ISSN:0975-9646

Preliminary Review of Swarm Intelligence: A Clever Algorithm and Data Clustering

NazneenTarannum S. H. Rizvi, Prof.R.R.Keole
CS&IT Dept, H.V.P.M COET, Amravati University, India

Abstract— **Swarm intelligence deals with the behavior of natural or artificial swarms. Swarms are systems that consist of many individuals that are organized and coordinated by principles of decentralized control, indirect communication, and self-organization. The concept is employed in work on artificial intelligence introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. This paper explores the role of SI in clustering different kinds of datasets. It finally describes a new SI technique for partitioning any dataset into an optimal numb er of groups through one run of optimization.**

Keywords— **Ant Colony Optimization, Data Clustering, Particle Swarm Optimization, Swarm Intelligence**

I. INTRODUCTION

Swarm intelligence is a relatively new subfield of artificial intelligence which studies the emergent collective intelligence of simple multi-agents. It is based on social, intelligent and self-organized behavior that can be observed in nature, such as ant colonies, flocks of birds, fish schools and bee hives, where a number of individuals with limited capabilities are able to come to intelligent solutions for complex problems. In recent years the swarm intelligence paradigm has received widespread attention in research, mainly as *1)Ant Colony Optimization* (ACO) that investigates probabilistic algorithms inspired by the stigmergy and foraging behavior of ants ,and 2)*Particle Swarm Optimization*(PSO) that investigates probabilistic algorithms inspired by the flocking, schooling and herding. The phenomenon of Swarm intelligence is of paramount significance and is an inspiration for several Swarm intelligence algorithms

A).Properties of a Swarm Intelligence System

The main properties of the collective behavior can be pointed out as follows and is summarized in Figure 1.

Homogeneity: every bird in flock has the same behavioral model. The flock moves without a leader, even though temporary leaders seem to appear.

Locality: its nearest flock-mates only influence the motion of each bird. Vision is considered to be the most important senses for flock organization.

Collision Avoidance: avoid colliding with nearby flock mates.

Velocity Matching: attempt to match velocity with nearby flock mates.

Flock Centering: attempt to stay close to nearby flock mates

Individuals attempt to maintain a minimum distance between themselves and others at all times. This rule is given the highest priority and corresponds to a frequently observed behavior of animals in nature .If individuals are not performing an avoidance maneuver they tend to be attracted towards other individuals (to avoid being isolated) and to align themselves with neighbors

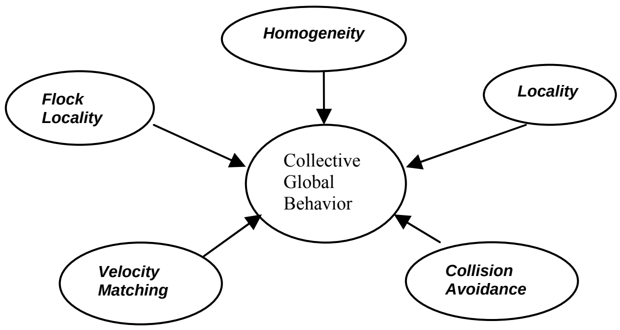


Fig.1. Main traits of collective behavior

II. LITERATURE SERVEY

It is an undeniable fact that all of us are optimizers and make decisions for the sole purpose of maximizing our quality of life, productivity in time, as well as our welfare in some way or another. Optimization is defined as a methodology to determine the best-suited solution for any problem under given state of affairs. Optimization plays very important role in almost every area of applications but data clustering has become a promising area of research work these days. Different researchers are proposing different approaches for optimizing clustering challenges. Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Stimulated Annealing (SA) are most popular optimization techniques till date. The basic classification of optimization techniques is studied and analyzed as depicted in the Fig. 1.

