International Journal of Industrial Engineering Computations 3 (2012) 435–444

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Rapid Ant based clustering-genetic algorithm (RAC-GA) with local search for clustering problem

Yaghub pirzadeh[a*], **Jamal shahrabi**[b], and **Mohamad taghi taghavifard**[c]

[a]*Graduate Faculty of Tehran-South Branch Azad University, Tehran,Iran*
[b]*Department of Industrial Engineering of Amirkabir University of Technology,Tehran,Iran*
[c]*Department of Management and Accounting, Allameh Tabatabai University, Tehran, Iran*

A R T I C L E I N F O

Article history:
Received 28 October 2011
Accepted December, 5 2011
Available online
14 December 2011

Keywords:
Genetic algorithm
Clustering problem
RAC-GA
Local search

A B S T R A C T

Clustering is a critical data analysis and it is a popular data mining technique. This paper presents a rapid Ant based clustering-genetic algorithm (RAC-GA) with local search to solve clustering problem. GA and local search are used as a global and local search to obtain better results. The proposed algorithm is evaluated by testing on some of the well-known real-world datasets, and the results are compared with other popular heuristics in clustering, such as GA, SA, TS, ACO and RAC. The results show strong improvement both in quality solution and process time area, especially in process time which is much less than previous algorithms.

1. Introduction

Data clustering is defined as a gathering the objects into some group such that, the objects in the particular group have the most similarity and differ from the objects of other groups (Larose, 2005). Regularly the objects have many attributes, which describe them, and the similarity and dissimilarity can be measured based on these attributes. There are many distance measurement equations, which could be utilized in the clustering methods. The data clustering has some roots in a number of areas; including data mining, machine learning, biology, and statistics (Zhang et al., 2009). Traditional clustering algorithms can be divided into two main categories: hierarchical and partitional (Kennedy &Eberhart, 1999; Niknam & Amiri, 2009). In this work, we concentrate on the heuristic partitional clustering. "Ant Colony based" clustering algorithm, developed in the recent years, is one of the most popular heuristic partitional clustering algorithms used in variety of domains.

The Ant Colony clustering is defined over continuous data (Shelokar et al., 2004). "Ant based clustering" has been used as a popular non-center based clustering method due to its efficiency, with nonlinear time complexity (Dorigo, 1991).However, the "Ant based clustering" algorithms have some defects, such as low convergence rate, inaccurate solutions and capturing in local optima. Besides, there are many other heuristic clustering algorithms introduced to solve such problem. A genetic based algorithm to solve the clustering problem is proposed by Mualik and Bandyopadhyay (2002)

* Corresponding author Tel: +98938-761-3856
E-mail: yaqubpirzadeh@gmail.com (P. Pirzadeh)

doi: 10.5267/j.ijiec.2011.12.004

and its result is presented on synthetic and reallife datasets to evaluate its performance. Krishna and Murty (1999) proposed a novel combinational algorithm called genetic K-means algorithm, which defines a basic mutation operator specific to clustering called distance based mutation. It has been proved that, through using the theory of finite Markov chain, GKA can converge to the best known optimum. A simulated annealing approach for solving the clustering problem is proposed by Selim and AlSultan (1991). Sung and Jin (2000) proposed a tabu search based heuristic for clustering. The particle swarm optimization, which simulates bird flocking was used for clustering by Kao et al. (2008). An ant colony clustering algorithm for clustering is presented by Shelokar et al. (2004). The algorithm employs distributed agents who mimic the way real ants find a shortest path from their nest to food source and back. Its performance was compared with GA, tabu search, and SA. Pirzadeh et al. (2011) introduced a novel heuristic based algorithm (RAC) which has improved the time efficiency of Ant clustering algorithm by implying multiple evaporation coefficients, their results have been compared with SA and GA and ACO, and they reduced the process time strongly by having a proper solutions.

