Jurnal Teknologi

Full paper

Data Analysis by Combining the Modified K-Means and Imperialist Competitive Algorithm

Mohammad Babrdelbonab[a,b*], Siti Zaiton Mohd Hashim[a], Nor Erne Nazira Bazin[a]

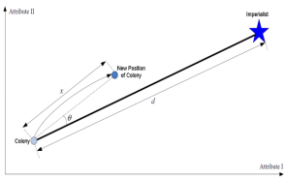
[a]*Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia*
[b]*Faculty of Computing, Islamic Azad University Bonab Branch*

*Corresponding author: bbmohammad2@live.utm.my

Article history

Received :1 January 2014
Received in revised form :
1 June 2014
Accepted :10 September 2014

Graphical abstract



Abstract

Data Clustering is one of the most used methods of data mining. The *k*-means Clustering Approach is one of the main algorithms in the literature of Pattern Recognition and Data Machine Learning which it very popular because of its simple application and high operational speed. But some obstacles such as the adherence of results to initial cluster centers or the risk of getting trapped into local optimality hinders its performance. In this paper, inspired by the Imperialist Competitive Algorithm based on the *k*-means method, a new approach is developed, in which cluster centers are selected and computed appropriately. The Imperialist Competitive Algorithm (ICA) is a method in the field of evolutionary computations, trying to find the optimum solution for diverse optimization problems. The underlying traits of this algorithm are taken from the evolutionary process of social, economic and political development of countries so that by partly mathematical modeling of this process some operators are obtained in regular algorithmic forms. The investigated results of the suggested approach over using standard data sets and comparing it with alternative methods in the literature reveals out that the proposed algorithm outperforms the *k*-means algorithm and other candidate algorithms in the pool.

Keywords: Data analysis; data clustering; k-means clustering; imperialist competitive algorithm

Abstrak

Pengelompokan data merupakan salah satu kaedah yang paling biasa digunakan dalam perlombongan data. Pendekatan kelompok *k-means* adalah salah satu daripada algoritma utama dalam literatur pengecaman corak dan pembelajaran data mesin yang sangat popular kerana fungsinya yang mudah dan kelajuan operasi yang tinggi. Tetapi beberapa halangan seperti pematuhan keputusan ke pusat-pusat kelompok awalan atau berisiko untuk terperangkap di optimaliti tempatan menghalang prestasinya. Dalam artikel ini, pendekatan baru telah dibangunkan dan ia diilhamkan dari algoritma imperialis kompetitif yang berdasarkan kepada kaedah *k-means*. Menggunakan kaedah ini pusat-pusat kelompok akan dipilih dan dikira dengan sewajarnya. Persaingan algoritma imperialis adalah satu kaedah dalam bidang pengiraan evolusi, yang cuba mencari penyelesaian optimum bagi pelbagai masalah pengoptimuman. Sifat-sifat asas algoritma ini diambil daripada proses evolusi pembangunan sosial, ekonomi dan politik negara-negara supaya sebahagian daripada pemodelan matematik proses ini dan beberapa pengendali diperolehi dalam bentuk algoritma biasa. Hasil dapatan analisis menggunakan pendekatan yang dicadangkan ke atas set data piawai serta perbandingan dengan kaedah alternatif dari kajian literatur mendedahkan bahawa prestasi algoritma yang dicadangkan melebihi daripada algoritma *k-means* serta algoritma pilihan yang lain.

Kata kuci: Analisis data; pengelompokan data; kelompok *k-means*; algoritma imperialis kompetitif

1.0 INTRODUCTION

In fact, data mining literally revolves around extracting knowledge from huge amounts of stored data in databases or other information sources in an automated (or semi-automated) manner. Clustering is an important technique in data mining. Clustering tries to organize datasets in some clusters so that the data objects inside one cluster have the most similarity with each other, and maximum diversity with those of other clusters. Clustering is a method of uncontrolled classification in which similar data are grouped regarding some parameters of interest, i.e. the most similar data objects are placed in one group [1],[2].

Generally, clustering algorithms and techniques are introduced in different ways, which can be classified generally into five categories: Partitioning, Hierarchical, Concentration-Based, Grid-Based and Model-Based methods. In a partitioning

method first k data partitions are built up with at least one data item (object) for each partition and k <= n. If the petitioning is of Hard-type a data item can contribute in one cluster while in the Fuzzy-type model of partitioning one data item can participate in several clusters albeit with different membership grades. Two famous heuristic algorithms for hard partitioning are *k*-means and *k*-methods algorithms. The Fuzzy partitioning counterparts of these algorithms are Fuzzy *k*-means and Fuzzy *k*-methods algorithms respectively [1],[2].

