International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org || PP—01-11

A Fundamental study on Swarm Intelligence algorithms

Mishra Jyoti Prakash [1], Mishra Anil Kumar [2]
[1]*(Computer Science & Engineering, GIET, Baniatangi/ BPUT,Odisha India)*
[2]*(Computer Science & Engineering, GIET,Baniatangi/ BPUT, Odisha, India)*

ABSTRACT :*The objective of this paper is to critically survey the existing and current research paper on single and multi-objective particle swarm optimization (PSO& MOPSO) algorithm and Artificial Bee Colony Optimization algorithm(ABC). Bio –inspired solutions are often applied to solve optimization problems. Swarm intelligence is a relatively new sub field of Soft Computing which studies the emergent collective intelligence group of simple agents. It is based on social behavior that can be observed in nature, such as ant colonies, flocks of birds, fish schools and bee hives, where the number of individuals with limited capabilities are able to come to intelligent solutions for complex problems.*

KEYWORDS –*ABC, PSO, Swarm intelligence, Soft Computing, Bio-Inspired algorithms.*

I. INTRODUCTION

In recent years the swarm intelligence paradigm has received wide spread intention in research, mainly as Ant colony optimization (ACO), Particle swarm Optimization (PSO), Artificial Bee Colony Optimization (ABC)[1]. Nature is of course a great immense of source of inspiration for solving hard and complex problems in Computer Science since it exhibits extremely diverse, dynamic, robust, complex and fascinating phenomenon. It always finds the optimal solution to solve its problem maintaining perfect balance among its components. This is the thrust behind the bio-inspired computing [2].Artificial Bee Colony algorithm simulating the intelligent foraging behavior of honey bee swarms is one of the most popular swarm based optimization algorithm. It has been introduced in 2005 and applied in several fields to solve different problems.

Particle Swarm Optimization (PSO) is a heuristic search technique (which is considered as an evolutionary algorithm by its authors) that simulates the movement of flock of birds which aim to find food. The relative simplicity of PSO and the fact that is a population based technique. Particle Swarm Optimization is developed by Dr.Eberhart and Dr. Kennedy in 1995.

Artificial Bee Colony (ABC)algorithm, proposed by Karaboga in2005 for real parameter optimization, is a recently introduced optimization algorithm and simulates the foraging behavior of bee colony for unconstrained optimization problems. The basic concept of ABC algorithm is described in this paper.

II. SINGLE AND MULTI OBJECTIVE OPTIMIZATION

Optimization algorithms are becoming in increasing popular in engineering design activities because of the availability and affordability of high speed computers. They are extensively used in those engineering design problems where the emphasizing is on maximizing or minimizing of certain goal. For example Optimizing algorithms are normally used in aerospace design activities to minimize the overall weight, simply because every element component adds to the overall weight of the aircraft. Thus , minimization of the weight of aircraft components is of the major concern to aerospace designer. Chemical Engineers, on the other hand, are interested in designing and operating a process plant for an optimum rate of Production. Civil engineers are involved in designing buildings, bridges, dams, and other structures in order to achieve a minimum overall cost or maximum safety or both. Telecommunication engineers are interested in designing the communication networks for communication from one node to another. Computer Science Engineers applying the optimization algorithms for solving the problems in various domains such as Data mining, Image Processing, Database Engineering, Operating System scheduling, Networking, Software Engineering etc. All the above mentioned tasks involve either minimization or maximization (Collectively known as optimization) of an objective.

When an optimization problem modeling a physical System involves only one objective function, the task of finding the optimal solution is called single-objective optimization. We have already explained in the previous section of this chapter regarding the inherent strength of ABC technique and Hybridized ABC/PSO technique to solve optimization problems in Numerical optimization and in the field of Content based image retrieval.