2. Clustering problem

Partitional clustering problem refers to grouping different objects into the same category. For a given input pattern,$X = \{X_1 \dots X_j \dots X_N\}$, where $X_j = \{x_{j1}, x_{j2}, \dots, x_{jn}\}^T \in \mathcal{R}^n$, there is K clusters which $K \le N$, such that:

1- $C_i \neq \emptyset , i = 1, \dots, K;$
2- $\bigcup_{i=1}^{K} C_i = X;$
3- $C_i \cap C_j = \emptyset, i, j = 1, \dots, K \; and \; i \neq j.$

Each pattern can be assigned to one cluster, at the end of process all the patterns should be assigned, and no cluster can be empty. Cluster center can be considered as the average of all objects, and it can be measured by calculating the average of each feature of input pattern for each cluster. Fitness metric can be composed as a sum of distances of objects from thecluster center.Euclidean distance function is the most popular function, and here, it has been used to measure solution quality. Based on Euclidean distance, the mathematical formulation of the data clustering problem can be considered as:

$$\min F(w,m) = \sum_{j=1}^{K}\sum_{i=1}^{N}\sum_{v=1}^{n} w_{ij} \left\| x_{iv} - m_{jv} \right\|^2 \tag{1}$$

$$\sum_{j=1}^{K} w_{ij} = 1, \quad i = 1, \cdots, N \tag{2}$$

$$\sum_{i=1}^{N} w_{ij} \ge 1, \quad j = 1, \cdots, K \tag{3}$$

where v_{th} is the feature of i_{th} object with value of x_{iv}. m_{jv} is the average of v_{th} feature of all patterns in j_{th} cluster and

$$w_{ij} = \begin{cases} 1 & \text{if cluster } j \text{ involves object } i \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \cdots, N; \; j = 1, \cdots, K. \tag{4}$$

The clustering problem aims to minimize the fitness function value (Xu & Wunsch, 2005).

3.Ant based clustering

While clustering problem is a kind of NP-hard problems, heuristic approaches have been applied to solve this kind of optimizing problems. Ant Colony approach uses stochastic mechanism to optimize the resulted problems, which is a non-center based approach. It means that the algorithm assigns the objects to cluster based on the heuristic and stochastic information rather than the distance from the cluster center. There are two main categories in this field; grid based algorithms and pheromone based algorithms.

3.1. Grid based Ant clustering

The grid based algorithms use pick and drop functions to put similar objects into the particular area of the grid. Since it is not necessary to determine the number of clusters, it has more flexibility and since it is more stochastic approach it has less accuracy than pheromone based approach algorithms. The following equations measures pick and drop probability:

$$P_{pick}(o_i) = \left(\frac{k_1}{k_1 + f(o_i)} \right)^2 \tag{5}$$

$$P_{drop(O_i)} == \begin{cases} 2f(O_i) & \text{if } f(O_i) < k_2 \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

The ant population searches the space randomly, the random search can be based on random walking through the grid or jumping based on short term memory. Whenever an ant faces with data object, it picks the object according to pick probability function. After picking an object, it searches again the space to find suitable place to drop the data object. Both pick and drop, use the similarity function to $f(o_i)$ measure the probabilities. There are also two parameters, k_1 and k_2 to control the effect of $f(o_i)$ on pick and drop functions.

$$f(o_i) = \max\left\{ 0, \frac{1}{s^2} \sum_{o_j \in Neigh_{s*s}(r)} \left[1 - \frac{d(o_i, o_j)}{\alpha(1 + \frac{v-1}{v_{\max}})} \right] \right\} \tag{7}$$

Similarity function defines local perception area for each ant, $d(o_i, o_j)$ is the function which measures the distance between o_i and o_j. α is similarity scaling factor, which belongs to (0,1]. By increasing the α similarity between the objects in the same cluster increases, and smaller clusters can be constructed by algorithm. In the other hand by decreasing the α the final clusters become larger and consequently the similarity between the objects decreases too (Kuntz et al., 1998).