The *k*-means algorithm is a major practical clustering method. The main purpose of *k*-means clustering method is to minimize the aggregate difference of all objects of a cluster from their related cluster centers [3].This method is generally used due to its simplicity and low repeat number. The *k*-means algorithm tries to find the cluster centers of $(c_1, c_2, \ldots, c_k)$ so that the sum of squared distances of every x_i from the nearest cluster center is minimum. The performance dependence of this algorithm on the initialization of cluster centers is a major problem. In this algorithm there is a strong connection between data points and nearest cluster centers which does allow the cluster centers to exit from local data dense areas but its ultimate solution is neither unique nor necessarily optimum[4].

Many different approaches such as the Genetic Algorithm [4], the Particle Swarm Optimization (PSO), the Colony of Ants, the Metal Annealing method, Taboo Search and some other combined methods [5], have been suggested for KM algorithm so far. In some approaches it is tried to select the initial cluster centers appropriately using some tricks [6]

In the literature, different techniques have been used to overcome this problem. Krishna and Murty (1999) combining k-means approach and genetic algorithm presented GKA method [7] so that computing the distance between the cluster centers as well as data distances from these centers is according to both methods. Kanungo *et al*. (2002) presented a more effective method to improve the structure of *k*-means clustering method is which using the Filtering Algorithm Method achieved a better and faster way, in order to reach the optimal number of clusters in a multi-dimensional space that this method is based on data sensitivity analysis toward the number and centers of clusters [8]. Kuo *et al.* (2005) combining two methods of *k*-means and ant colony algorithm presented a new method called "Ant K-means" [9], which was actually in search of the best cluster and is conducted based on ants' search theory and calculation of clusters centers. The *k*-means clustering has been demonstrated as "NP-hard" problems [3]. In addition to that, since innovative ways are very efficient in solving complex compound optimization problems, in many cases meta-heuristic methods have been applied to solve the problems. For example, it may be noted to [10] that the annealing simulated method and tabu search have been used in solving clustering problem [11]. From recent works have been conducted in this area combined method of Clustering Search [12], [13] and combined method based on neighborhood search method based on bound can be pointed out.

The rest of this paper comes as follows: In section (2) the concepts of clustering and the *k*-means algorithm is reviewed. In section (3) the imperialist competitive algorithm is introduced. In section (4) the suggested approach is introduced. In section (5) experimental results of implementing the suggested approach are reported and compared with those of former methods and finally the section (6) incorporates concluding remarks of this surveys.

2.0 CLUSTERING

Clustering is defined as grouping similar objects either physical or abstract. The groups inside one cluster have most similarity with each other and maximum diversity with other groups' objects [2].

2.1 Definition

Suppose the set $X = \{x_1, x_2, \ldots, x_n\}$ containing n objects. The purpose of clustering is to group objects in k clusters as $C = \{c_1, c_2, \ldots, c_k\}$ while each cluster satisfies the following conditions [3], [14]:

1) $C1 \cup C2 \cup \ldots \cup Ck = X$
2) $Ci \neq \emptyset \quad i = 1 \ldots k$
3) $Ci \cap Cj \neq \emptyset$

According to the above definition the possible states for clustering n objects in k clusters is obtained as follows :

$$NW(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i \binom{k}{i}(k-i)^n \qquad (1)$$

In most of approaches, the number of clusters i.e. k is specified by user. Relation (1) implies that even with a given k, finding out the optimum solution for clustering is not so simple. In addition, the number of possible solutions for clustering n objects in k clusters increases by the order of $k^n / k!$. So finding the best state for clustering n objects in k clusters is an intricate NP-Complete problem which needs to be solved by optimization techniques [15].

