IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol-6, Special Issue-8, Oct 2018 E-ISSN: 2347-2693

An Ant Colony Optimization based Evolutionary Multi-objective Clustering for Overlapping Clusters Detection (ACOEMCOC)

S. Punithavathy

Dept. of Computer Science, Pioneer College of Arts and Science, Coimbatore, India

Available online at: www.ijcseonline.org

Abstract- Identification of overlapping clusters in complex data has been remaining as the problem to tackle. To the best knowledge, no evolutionary and unsupervised clustering approach is able to detect it successfully. Most of the existing evolutionary clustering techniques fail to detect complex/spiral shaped clusters. This research adopts an optimization method called Ant Colony Optimization (ACO) with the existing algorithm called Evolutionary Multi-objective Clustering (EMC) for overlapping clusters detection. This work resolves the problem of overlapping clusters by enhancing the multi-objective evolutionary clustering approach with Genetic Algorithm (GA) with variable length chromosome & local search for feature selection. Combined with Evolutionary Multiobjective Clustering, Ant Colony Optimization (ACOEMCOC) approach succeeds in obtaining non-dominated and near-optimal clustering solutions in terms of different cluster quality measures like purity, and index etc., and classification performance.

Keywords **–** Evolutionary Multiobjective Clustering (EMC), EMCOC, FEMCOC, Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm.

I. INTRODUCTION

1.1DATA MINING

Data mining, "the extraction of hidden predictive information from large databases", is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems.

1.1.1 CLUSTERING

Clustering is an important real-world problem, and different clustering algorithms usually attempt to optimize some validity measure such as the compactness of the clusters, separation among the clusters, or a combination of both.

Typical objective functions in clustering formalize the goal of attaining high intra-cluster similarity (documents within a cluster are similar) and low inter-cluster similarity. This is an internal criterion for the quality of a clustering.

An alternative to internal criteria is direct evaluation in the application of interest. For search result clustering, it is needed to measure the time it takes users to find an answer with different clustering algorithms. This is the most direct evaluation, but it is expensive, especially if large user studies are necessary.

This section introduces two external criteria of clustering quality. The Rand index penalizes both false positive and false negative decisions during clustering. Purity is a simple and transparent evaluation measure.

Rand index

Given a set of *n*elements $S = \{O_1, \ldots, O_n\}$and two partitions of *S* to compare,

$X = \{x_1, \ldots, x_r\}$and

$Y = \{y_1, \ldots, y_s\}$, the following is defined:

- *a*, the number of pairs of elements in *S* that are in the same set in *X* and in the same set in *Y*
- *b*, the number of pairs of elements in *S* that are in different sets in *X* and in different sets in *Y*
- *c*, the number of pairs of elements in *S* that are in the same set in *X* and in different sets in *Y*
- *d*, the number of pairs of elements in *S* that are in different sets in *X* and in the same set in *Y*.

The Rand index, *R*, is:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

Intuitively, *a* + *b* can be considered as the number of agreements between *X* and *Y* and *c* + *d* as the number of disagreements between *X* and *Y*.

PURITY

The purity measure evaluates the coherence of a cluster, that is, the degree to which a cluster contains documents from a

single category .Purity can be interpreted as the classification rate under the assumption that all samples of the cluster are predicted to be members of the actual dominant class for the cluster. For an ideal cluster, which only contains documents from a single category, its purity value is 1. In general, the higher the purity value, the better the quality of the cluster is.

Purity is a performance measure used to evaluate clustering methods. The system compares clustering of a set of objects as given by an automatic system, to the gold standard clustering of the same set (e.g., created manually by human). Purity of a system's cluster ci is defined as the ratio between the number of items in ci that belong to the gold standard cluster dominant in ci, and the size of ci.

E.g., let's assume that a system produced two clusters: cluster 1 and cluster 2, and the gold standard cluster assignments are indicated by the x symbol:

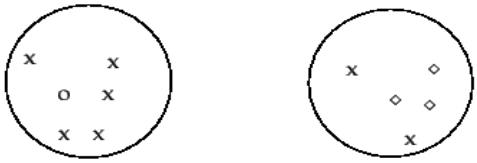

Fig1.1 Cluster 1 Cluster 2

For cluster 1, five of its six items belong to the dominant gold standard cluster in it; this means that purity (cluster 1) = 5/6 = 0.83. Similarly, purity (cluster 2) = 3/5 =0.6.
The purity of the entire clustering is defined as the average purity of clusters. For our example, purity = 0.83+0.6/2 = 0.71.

DISTANCE MEASURE
An important step in most clustering is to select a distance measure, which will determine how the similarity of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another.

