Research Journal of Applied Sciences, Engineering and Technology 11(3): 336-342, 2015
DOI: 10.19026/rjaset.11.1725
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2015 Maxwell Scientific Publication Corp.
Submitted: April 7, 2015 Accepted: May 22, 2015 Published: September 25, 2015

Research Article

Developing Clustering Based on Genetic Algorithm for Global Optimization

Khaled Batiha and Zeinab M. Olimat
Prince Hussein Bin Abdullah Faculty of Information Technology, Al al-Bayt University, Jordan

Abstract: Nowadays, databases are widely used over the world. The huge amount of data requires modern methods to make it useful meaning of information, clustering is one of the techniques that collects similar objects then put them in groups. Clustering is an approach appropriate for extracting useful meaning in large database. K-mean clustering is an algorithm characterized by simplicity and easy to implement and provides good results. However, it suffers from being trapped in local optimal solution. Some hybrid between two algorithms aims to combine the advantages of two algorithms to make optimization. In this thesis, we propose applying the same hybrid between k-mean clustering and Differential Evolution (DE) called Clustering based Differential Evolution CDE, but in the proposed method, we use Genetic Algorithm (GA) instead of Differential Evolution to find a globally optimal solution. This proposed method called Clustering based on Genetic Algorithm for Global Optimization (CGAGO), then we compare between them. In addition, we use a parameter called cluster period to improve k-mean clustering, in terms of finding the global optimum. Moreover, we test eleven Benchmark functions to validate the proposed method. Experimental results show that the proposed method CGAGO is slightly better and effective than CDE.

Keywords: Datamining, genetic algorithm, global optimization, k-mean clustering

INTRODUCTION

The global optimization problem can be defined as a pair of (S, F), where $S \leq R^D$ is a bounded set on R^D and F: S→R is a D- dimensional real valued function. The problem is to find a point $X^* \in S$ such that f (X^*) is a global minimum on S. So, it is required to find an $X^* \in S$ such that Cai *et al*. (2011):

$$\forall X \in S: f(X^*) \leq f(X). \quad (1)$$

Global optimization problems are employed in many fields such as science, data mining, biology and engineering (Cai *et al*., 2011; Weise, 2009). Global optimization is the ability to find global optimal solutions in the search space, the disadvantage of the data mining algorithm is trapped in local optimal solution when the search space is very wide. Many researchers have tried to find for methods solving global optimization problems.

Global optimization problems in data mining need to enhance clustering algorithm or to integrate it with another to get rid of being trapped in local optimal solution. In this thesis, we use k-mean clustering algorithm to make clustering of large data set because of its simplicity and ease of implementation, but it suffers from trapping in local optimal solution. Also, we use Genetic Algorithm (GA) which is a popular search algorithm that is able to find global optimal solutions (Goldberg, 1989).

Genetic Algorithm (GA) can be defined as an algorithm that is belonging to the Evolutionary Algorithm (EA) which is inspired by biological evolution such as reproduction, mutation, recombination and selection. Further, it is a part of Artificial Intelligence (AI). We have selected (GA) since it is the most common search algorithm for optimization (Goldberg, 1989).

In the previous researches, most of the researchers focused on optimizing the k-mean clustering algorithm for data mining, because of its simplicity, easy implementation and effectiveness. However, it suffers from trapping to local optimal solution (Elmasri and Navathe, 2010). The current researchers hope to optimize k-mean clustering to accomplish the global optimal solution. Optimization of k-mean clustering algorithm aims to find the optimal state of the problem exactly when databases are very large and this optimization helps the k-mean clustering make a search in the data to find the best solution compared to the local optimal search which used the data partially (Rusell and Norving, 2003; Smyth *et al*., 2001).

K-mean algorithm suffers from being trapped in local optimal solution (Sumathi and Sivanandam, 2006; Han and Kamber, 2011; Smyth *et al*., 2001). This local search is using a part of data to get the best of solution. Local optimal search uses a single current state and

Corresponding Author: Khaled Batiha, Prince Hussein Bin Abdullah Faculty of Information Technology, Al al-Bayt University, Jordan

generally moves only to neighbors of that state. This means that the k-mean algorithm finds a suboptimal solution (Rusell and Norving, 2003).