A single objective optimization problem can be stated in the general form(1.1):

Minimize/Maximize	$p_m(x)$,m=1,2…,M
Subject to	
$q_j(x)\geq 0$	j=1,2…,J;
$r_k(x)=0$	k=1,2…K;
$x_i^{(L)}\leq x_i\leq x_i^{(U)}$	i=1,2…n;

A solution X is a vector of n decision variables=$(x_1,x_2,\ldots,x_n)^T$. The last set of constraints are called variable bounds, restricting each each decision variable x_i to take a value within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bounds. These bounds constitute a decision variable space D, or simply the decision space. Associated with the problem are J inequality and K equality constraints and the terms $q_j(x)$ and $r_k(x)$ are called constraint functions. Although the inequality constraints are treated as $\geq$ type ,the $\leq$ constraints can also be considered in the above formulation by converting those to $\geq$ types simply by multiplying each constraint function by -1.[16]

In single objective optimization, there is a single goal –the search of an optimal solution (i.e., global one). Although the search may have number of local optimal solutions, the goal is always to find the global optimal solution. However, the most single objective optimization algorithms aims at finding one optimum solution, even when there exists a number of optimal solutions. In single objective optimization algorithm, as long as a new solution has a better objective function value than an old solution, the new solution can also be accepted. In single objective optimization, there is only one search space i.e. the decision variable space. An algorithm works in this space by accepting and rejecting solutions based on their function values.

A multi objective optimization problem can be stated in the general form (1.2):

Minimize/Maximize	$p_m(x)$,m=1,2,…,M	
Subject to		
$q_j(x)\geq 0$	j=1,2…,J;	(1.2)
$r_k(x)=0$	k=1,2…K;	
$x_i^{(L)}\leq x_i\leq x_i^{(U)}$	i=1,2…n;	

In multi-objective optimization, the M objective function $p(x)=(p_1(x),p_2(x),\ldots,p_m(x))^T$ can be either minimized or maximized or both. Other vectors are same as single objective optimization problem, Many optimization algorithms are developed to solve only one type optimization problems, such as e.g. minimization problems. When an objective is required to maximized by using such an algorithm, the duality principle[14] can be used to transform the original objective for maximization into an objective for minimization by multiplying the objective function by -1.It is to be noted that for each solution x in the decision variable space, there exists a point in the objective space, denoted by $p(x)=z=(z_1,z_2,\ldots,z_m)$.There are two goals in a multi-objective optimization: firstly, to find a set of solutions as close as possible to pareto-optimal font; secondly, to find a set of solutions as diverse as possible. Multi-objective optimization involves two search spaces i.e., the decision variable space and the objective space. Although these two spaces are related by a unique mapping between them, often the mapping is non-linear and the properties of the two search spaces are not similar. In any optimization algorithm, the search is performed in the decision variable space. However, the proceedings of an algorithm in the decision variable space can be traced in the objective space. In some algorithms, the resulting proceedings in the objective space are used to steer the search in the decision variable space. When this happens, the proceedings in both spaces must be co-ordinated in such a way that creation of a new solutions in the decision variable space is complementary to the diversity needed in the objective space.

A multi objective optimization problem by using various classical methods such as weighted sum approach[12],ϵ-constraint method[7],weighted metric method[9],value-function method[15], and goal programming method[5].In the weighted sum approach multiple objectives are weighted and summed together to create a composite objective function. Optimization of the composite objective results in the optimization of the individual objective functions. The outcome of such an optimization strategy depends on the chosen weights.

The ϵ-constraint method chooses optimizing one of the objective functions and treats rest of the objectives as constraints by limiting each of them within certain predefined limits. This fix up converts the multi-objective optimization problem into a single objective optimization problem. Here too the outcome of the single-objective constrained optimization results in solution which depends on the chosen constraint values. Weighted metric method suggests minimizing and L_p-metric constructed from all objectives. The value function method suggests maximizing an overall value function (or utility function) relating all objectives. Goal programming method suggest minimizing a weighted sum of deviations of objectives from user specified targets. These conversion methods result in single objective optimization problem, which must be solved by using a single objective algorithm. These classical multi objective optimization algorithms are having some difficulties particularly if the user wants to find multiple pareto-optimal solutions. First, only one pareto optimal solutions can be expected to be found in one simulation run of a classical algorithm. Second not all pareto optimal solutions can be found by some algorithms in non-convex MOPs. Third all the algorithms require some problem knowledge, such as suitable weights or ϵ or target values. Multi-objective optimization for finding multiple pareto-optimal solutions eliminates all fix-ups and can in principle find a set of optimal solutions corresponding to different weights and ϵ vectors. Although only one solution is needed for implementation, the knowledge of such multiple optimal solutions may help a designer to compare and choose a compromised optimal solution. A multi-objective optimization is ,in general, more complex than a single objective optimization, but the avoidance of multiple solution runs, no artificial fix-ups, availability of efficient population-based optimization algorithms, and above all, the concept of dominance help to overcome some difficulties and give the user the practical means to handle multiple objectives.