3.2. Pheromone based Ant clustering

The pheromone based Ant Clustering algorithm acts the same as original ACO algorithm implied previously for TSP problem. Like the original algorithm, there is a population of ants which act as the same as natural ants. At the initial step, algorithm distributes the population to discover the solutions according to ACO approach. Here there is a pheromone trail matrix $M_{N \times K}$, which is the basis of solution construction process,K and N are the number of clusters, and number of the objects, respectively and$R$ is the number of agents. The agents start to make a solution by an empty solution vector. This solution vector has N elements (equal with objectnumber) where the content of element i_{th} represents the corresponding cluster of object i_{th} . The agents use the pheromone trail information to assign each object to appropriate cluster. When the algorithm initializes the process, assigns a little random value to each element of pheromone matrix. Before solution construction phase, the algorithm makes the "FirstSolution". For constructing first solution, algorithm seeks each row of pheromone matrix M_{N*K} to find a maximum value M_{ij} for i_{th} row. Next, it fills i_{th} element of "FirstSolution" by j.To generate solutions there are two approaches. First approach just uses the "FirstSolution" to make a solution, and the second approach uses stochastic mechanism based on the information, which is obtained from pheromone trail matrix. In this approach, the algorithm measures the probability of assigning the object i to cluster j according to the following equation,

$$P_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^{K} \tau_{ik}}, j = 1, \cdots, K \tag{8}$$

In Eq. (8)τ_{ij} is the amount of pheromone between object i and cluster j and$P_{ij}$is the probability of assigning object i to cluster j. During the phase of creating solution, first the algorithm generates N random value$q_{1,\ldots,}q_N \in (0,1)$. For each object$o_i$, if q_i<Q then first approach is implied to make solution, otherwise, the second one is implemented. Q is a predefined threshold which establishes a tradeoff between two approaches. Shelokar and Kulkarni (2004) consider a high value, 0.98 for Q. Each agent, at each iteration, makes solution once and stores it in its solution vector. By making a complete solution by all agents, the algorithm calculates the fitness function value for all obtained solution vectors. Then a predefined percentage of the best solutions with lower fitness function value is qualified to be manipulated with local search process. If the result of the local search is better than the previous solutions, the algorithm will use the local search to update the pheromone matrix, otherwise, the original solutions will be chosen to update the pheromone trails. The update equation is as follows,

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{l=1}^{L} \Delta\tau_{ij}^{l}, i = 1, \cdots, N, j = 1, \cdots, K \tag{9}$$

where ρ is evaporation coefficient, which reduces the amount of pheromone trail to allow other solution to be examined (Xu, & Wunsch, 2005).

3.2.1. Rapid Ant clustering (RAC)

Heuristic approaches can solve the nonlinear problems appropriately, and they can obtain near optimal solutions, but they have some limitations. In some cases, the CPU time increases frequently, and this disadvantage have made it useless. Another disadvantage of such an approach is that it could trap in stagnation state. Pirzadeh et al. have proposed an improved approach to reduce the process time by having a proper solution quality. The solutions are built and their fitness measured. Then a predefined percentage of best solutions allowed updating the trails. Two major modifications are used to reduce process time and avoid from stagnation state. The first improvement is to update multiple evaporation coefficients, and the second one is to use pure stochastic solution making approach. The algorithm has a good talent to escape from local optimum by using the stochastic approach to make appropriate solution. It causes a smooth and nonstop moving toward global best solution. There are three evaporation coefficients to update pheromone trail and process can escape from stagnation. Note that only qualified agent can increase the amount of pheromone, and all ants decrease the matrix elements value according to ρ_2, which helps convergence of the algorithm once the pheromone reaches a minimum requirement level. The process sometimes is trapped in local optimum, and to solve this issue, the proposed algorithm uses a large evaporation coefficient ρ_3 after a while to restart the problem by having a proper initial state. ρ_1 and ρ_2 are just used for special elements which are participated in solution vectors but the third coefficient reduces all elements of trail matrix.