2.2 The K-Means Algorithm

There have been suggested many algorithms for addressing the clustering problem among them the *k*-means algorithm is one of the most practical and famous algorithms. In this method besides the input data sets, k samples are introduced into the algorithm as the initial centers of k clusters. These representing k's are usually the first k data samples [3]. The way of choosing these k representatives influences over the performance of *k*-means Algorithm, the four stages of this algorithm come as below:

Stage I: choose k data items randomly from $X = \{x_1, x_2, \ldots, x_n\}$ as cluster centers of $(m_1, m_2, \ldots, m_k)$

Stage II: Based on the relation (2) add every data item to a relevant cluster. I.E. if the following relation (2) holds, the object x_i from the set of $X = \{x_1, x_2, \ldots, x_n\}$ is added to the cluster cj

$$\|xi - mj\| < \|xi - mp\| \quad 1 \le p \le k, \qquad j \neq p \qquad (2)$$

Stage III: Now based on the clustering of stage II the new cluster centers *(m1*, m2*,…,mk*)* are calculated by using the relation (3) as below (ni is the number of objects in the cluster *i*)

$$m_i^* = \frac{1}{n_i}\sum_{x_j \in C_i} x_j \quad 1 \le i \le k \qquad (3)$$

Stage IV: If the cluster centers are changed, repeat the algorithm from Stage II, otherwise do the clustering based on the resulted centers.

Random selection of initial cluster centers makes this algorithm yield out different results for different runs over the same data sets which are considered as one of potential weak points of this algorithm [4].

3.0 THE IMPERIALIST COMPETITIVE ALGORITHM

The Imperialist Competitive Algorithm with mathematical modeling of social-political evolution is an algorithm for solving mathematical optimization problems, which was introduced in 2007. The main bases of this algorithm are Revolution, Imperialistic Competition and Assimilation. By following the social, economic and politic evolution paths of countries and by mathematical modeling some operators are obtained in regular algorithmic forms, which can be helpful in solving intricate optimization problems. In fact, these algorithms consider the optimization solutions as countries and try to improve the answers through repeating routines until culminating at the optimum solution [16].

In the Imperialist Competitive Algorithm, a population based algorithm, first we have a collection of countries with their own attributes. These countries are imaginarily-made but their attributes are exactly the same attributes we are to find their optimum values in optimization problems. In fact the purpose from executing ICA is to reach at a country which possesses the best potential attributes. For example in selecting the best country all economic, sport-related, scientific and social indices are considered with a given effectiveness weight for each of them. If Ai be the attributes of countries Country= {A_1, A_2,…, A_n} ,all these attributes are combined through a procedure. In optimization, the aim is to find an optimum solution based on the problem's variables. We arrange an array of problem variables, which should be optimized. This array is called a chromosome in the Genetic Algorithm while it is called a country here [16].

In this algorithm, some countries with different attributes are generated and some of these attributes are desired attributes such as the country's power which enables it to put some other countries under its colonial dominance. These countries are called Imperialist and the other countries under their dominance are called Colony.

At first step, we produce initial countries (N*Country*) in order to select N*imp* from the best countries (countries with the least cost function). The rest of countries are colonies with N*col* in number so that each of them belongs to one empire. To allocate initial colonies among the imperialists, each imperialist is assigned some colonies according to its power.

Generally, the purpose is to improve the empire while competing with each other. There are two kinds of competitions: One intra-group competition for empire position and the other among imperialist countries. Actually, we have two kinds of operators: the Assimilation Policy or moving toward imperialism and the other operator: Revolution, which refers to the changes in the country's position. The power of an empire is defined as the power of the imperialist country plus a percentage of the power of its colonies. The above-mentioned stages are summarized in the following pseudo-code [16].

Algorithm 1. Pseudo-code of the ICA Algorithm

1. **Begin**
2. Select some random points in order to generate the initial empires
3. Move colonies toward imperialist countries *(the Assimilation Policy)*
4. Apply Revolution operator
5. If there is a colony in one empire which its cost is lower than that of empire swap the positions of empire and colony
6. Compete the overall cost of an empire state *(including the cost of imperialist and its colonies)*
7. Select one *(or more)* from the weakest empires and delegate it to the empire with the most probability of possessing it
8. Remove the weak empire
9. If just one empire remains stop , otherwise repeat from *step 2*
10. **End**

4.0 THE PROPOSED ALGORITHM

In most of proposed clustering algorithms, primary cluster centers are selected randomly from data objects. Our main idea is that initially in the first stage data objects are clustered with the *k*-means algorithm based on their own attributes. The number of generated clusters of this stage is more than or equal to the number of main clusters in input data sets. Now the cluster centers inside every data set is selected from primary clusters in a non-repeated manner [3].