The main contribution of this research is to develop one-step k-mean clustering to find the global optimal solution. We propose a method that depends on a hybrid GA based on the one-step k-means clustering algorithm called Clustering Based Genetic Algorithm for Global Optimization (CGAGO) and using a parameter that is called the cluster period (Cai *et al*., 2011). This hybrid helps to prevent k-mean clustering from trapping to local optimal solution.

LITERATURE REVIEW

Jiang *et al*. (1997) proposed a method that depends on making optimization of Genetic Algorithm (GA) with Integer representation called (IGA). This method uses fitness scaling and proposed a new three selection operator, there are competition, self-reproduction and diversification. This IGA is used for the clustering problem. Rana *et al*. (2010) proposed a method that depends on combining Particle Swarm Optimization (PSO) and k-mean. K-mean suffering from being trapped to local optima, this reason occurs from initial points for k-mean. PSO suffers from its slow to reach the optimal solution, but it provides a global search. This proposed method uses sequential PSO. It helps k-mean to find the optimal solution. KumarMeena *et al*. (2012) proposed a method that depends on combining between Genetic Algorithm (GA) and Discrete Differential Evolution (DDE) to overcome the local search of k-mean clustering algorithm. This proposed method can get the best cluster center. Performance Evolution shows that the proposed method can present best fitness value in fewer iterations. Finally, Cai *et al*. (2011) proposed a method that dependson combination k-mean and Differential Evolution (DE). When the k-mean clustering produces k off spring's then the population updates the method using DE to update the population. This method is called clustering based Differential Evolution (CDE). It helps to discard local search and to reach global search.

Hybrid k-mean clustering and genetic algorithm: In this section we will explain the proposed method that depends on combining k-mean clustering and genetic algorithms, this hybrid aims to combine the advantages of the two algorithms to get global optimization.

The global optimal solution is the most important feature of a genetic algorithm and GA is the famous algorithm within searching algorithms. In this study, we propose a method called Clustering based on Genetic Algorithm for Global Optimization (CGAGO) as described in Fig. 1.

The proposed method is based on the combination of k-mean clustering algorithm and GA using a parameter called clustering period (denotes by m) (Cai *et al*., 2011). We will explain the proposed method as follows:

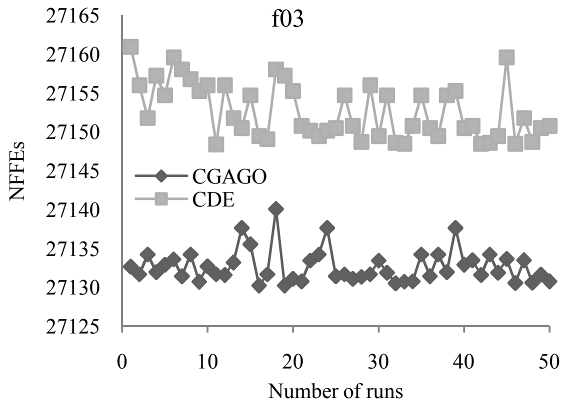

Fig. 1: Effect of NFFEs through 50 runs for (f03)

The initial population P is randomly generated from the dataset, then we evaluate the fitness function for each individual in the population P, then we initialize the generation counter (t) to start from 1 (t = 1). Then we use the GA and k-mean clustering if the halting criterion is not satisfied, in this algorithm we use the number of generations as a halting criterion, that means if the number of generations exceeds the specific number the algorithm is stopped, otherwise the algorithm does the following steps:

- We use GA to update the population and to apply clustering based on GA using the Pseudo code of k-mean clustering presented in Fig. 2.
  In clustering, each chromosome is represented as a set of clusters, each cluster is represented as allele value and each gene represents an object. We noticed that each gene with the same allele value is in the same cluster.
- We use a conditional statement that is based on the new parameter clustering period (m) and the number of generations that is if the number of generations (t) mod clustering period (m) equal zero we apply the following steps:
  - Generate the initial k clusters randomly, this random integer is between 2 and square numbers of the population (denotes by NP). We use initial k clusters randomly because no effect on the performance of the algorithm (Cai *et al*., 2011).
  - The next step is applying k-mean clustering algorithm as Fig. 3.
  - Choose K parents randomly from the population p and put them in the set (B).
  - Then combine set (A) and set (B) to choose the K best solutions and put them in a new set ($\overline{B}$), then update the population P by (P/B), then combine to set ($\overline{B}$) to become the update method as P = (P/B) $\bigcup \overline{B}$. Then end if.
  - Later, use generations counter (t) = t+1, then restart the previous step.

```
    Input: a database D, of m records, r1,..........r_m, and
    desired number of cluster k
    Output: set of k cluster
    Begin
(1)        Randomly choose krecords as the centroids for the
clusters;
Repeat
(2)        Assign each records, $r_i$, to a cluster such that the
distance between $r_i$ and cluster centroid (mean)is the smallest
among the k clusters;
(3)        Recalculate the centroid (mean) for each cluster
based on the records assigned to the cluster;
(4)        Until no change; output k of cluster;
```

Fig. 2: Pseudo code of k-means clustering algorithm

```
1-    Generate the initial population P;
2-    Evaluate the fitness for each individual in P;
3-    While The halting criterion is not satisfied do
4-    Select parents for reproduction;
5-    Perform crossover and mutation;
6-    Apply(GA) Forms;
7-    Repeat
8-    End
```

Fig. 3: Pseudo code of genetic algorithm

```
1-    Generate the initial population P randomly;
2-    Evaluate the fitness for each individual in P;
Initialize the generation counter
3-    t=1;
4-    while The halting criterion is not satisfied do
          5- Use GA to update the population (see lines
      3-7 in Algorithm 2)
      6-  If t%m==0 then
              Randomly generate k= rndint[2, $\sqrt{NP}$ ];
              Adopt the one-step k-means clustering to
      create k offspring(set A)
              Choose k-parents (the set B)randomly from the
      population P
            From the combined set A $\bigcup$ B, choose k-best
      solutions and put them
              in $\overline{B}$. Update P as P=(P/B) $\bigcup$ $\overline{B}$
         7- end if
     8-t=t+1
9-end while
```

Fig. 4: Pseudo code of CGAGO

Optimization in k-mean clustering to update the population of GA consists of four stages as follows:

* **Selection stage:** Select k individuals from the population randomly (step 1 in the k-means clustering).
* **Generation stage:** produce k-offspring (the set A) using k-means clustering (steps 2-4 k-mean clustering algorithm).
* **Replacement stage:** Select k solutions (the set B) from population randomly for replacement.
* **Update stage:** Combine A$\bigcup$B, select k best solutions and put them in the set $\overline{B}$, the change in P to become P = (P/B) $\bigcup$ $\overline{B}$. As shown in Fig. 4.

Clustering period: Clustering period is a condition for applying the k-mean clustering; when generation counter mod cluster period = 0 k-mean clustering begins working. In Fig. 4, we use a parameter called clustering period (m) (Cai *et al*., 2011). This parameter helps the GA to change the search space and clusters form periodically. This parameter works well if it has appropriate values. That is, this parameter works to change the search space to become more effective. Later we explain the performance evolution of this parameter.

EXPERIMENTAL RESULTS AND EVOLUTION

We will explain the experimental results and evolution of the proposed method. We will use two performance measures as follows:

* Absolute mean error and standard deviation.
* Average CPU time for each function.

To explain the performance of the proposed method, we compare the proposed method with a previous research (developing clustering based on differential evolution) that deals with the same clustering problem (Cai *et al*., 2011).

System specification: The research test was implemented on 2.4 GHz computer core i3, with 4 GB RAM, on Windows 7. We use in this research C# language for implementation.