$$\tau_{ij}(t+1) = (1-p_1)\tau_{ij}(t) + \sum_{l=1}^{L} \Delta\tau_{ij}^{l} \quad \text{if agent}_l \;\; i \;\text{ to }\; j \tag{10}$$

$$\tau_{ij}(t+1) = (1-\rho_2)\tau_{ij}(t) \qquad i = 1, \cdots, N; j = 1, \cdots, K \tag{11}$$

$$\tau_{ij}(t+1) = (1-\rho_3)\tau_{ij}(t) \qquad i = 1, \cdots, N; j = 1, \cdots, K \;\text{ with } \rho_3 > \rho_1 \gg p_2 \tag{12}$$

We use some lower and upper limits to prevent the pheromone to stay in required limit.

4-RAC-GA Clustering

Here we have proposed a novel combinational approach to improve the ACO algorithm for clustering problem. RAC-GA uses the advantages of RAC to be rapid enough to find proper solutions and it utilizes the robustness of GA to improve the solution quality and time efficiency. The prior proposed combinational algorithms have obtained better results than pure algorithms because multiple approaches can recover themselves. The main disadvantage of heuristics and metaheuristic approaches is to capture in local optimums, which leads us to have low quality solution. Most of algorithms commonly try to solve this inherent defect by increasing the process time, but they fail commonly to do that. As a metaheuristic approach, the ACO suffers from this disadvantage. Many solutions have been proposed to solve this issue, for example, the ACO and RAC have used local search and multiple updating rules, respectively. The proposed RAC-GA of this paper uses the RAC robustness to reduce the process time, and to optimize the solution quality by implying the GA operators as a global search method. Using the global search has some advantages; it extends the search space and generates different solutions. Consequently, it lets more solutions to be examined, and as result when it is trapped in local optimum it has high chance to escape.

There are some characteristics in GA and RAC approached, which makes them more desirable for hybrid method. RAC and ACO, both, are population based, ant based clustering algorithm solve the problems based on the result of the population of agents, while GA needs a population of solutions to generate better results. Ant based clustering algorithms generate a vector as a result which can represent itself as a chromosome to be implied in GA algorithm. GA needs to measure the fitness value for the results to evaluate and progress its process, while fitness of ant based clustering results is strongly measurable. Both RAC and GA are memory less approaches and results in step t depends on the results at step$(t-1)$. RAC and GA both use the stochastic approaches to generate the results.

4.1-RAC-GA Algorithm

The proposed RAC-GA incorporates of three main steps. In the first step, the initial pheromone matrix is generated. The proposed model creates a solution in the second phase where a solution based on the RAC algorithm is generated. RAC can make a solution rapidly, and the proposed algorithm uses this RAC advantage at the second phase. Once the solution procedure is generated in the second step, RAC-GA updates the pheromone trails based on a combinational mechanism, which has some root in GA and RAC algorithms. When all agents complete their solutions, RAC-GA starts the third step. In this step, the algorithm first calculates the fitness value for all generated solutions based on the rules given by Selim and AlSultan (1991) and then, it sorts the solutions based on the solution quality, which is determined with fitness value. Clustering algorithms aim to minimize this fitness value. A predefined percentage of the best solutions is qualified to update the pheromone matrix based on the method proposed by Sung and Jin (2000). Here RAC-GA utilizes the GA operators in this step to enhance the solutions. After sorting phase in this step, the RAC-GA implies the GA operators to generate a new population of solutions. The solution vectors are supposed as chromosomes and in this phase, the RAC obtains solutions considered as initial population for selection operator. Selection operator operates on this population of chromosomes and all generated solutions are candidate for mating. We have chosen the roulette wheel among different types of selection operators. According to predetermined crossover rate, based on a stochastic approach some of the chromosomes are selected to mate and generate offspring to generate new population. In this section, single point crossover is applied. Then mutation operator, by a little value, operates on the new generated population. The fitness value is calculated for new population and the RAC-GA sort the agents again based on the new fitness values. Now in the next phase a predefined percentage of the agents are qualified to update the pheromone trails according to the rules proposed by Dorigo (1991). RAC-GA repeats this process until meeting the termination condition. Termination condition can be specified based on the maximum iteration number or a certain number of the iteration with having no improvement on solution quality. Fig. 1 shows how the RAC-GA works.

