

# USING SWARM APPROACHES FOR STUDENT SELECTION PROCESS

Reetika Nagar<sup>1</sup>, Sachin Kumar<sup>2</sup>, Gaurav Pratap Singh<sup>3</sup>

<sup>1</sup>Assistant Professor, Department of CSE, ACET, Aligarh, India

<sup>2</sup>Assistant Professor, Department of CSE, SSITM, Aligarh, India

<sup>3</sup>Assistant Professor, Department of CSE, ACET, Aligarh, India

## Abstract

Swarm intelligence has been extensively used in research area because swarm systems are highly scalable, respond well to rapidly changing environments, work collectively without central control and consists of simple agents or particles. Swarm intelligence helps to design optimal solution scape and best results. A large number of applications have been proposed using swarm intelligence for different research area such as ant colony optimization (ACO), bee colony optimization (BCO), particle swarm optimization (PSO), bacterial foraging optimization (BFO), cat swarm optimization (CSO) and many more. In this paper, introduction of swarm intelligence, its variants ant colony optimization and particle swarm optimization, and student selection process has been given. Further, implementation of student selection process (SSP), it is a standardized process conducted to provide jobs to students within educational institutes or in a common job fair, is shown using ant colony optimization and particle swarm optimization.

**Keywords:** Swarm Intelligence, ACO, PSO, Student Selection Process.

\*\*\*

## 1. INTRODUCTION

Swarm intelligence (SI) focuses on the study of the collective behaviour of self-organized and decentralized systems, natural or artificial systems to solve complex problems [1]. A swarm is a group of homogenous, mobile agents which are liable to interact directly or indirectly with each other and their environment. This causes interesting coherent and functionally global behaviour patterns to emerge. Swarm-based algorithms have been recently emerged as a group of nature-inspired, population-based algorithms that are capable to obtain optimal, robust, less expensive and fast solutions to several complex problems [2], [3]. The motivation often comes from nature, especially biological systems. The agents follow very easy rules, and have no centralized control structure to control agent's behaviour to be local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behaviour, not known to the individual agents. Some natural examples of SI include bird flocking, ant colonies, animal herding, bacterial growth, fish schooling, spiders and many more. The global behaviours of swarm emerge in a non-linear manner from the behaviour of the individuals in that swarm and their interactions plays a vital role in designing the swarm's behaviour. In particular, the discipline focuses on the collective behaviour that result from the local interactions of the individuals with each other and with their environment and influence the development of many swarm intelligence variants such as ant colony optimization (ACO), particle swarm optimization (PSO), bee colony optimization (BCO), bacterial foraging optimization (BFO), cat swarm optimization (CSO), etc. In the past few years swarm intelligence has been extensively used in a wide area of real world problems and applications to get optimally best results in minimum time. In this paper,

swarm intelligence and its variant ant colony optimization (ACO) and particle swarm optimization (PSO) has been explained briefly. Further, implementation of swarm intelligence variants ACO and PSO is shown using real world students' selection process.

In student selection process, Students are selected on merit and knowledge basis on the published criteria that varies according to different placement committees. Placement committees ensure that throughout the selection process and admission, applicants are treated with courtesy. Entry criteria and application procedures are too published on the Institution's website for the information of students and those seeking to work with the organization. Ant colony optimization (ACO) is based on ant's behaviour. ACO aims to search for an optimally best path in a given solution/graph, as ants seek for a best path between their colony and food source. ACO can easily solve wide range of numerical and complex problems. Particle swarm optimization (PSO) was initially intended to simulate social behaviour [4]. PSO is a computational method used for problem optimization by continuously improving a candidate solution with respect to a given quality measure.

The paper is organized as follows. Section II discusses about the classification of research in swarm intelligence on the basis of different criteria. Section III gives a brief idea about the self-organization, an important property of swarm intelligent systems. Section IV and V gives overview ACO and PSO respectively. In Section VI comparison between ACO and PSO is given. Section VII gives a brief idea about student selection process. Section VIII describes the proposed solution. Section IX, discusses results and section X discusses conclusion.

## 2. CLASSIFICATION OF SI RESEARCH

According to different criteria's swarm intelligence research can be classified as follows-

1. On the basis of nature of the system under consideration-
  - **Natural-** Under natural swarm intelligence research natural and biological systems are studied
  - **Artificial-** Under artificial swarm intelligence research human artifacts are studied.
2. On the basis of the goals to be achieved –
  - **Scientific-** Under scientific swarm intelligence research main aim is to model swarm intelligence systems and understand the processes that initiate a system to behave in a coordinated manner as a result of local individual-environment and individual-individual interactions.
  - **Engineering-** Under engineering swarm intelligence main aim is to use the information of scientific stream to design systems that are capable to solve practical relevance problems.

