┘         │
```

Initialize number of candidates C in the cohort, quality variations t and set-up interval reduction factor r

Generate initial candidates

Perform K-means algorithm for each candidate

Perform mutation operation for each candidate

Calculate objection function for every candidate

Using roulette wheel approach, every candidate selects behavior to follow from within the C available choices

Every candidate shrinks the sampling interval of every quality i based on whether condition of saturation is satisfied

Every candidate forms t behaviors by sampling the qualities from within the updated sampling intervals

Every candidate follows the best behavior from within its t behaviors

Cohort behavior saturated?

Converged?

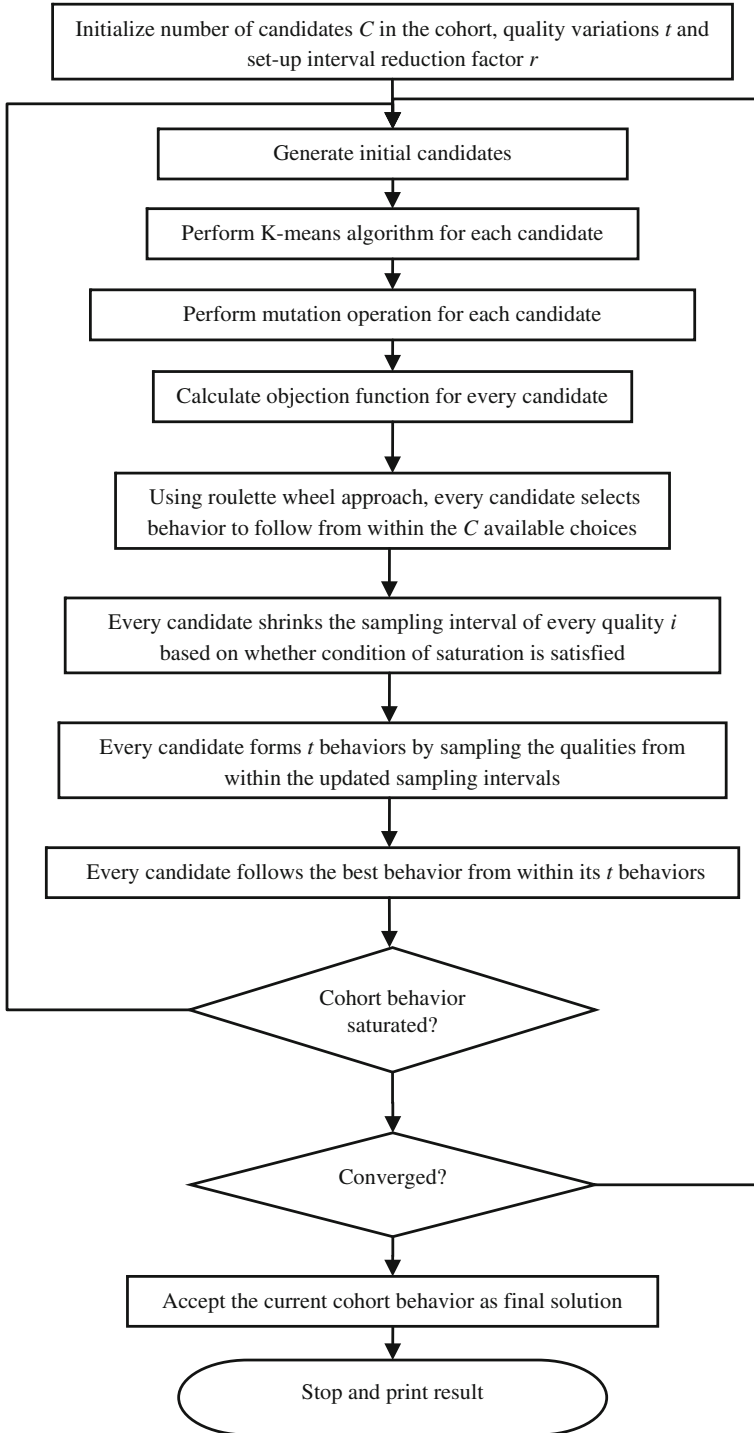Accept the current cohort behavior as final solution

Stop and print result

Fig. 4.2 The flow chart of the hybrid K-MCI

(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: δ (72 instances), a (89 instances), I (172 instances), u (151 instances), e (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 [31, 34], SA [27, 42], GA [29], TS [42], HBMO [43], K-means and K-means++. We have utilized two criteria to evaluate the performance of these algorithms: (i) the intra-cluster distances 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 is. As for the second criteria, 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 4.1. 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 4.2 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 4.2 indicates convergence speed of the respective algorithms. NFE is the number of times that the clustering algorithm has calculated the objective function to reach the optimal solution.