For example, in a 2-dimensional space, the distance between the point (x = 1, y = 0) and the origin (x = 0, y = 0) is always 1 according to the usual norms, but the distance between the point (x = 1, y = 1) and the origin can be 2, $\sqrt{2}$or 1 if you take respectively the 1-norm, 2-norm or infinity-norm distance.

1.2 THE PROBLEM DEFINITION AND THE PROPOSED STRATEGY

A single objective clustering algorithm cannot find all the clusters if different regions of the feature space contain clusters of diverse shapes, because its intrinsic criterion may not fit well with the data distribution in the entire feature space. A related problem is that virtually all existing clustering algorithms assume a homogeneous criterion over the entire feature space. As a result, all the clusters detected tend to be similar in shape and often have similar data density.

In this work an Ant Colony Optimization based evolutionary multi-objective method is proposed to detect the overlapping clusters.

As a consequence, the crossover and mutation operators are suitably modified to tackle the concept of composite chromosomes with variable lengths. Additionally, the proposed procedure with local search operation is introduced to refine the selection of fittest chromosomes. This approach move solutions towards successful identification of overlapping clusters. It also succeeds obtaining non-dominated and near-optimal clustering solutions in terms of different cluster quality measures and classification performance.

II. RELATED WORKS

2.1 EVOLUTIONARY MULTI OBJECTIVE CLUSTERING USING GENETIC ALGORITHM WITH FIXED LENGTH CHROMOSOME

Evolutionary clustering is one of the emergent and effective unsupervised clustering approaches in searching the near-optimal clustering solutions. At present, several genetic algorithm (GA) based clustering technique exists.

However, most of them are limited to a single objective and suffer from a number of problems. First, they usually ask the user to provide the number of clusters in advance, which is, in general, unknown to the users.

Also, many existing genetic clustering approaches, such as evolutionary fixed-length chromosomes that encode cluster centres as genes. Since the actual number of clusters is initially unknown, the fixed-length of chromosomes limits the GA to search for the near-optimal clustering solutions. Moreover, in complex data, the clustering may have different size, mixture of various data distribution, outlier, and linearly inseparable clusters.

The traditional clustering methods utilizing single criterion fail to solve these issues simultaneously. Also, single objective evolutionary clustering algorithms suffer from ineffective genetic search, which in turn get stuck at sub-optimal clustering solutions.

BASIC STEPS
The approaches where data points are assigned according to their distances from cluster centers, it is very usual for any data point that its distance from two or more clusters centers

are the same or equal as shown in Fig. 2.1(a). In that case, it is very difficult to select the cluster center into which that data pattern will be assigned. In this thesis, a new idea was proposed for solving this overlapping problem as shown in Fig. 2.1(b).

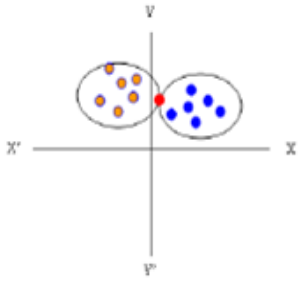

Fig 2.1(a) Overlapping problem.

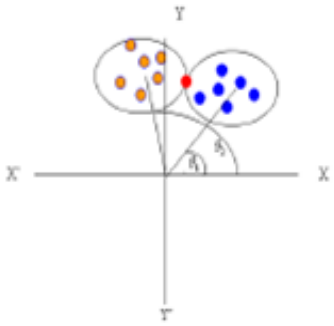

Fig 2.1(b) Removing problem.

Let θ 1 and θ2 are two angles between two cluster centers and reference point (red point) respectively. If θ 1 is less than θ 2, red point is assigned for blue cluster; otherwise it will be assigned for orange cluster. Reference point may vary for each chromosome. This method can apply not only for two clusters but also for more clusters centers when distance or similarity is equal. As per the distance calculation method each cluster is evaluated for overlapping.

In some other existing algorithms, crossover operation is performed each time on a single gene position. This might yield a total number of clusters (or 1's) smaller than Kmin. In that case, unreasonable offspring may often occur, and need to be repaired for many generations. To eliminate this problem, a modified version of crossover operator is introduced here.

The number of clusters in each of a parent pair is counted, say, NC1 and NC2 respectively, and a random integer NC is generated from the range [1,M], where M = Min(NC1, NC2), and then NC gene positions having allele 1's in each of the parents are randomly selected for crossover. The parent pair undergo crossover by exchanging alleles at the selected gene positions to introduce a pair of their offspring. The modified version of crossover operation is illustrated as follows:

Algorithm 1: Modified Crossover

For each pair of chromosomes Cha and Chb

1. Evaluate
   NC1 = number of 1's in Cha.
   NC2 = number of 1's in Chb.
   M = Min {NC1, NC2}
2. Generate a random integer NC from the range [1, M].
3. Randomly select NC gene positions among the genes with allele "1" from Cha and form a set
   Sa of indices of such selected positions. Randomly select NC gene positions among the genes
   switch allele "1" from Chb and form a set Sb of indices of such selected positions.
4. S = Sa ∪ Sb
5. for each index i in S
   Exchange the alleles of chromosomes Cha and Chb at gene position i.