Most multi objective optimization algorithms use the concept of domination. In these algorithms, two solutions are compared on the basis of whether one dominates the other solution or not. The concept of domination is described in the following definitions (assuming, without loss of generality, the objective function is to be minimized).

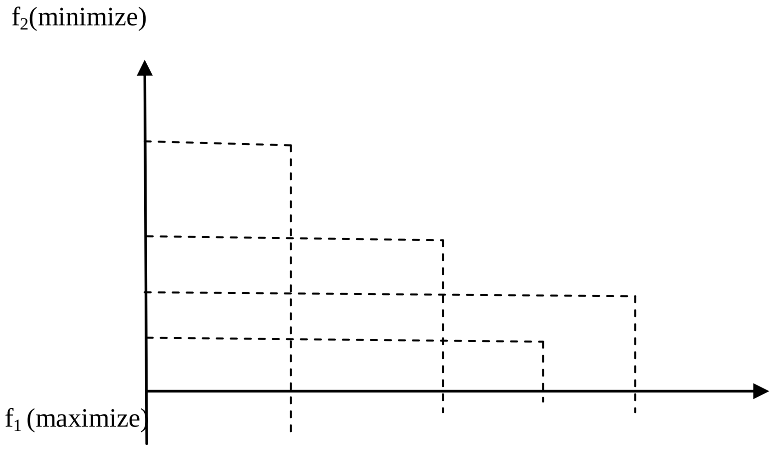
Definition 1. A solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$ if both the condition 1 and 2 are true:

1.The solution $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives or $f_j(x^{(1)})$ is not worse than $f_j(x^{(2)})$ for all j=1,2,…,M.

2.The solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective.

If any of the above condition is violated, the solution $x^{(1)}$ does not dominate the solution $x^{(2)}$.If $x^{(1)}$ dominates the solution $x^{(2)}$(or mathematically $x^{(1)} \preccurlyeq x^{(2)}$), it is also customary to write any of the following

- $X^{(2)}$ is dominated by $x^{(1)}$;
- $X^{(1)}$ is non-dominated by $x^{(2)}$, or;
- $X^{(1)}$ is non-inferior to $x^{(2)}$.



[Fig-1]

The figure illustrates a two objective optimization problem with five different solutions shown in the objective space. Let us assume that the objective function 1 needs to be maximized and objective function2 needs to be minimized. Five solutions with different objective function values shown in this figure. Since both the objective functions are importance to us ,it is difficult to find one solution which is best with respect to both the objectives. However, we can use the above definition of domination to decide which solution is better among any two given solutions in terms of both objectives .For example, if solutions 1 and 2 are to be compared, we observe that solution 1 is better than solution 2 in objective function1 and solution 1 is better than solution 2 in objective function 2.Thus both of the above conditions for domination are satisfied and we may write that solution 1 is better than solution 2.We take another instance of comparing solution 1 and 5.Here, solution 5 is better than solution 1 in the first objective and solution 5 is no worse than solution 1 in the second objective. Thus, both the above conditions for domination are satisfied and we may write that solution 5 dominates solution 1.

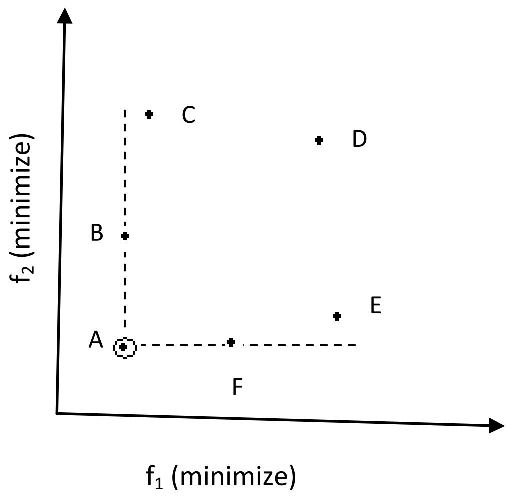
It is intuitive that if a solution $x^{(1)}$ another solution $x^{(2)}$, the solution $x^{(1)}$ is better than $x^{(2)}$, the solution $x^{(1)}$ is better than $x^{(2)}$ in the parlance of multi-objective optimization. Since, the concept of domination allows a way to compare solutions with multiple objectives; most multi-objective optimization methods use this concept to search for non dominated solutions.