The simulations results given in Table 4.2, 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

Table 4.1 Parameters of hybrid K-MCI, MCI and CI for data clustering

Data	CI			MCI			K-MCI		
	t	v	r	t	v	r	t	v	r
Iris	5	15	0.95	5	15	0.95	5	15	0.92
Wine	5	15	0.95	5	15	0.95	5	15	0.7
Cancer	5	15	0.95	5	15	0.95	5	15	0.95
Vowel	5	15	0.99	5	15	0.99	5	15	0.98
CMC	5	15	0.99	5	15	0.99	5	15	0.99
Glass	5	15	0.99	5	15	0.99	5	15	0.98

Table 4.2 Simulation results for clustering algorithms

Dataset	Criteria	K-means	K-means++	GA	SA	TS	ACO	HBMO	PSO	CI	MCI	K-MCI
Iris	Best	97.3259	97.3259	113.9865	97.4573	97.3659	97.1007	96.7520	96.8942	96.6557	96.6554	96.6554
	Average	106.5766	98.5817	125.1970	99.9570	97.8680	97.1715	96.9531	97.2328	96.6561	96.6554	96.6554
	Worst	123.9695	122.2789	139.7782	102.0100	98.5694	97.8084	97.7576	97.8973	96.6570	96.6554	96.6554
	S.D.	12.938	5.578	14.563	2.018	0.53	0.367	0.531	0.347	0.0002	0	0
	NFE	80	71	38,128	5314	20,201	10,998	11,214	4953	7250	4500	3500
Wine	Best	16,555.68	16,555.68	16,530.53	16,473.48	16,666.22	16,530.53	16,357.28	16,345.96	16,298.01	16,295.16	16,292.44
	Average	17,251.35	16,816.55	16,530.53	17,521.09	16,785.45	16,530.53	16,357.28	16,417.47	16,300.98	16,296.51	16,292.7
	Worst	18,294.85	18,294.85	16,530.53	18,083.25	16,837.53	16,530.53	16,357.28	16,562.31	16,305.06	16,297.98	16,292.88
	S.D.	874.148	637.14	0	753.084	52.073	0	0	85.497	2.118	0.907	0.13
	NFE	285	261	33,551	17,264	22,716	15,473	7238	16,532	17,500	16,500	6250
Cancer	Best	2988.43	2986.96	2999.32	2993.45	2982.84	2970.49	2989.94	2973.5	2964.64	2964.4	2964.38
	Average	2988.99	2987.99	3249.46	3239.17	3251.37	3046.06	3112.42	3050.04	2964.78	2964.41	2964.38
	Worst	2999.19	2988.43	3427.43	3421.95	3434.16	3242.01	3210.78	3318.88	2964.96	2964.43	2964.38
	S.D.	2.469	0.689	229.734	230.192	232.217	90.5	103.471	110.801	0.094	0.007	0
	NFE	120	112	20,221	17,387	18,981	15,983	19,982	16,290	7500	7000	5000
CMC	Best	5703.2	5703.2	5705.63	5849.03	5885.06	5701.92	5699.26	5700.98	5695.33	5694.28	5693.73
	Mean	5704.57	5704.19	5756.59	5893.48	5993.59	5819.13	5713.98	5820.96	5696.01	5694.58	5693.75
	Worst	5705.37	5705.37	5812.64	5966.94	5999.8	5912.43	5725.35	5923.24	5696.89	5694.89	5693.8
	S.D.	1.033	0.955	50.369	50.867	40.845	45.634	12.69	46.959	0.482	0.198	0.014
	NFE	187	163	29,483	26,829	28,945	20,436	19,496	21,456	30,000	28,000	15,000

(continued)

Table 4.2 (continued)

Dataset	Criteria	K-means	K-means++	GA	SA	TS	ACO	HBMO	PSO	CI	MCI	K-MCI
Glass	Best	215.73	215.36	278.37	275.16	279.87	269.72	245.73	270.57	219.37	213.03	212.34
	Mean	218.7	217.56	282.32	282.19	283.79	273.46	247.71	275.71	223.31	214.08	212.57
	Worst	227.35	223.71	286.77	287.18	286.47	280.08	249.54	283.52	225.48	215.62	212.8
	S.D.	2.456	2.455	4.138	4.238	4.19	3.584	2.438	4.55	1.766	0.923	0.135
	NFE	533	510	199,892	199,438	199,574	196,581	195,439	198,765	55,000	50,000	25,000
Vowel	Best	149,398.66	149,394.56	149,513.73	149,370.47	149,468.26	149,395.6	149,201.63	148,976.01	149,139.86	148,985.35	148,967.24
	Mean	151,987.98	151,445.29	159,153.49	161,566.28	162,108.53	159,458.14	161,431.04	151,999.82	149,528.56	149,039.86	148,987.55