2.2 EVOLUTIONARY MULTIOBJECTIVE CLUSTERING USING GENETIC ALGORITHM WITH VARIABLE LENGTH CHROMOSOME & LOCAL SEARCH

In GA applications, the unknown parameters are encoded in the form of strings, so-called chromosomes. A chromosome is encoded with binary, integer or real represented by positive integers; the chromosome is encoded with a unit (tuple) of positive integer numbers. Each unit represents a combination of brightness values, one for each band, and thus a potential cluster centroid.

CHROMOSOME INITIALIZATION

A population is the set of chromosomes. The typical size of the population can range from 20 to 1000. In the following an example is given to explain the creation of an initial population:

We assume to have a satellite image with three bands; Kmin is set to 2 and Kmax to 8.

At the beginning, for each chromosome I (i =1, 2,…,.P, where P is the size of population) all values are chosen randomly from the data space (universal data set; here: positive integers).

Such a chromosome belongs to the so-called parent generation. One (arbitrary) chromosomes of the parent generation is given here:

Nan (55) (150, 246, 23) Nan (11) Nan (100)

CROSSOVER AND MUTATION

Crossover: The purpose of the crossover operation is to create two new individual chromosomes from two existing chromosomes selected randomly from the current population. Typical crossover operations are one point crossover, two-point crossover, cycle crossover and uniform crossover. In this research, only the simplest one, the one-point crossover was adopted; the following example illustrates this operation (the point for crossover is after the 4th position):

Parent1: Nan (88) (226) Nan (104) (50) Nan(192)
Parent: Nan (127) (88) Nan (45) Nan (174) (101)
Child1: Nan (88) (226) Nan (45) Nan (174) (101)
Child2: Nan (127) (88) Nan(104) (50) Nan (192)

MUTATION

The non-uniform mutation operator is applied to the mutation operation. It selects one of the parent chromosome genes gi and adds to it a random displacement. The operator uses two uniform random numbers r1 and r2 drawn from the interval [0, 1]. The first (r1) is used to determine the direction of the displacement while the other (r2) is used to generate the magnitude of the displacement.

Assuming that gi € [ai, bi], where ai and bi are the gene lower and upper bounds, respectively, the new variable becomes

$$q_i = \begin{cases} g_i + (b_i - g_i) f(G), r_1 < 0.5 \\ g_i - (g_i - a_i) f(G), otherwise \end{cases}$$

Where f (G) = [r2 (1 − (G/Gmax))] p, G is the current generation, Gmax is the maximum number of generations, and p is a shape parameter.

FITNESS FUNCTION

Based on crossover and mutation the chromosomes, once initialized, iteratively evolve from one generation to the next. In order to be able to stop this iterative process, a fitness function needs to be defined to measure the fitness or adaptability of each chromosome in the population. The population then evolves over generations in the attempt to minimize the value of fitness, also called index.

APPLY VARIABLE LENGTH

This work has used the same chromosome representation and crossover operation. Mathematical operations between an integer centroid value and NaN we used the following logic. Generated a random number σ between [0 1] and if the value of σ is greater than 0.5 then we take the integer centroid as resultant gene in the child chromosome., and when the value of σ is less than 0.5 NaN is taken as the resultant gene in the child chromosome. The NaN or integral value in the child gene occurs with equal probability; hence the natural randomness of evolution is preserved.

2.3 ANT COLONY OPTIMIZATION

The Ant Colony Optimization (ACO) is a metaheuristic inspired by the behaviour of real ants.Ants and other insects that live in a colony, like bees, termites and wasps, can be seen as distributed systems, that in spite of the simplicity of each individual, present a high level of social organization when observed together.

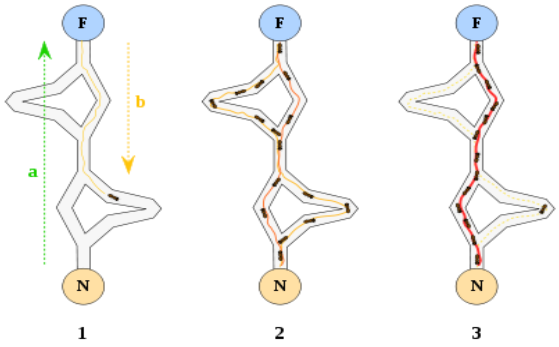

Figure 2.3.1 Simulated ants

The original idea comes from observing the exploitation of food resources among ants, in which ants' individually limited cognitive abilities have collectively been able to find the shortest path between a food source and the nest.
1. The first ant finds the food source (F), via any way (a), then returns to the nest (N), leaving behind a trail pheromone (b)
2. Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route.
3. Ants take the shortest route; long portions of other ways lose their trail pheromones.