Most multi-objective optimization algorithms use the concept of domination. In these algorithms, two solutions are compared on the basis of whether one dominates the other solution or not. The concept of domination is described in the following definitions (assuming, without loss of generality, the objective functions to be minimized).

Definition 2. A solution *x* strongly dominates a solution *y* (denoted by $x \prec y$), if solution *x* is strictly better than solution *y* in all M objectives.

[The dominance relationship between the two solutions defined is sometimes referred to as weak dominance relation. The definition is modified and a strong dominance relation is derived]

Figure 1 illustrates a particular case of the dominance relation in the presence of two objective functions.



(Solution A weakly dominates solutions B and F and it strongly dominates solutions C, D and E)

Fig. 2. Dominance relation in a two- objective space

However, if a solution *x* strongly dominates a solution *y*, the solution *x* also weakly dominates solution *y*, but not vice versa.

Definition 3. The decision vector $x \in P$ (where P is the set of solution or decision vectors) is non-dominated with respect to set P, if there does not exit another $x^t \in P$ such that $f(x^t) \preccurlyeq f(x)$

Definition 4. Among a set of solution or decision vectors P, the non-dominated set of solution or decision vectors P′ are those that are not dominated by any member of the set P.

Definition 5. A decision variable vector $x \in P$ where P is the entire feasible region or simply the search space, is Pareto-Optimal if it is non-dominated with respect to P.

Definition 6. When the set P is the entire search space, the resulting non-dominated set P′ is called the Pareto-Optimal set. In other words, $P' = \{x \in P \mid xisPareto - Optimal\}$

The non-dominated set P′ of the entire feasible search space P is the globally

Pareto-Optimal set.

Definition 7. All Pareto-Optimal solutions in a search space can be joined with a curve (in two-objective space) or with a surface (in more than two-objective space). This curve or surface is termed as Pareto optimal front or simply Pareto front. In other words, $PF = \{f(x) \mid x \in P'\}$

Figure 2 illustrates a particular case of the Pareto front in the presence of two objective functions

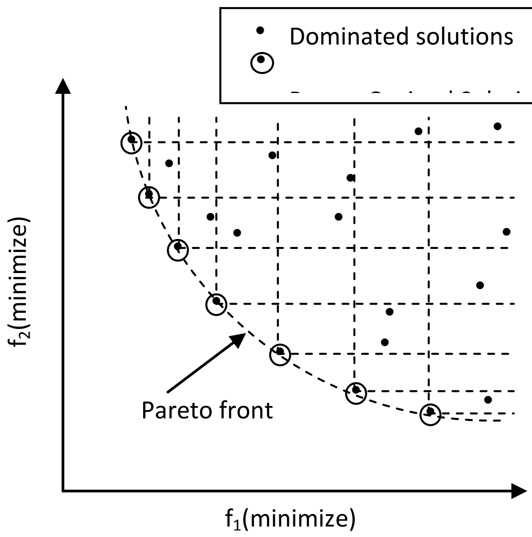


Fig. 3. The Pareto front in a two

We thus wish to determine the Pareto optimal set form the set P of all the feasible decision variable vectors that satisfy (6) to (8). It is to be noted that in practice, the complete Pareto Optimal set is not normally desirable (e.g. it may not be desirable to have different solutions that map to the same values in objective function space) or achievable. Thus a preferred set of Pareto optimal solutions should be obtained from practical point of view.

III. DIFFERENT SWARM INTELLIGENCE TECHNIQUES

To solve the single and multi objective optimization problems has been many Swarm Intelligence methodologies proposed in the literature.And different Swarm Intelligence techniques has been described in Table Number.1. One among such technique is Artificial Bee Colony Optimization proposed by Dervis Karaboga . We Provide the basic philosophy of Artificial Bee Colony Optimization (ABC),Particle Swarm optimization(PSO) and the hybridization of the ABC and PSO for single objective optimization and Artificial Bee Colony Optimization for Multi-Objective optimization .