	Worst	162,455.69	161,845.54	165,991.65	165,986.42	165,996.42	165,939.82	165,804.67	158,121.18	150,468.36	149,102.38	149,048.58
	S.D.	3425.25	3119.751	3105.544	2847.085	2846.235	3485.381	2746.041	2881.346	495.059	43.735	36.086
	NFE	146	129	10,548	9423	9528	8046	8436	9635	15,000	13,500	7500

to other methods. In short, the results highlighted the precision and robustness of the proposed K-MCI as 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, KMCI 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 148,967.24, 148,987.55, 149,048.58 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 4.2. For instance, MCI has achieved a best, average, worst solutions of 16,295.16, 16,296.51 and 16,297.98 with a standard deviation of 0.907 for Wine data set while CI has obtained best, average, worst solutions of 16,298.01, 16,300.98 and 16,305.60 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 4.2 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 16,292.44, 16,295.16 and 16,298.01. These results prove that K-MCI will provide 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 standalone K-means

Table 4.3 The achieved best centers on Cancer data set	Dataset	Center 1	Center 2
	Cancer	7.11701	2.88942
		6.64106	1.12774
		6.62548	1.20072
		5.61469	1.16404
		5.24061	1.99334
		8.10094	1.12116
		6.07818	2.00537
		6.02147	1.10133
		2.32582	1.03162

Table 4.4 Achieved best centers on the glass and vowel data set

Dataset	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
Glass	1.52434	1.51956	1.51362	1.52132	1.51933	1.51567
	12.03344	13.25068	13.15690	13.74692	13.08412	14.65825
	0.01215	0.45229	0.65548	3.51952	3.52765	0.06326
	1.12869	1.53305	3.13123	1.01524	1.36555	2.21016
	71.98256	73.01401	70.50411	71.89517	72.85826	73.25324
	0.19252	0.38472	5.33024	0.21094	0.57913	0.02744
	14.34306	11.15803	6.73773	9.44764	8.36271	8.68548
	0.23039	0.00433	0.67322	0.03588	0.00837	1.02698
	0.15156	0.06599	0.01490	0.04680	0.06182	0.00382
Vowel	506.98650	623.71854	407.89515	439.24323	357.26154	375.45357
	1839.66652	1309.59677	1018.05210	987.68488	2291.44000	2149.40364
	2556.20000	2333.45721	2317.82688	2665.47618	2977.39697	2678.44208

Table 4.5 The archived best centers on the Iris, Wine and CMC data set

Dataset	Center 1	Center 2	Center 3
Iris	5.01213	5.93432	6.73334
	3.40309	2.79781	3.06785
	1.47163	4.41787	5.63008
	0.23540	1.41727	2.10679
Wine	13.81262	12.74160	12.50086
	1.83004	2.51921	2.48843
	2.42432	2.41113	2.43785
	17.01717	19.57418	21.43603
	105.41208	98.98807	92.55049
	2.93966	1.97496	2.02977
	3.21965	1.26308	1.54943
	0.34183	0.37480	0.32085
	1.87181	1.46902	1.38624
	5.75329	5.73752	4.38814
	1.05368	1.00197	0.94045
	2.89757	2.38197	2.43190
	1136.97230	687.01356	463.86513
CMC	43.64742	24.41296	33.50648
	2.99091	3.03823	3.13272
	3.44673	3.51059	3.55176
	4.59136	1.79036	3.65914
	0.80254	0.92502	0.79533
	0.76971	0.78935	0.69725
	1.82586	2.29463	2.10130
	3.42522	2.97378	3.28562
	0.10127	0.03692	0.06151
	1.67635	2.00149	2.11479

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 4.2 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 4.3, 4.4 and 4.5 to illustrate the best centers found by K-MCI in the test data.