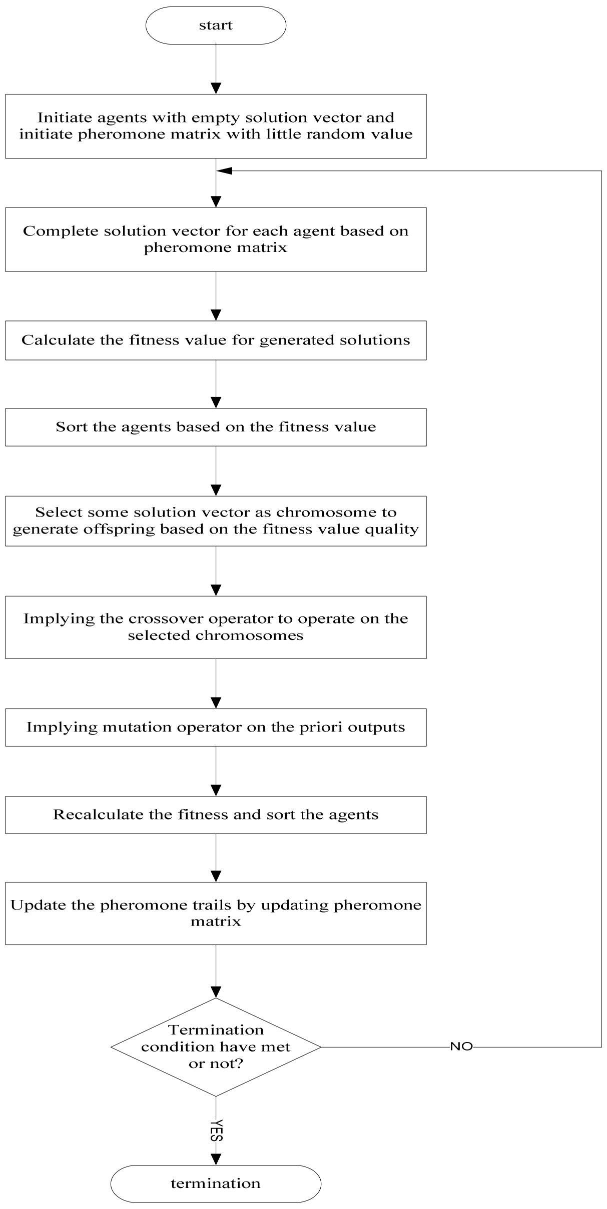
start
Initiate agents with empty solution vector and initiate pheromone matrix with little random value
Complete solution vector for each agent based on pheromone matrix
Calculate the fitness value for generated solutions
Sort the agents based on the fitness value
Select some solution vector as chromosome to generate offspring based on the fitness value quality
Implying the crossover operator to operate on the selected chromosomes
Implying mutation operator on the priori outputs
Recalculate the fitness and sort the agents
Update the pheromone trails by updating pheromone matrix
Termination condition have met or not?
NO
YES
termination

Fig.1. RAC- GA algorithm

5. Results

We have tested our approach on three different data sets with different scales and the results are compared with some of the well known algorithms. The proposed algorithms are implemented in C++ language and simulated on personal computers with the Pentium IV, 2.8HZ and 512GB RAM. We have chosen three well known data sets to test the RAC-GA and evaluate the results. Iris, Wine and CMC are the chosen data sets.

Dataset 1: The Iris dataset is the best known and the most popular database for evaluating the algorithms. This dataset contains three categories, and there are 50 records in each category, where each category is associated with a certain type of Iris plant. Two categories cannot be separated from each other by linear functions and one category is linearly separable. There are 150 instances with four numeric features in iris data set. There is no missing attribute value.