In a series of experiments on a colony of ants with a choice between two unequal length paths leading to a source of food, biologists have observed that ants tended to use the shortest route. A model explaining this behavior is as follows:

1. An ant (called "blitz") runs more or less at random around the colony;
2. If it discovers a food source, it returns more or less directly to the nest, leaving in its path a trail of pheromone;
3. These pheromones are attractive, nearby ants will be inclined to follow, more or less directly, the track;
4. Returning to the colony, these ants will strengthen the route;
5. If there are two routes to reach the same food source then, in a given amount of time, the shorter one will be traveled by more ants than the long route;
6. The short route will be increasingly enhanced, and therefore become more attractive;
7. The long route will eventually disappear because pheromones are volatile;
8. Eventually, all the ants have determined and therefore "chosen" the shortest route.

Ants use the environment as a medium of communication. They exchange information indirectly by depositing

pheromones, all detailing the status of their "work". The information exchanged has a local scope, only an ant located where the pheromones were left has a notion of them.

This system is called "Stigmergy" and occurs in many social animal societies. The mechanism to solve a problem too complex to be addressed by single ants is a good example of a self-organized system.

This system is based on positive feedback (the deposit of pheromone attracts other ants that will strengthen it themselves) and negative

Theoretically, if the quantity of pheromone remained the same over time on all edges, no route would be chosen. However, because of feedback, a slight variation on an edge will be amplified and thus allow the choice of an edge. The algorithm will move from an unstable state in which no edge is stronger than another, to a stable state where the route is composed of the strongest edges.

The basic philosophy of the algorithm involves the movement of a colony of ants through the different states of the problem influenced by two local decision policies, viz., *trails* and *attractiveness*. Thereby, each such ant incrementally constructs a solution to the problem. When an ant completes a solution or during the construction phase,the ant evaluates the solution and modifies the trail value on the components used in its solution. This pheromone information will direct the search of the future ants.

Furthermore, the algorithm also includes two more mechanisms, viz., *trailevaporation* and *daemonactions*. *Trailevaporation* reduces all trail values over time thereby avoiding any possibilities of getting stuck in local optima. The *daemonactions* are used to bias the search process from a non-local perspective.

Example pseudo-code and formulae

```
procedure ACO_MetaHeuristic
while(not_termination)
generateSolutions()
daemonActions()
pheromoneUpdate()
end while
end procedure
```

EDGE SELECTION

An ant is a simple computational agent in the ant colony optimization algorithm. It iteratively constructs a solution for the problem at hand. The intermediate solutions are referred to as solution states. At each iteration of the algorithm, each ant moves from a state x to state y, corresponding to a more complete intermediate solution. Thus, each ant k computes a set Ak(x) of feasible expansions to its current state in each iteration, and moves to one of these in probability.

For ant *k*, the probability p_{xy}^k of moving from state *x* to state *y* depends on the combination of two values, viz., the *attractiveness* η_{xy} of the move, as computed by some heuristic indicating the *a priori*desirability of that move and the *trail level* τ_{xy} of the move, indicating how proficient it has been in the past to make that particular move.

The *trail level* represents a posteriori indication of the desirability of that move. Trails are updated usually when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions, respectively.

In general, the k^{th} ant moves from state *x* to state *y* with probability

$$p_{xy}^k = \frac{(\tau_{xy}^\alpha)(\eta_{xy}^\beta)}{\sum(\tau_{xy}^\alpha)(\eta_{xy}^\beta)}$$

Where τ_{xy} is the amount of pheromone deposited for transition from state *x* to *y*, $0 \leq \alpha$ is a parameter to control the influence of τ_{xy}, η_{xy} is the desirability of state transition *xy* (*a priori* knowledge, typically 1 / d_{xy}, where *d* is the distance) and $\beta \geq 1$ is a parameter to control the influence of η_{xy}.

PHEROMONE UPDATE

When all the ants have completed a solution, the trails are updated by

$$\tau_{xy}^k = (1-\rho)\tau_{xy}^k + \Delta\tau_{xy}^k$$

Where τ_{xy}^k is the amount of pheromone deposited for a state transition *xy*, ρ is the *pheromone evaporation coefficient* and $\Delta\tau_{i,j}^k$ is the amount of pheromone deposited.