In this survey by the help of combining these two the *k*-means algorithm and the Imperialist Competitive Algorithm and the proposed approach we can reach at the premised goal i.e. information clustering. The *Algorithm 2* shows the proposed clustering imperialist algorithm:

Algorithm 2. Pseudo-code of the Proposed ICAK K-Means Clustering Algorithm

1- ***Input*** : the data set $X = \{x1, x2, \ldots, xn\}$, the number of attributes and the number of clusters
2- ***Output*** : The set of cluster centers *C={C1,C2,…,Ck}*
3- Finding seed cluster centers
4- *Initialize-Population* **:** selecting the best countries as imperialists
5- The assignment of other countries as colony
6- While r < Iteration
   - *6.1-* Completing Assimilation Operation */* see section 4.4*/*
   - *6.2-* Completing Revolution Operation */* see section 4.5*/*
   - *6.3-* Comparing colonies with imperialist countries *(if the colony is better than its imperialist replaces it)*
   - *6.4-* Evaluating empires *(calculating the index of each empire) /* see section 4.6*/*
   - *6.5-* According to the index a colony is removed from the weakest empire and transmitted to another empire
   - *6.6-* Reporting the best outcomes

7- End

4.1 The Imperialist Competitive Algorithm

In this algorithm for finding the initial cluster centers for using in the countries the following approach is prosed as comes below, first all data objects are clustered according to their attributes and by using the *k*-means algorithm. Then, according to the generated clusters and based on each attribute a pattern for a data object is made at every stage. The objects with the same patterns are put in one cluster; following this, all objects are clustered. The obtained clusters of this stage are more than other stages. The base of this approach roots back to reference [6]. In this reference, clustering is completed in two stages. The first stage is completed as discussed above and in the second stage; similar clusters are integrated with each other until reaching at a given number of clusters. *Algorithm 3* shows the proposed approach for initial clustering of data item. The resulted cluster centers are named "*seed centers for clusters*".

As seen in the *Algorithm 3*, for each attribute of data object, a cluster label is produced for each data object and this label is added to the data object pattern. The objects with identical patterns locate in one cluster. For producing the label of each attribute first the mean and standard deviation values of that attribute are computed for all data objects. Then based on the mean and standard deviation the range of in-question attribute values is divided into k identical ranges so at the end of each range is the initial center of clusters. Now based on the initial centers, all the data is clustered by the *k*-means method.

4.2 The Structure of Countries

For presenting countries with k number of clusters and q number of attributes a matrix of $q \times k$ order is used as defined below:

$$M = [m_{11}, m_{12}, \dots, m_{1q}, m_{21}, m_{22}, \dots, m_{2q}, \dots, m_{1k}, m_{2k}, \dots, m_{kq}]$$

According to the above structure $m_i = [m_{i1}, m_{i2}, \dots, m_{iq}]$ is called the center of *I*'s cluster. In the beginning of the proposed algorithm, a certain number of countries are generated with the above structure. The cluster centers inside one country are randomly chosen from the seed centers, which were obtained by the discussed *Algorithm 3* in a non-repeated manner.

Algorithm 3. pseudo-code of the Proposed Find_Seed_Cluster_Center Algorithm

1. **Input**: Data Set ($X = \{x1, x2, \dots, xn\}$), Attribute Set ($A = \{A1, A2, \dots, Aq\}$), Cluster Number (K),
2. ***Output***: Clusters Seed Set ($SC = \{sc1, sc2, \dots, scH\}$)
3. **Begin**
4. While ($\forall Aj \in A$)
   - 4.1. Compute Mean (μ_j) and Standard Deviation (σ_j)
   - 4.2. Compute Cluster Center (***e***= 1,2,..., *k*)
     $X_e = Z_e * \sigma_j + \mu_j \qquad Z_e = \frac{2e-1}{2*k}$
   - 4.3. Execute K-means on this attribute
   - 4.4. Allocate cluster labels obtained from step 4.3 to every data pattern
5. Find unique patterns ($H \geq k$) and clustering each data whit obtained patterns.
6. Return SC
7. **End**

4.3 Evaluating the Fitness of Countries

For evaluating the fitness criterion of each country first based on the cluster centers inside each country the assignment operator of objects to clusters is executed according to the Relation (2). Now based on the completed clustering, new cluster centers are obtained through the Relation (3) and replaced as a country. Based on the new cluster centers the fitness of countries is computed by the Relation (4).