Table 1
Different Swarm Intelligence Algorithms

SL No.	Name of algorithm	Year of development	Based on Technique
1	Altruism	Foster KR,Wenseleers T,Ratnicks L W 2006	Hamilton's rule of kin selection
2	Ant colony optimization	Marco Dorigo 1992	Ant
3	Artificial Bee Colony	Karaboga 2005	Honey Bee
4	Artificial Immune System	De castro & Von Zuben's and Nicosia and Cuttello' 2002	Abstract structure and function of Immune System
5	Charged System Search(CSS)	Kaveh A. & Taltahari S. 2010	Based on some principles from Physics and mechanics
6	Cuckoo Search(CS)	Yang Xin-She & Deb Suash	Mimic some Breeding behavior

		2009	of some Cuckoo species
7	Firefly algorithm(FA)	Yang Xin-She 2008	Inspired by flashing behavior of fireflies
8	Gravitational search algorithm(GSA)	Rashedi,Nezamabadipour & Saryazdi	Based on law of gravity and the motion of mass interaction
9	Intelligent water drops(IWD)	Shah-Hosseinni Hamed 2009	Inspired by natural rivers and how they find almost optimal paths to their destination
10	Particle swarm optimization(PSO)	Kennedy & Eberhart(1995)	Inspired by swarms
11	River formation dynamics(RFD)	Gradient version of ACO	Based on the Principle How water forms rivers

Particle swarm Optimization

PSO is an evolutionary computation technique developed by Eberhart and Kennedy[2] in 1995.It was originally proposed for single objective optimization problems. This was inspired by the social behavior of bird flocking and fish schooling. PSO has its root in artificial life and social psychology, as well as in engineering and computer science. It utilizes a "population" of particles that fly through the problem hyperspace with given velocity. At each iteration, the velocities of individual particles are stochastically adjusted according to their historical best position for the particle itself and the neighborhood best are derived according to the user defined fitness function. The movement of each particle naturally evolves to an optimal or near-optimal solution. The word "swarm" comes from the irregular movements of the particle in the problem space, now more similar to a swarm of mosquitoes rather than a flock of birds or a school of fish[3]. PSO is a computational intelligence-based technique that is not largely affected by the size and nonlinearity of the problem, and can coverage to the optimal solution in the problem, and can coverage to the optimal solution in many problems where most analytical methods fail to converge. It can therefore, be effectively applied to different optimization problems in power systems, Data Mining, image processing, Numerical optimization. A number papers have been published in the past few years that focus on many issues. Moreover, PSO has some advantages over other similar optimization techniques such as GA, namely the following.

1. PSO is easier to implement and there are fewer parameters to adjust.
2. In PSO, every particle remembers its own previous best value as well as the neighborhood best; therefore it has a more effective memory capability than the GA.
3. PSO is more efficient in maintaining the diversity of the swarm(more similar to the ideal social interaction in a community), since all particles use the information related to the most successful particle in order to improve themselves, whereas GA, the worse solution are discarded and only good ones are saved; therefore in GA the population evolves around a subset of best individuals.

Basic Concepts and Formulation

PSO is based on two fundamental disciplines: social science and computer science. In addition, PSO uses the swarm intelligence concept, which is the property of a system, whereby the collective behaviors of unsophisticated agents that are interacting locally with their environment create global functional patterns. In addition, PSO uses the swarm intelligence concept, which is a property of a system, whereby the collective behaviors of unsophisticated agents that are interacting locally with their environment create coherent global functional patterns.

1. Social Concepts: it is known that "human intelligence results from social interaction". Evaluation, comparison, and limitation of others, as well as learning from experience allow humans to adapt to the environment and determine optimal patterns of behavior, attitudes and such like. In addition, a second fundamental social concept indicates that " culture and cognitions are in separable consequences of human sociability." Culture is generated when individual become more similar due to social learning. The sweep of culture allows individuals towards more adaptive patterns of behavior.
2. Swarm Intelligence principle: Swarm Intelligence can be described by considering five fundamental principles.
   a. Proximity Principle: the population should be able to carry out simple space and time computations.
   b. Quality Principle: The population should be able to respond to quality factors in the environment.
   c. Diverse Response principle: The population should not commit its activity along excessively narrow channels.
   d. Stability Principle: The Population should not change its mode of behavior every time the environment changes.

e. Adaptability Principle: The population should be able to change its mode of behavior when it is worth the computational price.