Dataset 2: The wine dataset contains 178 records with 13 features in each record, which represent the chemical characteristics of different types of the wines. Some of the characteristics are: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyaninsm, color intensity, hue, OD280/OD315 of diluted wines and praline. The number of records in each cluster differs from others in this dataset and there are 59, 71 and 48 records in three clusters.

Dataset 3: Contraceptive method choice dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey, which has 1473 records and 10 features in each instance with three clusters. The problem is to predict the current contraceptive method choice (no use, long term methods, or short term methods) of a woman based on her demographic and socioeconomic characteristics.

To evaluate the performance of the RAC-GA algorithm, we have compared it with the following well known clustering algorithms: GA (Murthy & Chowdhury, 1996), TS (Al Sultan, 1995), SA (Selim & AlSultan, 1991), ACO (Shelokar et al., 2004), RAC (pirzadeh et al., 2011).The obtained results show that, RAC-GA solution quality is improved and process time is strongly reduced. GA operators can be considered as a global search and expand the search space in such a way that there are many more solutions to examine and in each iteration it can generate completely different solutions, so the RAC-GA has a high chance to escape from the local optimums. Here we have represented a comparison between the RAC-GA result and some other well known algorithms in Table 1 to Table 3.

We have selected three data sets with different sizes; first data set is Iris, which is a small data set, with few records and few features in each record. The Wine dataset is a dataset with few records, while each record has many features. The third data set, CMC, is a big data set with too many big records. The results show that, RAC-GA can achieve the best result on the 3 examined datasets. The main improvement of RAC-GA is process time reduction. We can see obviously the process time is strongly reduced and it is less than other well known algorithms. Fig. 2 shows RAC-GA solution making rate from the beginning to the end. As we can see in Fig. 2.a RAC-GA makes the results and optimizes them rapidly in initial states, but it slows down in next steps.

For Iris dataset which is a small dataset, it attacks to the best solution in the first steps, but it takes a time to achieve the best solution. It is because the variance of the object features is a small value, which makes it difficult to find the best possible solution. However, the RAC-GA can reach better solutions after spending a bit more time. In Fig. 2.b RAC-GA has the same pattern to attack to the best and it has found it straightly because the variance of the object features is high and clustering on this dataset is easier than Iris dataset. Fig. 3.c represents the RAC-GA results on the CMC dataset. As we can observe, CMC dataset is a big dataset rather than two other datasets but it follows the same rule in solution making. By considering CMC size, the variance of the objects is low in this dataset and its curve is more similar to Iris dataset than the Wine.

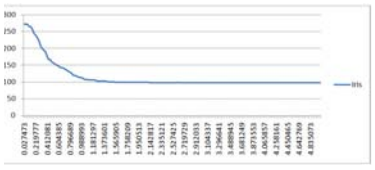

Fig. 2.a. RAC-GA solution making rate curve on the Iris

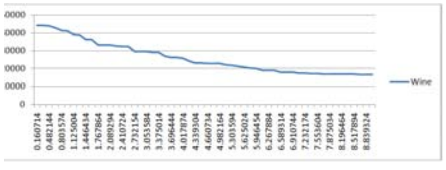

Fig. 2.b. RAC-GA solution making rate curve on the Wine

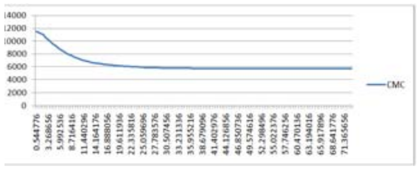

Fig. 2.c. RAC-GA solution making rate curve on the CMC

By using GA, proper solutions are obtained in reasonable amount of time although the local search uses relatively strong mechanism to obtain other optimum points in current state vicinity. Table 1, Table 2 and Table 3 show the local search effects, compared with other approaches. It shows strong improvement in solution quality and process time compared with other heuristic algorithms.