MODIFICATION ALGORITHM

Modification strategies act upon the search set of the complete solutions: starting with a complete solution and proceeding by modifications of it (e.g., Local Search)

GENERIC LOCAL SEARCH

```
procedure Modification heuristic()
define neighborhood structure();
s ← get initial solution(S);   sbest ← s;
while (¬ stopping criterion)
s0 ←select solution from neighborhood(N(s));
if (accept solution(s0))
s ←s0;   if (s < sbest)
```


sbest ← s;
end if end if
end while return sbest;

ANT COLONY OPTIMIZATION WITH LOCAL SEARCH

As a matter of fact, the best instances of ACO algorithms for (static/centralized) combinatorial problems are those making use of a problem-specific local search daemon procedure. It is conjectured that ACO's ants can provide good starting points for local search.

More in general, a construction heuristic can be used to quickly build up a complete solution of good quality, and then a modification procedure can take this solution as a starting point, trying to further improve it by modifying some of its parts

This hybrid two-phase search can be iterated and can be very effective if each phase can produce a solution which is locally optimal within a different class of feasible solutions.

III. PROPOSED METHODOLOGY

3.1 AN ANT COLONY OPTIMIZATION BASED EVOLUTIONARY MULTIOBJECTIVE CLUSTERING WITH VARIABLE LENGTH CHROMOSOMES & LOCAL SEARCH

An Ant Colony Optimization algorithm (ACO) is essentially a system based on agents which simulate the natural behaviour of ants, including mechanisms of cooperation and adaptation. Use of this kind of system as a new metaheuristic was proposed in order to solve combinatorial optimization problems. This new metaheuristic has been shown to be both robust and versatile – in the sense that it has been successfully applied to a range of different combinatorial optimization problems.

In this research work, the Ant Colony Optimization approach is used to produce best optimal clusters, and also detects the overlapping in clusters. Combining the evolutionary muti-objective clustering method with Ant Colony Optimization reduces the overlapping using the distance calculation mechanism. This method successfully reduces the overlapping clusters compared to other existing methods. For comparison this approach uses six different data sets such as Pima, Iris, Spiral, Cancer etc., Also this approach is compared with other two methods for clustering evaluation measures like purity, rand index and time taken for clustering.

First step in this work is to input the parameters for number of cluster centers, population rate, and number of generations, crossover rate and mutation rate. Population rate indicates the number of ants and the generation count tells the number of clusters. Crossover and mutation rate is give for feature selection using genetic algorithm.

Second step is the selection of data set for clustering. First the existing evolutionary multiobjective clustering with fixed length chromosomes method is applied on the data set and the rate of overlapping, purity, rand index and time taken for clustering are evaluated. Second, the Evolutionary multiobjective clustering with variable length chromosomes method is applied.

In the next step, Ant Colony Optimization based evolutionary multi-objective clustering is applied on the data set. Ant colony optimization produces best optimal clusters and detects the overlapping clusters. Using the distance calculation method the cluster element which is common to two clusters is taken as point and the distance of that point is calculated from both cluster centers. In that, the shortest distance will be taken and the element is added to that cluster. The problem of overlapping is reduced by this way.

ANT COLONY ALGORITHM PSEUDOCODE

Training set = all training cases; attributes that are not yet used by the ant.
WHILE (No. of cases in the Training set > max_uncovered_cases)
    i=0;
    REPEAT i=i+1;
    Ant i incrementally constructs a classification rule;
    Prune the just constructed rule;
    Update the pheromone of the trail followed by Ant i;
    UNTIL (i ≥ No_of_Ants) or (Ant i constructed the same rule as the previous No_Rules_Converg-1 Ants)
    Select the best rule among all constructed rules;
    Remove the cases correctly covered by the selected rule from the training set;
END WHILE

ANT COLONY CROSSOVER

Crossover operator is used to generate new individual and it can retain good features from the current generation. For the purpose of crossover, the cluster centers are considered to be indivisible, i.e., the crossover points can only lie in between two cluster centers.

The crossover operation, applied stochastically, must ensure that information exchange takes place in such a way that both the offspring's encode the centers of at least two clusters.