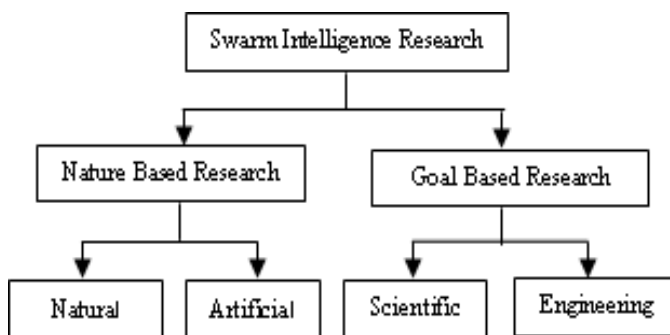


Fig -1: Classification of SI Research

## 3. SELF-ORGANIZATION

Self-organization plays a vital role in swarm intelligence as it allows interaction among swarm individuals. Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern [5]. Self-organization depends on activity amplification, facilitation and self-catalysis by positive feedback; activity balancing and stabilizing by negative feedback; amplification of random fluctuations and multiple interactions to transfer information between individual. Interaction between individual can occur in different ways such as by means of signals, by means of indices, by direct communication and by stigmergy, that is, modifying the environment to act together to solve complex problems.

Self-organization property has been applied to a wide range of areas, such as, computers, biology, chemistry, politics, etc. because of its adaptive and robust nature. Self-organization system identifies themselves to changing

demands and thus keeps on functioning despite of perturbations. Because of this, self-organization has been used as a paradigm to design adaptive and robust artificial systems [6].



Fig -2: Example of self-organization in biological systems- Birds flocking

## 4. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) designed to solve discrete optimization problems was introduced by M. Dorigo et al. [7], [8], [9]. It is a Multi-agent approach for solving complex combinatorial optimization problems to get a more optimal solution. ACO's iconic member is ANT System. It is observed naturally that ants are blind, small in size and still are able to find the shortest route to their food source. They make the use of antennas and pheromone liquid to interact with each other. ACO inspired from the social behaviour of live ants, are able to synchronize with searching solutions for local problems to maintaining previous information collected by each ant by maintaining array list.

### 4.1 Ant Behaviour

Ants, like many other social insects, communicate with each other using volatile chemical substances known as pheromones [10], which are direction and intensity can be perceived with their long, mobile antennae [11]. Ant behaviour is stochastic that probabilistically build a solution by iteratively adding solution components to partial solutions by indirect communication (pheromone paths) – Stigmergy [12] and heuristics information. Ants act concurrently and independently and help to develop high quality solutions through global cooperation.

### 4.2 ACO Algorithm

- **General Structure**  
**procedure** ACO meta-heuristic()  
**while** (stopping criterion)  
**schedule activities**  
     ants construct solutions using pheromone();

```

    update pheromone ();
    daemon actions(); /*OPTIONAL*/
end schedule activities
end while
return best solution generated;

```

- **Basic Flow:**
  1. Construction graph are used to represent the solution space
  2. Set ACO parameters and initialize pheromone trails
  3. Generate ant solutions from each ant’s walk through construction graph mediated by pheromone trails.
  4. Update pheromone intensities.
  5. Go to step 3, and repeat until termination conditions.

The basic flow of ACO shows that the main goal of ACO is to find the best quality paths on the problem’s construction graph by contingently moving through neighbour nodes.

### 5. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a heuristic optimization technique introduced by Kennedy and Eberhart [13], [14]. It was originally designed to solve non-linear continuous optimization problems, but latter practical and real-life application problems. PSO is a robust and stochastic optimization technique based on the swarms movement and intelligence. PSO inspired from the sociological behaviour of bird flocking can obtain optimal results. In PSO each particle moves to search for the optimum solution and hence has a *velocity*. PSO treats each particle as a point in a N-dimensional space which accordingly adjusts its “flying” based on experiences and each particle has a memory to store its previous best solutions (experiences). The particles in the swarm *co-operate* and exchange their experiences. In basic PSO co-operation is very easy and simple like this:

- Particles have neighborhoods associated with it.
- Particles know the fitness’s of neighboring particles, and use the best fitness position particle.
- This best fitness position is simply used to adjust the velocity of particles.