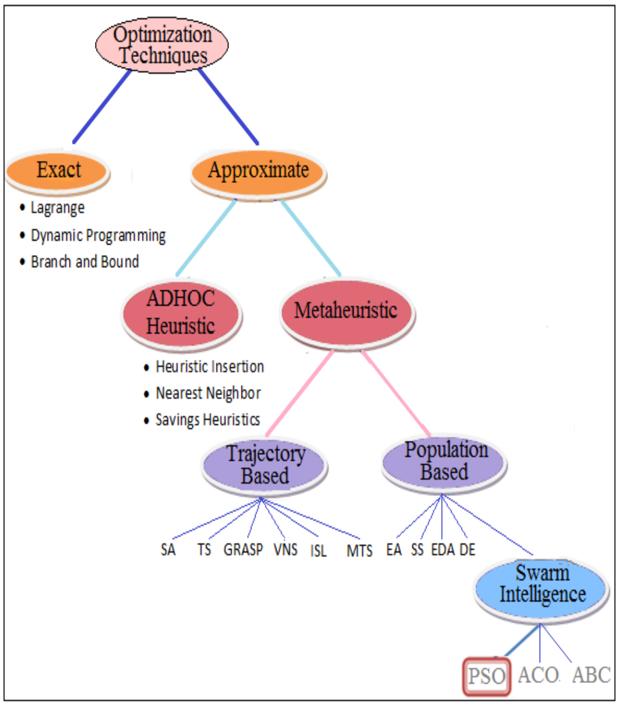


Fig.2. Analyzed Classification of Optimization Techniques

A). PARTICLE SWARM OPTIMISATION

Particle swarm optimization (Kennedy and Eberhart 1995; Kennedy, Eberhart and Shi, 2001) is a population based stochastic optimization technique for the solution of continuous optimization problems. It is inspired by social behaviors in flocks of birds and schools of fish. In particle swarm optimization (PSO), a set of software agents called particles search for good solutions to a given continuous optimization problem. Each particle is a solution of the considered problem and uses its own experience and the experience of neighbor particles to choose how to move in the search space. In practice, in the initialization phase each particle is given a random initial position and an initial velocity. The position of the particle represents a solution of the problem and has therefore a value, given by the objective function. While moving in the search space, particles memorize the position of the best solution they found. At each iteration of the algorithm, each particle moves with a velocity that is a weighted sum of three components: the old velocity, a velocity component that drives the particle towards the location in the search space where it previously found the best solution so far, and a velocity component that drives the particle towards the location in the search space where the neighbor particles found the best solution so far. PSO has been applied to many different problems and is another example of successful artificial/engineering swarm intelligence system.

In PSO, a population of conceptual `particles' is initialized with random positions X*i* and velocities V*i*, and a function, *f*, is evaluated, using the particle's positional coordinates as input values. In an n-dimensional search space,
X*i* = (x*i*1, x*i*2, x*i*3,...,x*in*) and V*i*= (v*i*1, v*i*2, v*i*3,...,v*in*): Positions and velocities are adjusted, and the function is evaluated with the new coordinates at each time-step.

After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).
v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[])---------------(a)
present[]=present[]+v[]----------(b)

Where:-

v[]	is the particle velocity,
persent[]	is the current particle (solution).
pbest[] and gbest[]	are defined as stated before. i.e. personal best and global best resp.
rand ()	is a random number between (0,1).
c1, c2	are learning factors. Usually c1 = c2 = 2

Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.

Once the iterations are terminated, most of the particles are expected to converge to a small radius surrounding the global optima of the search space.