4.6   Conclusion

CI is a new emerging optimization method, which has a great potential to solve many optimization problems including for data clustering. However, CI may converge slowly and prematurely converge to local optima when the dimensionality of data and number of cluster centers 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 solutions and the convergence speed. Finally in this chapter, we proposed a novel hybrid K-MCI algorithm for data clustering. This new algorithm tries to exploit the merits of the two algorithms simultaneously, where the K-means is utilized to improve the candidates of MCI 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 proposed algorithm can produce a higher quality clusters with a smaller standard deviation on the selected data set compare to other clustering methods. Moreover, the convergence speed to global optima of the proposed algorithm is better than other heuristic algorithms. In other words, our proposed method can be considered as an efficient and reliable method to find the optimal solution for clustering problems.

There are a number of future research directions can be considered to improve and extend this research. The computational performance is governed by parameters such as the sampling interval reduction factor r. Thus, a self-adaptive scheme can be introduced to fine tune the sampling interval reduction. In this research, we assumed the number of clusters are known a prior when solving the clustering problems. Therefore, we can further modify our algorithm to perform automatic clustering without any prior knowledge of 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 [44], traveling salesman problem [45], process planning and scheduling [46] and load dispatch of power system [47].

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**, 264323 (1999)
2. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics). Wiley-Interscience (2005)
3. Fan, S., Chen, L., Lee, W.-J.: Machine learning based switching model for electricity load forecasting. Energy Convers. Manag. **49**, 1331–1344 (2008)
4. Anaya, A.R., Boticario, J.G.: Application of machine learning techniques to analyse student interactions and improve the collaboration process. Expert Syst. Appl. **38**, 1171–1181 (2011)
5. Das, S., Konar, A.: Automatic image pixel clustering with an improved differential evolution. Appl. Soft Comput. **9**, 226236 (2009)
6. Siang Tan, K., Mat Isa, N.A.: Color image segmentation using histogram thresholding fuzzy c-means hybrid approach. Pattern Recogn. **44**, 1–15 (2011)
7. Portela, N.M., Cavalcanti, G.D., Ren, T.I.: Semi-supervised clustering for MR brain image segmentation. Expert Syst. Appl. **41**, 1492–1497 (2014)
8. Zhao, F., Fan, J., Liu, H.: Optimal-selection-based suppressed fuzzy c-means clustering algorithm with self-tuning non local spatial information for image segmentation. Expert Syst. Appl. **41**, 4083–4093 (2014)
9. Ci, S., Guizani, M., Sharif, H.: Adaptive clustering in wireless sensor networks by mining sensor energy data. Netw. Coverage Routing Schemes Wirel. Sens. Netw. **30**, 2968–2975 (2007)
10. Carmona, C., Ramrez-Gallego, S., Torres, F., Bernal, E., del Jesus, M., Garca, S.: Web usage mining to improve the design of an ecommerce website: Orolivesur.com. Expert Syst. Appl. **39**, 11243–11249 (2012)
11. Yuan, T., Kuo, W.: Spatial defect pattern recognition on semiconductor wafers using model-based clustering and bayesian inference. Eur. J. Oper. Res. **190**, 228–240 (2008)
12. Bassiou, N., Kotropoulos, C.: Long distance bigram models applied to word clustering. Pattern Recogn. **44**, 145158 (2011)
13. Bhattacharya, A., De, R.K.: Average correlation clustering algorithm (ACCA) for grouping of co-regulated genes with similar pattern of variation in their expression values. J. Biomed. Inform. **43**, 560–568 (2010)
14. Macintyre, G., Bailey, J., Gustafsson, D., Haviv, I., Kowalczyk, A.: Using gene ontology annotations in exploratory microarray clustering to understand cancer etiology. Pattern Recogn. Lett. **31**, 2138–2146 (2010)
15. Zheng, B., Yoon, S.W., Lam, S.S.: Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms. Expert Syst. Appl. **41**, 1476–1482 (2014)
16. Cheng, Y.-M., Leu, S.-S.: Constraint-based clustering and its applications in construction management. Expert Syst. Appl. **36**, 5761–5767 (2009)