Data collection: In this research, we use the dataset from UCI repository ((http://archive.ics.uci.edu/ml/) for data miming to test the proposed method (UCI, 2014).

Performance measurement: Four performance measures are selected from (Cai *et al*., 2011) to test the performance of the algorithm. These measures are characterized as follows:

Error: The error of solution X is evaluated as f(x)-f(x*), where X* is a global optimum of the function. The minimum error is recorded when reached to 50 runs. We calculate the average of error values and standard deviation of error values.

NFFE$_S$: The Number of Fitness Function Evaluation (NFFES). We use D*5000 where D is a number of decision variables (Cai *et al*., 2011).

Number of Successful Runs (SR): The number of successful runs is recorded when the VTR is reached before the Max-NFFE$_S$ condition terminates the test.

Convergence graphs: The convergence graph explains the mean error performance of the total runs and the fitness number of total runs.

Table 1: Parameter settings of proposed method

Population size	100
Scaling factor	0.5
Crossover probability	0.9
Mutation probability	0.05
Value To Reach (VTR)	VTR = 10^{-8} for all functions, except f07 = 10^{-2}
Max-NFFS	D*5000, where D is a decision variable of function (Cai *et al*., 2011)

Table 2: Benchmark test functions

Number	Test functions	S	D	Optimal
f01	$\sum_1^D x^2$	$[-100,100]^D$	30	0
f02	$\sum_{i=1}^{D} \lvert x_I \rvert + \prod_{i=1}^{D} \lvert x_I \rvert$	$[-10,10]^D$	30	0
f03	$\sum_{i=1}^{D}\left(\sum_{j=1}^{D} x_J\right)$	$[-100,100]^D$	30	0
f04	$\max\{\lvert x_I \rvert, 1 \le i \le D\}$	$[-100,100]^D$	30	0
f05	$\sum_{i=1}^{D}\left[100\left(x_i+1-x_i^{2}\right)^2+\left(x_i-1\right)^2\right]$	$[-30,30]^D$	30	0
f06	$\sum_{i=1}^{D}\left(x_i+0.5\right)^2$	$[-100,100]^D$	30	0
f07	$\sum_{i=1}^{D} x_i^{4} + random[0,1)$	[-1.28,1.28]	30	0
f08	$\sum_{i=1}^{D}\left(-x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right)$	$[-500,500]^D$	30	-418.982887*D
f09	$\sum_{i=1}^{D}\left(x_i^{2}-\cos\left(2\pi x_i\right)+10\right)$	[-5.12,5.12]	30	0
f10	$-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^{2}}\right)-\exp\left(\frac{1}{D}\sum\cos\left(2\pi x_i\right)+20+\exp(1)\right)$	$[-32.32]^D$	30	0
f11	$\frac{1}{4000}\sum_{i=1}^{D} x_i^{2} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^D$	30	0

Table 3: Comparisons between proposed method and CDE in mean absolute error and Std.dev and D = 30 (D*5000) for all experiments and average of time in seconds

		CGAGO				CDE			
Function test	MaxNFFEs	Mean	S.D.	SR	Time (s)	Mean	S.D.	SR	Time (sec)
f01	150000	0.04	0.14	50	60.22	0	0	50	5.36
f02	150000	2.17	2.64	50	80.22	2.80	2.78	50	30.37
f03	150000	2.47	1.82	50	60.44	4.28	3.59	50	24.73
f04	150000	0.02	0.04	50	73.00	0	0	50	15.61
f05	150000	0	0	50	45.14	0	0	50	22.12
f06	150000	0.06	0.06	50	55.45	0	0	50	23.11
f07	150000	1.90	1.3	50	75.12	3.29	2.69	50	35.1
f08	150000	0.5	0.41	50	60.52	1.16	0.69	50	33.31
f09	150000	3.19	2.33	50	68.33	5.75	2.87	50	20.71
f10	150000	0	0	50	45.00	0	0	50	30.31
f11	150000	1.13	1.3	50	60.22	1.02	1.28	50	32.11