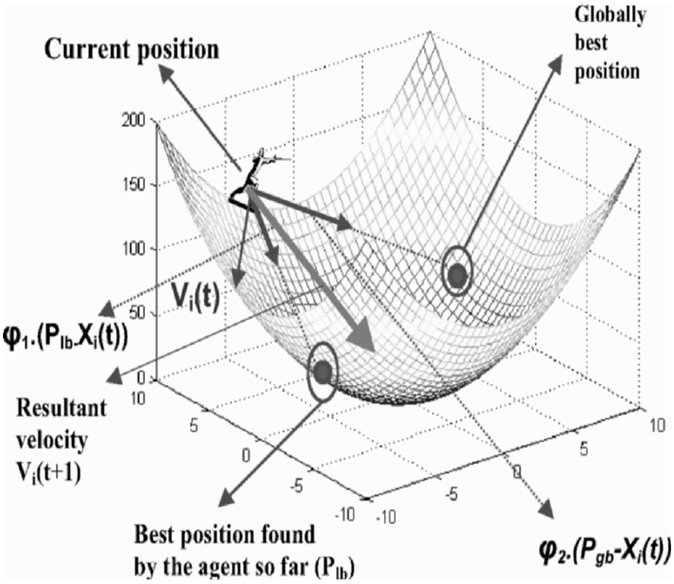


Fig. 3. Illustrating the velocity updating scheme of basic PSO

A pseudo code for the PSO algorithm is presented in Algorithm 1

Algorithm 1: The PSO Algorithm

Input: Randomly initialized position and velocity of the particles: X*i*(0) and V*i*(0)
Output: Position of the approximate global optima X¤
while terminating condition is not reached do
  for *i* = 1 to *numberofparticles* do
    Evaluate the ¯tness: =*f*(X*i*(*t*));
    Update P(*t*)andg(*t*);
  Adapt velocity of the particle using Equation 3;
  Update the position of the particle;
  end for
end while

B) ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.

Metaheuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. The metaheuristic part permits the low level heuristic to obtain solutions better than those it could have achieved alone, even if iterated.

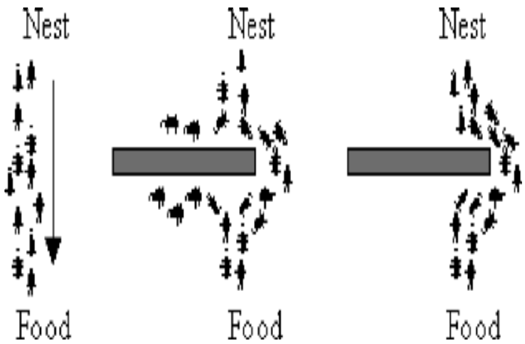


Fig. 4. Illustrating the behavior of real ant movements.

1).Working Of Ant Colony System

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 $A_k(x)$ 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 kth ant moves from state x to state y with probability

$$p_{xy}^k = \frac{(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}{\sum_{y \in \mathrm{allowed}_y} (\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}$$

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}. τ_{xy} and η_{xy} represent the attractiveness and trail level for the other possible state transitions.

Pheromone update

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

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

where τ_{xy} is the amount of pheromone deposited for a state transition xy, ρ is the pheromone evaporation coefficient and $\Delta\tau_{xy}^k$ is the amount of pheromone deposited by kth ant, typically given for a TSP problem (with moves corresponding to arcs of the graph) by

$$\Delta\tau_{xy}^k = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases}$$

where L_k is the cost of the kth ant's tour (typically length) and Q is a constant.

The characteristic of ACO algorithms is their explicit use of elements of previous solutions. ACO algorithms are inspired from the behavior of real ant colonies, when ants search for food from their nest to food sources. Ants cooperate and communicate indirectly via their pheromone, where they leave a trail to the path they explore. The more pheromone on a specific trail, the higher the possibility of that trail to be followed by the ants. This behavior inspired researchers to develop the first ACO algorithm, called the ant system (AS), which has been applied to the well-known travelling salesman problem (TSP). Moreover, AS has been applied to other combinatorial optimization problems, such as the quadratic assignment problem the job scheduling problem [7], the vehicle routing problem (VRP) and many other optimization problems.

The main steps of ACO algorithm are presented in Algorithm 2.

Algorithm 2: Procedure ACO

1: Initialize pheromone trails;
2: repeat {at this stage each loop is called an iteration}
3: Each ant is positioned on a starting node
4: repeat{at this level each loop is called a step}
5: Each ant applies a *state transition rule like rule (2)* to incrementally build a solution and a *local pheromone-updating rule like rule (4);*
6: until all ants have built a complete solution
7: global pheromone-updating rule like rule (5) is applied.
8: until terminating condition is reached

C). ARTIFICIAL BEE COLONY(ABC) OPTIMISATION

Karaboga proposed Artificial Bee Colony (ABC) in 2005 based on inspecting the behaviors of real bees on finding nectar and sharing the information of food sources to the bees in the nest. Three kinds of bees are defined as the artificial agents. Every kind of bee plays different and important roles in the optimization process. The artificial agents are called the employed bee, the onlooker, and the scout with distinct responsibilities. The employed bee stays on a food source, which represents a spot in the solution space, and provides the coordinate for the onlookers in the hive for reference. The onlooker bee receives the locations of food sources and selects one of the food sources to gather the nectar. The scout bee moves in the solution space to discover new food sources.