$$Fitness(c) = \sum_{i=1}^{k} \sum_{x_j \in c_i} \|x_j - m_i^*\| \qquad (4)$$

4.4 The Assimilation Operator

Following this policy, the colony moves x units in the connecting line of the colony toward the Imperialist and anchors at the New Position of Colony (see Figure 1). The x is a random number with uniform distribution (or any other appropriate distribution form) If the distance between the colony and the Imperialist is shown as d, Normally for 'd' we have [16]:

$$X \sim U(0, ß * d)$$

In which ß is a number greater that *"1"* and close to "*2*", a suitable selection could be *ß=1*. If *ß≥1* is taken, the colony country can approach the imperialist country from different directions. Also beside this movement, a small angular deviation with uniform distribution is added to the movement path. A graphical perspective of applying the Assimilation Policy in the Imperialist Competitive Algorithm is shown as below:

$$Colony' = Colony + ß.(imperialist - Colony) \qquad (5)$$

Now in order not to be fixed at a point in the path toward the imperialist country and to be potentially omnipresent in all points we multiply it by r, which is a vector of random numbers of $1 \times dim$ in size and dim is a number depending on the dimension of solution. That is, the movement toward empire is calculated by the Relation (4) and the deviation is computed as a coefficient of perpendicular vectors to the curve as shown in the Relation (6).

$$Colon' = Colon + ß.r.(imperialist - Colony) \qquad (6)$$

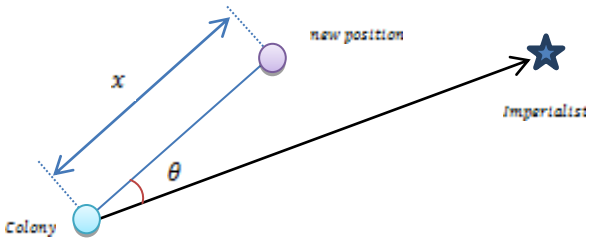


Figure 1 Moving colony toward imperialist

4.5 The Revolution Operator

The occurrence of revolution imposes sudden changes over the social and, political attributes of a country. In the Imperialist Competitive Algorithm, the Revolution is modeled by the random replacement of a colony country with a new random position. From the algorithmic point of view the evolutionary movement prevents from getting trapped in local optimal valleys and sometimes it improves the country's position and brings it to a better optimality range. The revolution occurrence rate should be μ, the changed variables are μ × nVar standing for observed changes and nVar is our Decision Variables Matrix Size. Our engaged procedure for the Revolution operator is a normal distribution as:

$$new\ position\ of\ colony \sim N(Colony, \sigma^2) \qquad (7)$$

Statistically the Relation (7) can be re-written as:

$$new\ position\ of\ colony \sim Colony + \sigma.N(0,1) \qquad (8)$$

So we generate a standard normal figure and multiply it by an "σ "which indicates the number of our steps and is called the *step size*:

$$\sigma = \eta\ .\ (ColonyMax - ColonyMin) \qquad (9)$$

Therefore η=0.1 is taken and *ColonyMin* & *ColonyMax* are respectively the least and most values among the countries with the same space width. Therefore, in overall for $\mu \times nVar$ components the changes are according to the Relation (8). We perform this task over both colonies and empires if it improves the situation.

4.6 The Imperialistic Competition

The power of an empire is defined as the power of the imperialist country f(imp) plus a percentage of ξ the mean power of its all colonies $Mean(f(col))$.

$$empire\ target\ function = f(imp) + \xi \times Mean(f(col)) \quad (10)$$

■5.0 EVALUATION

The suggested algorithm is coded by an appropriate programming language and is run in a Pentium IV computer with 3.03 GHz microprocessor speed and 512 MB main memory. For measuring the efficiency of the proposed algorithm, the standard data items of Table 1 are used.

The execution results of the proposed algorithm over the selected check data sets as well as the comparison figures relative to reported K-Means, PSO, K-NM-PSO results in Ref [17] are tabulated in the Table 2.

Table 1 Table type styles

Name of Data set	*Data Set attribute*		
	Size of data set	*No. of Cluster*	*No. Of attribute*
Iris	150(50,50,50)	3	4
Wine	178(59,71,48)	3	13
CMC	1473 (629, 334, 510)	3	9
Glass	214 (70, 17, 76, 13, 9, 29)	6	9

As easily seen in the Table 2, the suggested algorithm provides superior results relative to k-means and PSO algorithms.