Parameters setting: In the proposed method CGAGO, there are four basic parameters; population size (NP), scaling factor (F), crossover probability (CR) and Mutation Probability (MR). In Cai *et al*. (2011) and Liu and Lampinen (2005) both researcher used the same parameters in their test, in Cai *et al*. (2011) the researcher tests the proposed method using the parameters; crossover probability, mutation probability, population size, Value To Reach (VTR) and MAX-NFFEs to evaluate the mean error and standard deviation. In Cai *et al*. (2011) the researcher proposed a method to develop Clustering based Differential Evolution (CDE) and calculate the mean error and standard deviations, then apply the proposed method in high dimension and different clustering period.

In this research, we use all parameters in Cai *et al*. (2011) and Liu and Lampinen (2005) with the same values. Table 1 Explains the parameter settings to test the proposed method.

Fitness functions: The fitness function is the most important step because any optimization problem depends on the chosen the fitness function (Goldberg, 1989). In this research, we use eleven functions with high dimensions to compare the CDE and proposed method CGAGO and to explain which method is most effective and obtain a good solution. We use eleven functions from Benchmark Test Functions to verify the performance of our approach (Cai *et al*., 2011). Table 2 explains the functions that are used as fitness functions.These eleven functions are used in our experiments, where D is the number of variables, "optimal" is the minimum value of the function and $S \leq R^D$. We described all functions in Table 2.

Comparisons between CGAGO and CDE: In this section, we present the performance evolution of the proposed method called CGAGO and CDE, which are used in developing clustering based differential evolution. Table 3 explains the comparison between CGAGO and CDE.

We present the comparison accordingly to the Table 3 as follows:

Mean error: From Table 3 we can notice that the performance evaluation shows that CGAGO is slightly better than CDE, where CGAGO records the lower mean error values than CDE. Also, From the Table 3, we can notice that f02, f03, f07, f08, f09 get the lower mean error that shows these functions are the most appropriate as a fitness function for CGAGO.

Standard deviation of errors: From Table 3 we can notice that CGAGO is better than CDE in the five functions in variance between the fitness values; (f02, f03, f07, f08 f09).

VTR (Value to Reach): We can get the VTR when computing the error, then compare the values with the specific value as the follows:

VTR (i.e., $f(x) - f(x^*) < 10^{-8}$) for all functions except (f07) = 10^{-2}

CGAGO and CDE can reach the VTR for (f07, f02, f05, f10) before 50 runs, CGAGO can reach the VTR for (f09), CDE can reach the VTR for (f08) and CGAGO and CDE cannot reach VTR for (f03).

MAX-NFFEs: From Table 3 we can notice that CGAGO and CDE reach to the MAX-NFFEs for (f05, f10), where MAX-NFFEs is extended to 150000, however CGAGO needs to NFFEs is less thanCDE.

CPU time: We can notice that the CPU time of CDE is lower than theCPU time for CGAGO because the difference in the process cycle (selection, crossover, mutation) for CGAGO and CDE. CGAGO starts with selection, crossover, mutation, but CDE start with mutation, crossover, selection. Otherwise, CGAGO needs a parallel computing.

Influence of population size: In general, any algorithm is sensitive for the choice of the population size, because increasing the population size leads to increase the difference of possible solution and expanding of search space, so the probability to find the best movement of search decreases too much. The influence of population size for CGAGO and CDE are explained in Table 4 using the same parameters were mentioned. From the Table 4, we use two different population sizes (NP) to compare between CGAGO and CDE as the following:

Population size (NP) = 50: For (NP) = 50, CGAGO is better than CDE in seven functions (f02, f03, f05, f07, f08, f09), however, for four functions (f01, f04, f06, f10) CDE is better than CGAGO.

Population size (NP) = 200: For (NP) = 200, CGAGO is better than CDE in the five functions (f01, f02, f03, f05, f08).