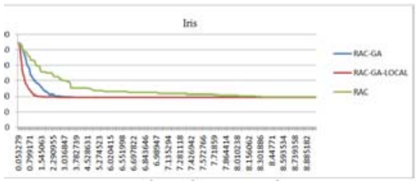

Fig. 3.a. solution making rate curve on the Iris

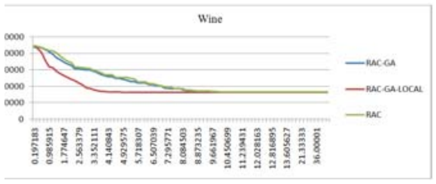

Fig. 3.b. solution making rate curve on the Wine

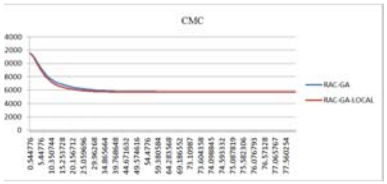

Fig. 3.c. Solution making rate curve on the CMC

Fig 3 shows the results of RAC-GA with and without local search. Fig. 3.a, b. and c. represent the results of the mentioned datasets, Iris, Wine and CMC, respectively. We can obviously see that by using local search there are more attitudes to attack to local optimums. Because of this, by using local search, the RAC-GA shows more attitudes to escape from local optimum and optimizes the best solution achieved. However, it has an overhead process and consequently it increases the process time. An important point is solution making rate, which is higher in RAC-GA than pure RAC, and in RAC-GA with local search is the highest among represented approaches. Local search can adaptively join to RAC-GA, because it can be considered as a supplementary section of global search approach (GA). Using local search causes more rapid finding solution rate, but the algorithm is trapped in some local optimums. However, the implementation of GA helps escape from local points very easily. Consequently, in order to achieve the best optimum, the proposed method needs to spend more time than pure RAC-GA. Fig. 3 shows the above mentioned points.

As mentioned in previously, RAC-GA along with local search provides us proper solutions. RAC-GA results with and without local search have represented in Table3. In all examined datasets we observe that RAC-GA can easily obtain the best process time and the best solution quality. According to above descriptions, RAC-GA in all datasets and without local search can optimize the process time strongly, but by implying the local search, it can achieve the best solution quality. In Table1, we can see that the proposed algorithm on Iris data set can optimize both the process time and solution quality compared with other well known algorithms. RAC-GA with local search can achieve 96.9296, which is the best found solution, and in process time point of view it can achieve the best solution with local search in 6.5Sec and it can also obtain near optimal solution in 5Sec. The results of the second dataset is the same as first one. The proposed algorithm obtains the absolute best solution in a relatively short amount of time. The achieved time for the RAC-GA with local search is 16Sec and without local search is 9Sec, while the process times for other algorithms of GA and TS are more than 100Sec. After RAC-GA, the best CPU time belongs to SA, which is 32Sec and it is much more than our obtained results. The results of our implementation without local search is about 16502 and for the algorithm with local search is about 16481, which are much less than the next best found solution, 16530, which belongs to ACO and SA. The results of the CMC, for RAC-GA follow the same pattern. For this dataset, RAC-GA can also obtain the best in 73 an78Sec with and without local search, respectively.

Table 1
RAC-GA with and without local search result on Iris data set

Method	Best result	CPU-time(s)
RAC-GA-LOCAL SEARCH	96.9296	6.5
RAC-GA	96.9681	5
RAC	97.2221	9
GA	113.9865	140
TS	97.3659	135
SA	97.4573	32
ACO	97.1007	75

Table 2
RAC-GA with and without local search result on Wine data set

Method	Best result	CPU-time(s)
RAC-GA-LOCAL SEARCH	16481.3	16
RAC-GA	16502.21342	9
RAC	16523.10000	14
GA	16530.53381	226
TS	16666.22699	161
SA	16530.53381	57.28
ACO	16530.53381	68.29