The process of the ABC optimization is described as follows:
(1) Initialization. Spray ne percentage of the populations into the solution space randomly, and then calculate their fitness values, called the nectar amounts, where ne represents the ratio of employed bees to the total population. Once these populations are positioned into the solution space, they are called the employed bees. Evaluate the fitness of the employed bees and take the fitness to be their amount of nectar.

(2)Move the onlookers. We calculate the probability of selecting a food source, where θ_i denotes the position of the i-th employed bee, $F(\cdot)$ is the fitness function, S represents the number of employed bees, and Pi is the probability of selecting the i-th employed bee. Then we select a food source to move to by roulette wheel selection for every onlooker bee and determine the nectar amounts. The onlookers are moved where xi denotes the position of the i-th onlooker bee, t denotes the iteration number, θ_k is the randomly chosen employed bee, j represents the dimension of the solution, and $\varphi(\cdot)$ produces a series of random variable in the range $[-1, 1]$.

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^{S} F(\theta)}$$

$$x_{ij}(t+1) = \theta_{ij}(t) + \varphi(\theta_{ij}(t) - \theta_{kj}(t)) \ldots \quad (12)$$

(3)Update the best food source found so far. We record the best fitness value and the position, which are found by the bees.
(4)Move the scouts. If the fitness values of the employed bees are not improved by a consecutive number of iterations, called “Limit,” those food sources are abandoned, and these employed bees become the scouts. The scouts are moved by Eq. (11), where r is a random number and $r \in [0, 1]$.

$$\theta_{ij} = \theta_{j\,min} + r \cdot (\theta_{j\,max} - \theta_{j\,min}) \quad (13)$$

(5)Termination checking. If the amount of the iterations satisfies the termination condition, we terminate the program and output the results; otherwise, go back to Step (2).

Clustering means the act of partitioning an unlabeled dataset into groups of similar objects. The goal of clustering is to group sets of objects into classes such that similar objects are placed in the same cluster while dissimilar objects are in separate clusters. Clustering is used as a data processing technique in many different areas, including artificial intelligence, bioinformatics, biology, computer vision, city planning, data mining, data compression, earthquake studies, image analysis, image segmentation, information retrieval, machine learning, marketing, medicine, object recognition, pattern recognition, spatial database analysis, statistics and web mining. Clustering is helpful whenever there is huge amount of data.

III. DATA CLUSTERING: AN OVERVIEW

A). Cluster Analysis Using PSO

Cluster analysis is a collection of methods, which identifies groups of instances that have similar characteristics. Clustering can be achieved by various algorithms [7]-[8] that differ significantly in their methods of generation of cluster. Typically a cluster includes groups with low distances among the cluster members, dense areas of the data space [4] or particular distributions [10]. The appropriate clustering algorithm and parameter settings depend on the individual data set being used for clustering and intended use of the results [11]. In this paper, a cluster analysis model is proposed which is based on most popular nature inspired technique known as PSO. It has following steps (Algorithm-1). In this algorithm, X is the dataset, $C=\langle C_1, C_2, \ldots C_n \rangle$ is the cluster centers vector, C_i is the ith cluster center, n is the expected total number of clusters in dataset X, $V=\langle V_1, V_2, \ldots V_n \rangle$ represents vector of random velocities. V_i is the velocity vector of C_i. V_{new} and C_{new} is new velocity and next cluster center position respectively. The Algorithm computes n number of clusters in a given dataset. Here n (n>1) is the number of cluster provided by the user. Initial cluster centers will be selected randomly. Euclidian distances are computed from randomly selected cluster center. Fitness of all instances of generated clusters has been calculated as it is used as lbest.