Table 4: Mean error for different population size

	NP = 50		NP = 200	
Function	CGAGO	CDE	CGAGO	CDE
f01	1.28	1.15	3.13	3.87
f02	1.32	1.75	3.86	4.22
f03	1.71	2.11	4.15	4.78
f04	1.15	1.05	4.45	3.75
f05	1.77	1.82	4.91	5.39
f06	1.43	1.29	3.23	2.81
f07	1.33	2.47	3.89	3.21
f08	2.3	2.67	4.34	5.25
f09	2.25	2.81	4.71	4.12
f10	2.37	2.31	4.82	3.57
f11	2.22	2.87	5.25	4.75

Table 5: Mean error for different dimensionality

	D = 10		D = 20	
Function	CGAGO	CDE	CGAGO	CDE
f01	4.38	4.75	2.95	2.05
f02	6.75	5.06	2.55	2.89
f03	5.34	5.72	3.22	3.12
f04	4.85	4.76	2.65	2.44
f05	1.23	2.45	1.12	1.87
f06	3.71	3.55	2.88	3.31
f07	4.17	4.28	1.55	1.69
f08	3.35	2.49	2.27	3.81
f09	4.35	5.65	2.71	3.31
f10	3.51	4.21	2.11	3.71
f11	6.31	6.25	3.72	2.91

Table 6: Mean error for different clusters period

Function	m = 5	m = 15	m = 20
f01	4.55	3.31	4.88
f02	3.44	3.28	2.41
f03	5.31	4.55	4.22
f04	3.21	3.56	3.85
f05	2.12	2.62	1.81
f06	5.81	4.32	2.51
f07	2.33	1.99	2.1
f08	4.84	4.25	3.63
f09	3.41	3.05	2.49
f10	3.86	2.55	1.65
f11	2.34	2.24	2.65

We can notice from the previous results that CGAGO can give better results for a large and small population size; we can conclude the results of CGAGO is the most appropriate for different population sizes.

Scalability study: Any algorithm when we measure the important factors that affect in the performance, we can find the scalability measure is very important until we know how the algorithm works, when the problem dimensionality is changed. Table 5 shows (CGAGO) and (CDE) working in different dimensionality (D), D = 10 and D = 20 by recording the mean error. These changes in dimensionality affect Max-$NFFE_S$ (D*5000). All results use the same parameters tested in above section.

Effect of clustering period: In the proposed method (CGAGO), we add a new parameter called cluster period (denotes by m) (Damavandi and Safavi-Naeini, 2005). This parameter makes a control to k-mean clustering to perform periodically. Table 6 explains the effect of this parameter in (CGAGO) using the same parameters. Table 6 modifies cluster period as m = 5, m = 15, m = 20 for the proposed method. We can see from Table 6 that (CGAGO) provides best results when (m≥15), where (m) make k-mean clustering to work efficiently.

Convergence graphs: The convergence graphs show the errors and NFFEs of the 50 independent runs. We use f03 as an example to explain the effect of NFFEs through 50 independent runs and these figures show the effect of error through 50 independent runs for CGAGO and CDE for comparisons. From Fig. 1 and 5, we can notice that (CGAGO) needs less NFFEs than (CDE) for

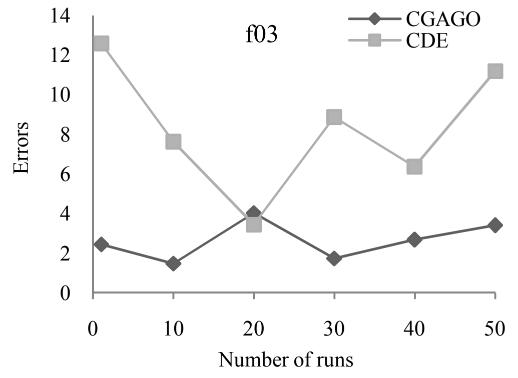

Fig. 5: Effect of errors through 50 runs for f (03)

(f03), from the figure CDE gives absolute error for (f03) greater than (CGAGO) and (CDE) up to maximum absolute error in run number 3 and back to records the maximum absolute error in run number 30 and 50, but (CGAGO) values converge with each. The two algorithms cannot reach to the VTR for (f03).