#### 5.1 PSO Algorithm

Basic PSO algorithm works on population of swarm where each individual is called a "particle", and can move in a multidimensional space that represents the solution space. PSO algorithm is of two types- Continuous PSO and Binary PSO.

Continuous PSO algorithm uses a real-valued multidimensional space, and determines the position of each particle in that space using the following equations [15]:

$$V_{id} = v_{id} + c_1 \text{rand}() (p_{id} - X_{id}) + c_2 \text{Rand}() (p_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

where,  $c_1$  and  $c_2$  are positive constants,  $\text{rand}()$  and  $\text{Rand}()$  are two random functions ranging from 0 to 1,  $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$

,  $x_{id}$ ) represents the  $i_{th}$  article,  $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$  represents the previous best fitness value position of the  $i_{th}$  particle,  $g$  represents best particle index of the population,  $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$  represents the rate of the position change for particle  $i$ .

Binary PSO represents particle position using binary number 0 or 1. Particle position in a dimension is randomly generated using logistic function distribution. Here velocity is used as a probability to determine whether  $x_{id}$  a particle bit will be in one state or another. They squashed vid using a logistic function  $s(v) = 1 / (1 + \exp(-v))$  [16] while the velocity is calculated using equation (1).

- **Basic Flow:**
  1. Initialize the swarm by randomly assigning initial velocity and a position to each particle in each dimension.
  2. Evaluate the desired fitness function for each particle’s position to be optimized.
  3. For each individual particle, update its historically best position so far;  $P_i$ , if its current position is better than its historically best one.
  4. Update the swarm’s globally best particle, and reset its index as  $g$  and its position at  $P_g$ .
  5. Update the velocities of particles using equation (1).
  6. Update each particle position using equation (2).
  7. Repeat steps 2–6 until stopping criterion is satisfied

The PSO helps to achieve a sufficiently good fitness solution and provides optimized results.

### 6. COMPARISON BETWEEN ACO AND PSO

Table -1: Comparison between ACO and PSO

| Criteria                          | ACO  | PSO  |
|-----------------------------------|--|--|
| <b>Communication Mechanism</b>    | ACO uses stigmergy an indirect communication mechanism.              | Communication among particles is rather direct without altering the environment. |
| <b>Problem Types</b>              | ACO was originally used to solve combinatorial optimization problems | PSO was originally used to solve continuous problems                             |
| <b>Modified Problem Type</b>      | Modified to adapt continuous problems.                               | Modified to adapt binary optimization problems.                                  |
| <b>Problem Representation</b>     | Construction graph represents solution space                         | Set of n-dimensional points represents solution space.                           |
| <b>Algorithm Application Area</b> | ACO is more applicable to  | PSO is more applicable to  |

|                       |  |   |
|-----------------------|--|---|
|                       | problems where source and destination are predefined and specific. | problems where previous and next particle positions at each point are clear and uniquely defined. |
| <b>Algorithm Goal</b> | ACO's goal is to search for an optimal path.                       | PSO's goal is to find the location of an optimal point.   |
| <b>Examples</b>       | Sequential ordering, scheduling, etc.                              | Track dynamic systems, evolve NN weights, etc.  |

### 7. STUDENT SELECTION PROCESS

Student selection process (SSP) is the process conducted within educational institutes or in a common place. This is an organized process to provide jobs to students pursuing or about to complete the programme. In this process, companies/industries visit number of colleges to select students depending on their knowledge, capability, focus and aim. Its main purpose is to identify qualified and talented student having working capability. Selection of students can be done through various procedures such as qualification criteria, group discussion, technical interview and many more. Students should satisfy minimum specific criteria before getting selected. Example-Qualification criteria include marks or grade range, specific program. Here, student selection process is implemented using swarm intelligence to get optimal selection result.

### 8. PROPOSED SOLUTION

Let us consider a real world student selection process (SSP). Here students are selected through placement committees of reputed organizations for implementing ACO and PSO. A criterion that is fitness value used is the knowledge power and marks of students in different fields.

In ACO, goal is to find the shortest path of traverse all the given colleges for best student's selection in minimum time. Given solution space is the collection of colleges to be traversed and placement committees represent agents/ant to implement the algorithm. Parameter that agents use for selection is the knowledge and marks of students of respective college. Number of students selected represents the pheromone trail. Different pattern of selection is formed after number of iteration. The pattern of selection with maximum number students within minimum time represents the shortest selection path, that is, optimal result in best minimum time.