For better study and analysis of the proposed approach, the execution results of the proposed approach along with TS, ACO, GA, HBMO, PSO-SA,ACO-SA,K-Means, PSO-ACO-K, PSO-ACO,PSO and SA results which are reported in ref [17] are tabulated in the Tables 3 to 6. It is worth mentioning to note that the investigated algorithms of Ref [17] are implemented with Matlab 7.1 over a Pentium IV system of 2.8GHz CPU speed and 512 MB main memory.

Table 2 The obtained results from implementing the suggested algorithm over selected data sets

Data set	*K-Means [Yi-Tung et al. 2008]*	*PSO [Yi-Tung et al. 2008*	*K-NM-PSO [Yi-Tung et al. 2008*	*Proposed Alg.*	
				Result	*CPU*
				time(S)	
Iris	97.33	96.66	96.66	96.5403	~15
Wine	16555.68	16294.00	16292.00	16292.94	~27
CMC	5542.20	5538.50	5532.40	5532.42	~109
Glass	215.68	271.29	199.68	234.8546	~31

In Tables (3) to (6) best, worst and average results are reported for 100 runs respectively. The resulted figures represent the distance of every data object from the center of the cluster to which it belongs and is computed by using the Relation (4). As observed in the table, regarding the execution time the proposed algorithm generates acceptable solutions.

Table 3 The results of implementing the algorithms over iris test data for 100 runs

Entry	**Method**	**Result**			**CPU Times(S)**
		Best	**Average**	**Worst**	
1	PSO–ACO–K	96.650	96.650	96.650	~16
2	PSO–ACO	96.654	96.654	96.674	~17
3	PSO	96.8942	97.232	97.897	~30
4	SA	97.457	99.957	102.01	~32
5	TS	97.365	97.868	98.569	~135
6	GA	113.986	125.197	139.778	~140
7	ACO	97.100	97.171	97.808	~75
8	HBMO	96.752	96.953	97.757	~82
9	PSO_SA	96.66	96.67	96.678	~17
10	ACO–SA	96.660	96.731	96.863	~25
11	k-Means	97.333	106.05	120.45	0.4
12	MY Proposed ALG.	96.5403	96.5404	96.5405	~15

Table 4 The results of implementing the algorithms over wine test data for 100 runs

Entry	Method	Result			CPU Times(S)
		Best	Average	Worst	
1	PSO–ACO–K	16,295.31	16,295.31	16,295.31	~30
2	PSO–ACO	16,295.34	16,295.92	16,297.93	~33
3	PSO	16,345.96	16,417.47	16,562.31	~123
4	SA	16,473.48	17,521.09	18,083.25	~129
5	TS	16,666.22	16,785.45	16,837.53	~140
6	GA	16,530.53	16,530.53	16,530.53	~170
7	ACO	16,530.53	16,530.53	16,530.53	~121
8	HBMO	16,357.28	16,357.28	16,357.28	~40
9	PSO_SA	16,295.86	16,296.00	16,296.10	~38
10	ACO–SA	16,298.62	16,310.28	16,322.43	~84
11	k-Means	16,555.68	18,061.01	18,563.12	0.7
12	MY Proposed ALG.	16,292.94	16,293.49	16,293.87	~27

Table 5 The results of implementing the algorithms over cmc test data for 100 runs

Entry	Method	Result			CPU Times(S)
		Best	Average	Worst	
1	PSO–ACO–K	5,694.28	5,694.28	5,694.28	~31
2	PSO–ACO	5,694.51	5,694.92	5,697.42	~135
3	PSO	5,700.98	5,820.96	5,923.24	~131
4	SA	5,849.03	5,893.48	5,966.94	~150
5	TS	5,885.06	5,993.59	5,999.80	~155
6	GA	5,705,63	5,756.59	5,812.64	~160
7	ACO	5,701.92	5,819.13	5,912.43	~127
8	HBMO	5,699.26	5,713.98	5,725.35	~123
9	PSO_SA	5,696.05	5,698.69	5,701.81	~73
10	ACO–SA	5,696.60	5,698.26	5,700.26	~89
11	k-Means	5,842.20	5,893.60	5,934.43	0.5
12	MY Proposed ALG.	5,532.422	5,532.947	5,533.404	~109

Table 6 The results of implementing the algorithms over class test data for 100 runs