Choose a random branch, B, from root to a leaf in program tree **Pn**

For every edge i, j in B
Probability of choosing node i as root of subtree Sn, where i is parent and j is a child node is given by:

$$p(i, n) = (\tau \max(n) - \tau \min(n) + \tau_{i,j}(n)) / T(n)$$

Choose random branch, B, from root to a leaf in program tree **Pm**

For every edge i, j in B

Probability of choosing node i as root of subtree Sm,

$p(i, m) = (\tau \max(m) - \tau \min(m) + \tau_{i,j}(m)) / T(m)$

Where T(k) is given by:

$T(k) = \Sigma_{i,j \in E(k)} (\tau \max(k) - \tau \min(k) + \tau_{i,j}(k))$

And

$\tau_{i,j}(k) = C(V(k,i),V(k,j))$

$\tau \max(k) = \max_{i,j \in E(k)} (\tau_{i,j}(k))$

$\tau \min(k) = \min_{i,j \in E(k)} (\tau_{i,j}(k))$ and

$E(k) = \{$ edges in kth program subtree $\}$

ANT COLONY OPTIMIZATION ALGORITHM

Step 1: Initialize the required variables, functions, input trajectory, output trajectory

Step 2: Set the initial pheromone trails value.

Step 3: Each ant is individually placed on initial state with empty memory

Step 4: For each ant go to step 5

Step 5: choose in probability the state to move into

Step 6: append the chosen move to the k-th ant's set tabk and goto Step 7

Step 7: If the ant k completed the solution move to Step 8 else move to Step 5

Step 8: For each ant move ($l\psi$) and go to step 9

Step 9: compute $\Delta\tau l\psi$ goto Step 10

Step 10: Update the trail matrix and if all ant moved goto Step 11 else go to Step 8

Step 11: If the termination condition is satisfies then stop the process else go to Step 4

After clustering and overlapping detection, cluster evaluation is done by calculating Purity, Rand Index, and Rate of overlapping and the Time taken for clustering using respective calculations. Results are stored in text files for all the six data sets and the graphs are generated for each measure. At last a comparison table shows the result of all the measures for three approaches and the result proves that Ant Colony approach is best method for overlapping detection.

IV. RESULT ANALYSIS

The Ant Colony Optimization based evolutionary multi-objective clustering for overlapping cluster detection is tested on six benchmark data sets known as Wine, Pima, Glass, Iris, Cancer and Spiral. Names and characteristics of all datasets available on UCI machine learning repository. These data sets are used to evaluate the performance of the Ant Colony Optimization based Evolutionary Multiobjective Clustering.

The performance of Ant Colony Optimization based Evolutionary Multiobjective Clustering is measured by different clustering validity metrics: Purity, Rand index, Rate of overlapping clusters and also the time taken for clustering. We compare the proposed algorithm with the existing Fuzzy Genetic-based Evolutionary Multiobjective Algorithm (FEMCOC) with fixed length and variable-length Evolutionary Multiobjective Clustering algorithm.

Parameters like number of centers, population rate, number of generation, crossover rate and mutation rate are given as user input.

4.1 COMPARISION GRAPH

The performance comparison of proposed algorithm with Evolutionary Multiobjective Clustering is given in the following figures. From the figures, it can be observed that Ant Colony Optimization based Evolutionary Multiobjective Clustering obtains very good values in comparison to Evolutionary Multiobjective Clustering and variable length Evolutionary Multiobjective Clustering.