**Algorithm 1:** ACO algorithm for SSP

```

procedure
  Set knowledge and marks of students ; /*
parameter*/
  initialize number of students; /* pheromone trails*/
  while (Max number of students ) do
    Construct collection of colleges; /*solution space*/
    
```

```

  Apply Local Search;
  Update number of students;
  end
end ACO algorithm for SSP
    
```

In PSO, goal is to find the colleges where best students are selected in minimum time. Given solution space is the collection of colleges to be traversed having students and placement committees represent agents/particles to implement the algorithm. Fitness value is the knowledge and marks of students. Position and velocity is updates if the present value of knowledge and marks is better than previous one. Thus, optimal pattern of student's selection is obtained.

**Algorithm 2:** PSO algorithm for SSP

```

procedure
  for each placement committee /* particles*/
    Initialize number of colleges; /* position*/
    Initialize number of students; /*velocity*/
  end
  do
    for each placement committee
      Determine knowledge and marks; /*fitness value
      if the fitness value is better than its personal best
      set current value as the new pBest;
    end
    for each particle
      Update particle velocity;
      Update particle position;
    end
  end
  While (maximum number of students and minimum
time)
    
```

### Implementation

Consider group of four colleges for selection process.

Database of each college is as given:

- First college named as A consists of 100 students

| Number of Students | Average Marks (%)       | Average Knowledge |
|--------------------|-------------------------|-------------------|
| 10                 | Greater and equal to 75 | O.S               |
| 50                 | 70 to 74.9              | O.S, D.S          |
| 40                 | Less than 70            | DAA               |

- First college named as B consists of 50 students

| Number of Students | Average Marks (%)       | Average Knowledge |
|--------------------|-------------------------|-------------------|
| 15                 | Greater and equal to 75 | D.S               |
| 15                 | 70 to 74.9              | O.S, DAA          |
| 20                 | Less than 70            | O.S, D.S          |

- First college named as C consists of 150 students

| Number of Students | Average Marks (%)       | Average Knowledge |
|--------------------|-------------------------|-------------------|
| 20                 | Greater and equal to 75 | O.S, DAA, D.S     |
| 80                 | 70 to 74.9              | O.S               |
| 50                 | Less than 70            | DAA, D.S          |

- First college named as D consists of 100 students

| Number of Students | Average Marks (%)       | Average Knowledge |
|--------------------|-------------------------|-------------------|
| 5                  | Greater and equal to 75 | DAA,D.S           |
| 40                 | 70 to 74.9              | D.S,O.S           |
| 55                 | Less than 70            | DAA               |

Distance between each college is A to B –10 km, A to C -15 km, A to D -5 km, B to C -10 km, B to D -10 km, C to D -20 km.

**Assumptions**

- Selection parameter of each placement committees/agents is assumed to be same as average marks greater than and equal to 75% and average knowledge of O.S, D.S and DAA.
- Distance between colleges is considered as execution time (denoted in units).
- Self-looping is not considered.

**Student Selection by ACO:**

In ACO, optimal selection path is to be searched, that is, path having shortest execution time and fulfilling all criteria's.

**Table -2:** Results after sample run on SSP for ACO

| R un | Iteration | No of Students | Average Knowledge | Path    | Execution time |
|------|-----------|----------------|-------------------|---------|----------------|
| 1    | 1         | 50             | O.S, D.S, DAA     | A-B-C-D | 40             |
|      | 2         | 50             | O.S, D.S, DAA     | A-C-D-B | 45             |
|      | 3         | 50             | O.S, D.S, DAA     | A-D-C-B | 35             |
| 2    | 1         | 50             | O.S, D.S, DAA     | B-A-C-D | 45             |
|      | 2         | 50             | O.S, D.S, DAA     | B-C-D-A | 35             |
|      | 3         | 50             | O.S, D.S, DAA     | B-D-A-C | 30             |
| 3    | 1         | 50             | O.S, D.S, DAA     | C-D-A-B | 35             |
|      | 2         | 50             | O.S, D.S, DAA     | C-A-B-D | 35             |
|      | 3         | 50             | O.S, D.S, DAA     | C-B-D-A | 25             |

|   |   |    |               |         |    |
|---|---|----|---------------|---------|----|
| 4 | 1 | 50 | O.S, D.S, DAA | D-A-B-C | 25 |
|   | 2 | 50 | O.S, D.S, DAA | D-B-C-A | 35 |
|   | 3 | 50 | O.S, D.S, DAA | D-C-A-B | 45 |

Thus, two optimal paths C-B-D-A and D-A-B-C are obtained with minimum execution time of 25 units.