$$F_{X_i} = \frac{1}{\sum_{k=1}^{N} |X_i - X_k|^2}$$

Here n number of lbest are generated and next velocity vectors have been computed by using initial velocity, lbest and gbest, Next positions of cluster centers are generated by using new velocity. These steps will be repeatedly executed until and unless the target clusters are found. The positions and velocities of the particles initially in search space denoted by V and X. The new velocities and positions of the particles for next iterations [5] can be evaluated by using the equations 1 and 2. Fig-1 and fig-2 demonstrates generation of initial cluster and target cluster respectively. Flowchart of this procedure is shown on fig-3.

$$F_{CT} = \frac{k}{\left({}^{1}\!/\!\sum_{i=1}^{N} \sum_{j=1}^{n} |C_j - X_i|^2 \right) + d} \quad (4)$$

ALGORITHM-3 PSO CLUSTERING (X, n, S)

X – Dataset to be clustered, n – Number of cluster.

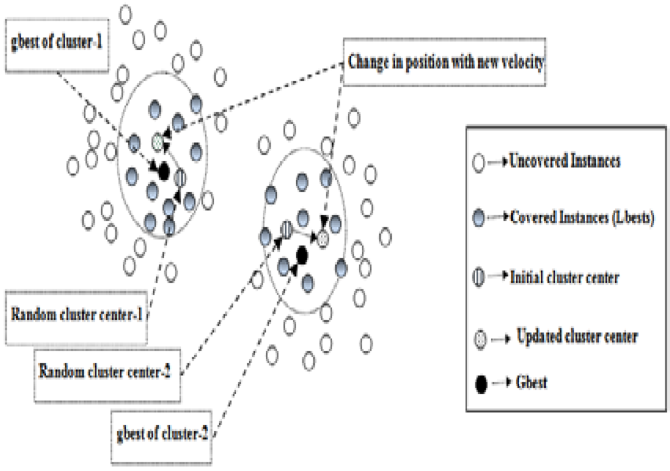
S – Small positive valued constant

V=<v1,v2,….vk>. Here n is the number of cluster and k is dimension of dataset.VR1R,VR2R,VR3R ,…VRnR R Rare initial random velocity vector for CR1R, CR2R, CR3R,…CRnR respectively .

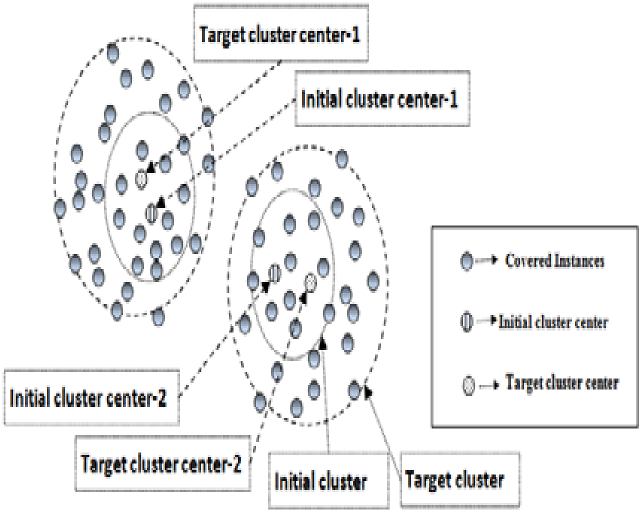
1. load dataset X and set number of cluster 'n' to be found. 2. Set initial random cluster center vector <CR1R,CR2R,….CRnR> and random velocity V=<VR1R,VR2R,….VRnR>.
3. Compute Euclidian distance from all clusters<CR1R,CR2R,….CRnR> to all the instances of X and
4. Create clusters based on Euclidian distances.
5. Calculate fitness of all instances (FRxiR) of clusters by using the equation-3 and generate lbest.
6. Choose the instance having highest fitness in each cluster is chosen as gbest of that cluster. Generate n number of gbest.
7. Compute new velocity VRNEW Rout of initial velocity, lbest and gbest by use of equation-1.
8. Update the position of all cluster centers (centroid) with new velocity VRNEWR and generate CRNEW Rby using equation-2.
9. if (Euclidian distance(C,CRNEWR) <= S)
10. goto step-3
11. else display final clusters
12. goto step-14
13. end if
14. Calculate the performance of PSO (FRCTR) using equation-4.
15. end