Entry	Method	Result			CPU Times(S)
		Best	Average	Worst	
1	PSO–ACO–K	199.53	199.53	199.53	~31
2	PSO–ACO	199.57	199.61	200.01	~35
3	PSO	270.57	275.71	283.52	~400
4	SA	275.16	282.19	287.18	~410
5	TS	279.87	283.79	286.47	~410
6	GA	278.37	282.32	286.77	~410
7	ACO	269.72	273.46	280.08	~395
8	HBMO	245.73	247.71	249.54	~390
9	PSO_SA	200.14	201.45	202.45	~38
10	ACO–SA	200.71	201.89	202.76	~49
11	k-Means	215.74	235.5	255.38	~1
12	MY Proposed ALG.	234.8546	247.5682	260.0506	~31

6.0 CONCLUSION

In this paper, a new method for the Imperialist Competitive Algorithm in clustering data objects is introduced based on the *k*-means method. In this research by generating countries with the help of seed centers the main challenge of the *k*-means method can be partially resolved. The obtained results from implementing the proposed approach over the selected check data sets proves the higher efficiency of the proposed approach over massive data sets when compared with the other studied algorithms in this paper. But still this proposed approach selects the cluster centers in the form of countries randomly. However, this random selection is among seed centers, which are very less than the number of data objects. We should notify that closer data objects are integrated with each other and are taken as seed centers for clusters. In future works, we can define a certain criterion so that among the obtained seed centers the centers are choses, which may improve the proposed approach.

Acknowledgement

The authors would like to express their cordial thanks to University Technology Malaysia for International Doctoral Fellowship.

References

[1] Han, J., M. Kamber, and J. Pei. 2006. *Data Mining: Concepts and Techniques.* Morgan kaufmann.
[2] Gan, G., C. Ma, and J. Wu. 2007. Data Clustering: Theory, Algorithms, and Applications. 20.
[3] Bonab, M. B. 2011. *Modified K-Means Algorithm for Genetic Clustering.* 11(9): 5.
[4] Bandyopadhyay, S. and U. Maulik. 2002. An Evolutionary Technique based on K-Means Algorithm for Optimal Clustering in RN. *Information Sciences.* 146(1–4): 221–237.
[5] Kao, Y.-T., E. Zahara, and I.W. Kao. 2008. A Hybridized Approach to Data Clustering. *Expert Systems With Applications.* 34(3): 1754–1762.
[6] Khan, S. S. and A. Ahmad. 2004. Cluster Center Initialization Algorithm for K-means Clustering. *Pattern Recognition Letters.* 25(11): 129–1302.
[7] Krishna, K. and M. N. Murty. 1999. Genetic K-means Algorithm. Systems, Man, and Cybernetics, Part B: Cybernetics. *IEEE Transactions on.* 29(3): 433–439.
[8] Kanungo, T., *et al*. 2002. An Efficient k-means Clustering Algorithm: Analysis and Implementation. Pattern Analysis and Machine Intelligence. *IEEE Transactions on*. 24(7): 881–892.
[9] Kuo, R. J., *et al*. 2005. Application of Ant K-means on Clustering Analysis. *Computers & Mathematics with Applications.* 50(10–12): 1709–1724.
[10] Sun, L.-X., *et al*. 1994. Cluster Analysis by the K-means Algorithm and Simulated Annealing. *Chemometrics and Intelligent Laboratory Systems.* 25(1): 51–60.
[11] Osman, I.H. and N. Christofides. 1994. Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search. *International Transactions in Operational Research*. 1(3): 317–336.
[12] Güngör, Z. and A. Ünler. 2008. K-Harmonic Means Data Clustering with Tabu-search Method. *Applied Mathematical Modelling*. 32(6): 1115–1125.
[13] Güngör, Z. and A. Ünler. 2007. K-harmonic Means Data Clustering with Simulated Annealing Heuristic. *Applied Mathematics and Computation*. 184(2): 199–209.
[14] Alpaydin, E. 2004. *Introduction to Machine Learning*. MIT press.
[15] Hruschka, E. R. and N. F. Ebecken. 2003. A Genetic Algorithm For Cluster Analysis. *Intelligent Data Analysis*. 7(1): 15–25.
[16] Atashpaz-Gargari, E. and C. Lucas. 2007. Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.
[17] Niknam, T. and B. Amiri. 2010. An Efficient Hybrid Approach Based On PSO, ACO and K-Means for Cluster Analysis. *Applied Soft Computing*. 10(1): 183–197.