**Student Selection by PSO:**

In PSO, colleges where best students are selected in minimum time are to be searched, that is, colleges having shortest execution time and fulfilling all criteria's.

**Table -3:** Results after sample run on SSP for PSO

| R un | Iteration | No of Students | Average Knowledge | Path                | Execution time |
|------|-----------|----------------|-------------------|---------------------|----------------|
| 1    | 1         | 45             | O.S, D.S, DAA     | A-B-C               | 25             |
|      | 2         | 30             | O.S, D.S, DAA     | A-C                 | 15             |
|      | 3         | 15             | O.S, D.S, DAA     | A-D                 | 5              |
| 2    | 1         | 45             | O.S, D.S, DAA     | B-A-C               | 25             |
|      | 2         | 35             | O.S, D.S, DAA     | B-C                 | 10             |
|      | 3         | 30             | O.S, D.S, DAA     | B-D-A               | 15             |
|      | 4         | 40             | O.S, D.S, DAA     | B-D-C               | 30             |
| 3    | 1         | 20             | O.S, D.S, DAA     | C-C(not considered) | -              |
|      | 4         | 15             | O.S, D.S, DAA     | D-A                 | 5              |
|      | 2         | 25             | O.S, D.S, DAA     | D-C                 | 20             |

Thus, optimal pattern in which colleges is to be traversed for student selection in minimum time are A-D or D-A with execution time of 5 units.

**9. RESULTS**

From the observations obtained in Section VII, it is very clear that optimal results have been obtained in minimum time. ACO determines the shortest path of traversing all the colleges in optimal time. PSO determines the colleges to be traversed for selection in minimum time. Final outcome of student selection process considered is represented in graphical form.

In figure 3, colleges are figured as the nodes in the graph and only those edges are shown which form shortest paths. Coloured lines are used to outline the final path found after

running the ACO algorithm. Coloured lines denoted two optimal shortest paths are selected. Final shortest paths of ACO are D-A-B-C and C-B-D-A obtained with execution time of 25 units.

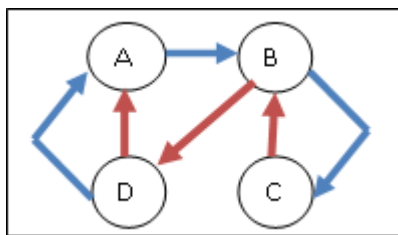


Fig -3 (a)

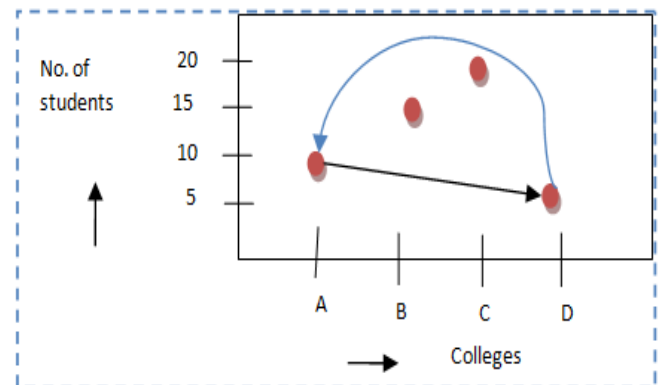


Fig -4(b)

Fig -4: Graphs representing final outcomes for PSO (a) connected graph (b) line graph

Table -4: Comparison between ACO-SSP and PSO-SSP

| Criteria                   | ACO-SSP  | PSO-SSP   |
|----------------------------|--|---|
| Algorithm Goal             | ACO's goal is to search optimal selection path.  | PSO's goal is to find the optimal colleges with best students.  |
| Velocity                   | Velocity is not considered.  | Each particle is associated with velocity that is updated according to fitness value.                                       |
| Problem Types              | ACO was originally used to solve combinatorial optimization problems                         | PSO was originally used to solve continuous problems  |
| Updating                   | Numbers of students are updated.   | Number of students and colleges traversed are updated   |
| Problem Representation     | Construction graph represents solution space   | Set of n-dimensional points represents solution space.  |
| Algorithm Application Area | ACO is more applicable to problems where source and destination are predefined and specific. | PSO is more applicable to problems where previous and next particle positions at each point are clear and uniquely defined. |

### 10. CONCLUSION

ACO and PSO both are the promising variants of swarm intelligence. In this paper, ACO algorithm for SSP and PSO algorithm for SSP has been proposed and implemented successfully. It is able to give an optimally best results in minimum time for the desired problem. The approach used for getting and representing best results is simple and easy to understand. Later it can be implemented using large sample of database. Here, Both ACO and PSO help to determine an optimized result in optimal time.