$$Fc = \frac{1}{\sum_{i=1}^{N} \sum_{j=1}^{n} |C_j - X_i|^2} \quad (5)$$

FRXiR represents fitness of an instance, where X is the dataset, N is the number of instances in X, XRi Ris the iPthP instance of X.FRC Rrepresents fitness of cluster center vector, where X is the dataset, N is the number of instances in X, XRi Ris the iPthP instance of X.FRCTR represents fitness of particular clustering method of technique, where X is the dataset used, N is the number of instances in X, XRi Ris the iPthP instance of X, k is a positive constant and d is a small-valued constant. Most of time the PSO algorithm stops based on two parameter.1- exceed maximum velocity range and 2- maximum number of iterations. Our proposed model will stop in neither of these conditions. It will stop when it reaches a value (S) . S is the difference between old cluster center and new cluster center. Here s is small valued constant. Value of S depends upon dataset being used. During experiments, values of S has been chosen for different datasets and listed at table-2 and table-3.



(Fig-5: Initial cluster centers, bests and gbest)



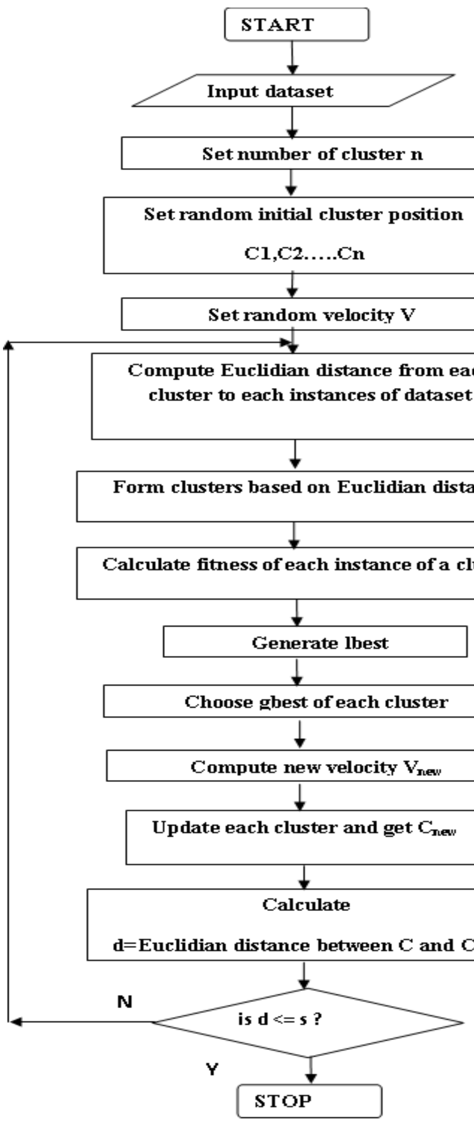
(Fig-6: Formation of Target cluster)

B). Clustering With SI Algorithms

In this section, We outline a new algorithm which employs the PSO model to automatically determine the number of clusters in a previously unhandled dataset.

1). PSO Based Clustering Algorithms

Data clustering can be viewed as an optimization problem. This view offers us a chance to apply PSO algorithm for evolving a set of candidate cluster centroids and thus determining a near optimal partitioning of the dataset at hand. An important advantage of the PSO is its to cope with local optima by maintaining, recombining and comparing several candidate solutions simultaneously. In contrast, local search heuristics, such as the simulated annealing algorithm only refine a sing
le candidate solution and are notoriously weak in coping with local optima. Deterministic local search, which is used in algorithms like the K-means, always converges to the nearest local optimum from the starting position of the search.



(Fig-7: Flowchart for cluster analysis using PSO)

The PSO clustering algorithm is summarized in Algorithm 4.