CONCLUSION AND RECOMMENDATIONS

Data mining helps to extract useful meaning from large databases using k-mean clustering algorithms. K-mean suffers from trapped in local optimal solution, to solve this problem we need to make optimization to get a global optimal solution using a hybrid approach between Genetic Algorithm and k-mean clustering.

Hybrid between k-mean clustering and genetic algorithms aims to make k-mean clustering more efficient and more effective. The proposed method CGAGO depends on combining k-mean clustering and genetic algorithms using a new parameter called clustering period. CGAGO had been applied in eleven Benchmark functions to test which function gives a good results using to dataset from UCI repository for data mining. After evaluating the performance of CGAGO we made another test to know the effect of the population size, the effect of a new parameter cluster period and compared the results with CDE. The experimental results indicated that CGAGO had reached the global optimal solution with less NFFE than CDE with a mean error and standard deviations in 6 functions for (CGAGO) better than (CDE).

In our future work, we can study the effect of using different schemes and can make many possible directions to optimize clustering such as the following:

- We can apply some other clustering algorithm such as k-prototype.
- We can apply particle swarm optimization as a hybrid with k-mean algorithm.
- We can use different distance measures and compare which gives a good solution.

- We can use different crossover and mutation operators.

ACKNOWLEDGMENT

This study is a part of M.SC. thesis of Zeinab M. Olimat under the supervision of Dr. Khaled Batiha Al al-Bayt University.

REFERENCES

Cai, Z., W. Gong, C. Ling and H. Zhang, 2011. A Clustering-based differential evolution for global optimization. Appl. Soft Comput., 11(1): 1363-1379.

Damavandi, N. and S. Safavi-Naeini, 2005. A hybrid evolutionary programming method for circuit optimization, IEEE Transaction on Circuits and Systems, 52(5): 902-910.

Elmasri, R and S.B. Navathe, 2010. Fundamentals of Database Systems. 6th Edn., Addison-Wesley, Boston, pp: 1035-1057.

Goldberg, D., 1989. Genetic Algorithms in Search: Optimization and Machine Learning. 1st Edn., Addison-Wesley, New York.

Han, J. and M. Kamber, 2011. Data Mining: Concepts and Techniques. 3rd Edn., Morgan Kaufmann, USA, pp: 1-32.

Jiang, J., J. Wang, X. Chu and R. Yu, 1997. Clustering data using a modified Integer Genetic Algorithm (IGA). Anal. Chim. Acta, 354(1-3): 263-274.

KumarMeena, Y., Shashank and V.P. Singh, 2012. Text documents clustering using genetic algorithm and discrete differential evolution. Int. J. Comput. Appl., 43(1): 16-19.

Liu, J. and J. Lampinen, 2005. A fuzzy adaptive differential evolution algorithm. Soft Computing. 2005; 9(6): 448-462.

Rana, S., S. Jasola and R. Kumar, 2010. A hybrid sequential approach for data clustering using k-mean and particle swarm optimization algorithm. Int. J. Eng. Sci. Technol., 2(6): 167-176.

Rusell, S.I. and P. Norving, 2003. Artificial Intelligence: Modern Approach. 2nd Edn., Pearson Education, London, pp: 116-119.

Smyth, P., D.J. Hand and H. Mannila, 2001. Principles of Data Mining. MIT Press, Cambridge, Mass, ISBN: 026208290X, pp: 546.

Sumathi, S. and S.N. Sivanandam, 2006. Introduction to Data Mining and its Applications. Springer, Berlin, Heidelberg, New York, ISBN: 3540343504, pp: 828.

UCI, 2014. Retrieved from: http://archive.ics.uci.edu/ml/. (Accessed on: March 25, 2014)

Weise, T., 2009. Global Optimization Algorithm-theory and Application. 2nd Edn., Retrieved from: http://www.it-weise.de/. (Accessed on: February 10, 2009)