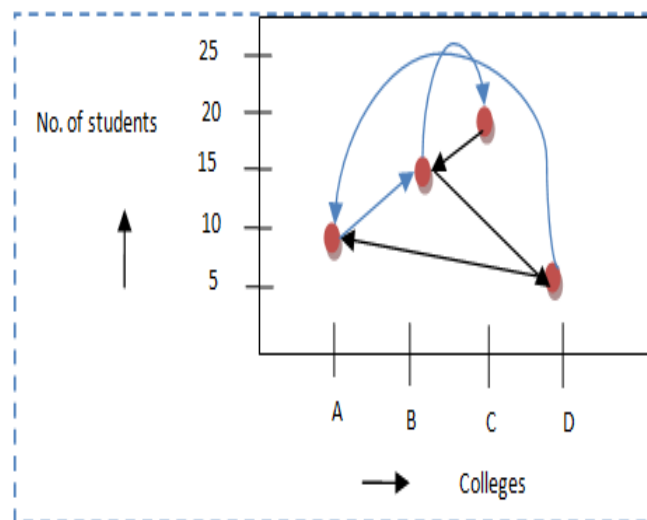


Fig -3 (b)

Fig - 3: Graphs representing final outcomes for ACO (a) connected graph (b) line graph

In figure 4, colleges are figured as the nodes in the graph and only those edges are shown which shows the pattern to traverse different colleges. Coloured lines are used to outline the final path found after running the PSO algorithm. Coloured lines denoted two optimal patterns are selected. Final patterns of PSO are D-A and A-D with execution time of 5 units.

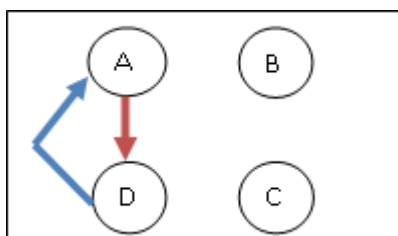


Fig -4 (a)

**REFERENCES**

- [1] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems," Oxford University Press, New York, NY, 1999.
- [2] C. Blum and D. Merkle (eds.), "Swarm Intelligence – Introduction and Applications. Natural Computing," Springer, Berlin, 2008.
- [3] M. Belal, J. Gaber, H. El-Sayed, and A. Almojel, "Swarm Intelligence, In Handbook of Bioinspired Algorithms and Applications," Series: CRC Computer & Information Science, Chapman & Hall Eds, ISBN 1-58488-477-5, vol. 7, 2006.
- [4] Kennedy, J., "The particle swarm: social adaptation of knowledge," Proceedings of IEEE International Conference on Evolutionary Computation. pp. 303–308, 1997.
- [5] Camazine, S., Deneubourg, J., Franks, N., Sneyd, J., Theraulaz, G., and Bonabeau, E., "Self-Organization in Biological Systems." Princeton University Press, 2000.
- [6] Gershenson C., " Design and Control of Self-organizing Systems," CopIt ArXives, Brussel, 2007.
- [7] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," Dipartimento di Elettronica, Politecnico di Milano, Italy, Tech. Rep.91-016, 1991.
- [8] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
- [9] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian. "Ant System for Job-shop Scheduling," Belgian Journal of Operations Research, Statistics and Computer Science, vol.34 (1), pp.39-53, 1994.
- [10] M. Dorigo, V. Maniezzo and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics-Part B, vol. 26(1), pp. 29-41, 1996
- [11] D. E. Jackson, F. L. "Ratnieks, Communication in ants," Current Biology, vol. 16, no. 15, pp. R570–R574, 2006.
- [12] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," Future Generation Computer Systems, vol. 16, no. 9, pp. 851–871, 2000.
- [13] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948, 1995
- [14] R. C. Eberhart and J. Kennedy, " A new optimizer using particle swarm theory," in Proc. of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43, 1995.
- [15] Kennedy, J., and Eberhart, R. C., "Particle swarm optimization," in Proc. of IEEE International Conference on Neural Networks (ICNN), vol.4, pp.1942-1948, 1995.
- [16] Kennedy, J., and Eberhart, R. C., "A discrete binary version of the particle swarm algorithm," in Proc. Conf. on Systems, Man, and Cybernetics, Piscataway, NJ: IEEE Service Center. p. 4104- 4109, 1997.