Algorithm 4: The PSO Clustering Algorithm

Initialize each particle with K random cluster centers.
for iteration count = 1 to maximum iterations do
for all particle i do
for all pattern X*p* in the dataset do
calculate Euclidean distance of X*p* with all cluster centroids
assign X*p* to the cluster that have nearest centroid to X*p*
end for
calculate the ¯tness function *f*(Z*i;Mi*)
end for
Find the personal best and global best position of each particle.
Update the cluster centroids according to velocity updating and coordinate updating formula of PSO.
end for

PSO based method outperformed K-means, FCM and few other state-of-the-art clustering algorithms.

2. The Ant Colony Based Clustering Algorithms

Ant colonies provide a means to formulate some powerful nature-inspired heuristics for solving the clustering problems. Among other social movements, researchers have simulated the way, ants work collaboratively in the task of grouping dead bodies so, as to keep the nest clean (Bonabeau *et al*., 1999). It can be observed that, with time the ants tend to cluster all dead bodies in a specific region of the environment, thus forming piles of corpses.

Larval sorting and corpse cleaning by ant accomplish certain tasks in robotics .This inspired the Ant-based clustering algorithm. Lumer and Faieta modified the algorithm using a dissimilarity-based evaluation of the local density, in order to make it suitable for data clustering .
This introduced standard Ant Clustering Algorithm (ACA). It
has subsequently been used for numerical data analysis , data-mining ,graph-partitioning, and text-mining Many authors proposed a number of modifications to improve the convergence rate and to get optimal number of clusters. The standard ACA pseudo-code is summarized in Algorithm 3.

Algorithm 5: Procedure ACA

Place every item X*i* on a random cell of the grid;
Place every ant k on a random cell of the grid unoccupied by ants;
iteration count ← 1;
while iteration count < maximum iteration do
for *i* = 1 to *no of ants* do
if unladen ant and cell occupied by item X*i* then
compute *f*(X*i*) and *Ppick¡up*(X*i*);
else
if ant carrying item xi and cell empty then
compute *f*(X*i*) and *Pdrop*(X*i*);
drop item X*i* with probability *Pdrop*(X*i*);

```
            end if
      end if
    move to a randomly selected, neighboring and
    unoccupied cell ;
      end for
      t ← t + 1
end while
print location of items;
```

Like a standard ACO, ant-based clustering is a distributed process that employs positive feedback. Ants are modeled by simple agents that randomly move in their environment. The environment is considered to be a low dimensional space, more generally a two-dimensional plane with square grid. Initially, each data object that represents a multi-dimensional pattern is randomly distributed over the 2-D space.

IV. Applications Of Swarm Intelligence

A). Text Document Clustering

Text document clustering is a fundamental operation used in unsupervised document organization, automatic topic extraction, and information retrieval. Clustering involves dividing a set of objects into a specified number of clusters .

The motivation behind clustering a set of data is to find inherent structure inside the data and to expose this structure as a set of groups. The data objects within each group should exhibit a large degree of similarity while the similarity among different clusters should be minimized. The document clustering technology is different from the classifying technology because the task of document clustering is finding natural groups that were previously unknown in the existing document achieves rather than classifying the document into different categories on the basis of pre-defined categories or some externally imposed artificial criteria. Recent studies have shown that partitional clustering algorithms are more suitable for clustering large datasets. The K-means algorithm is the most commonly used partitional clustering algorithm because it can be easily implemented and is the most efficient one in terms of the execution time. The major problem with this algorithm is that its result is sensitive to the selection of the initial partition and may converge to local optima.

B). Crowd Simulation

Artists are using swarm technology as a means of creating complex interactive systems or simulating crowds. Stanley and Stella in: Breaking the Ice was the first movie to make use of swarm technology for rendering, realistically depicting the movements of groups of fish and birds using the Boids system. Tim Burton's Batman Returns also made use of swarm technology for showing the movements of a group of bats. The Lord of the Rings film trilogy made use of similar technology, known as Massive, during battle scenes. Swarm technology is particularly attractive because it is cheap, robust, and simple. Airlines have used swarm theory to simulate passengers boarding a plane. Southwest Airlines researcher Douglas A. Lawson used an ant-based computer simulation employing only six interaction rules to evaluate boarding times using various boarding methods.