In PSO, the term "particles" refers to population members which are mass-less volume-less (or with arbitrarily small mass or volume) and are subject to velocities and acceleration towards a better mode of behavior.

3. Computational Characteristics: Swarm Intelligence provides a useful paradigm for implementing adaptive systems. It is an extension of evolutionary computation and includes the softening parameterization of logical operators like AND, OR and NOT. In particular PSO is an extension, and a potentially important incarnation of cellular automata (CA). The particle swarm can be conceptualize as cells in CA, whose state change in many dimensions simultaneously. Both PSO and CA share the following computational attributes.

i. Individual particles are updated in parallel.
ii. Each value depends only on the previous value of the particle (cell) and its neighbors.
iii. All updates are performed according to the same rule.

As a basic principle, in PSO, a set of randomly generated particles in the initial swarm are flown (have their parameters adjusted) through the hyper-dimensional search space (problem space) according to their previous flying experience. Changes to the position of the particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals. Each particle represents a potential solution to the problem being solved. The position of a particle is determined by the solution it currently represents. The position of each particle is changed according to its own experience and that of its neighbors. These particles propagate towards the optimal solution over a number of generations (moves) based on large amount of information about the problem space that is assimilated and shared by all members of the swarm. PSO algorithm finds the global best solution by simply adjusting the trajectory of each individual toward its own best location (*pbest*) and towards the best particle of the entire swarm (*gbest*) at each time step (generation). In this algorithm, the trajectory of each individual in the search space is adjusted by dynamically altering the velocity of each particle according to its own flying experience and the flying experience of the other particles in the search space.

Each of these studies implements MOPSO in a different fashion. However, the PSO heuristic puts a number of constraints on MOPSO. In PSO itself the swarm population is fixed in size, and its members can not be replaced, only adjusted by their *pbest* and *gbest*, which are by themselves easy to define. However, in order to facilitate an multi-objective approach to PSO, a set of non-dominated solutions (the best individuals found so far using the search process) must replace the single global best individual in the standard single-objective PSO case. Besides, there may be no single previous best individual for each member of the swarm. Choosing which *gbest* and *pbest* to direct a swarm member's flight is therefore important in MOPSO. Main focus of various MOPSO algorithms is how to select *gbest* and *pbest* with a separate divergence on whether an elite archive is maintained.

In order to apply the PSO strategy for solving multi-objective optimization problems, the original scheme has to be modified. The algorithm needs to search a set of different solutions (the so-called Pareto front) instead of a single solution (as in single objective optimization). We need to apply Multi-Objective Particle Swarm Optimization (MOPSO) to search towards the true Pareto front (non-dominated solutions). Unlike the single objective particle swarm optimization, the algorithm must have a solution pool to store non-dominated solutions found by searching upto stopping criterion (say, upto iteration I_{max}). Any of the solutions in the pool can be used as the global best (*gbest*) particle to guide other particles in the swarm during the iterated process. The plot of the objective functions whose non-dominated solutions are in the solution pool would make up for the Pareto front. The pseudo code for a general MOPSO is illustrated in Algorithm 2.

Algorithm 2 General Multi-Objective Particle Swarm Optimization Algorithm

```
01. Begin
02.         Parameter Settings and initialize Swarm
03.         Evaluate Fitness and initialize leaders in a leader pool or external archive
04.         Archive the top best leader from the external archive through evaluation of      some sort of
    quality measure for all leaders.
```

```
05.        I = 0                /* I = Iteration count */
06.        While (the stopping criterion is not met, say, $I < I_{max}$)
07.       Do
08.           For each particle
09.                  Select leader in the external archive
10.                  Update velocity
11.                  Update position
12.                  Mutate periodically              /* optional */
13.                  Evaluate Fitness
14.                  Update pbest
15.           End for
16.           Crowding of the leaders
17.           Update the top best into external archive
18.        I++
19.        End While
20.        Report results in the external archive
21.  End
```