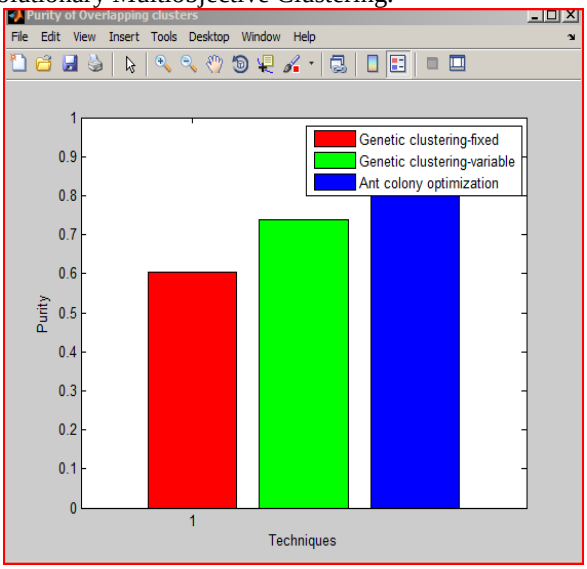

Fig 4.1.1 Purity of Clusters

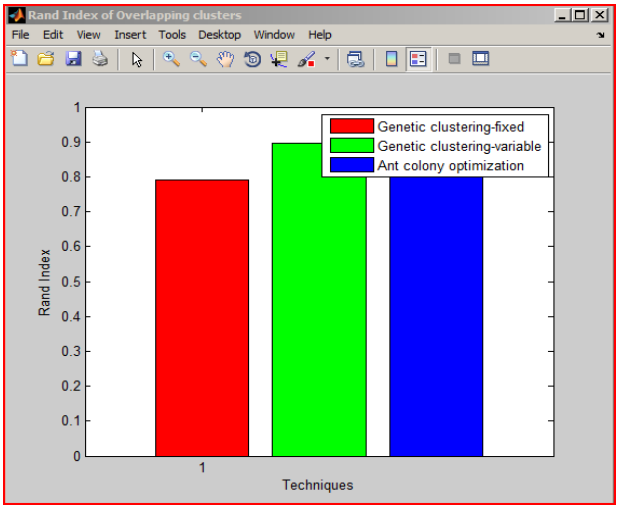

Fig 4.1.2 Rand Index of Clusters

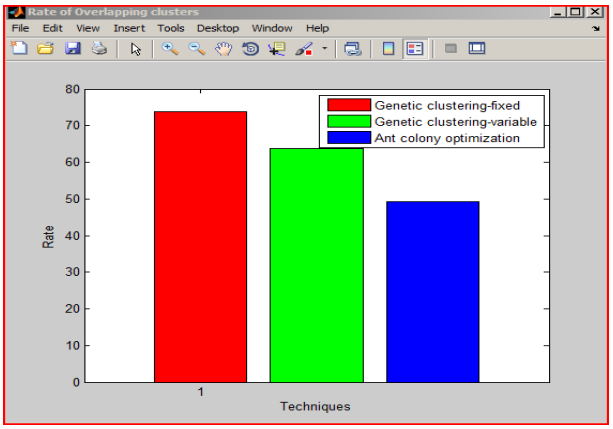

Fig 4.1.3 Rate of overlapping clusters

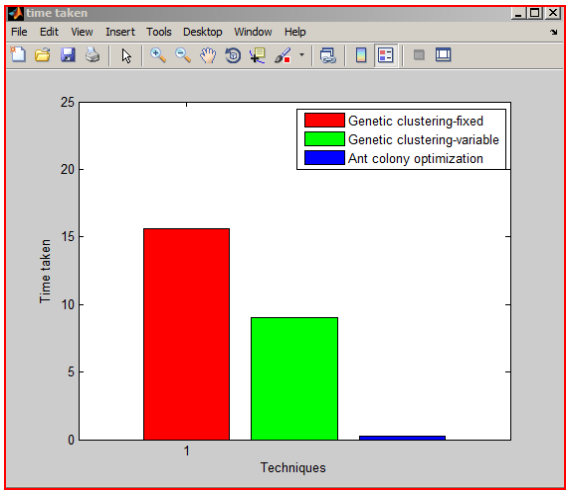

Fig 4.1.4 Time Taken for clustering

4.3 COMPARISON TABLE FOR PERFORMANCE ANALYSIS

A comparison of Ant Colony Optimization based Evolutionary Multiobjective Clustering, Evolutionary Multiobjective Clustering and Variable-length Evolutionary Multiobjective Clustering in the cases of Wine, Pima, Glass, Iris, Cancer and Spiral datasets for the Purity, Rand index, Rate of overlapping and time taken for clustering are given in Table 4.3.1.

Table 4.3.1: The Comparison Table

Dataset	Wine	Glass	Pima	Iris	Cancer	Spiral
Purity Fixed	6.030e-001	6.866e-001	6.532e-001	6.503e-001	6.963e-001	6.511e-001
Purity Variable	7.394e-001	7.857e-001	7.961e-001	7.762e-001	7.189e-001	7.869e-001
Purity ACO	9.754e-001	9.350e-001	9.206e-001	9.947e-001	9.171e-001	9.058e-001
R.I Fixed	7.901e-001	7.831e-001	7.724e-001	7.053e-001	7.109e-001	7.540e-001
R.I Variable	8.947e-001	8.298e-001	8.553e-001	8.105e-001	8.641e-001	8.001e-001
R.I ACO	9.339e-001	9.798e-001	9.613e-001	9.606e-001	9.114e-001	9.910e-001
Rate Fixed	7.379e+001	7.369e+001	7.256e+001	7.425e+001	7.231e+001	7.354e+001
Rate Variable	6.373e+001	6.147e+001	6.021e+001	6.407e+001	6.224e+001	6.382e+001
Rate ACO	4.925e+001	4.859e+001	4.825e+001	4.515e+001	4.848e+001	4.770e+001
Time Fixed	1.560e+001	1.825e+001	1.446e+001	1.283e+001	1.766e+001	1.180e+001
Time Variable	8.999e+000	1.390e+001	8.226e+000	6.087e+000	1.032e+001	5.803e+000
Time ACO	2.529e-001	1.589e-001	2.636e-001	1.563e-001	2.359e-001	1.621e-001

V. CONCLUSIONS AND FUTURE ENHANCEMENT

The experimental results demonstrate that comparing to the existing Evolutionary Multiobjective Clustering algorithms, the Ant Colony Optimization based Evolutionary Multiobjective Clustering successfully identifies the overlapping clusters in complex data set. It evaluates the Purity, Rand index, Rate of overlapping clusters and time taken for clustering.