C). Adaptive Routing in Telecommunications Networks

Ant Colony Routing is a general framework in which most swarm intelligence routing algorithms can be placed. SI design has been applied to a wide variety of problems in combinatorial and continuous optimization, telecommunications, robotics, etc., often with excellent results. The task of a routing algorithm is to find paths through the network graph that connect source and destination nodes, while optimizing predefined criteria and possibly satisfying certain constraints. ACR describes SI routing from a perspective that is complementary to the one that is commonly followed in the field, emphasizing the aspects of distributed and cooperative reinforcement learning, and active and passive information sampling. The network nodes are seen as reinforcement learning agents that adaptively learn about network status through passive monitoring of their local traffic and connection topology, and active gathering of non-local information through, for instance, the generation of ant-like agents.

D) Swarm robotics

Swarm robotics is a new approach to the coordination of multirobot systems which consist of large numbers of mostly simple physical robots. It is supposed that a desired collective behavior emerges from the interactions between the robots and interactions of robots with the environment. This approach emerged on the field of artificial swarm intelligence, as well as the biological studies of insects, ants and other fields in nature, where swarm behavior occurs.

Unlike distributed robotic systems in general, swarm robotics emphasizes a *large* number of robots, and promotes scalability, for instance by using only local communication. That local communication for example can be achieved by wireless transmission systems, like radio frequency or infrared.

Potential applications for swarm robotics include tasks that demand for miniaturization (nano robotics, macrobiotics), like distributed sensing tasks in micro machinery or the human body. On the other hand swarm robotics can be suited to tasks that demand cheap designs, for instance mining tasks or agricultural foraging tasks. Also some artists use swarm robotic techniques to realize new forms of interactive art.

V. Conclusions

SI systems consist typically of a population of simple agents or boids interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems In this Paper, we introduced some of the preliminary concepts of Swarm Intelligence (SI) with an emphasis on particle swarm optimization and ant colony optimization algorithms. We then described the basic data clustering terminologies and also illustrated some of the past and ongoing works, which apply different SI tools to pattern clustering problems.

.ACKNOWLEDGEMENT

My thanks to the Guide, Prof. R.R.Keole and Principal Dr.A.B.Marathe, who provided me constructive and positive feedback during the preparation of this paper.

REFERENCES

[1] Ant Algorithms for Discrete Optimization Artificial Life M. Dorigo, G. Di Caro & L. M. Gambardella (1999).

[2] Swarm Intelligence, From Natural to Artificial Systems M. Dorigo, E. Bonabeau, G. Theraulaz

[3] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie, 1992.

[4] T. Stützle et H.H. Hoos, MAX MIN Ant System, Future Generation Computer Systems, volume 16, pages 889-914, 2000

[5] M. Dorigo et L.M. Gambardella, Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, volume 1, numéro 1, pages 53-66, 1997.

[6] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948. doi:10.1109/ICNN.1995.488968

[7] Reynolds, C. W. (1987). "Flocks, herds and schools: A distributed behavioral model". Computer Graphics 21 (4): 25–34.

[8] Lewis, M. Anthony; Bekey, George A. "The Behavioral Self-Organization of Nanorobots Using Local Rules". Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems.

[9] Martens, D.; Baesens, B.; Fawcett, T. (2011). "Editorial Survey: Swarm Intelligence for Data Mining". Machine Learning 82 (1): 1–42.

[10] L. M. Gambardella, E. Taillard and M. Dorigo, "Ant colonies for the quadratic assignment problems," *Journal of Operations Research Society*, vol. 50, pp. 167–176, 1999.

[11] Sandeep Rana, Sanjay Jasola, and Rajesh Kumar ―A review on particle swarm optimization algorithms and their applications to data clustering‖, Springer 24 Nov 2010, pp. 211-222.

[12] D. V. D. Merwe and A. Engelbrecht. Data clustering using particle swarm optimization. In Proceedings of IEEE Congress on Evolutionary Computation 2003, pages 215–220. IEEE, 2003.

.