Table 3
RAC-GA with and without local search result on CMC data set

Method	Best result	CPU-time(s)
RAC-GA-LOCAL SEARCH	5699.15	78
RAC-GA	5699.8312	73
GA	5705.6301	160
TS	5885.0261	155
SA	5849.0380	150
ACO	5701.9230	127

6. Conclusion

Previous researches on heuristic algorithms had improved the solutions on the nonlinear problems, but recently proposed approaches are so better than traditional approaches. For example ACO shows a good improvement on the quality solution or some combinational algorithms even obtained better results. Some novel modifications on ACO, leads to introduce the RAC which reduces the process time besides having a proper quality. Although, RAC just obtained acceptable results in proper time just on little problems. As mentioned before, combinational approaches shows better results, and according to inherent adaption between GA and ACO, RAC-GA obtained better results on all kind of data sets with different characteristics. Implying the local search on RAC-GA leads to even better solutions but because of process overhead process time is a little increased. RAC-GA has a combinational approach which is so successful to solve nonlinear problems. It strongly reduces the time and improves the result quality. RAC-GA uses the advantages of the ACO based approaches to be fast and utilizes the GA operator to improve the solution quality. Results show that by implying the local search, process time increases but it enhance the solution quality. In the other hand applied approaches ACO, GA and local search are inherently adaptive with each other and the comparison between obtained results and some other well known algorithms show this claim.

Reference

Zhang, C., Ouyang, D., & Ning, J. (2010). An artificial bee colony approach for clustering. *Expert Systems with Applications*, 37(7), 4761-4767.

Larose, D. T. (2005). *Discovering Knowledge in Data*. A John Wiley & Sons, Inc.

Lumer, E.D., & Faieta, B. (1994). *From Animals to Animats3. In: D. Cliff, P.Husbands, J.A. Meyer, W.Stewart (Eds.).* MIT Press, Cambridge, MA, 501–508.

Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., & Chretien, L. (1991). *From Animals to Animats1.In: J.A. Meyer et, S.W.Wilson (Eds.), MIT Press*, Cambridge, MA, 356–363.

Kennedy, J., & Eberhart, R. (1999). *Particle swarm optimization*. In: Proceedings of theIEEE International Conference on Neural Networks, Piscataway, NJ., 1942–1948.

Krishna, K., & Murty. (1999). Genetic K-means Algorithm. IEEE *Transactions on Systems Man and Cybernetics B Cybernetics*, 29, 433–439.

Kao, Y.T., Zahara, E., & Kao, I.W. (2008).A hybridized approach to data clustering. *Expert Systems with Applications*, 34(3), 1754-1762.

Dorigo, M. (1991). Optimization learning and natural algorithms. (in italian) Ph.D thesis.

Mualik, U., & Bandyopadhyay, S. (2002). Genetic algorithm based clustering technique. *Pattern Recognition*, 33(9), 1455–1465.

Kuntz, P., Layzell,P., & Snyers, D. (1998) a stochastic heuristic for visualizing graph. Clusters in a bi-dimensional space prior to partitioning. *Journal of Heuristics*, 5(3), 327–351.

Shelokar, P.S., Jayaraman, V.K., & Kulkarni, B.D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, 509(2), 187–195.

Pirzadeh, Y., Shahrabi, J., & Anvari, M.T. (2011). Rapid Ant clustering Algorithm with multiple evaporation coefficients, the fifth Conference of data mining, Iran.

Xu, R., &Wunsch, D. (2005). Survey of clustering algorithms. *IEEE transactions on neural networks*, 16(3), 645-678.

Selim, S. Z., & Al Sultan, K.S. (1991). A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, 24, 1003–1008.

Sung, C. S., & Jin, H. W. (2000). A Tabu-search-based heuristic for clustering. *Pattern Recognition*, 33, 849–858.

Niknam, T. & Amiri, B. (2010). An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied soft computing*, 10(1), 183-197.