In the above general MOPSO algorithm (Algorithm 1), first the swarm is initialized. Then, a set of leaders is also initialized with the non-dominated particles from the swarm. These set of leaders are stored in an external archive. Later on, some sort of quality measure is calculated for all the leaders in order to select usually one leader for each particle of the swarm. At each generation, for each particle, a leader is selected and the flight is performed. Most of the existing MOPSOs apply some sort of mutation operator after performing the flight. Then, the particle is evaluated and its corresponding *pbest* is updated. A new particle replaces its *pbest* particle usually when this particle is dominated or if both are incomparable (i.e. they are both non-dominated with respect to each other). After all the particles have been updated, the set of leaders is updated, too. Finally, the quality measure of the set of leaders is re-calculated. This process is repeated for a certain fixed number of iterations.

In the case of multi-objective optimization problems, each particle might have a set of different leaders from which just one can be selected in order to update its position. Such set of leaders is usually stored in a different place from the swarm, that is called external archive. This is a repository in which the non-dominated solutions formed so far are stored. The solutions contained in the external archive are used as leaders when the positions of the particles of the swarm have to be updated. Furthermore, the contents of the external archive is also usually reported as the final output of the algorithm.

Artificial Bee Colony Optimization

Background[The Honey Bee Danc Language NC State University]

Honey Bee dancing , perhaps the most intriguing aspect of their biology, is one of the most fascinating behaviors in animal life. Performed by a worker bee that has returned to the honey comb with pollen or nectar, the dances , in essence , contribute a language that “tells” other workers where the food is. By signaling both distance and direction with particular movements , the worker bee uses the dance language to recruit and direct other workers in gathering pollen and nectar.

The late Karl von Frish, a professor of zoology at the university of Munich in Germany, is Credited with interpreting the meaning of honey bee dance movement. He and his students carried out decades of research in which they have carefully described the different components of each dance. Their experiments typically used glass-walled observation hives and paint marked bee forgers. First they trained the forgers to find food sources placed at known distances from colony. When bees returned from gathering from those sources ,von Frisch and his students carefully measured both the duration and angle of dances the forgers performed to recruit other bees to help gather food. Their findings led them to the concept of dance language .

As Swarm intelligence has become a research interest to many research scientists of the related fields in the recent years. The Swarm intelligence is defined as “… any attempt to design algorithms or distributed problem solving devises inspired by the collective behavior of social insect colonies and other animal socities…” by Bonabeau et al[15].Bonabeau et al. emphasize their view point on social insects alone, such as termites, bees, wasps as well as different ant species. However, the term swarm is used in a general manner to refer any restrained collection of interacting agents or individuals. The classical example of a swarm is bees swarming around their hive. The Karl von Frish led the foundation for Bees to come under the study under swarm intelligence. For example an ant colony can be thought of as a swarm whose individual agents are ants; a flock of birds is a swarm of birds; an immune system is a swarm of cells as well as a crowd is a swarm of people.

Artificial Bee Colony (ABC) algorithm, proposed by Karaboga in 2005 for real parameter optimization, is an optimization algorithm and simulates the foraging behavior of bee colony for unconstrained optimization problem. For solving constrained optimization problems, a constraint method was incorporated with the algorithm. In a real bee colony, there are some tasks perform by specialized individuals .These specialized bees try to maximizing the nectar amount stored in the hive by performing efficient division of labour and self organization. The minimal model of swarm-intelligent forge selection in a honey bee colony, that ABC algorithm adopts, consists of three kinds of bees: employed bees, on looker bees and scout bees. Half of the colony comprises employed bees and other half includes the on looker bees. Employed bees are responsible from exploiting the nectar sources explored before and giving information to other waiting bees (on looker bees) in the hive about the quality of food source site which they are exploiting. Onlooker bees wait in the hive and decide a food source to exploit depending on the information shared by the employed bees. Scouts randomly search the environment in order to find a new food source depending on the internal motivation or possible external clues randomly.

Behavior of real bees
The minimal model of forge selection that lead to the emergence of collective intelligence of honey bee swarms consists of three essential components: food source, employed forgers, unemployed forgers, and defines two leading modes of behavior: recruitment to the nectar source and abandonment of a source .

(i)Food Source: the value of food source depends on many factors, such as proximity to the nest, richness or concentration of energy and ease of extracting this energy. For the simplicity, the profitability of the food source can be represented with a quantity.