Further to enhance the work, we can apply the proposed procedure for real data sets with various fitness functions and evaluation measures.

REFERENCES

[1]. K.P Malarkodi, S.Punithavathy, "A Fuzzy Based Evolutionary Multi-objective Clustering for overlapping Clusters Detection" IJSER Vol 2, Issue 9, Sept 2011. ISSN 2229-5518.

[2]. E. Falkenauer. Genetic Algorithms and Grouping Problems. John Wiley & Sons,1998

[3]. U. Maulik and S. Bandyopadhyay. Genetic algorithm-based clustering technique. Pattern Recognition, 33:1455{1465, 2000.

[4]. K. Deb. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester, UK,

[5].L. MacQueen. Some methods for classification and analysis of multivariate observations.In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281-297. University of California Press, Berkeley, 1967.

[6]. D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization.In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), pages 283 to 290. Morgan Kaufmann Publishers, 2001.

[7]. D. W. Corne, J. D. Knowles, and M. J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 839-848. Springer,2000.

[8]. N. J. Radcli_e. Equivalence class analysis of genetic algorithms. Complex Systems, 5:183 {205, 1991}.

[9].A. Topchy, A. K. Jain, and W. Punch. A mixture model for clustering ensembles.In Proceedings SIAM Conf. on Data Mining, 2004.
[10].J.D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Ph.D. Thesis, Vanderbilt University, Nashville, TN, 1984.
[11]. A. Topchy, A. K. Jain, W. Punch, "Clustering ensembles: models of consensus and weak partitions,"IEEE Intelligence, vol. 27, no. 2, pp. 1866- 1881, 2005
[12]. R.Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, In Proceedings of ACM-SIGMOD 1998 Int. Conf. on Management of Data: 94-105, 1998.
[13]. C.-L.Hwang and A.S.Masud. Multiple Objective Decision Making—Methods and Applications. Springer- Verlag, Heidelberg, 1979.
[14]. K. S. N. Ripon, "Real jumping gene genetic algorithm (RJGGA) for evolutionary multi-objective optimization problems," M.Phil Thesis, Department of Computer Science, City University of Hong Kong, Hong Kong, 2006
[15]. K. S. N. Ripon, C. –H. Tsang, and S. Kwong, "Multi-objective data clustering using variable-length real jumping genes genetic algorithm and local search method," in Proc. International Joint Conference on Neural Networks (ICJNN'06), Vancouver, Canada, 2006, pp. 3609-3616.
[16]. J. Handl, and J. Knowles, "Evolutionary multi-objective clustering," in Proc. Eighth Int. Conf. on Parallel Problem Solving from Nature, 2004, pp. 1081-1091.
[17]. Ajith Abraham and Lakhmi Jain, Evolutionary Multiobjective Optimization, Oklahoma State University, USA, 2007
[18]. M. Dorigo, V. Maniezzo, et A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics--Part B , volume 26, numéro 1, pages 29-41, 1996.
[19]. A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, acts de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.
[20]. S. Goss, S. Aron, J.-L. Deneubourg et J.-M. Pasteels, *The self-organized exploratory pattern of the Argentine ant*, Naturwissenschaften, volume 76, pages 579-581, 1989.
[21]. J.-L. Deneubourg, S. Aron, S. Goss et J.-M. Pasteels, *The self-organizing exploratory pattern of the Argentine ant*, Journal of Insect Behavior, volume 3, page 159, 1990.
[22]. M. Zlochin, M. Birattari, N. Meuleau, et M. Dorigo, Model-based search for combinatorial optimization: A critical survey, Annals of Operations Research, vol. 131, pp. 373-395, 2004
[23]. M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computation", Vol. 1(1), pages 53-66, 1997.
[24]. A. Banerjee, S. Basu, C. Krumpelman, J. Ghosh, and R. Mooney. Model-based overlapping clustering. Proceedings of KDD2005, pages 100–106, 2005.
[25]. M. Deodhar, *Consensus clustering of microarray data*, 2006.

AUTHOR PROFILE

Name: S.PUNITHAVATHY Qualification: M.Sc., M.Phil.
Department: Computer Science
Institution: Pioneer college of Arts & Science, Coimbatore.
Years of Experience: 6
Email-ID:punithavathy26@gmail.com