17. Kuo, R., An, Y., Wang, H., Chung, W.: Integration of selforganizing feature maps neural network and genetic k-means algorithm for market segmentation. Expert Syst. Appl. **30**, 313–324 (2006)
18. Kim, K.-J., Ahn, H.: A recommender system using GA k-means clustering in an online shopping market. Expert Syst. Appl. **34**, 1200–1209 (2008)
19. Jun, S., Park, S.-S., Jang, D.-S.: Document clustering method using dimension reduction and support vector clustering to overcome sparseness. Expert Syst. Appl. **41**, 3204–3212 (2014)
20. Gunes, S., Polat, K., Yosunkaya, S.: Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. Expert Syst. Appl. **37**, 7922–7928 (2010)
21. Hung, Y.-S., Chen, K.-L.B., Yang, C.-T., Deng, G.-F.: Web usage mining for analysing elder self-care behavior patterns. Expert Syst. Appl. **40**, 775–783 (2013)
22. Chan, C.-C.H.: Intelligent spider for information retrieval to support mining-based price prediction for online auctioning. Expert Syst. Appl. **34**, 347–356 (2008)

23. Dhanapal, R.: An intelligent information retrieval agent. Know. Based Syst. **21**, 466–470 (2008)
24. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc. (2005)
25. Selim, S.Z., Ismail, M.A.: K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 81–87 (1984)
26. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms SODA '07, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2007)
27. Selim, S.Z., Alsultan, K.: A simulated annealing algorithm for the clustering problem. Pattern Recogn. **24**, 1003–1008 (1991)
28. Sung, C., Jin, H.: A tabu-search-based heuristic for clustering. Pattern Recogn. **33**, 849–858 (2000)
29. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recogn. **33**, 1455–1465 (2000)
30. Fathian, M., Amiri, B., Maroosi, A.: Application of honey-bee mating optimization algorithm on clustering. Appl. Math. Comput. **190**, 1502–1513 (2007)
31. Shelokar, P.S., Jayaraman, V.K., Kulkarni, B.D.: An ant colony approach for clustering. AnalyticaChimicaActa **509**, 187–195 (2004)
32. Chen, C.-Y., Ye, F.: Particle swarm optimization algorithm and its application to clustering analysis. In: IEEE International Conference on Networking, Sensing and Control, vol. 2, pp. 789–794 (2004)
33. Cura, T.: A particle swarm optimization approach to clustering. Expert Syst. Appl. **39**, 1582–1588 (2012)
34. Kao, Y.-T., Zahara, E., Kao, I.-W.: A hybridized approach to data clustering. Expert Syst. Appl. **34**, 1754–1762 (2008)
35. Zhang, C., Ouyang, D., Ning, J.: An artificial bee colony approach for clustering. Expert Syst. Appl. **37**, 4761–4767 (2010)
36. Hatamlou, A.: Black hole: a new heuristic optimization approach for data clustering. Inf. Sci. **222**, 175–184 (2013)
37. Kulkarni, A.J., Durugkar, I.P., Kumar, M.: Cohort intelligence: a self supervised learning behavior. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1396–1400 (2013)
38. Lee, Z.-J., Su, S.-F., Chuang, C.-C., Liu, K.-H.: Genetic algorithm with ant colony optimization (ga-aco) for multiple sequence alignment. Appl. Soft Comput. **8**, 55–78 (2008)
39. Zhao, N., Wu, Z., Zhao, Y., Quan, T.: Ant colony optimization algorithm with mutation mechanism and its applications. Expert Syst. Appl. **37**, 4805–4810 (2010)
40. Stacey, A., Jancic, M., Grundy, I.: Particle swarm optimization with mutation. In: The 2003 Congress on Evolutionary Computation, 2003 (CEC '03), vol. 2, pp. 1425–1430 (2003)
41. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences (2013)
42. Niknam, T., Amiri, B.: An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Appl. Soft Comput. **10**, 183–197 (2010)
43. Fathian, M., Amiri, B.: A honeybee-mating approach for cluster analysis. Int. J. Adv. Manuf. Technol. **38**, 809–821 (2008)
44. Bhandari, A.K., Singh, V.K., Kumar, A., Singh, G.K.: Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using kapurs entropy. Expert Syst. Appl. **41**, 3538–3560 (2014)
45. Albayrak, M., Allahverdi, N.: Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. Expert Syst. Appl. **38**, 1313–1320 (2011)
46. Seker, A., Erol, S., Botsali, R.: A neuro-fuzzy model for a new hybrid integrated process planning and scheduling system. Expert Syst. Appl. **40**, 5341–5351 (2013)
47. Zhisheng, Z.: Quantum-behaved particle swarm optimization algorithm for economic load dispatch of power system. Expert Syst. Appl. **37**, 1800–1803 (2010)