(ii) Employed foragers: they are associated with a particular food source which they are currently exploiting or the “employed” at. They carry with them information about this particular food source , its distance and direction from the nest and profitability of the source and share the information with certain probability.

(iii) Unemployed forgers: they are looking for a food source to exploit. There are two type of unemployed forgers – scouts searching the environment surrounding the nest for new food source and onlookers waiting in the nest and finding a food source through the information shared by employed forgers. The mean number of scouts averaged about conditions is about 5-10%.

The exchange of information among bees is the most important occurrence in the information collective knowledge. While examining the entire hive, it is possible to distinguish some parts that are commonly exist in all hives. The most important part of the hive with respect to exchanging information is the dancing area. Communication among bees related to the quality of food sources occurs in the dancing area. The related dance is called as waggle dance. Since information about all the current rich sources is available to the onlooker on the dance floor, she probably could watch numerous dances and choose to employ herself at the most profitable source. Employed foragers share their information with a probability ,which is proportional to the profitability of the food source, and sharing of this information through a waggle dancing is longer in duration. Hence, the recruitment is proportional to profitability of a food source.

In order to understand the basic behavior characteristics of foragers better, let’s examine the figure 4.Assume that there are two discovered food sources: A and B. At very beginning, a potential forager will start as unemployed forager. That bee will have no knowledge about the food sources around the nest. There are two possible options for such bee:

(a) It can be a scout and start searching around the nest spontaneously for a food due to some internal motivation or possible external clue(‘S’ in figure.4).
(b) It can be recruit after watching the waggle dances and starts searching for a food source(‘R’ in figure.4)

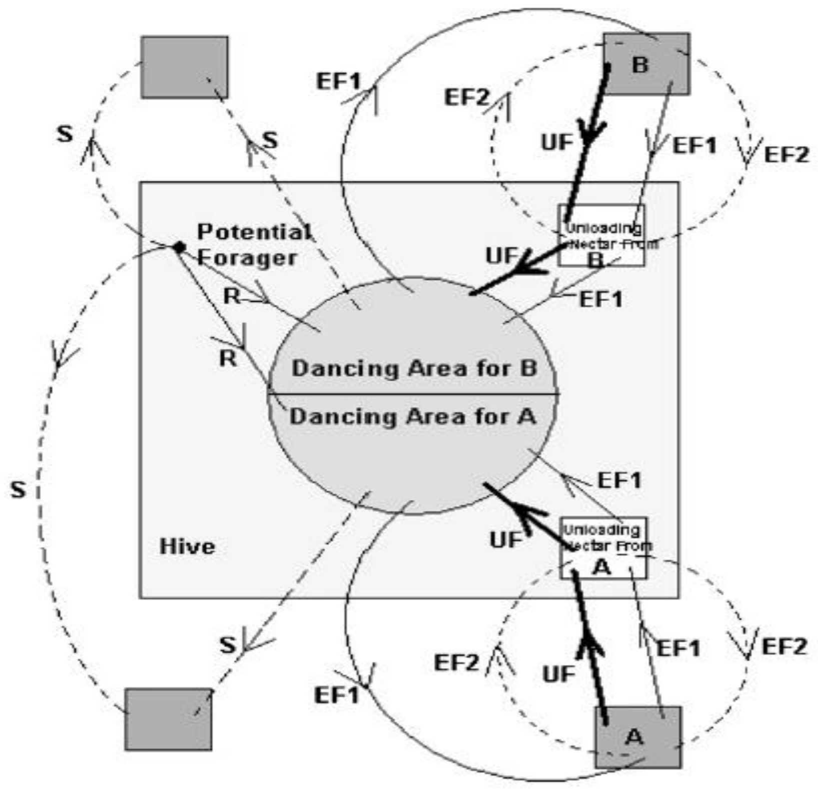


Figure-4.Behavior of honeybee foraging for nectar

ABC algorithm
1. Initialize the food source positions.
2. Each employed bee produces a new food source in her food source site and exploits the better source.
3. Each onlooker bee selects a source depending on the quality of her solution, produces new food source in the selected food source site and exploits the better source.
4. Determine the source to be abandoned and allocate its employed bee as scout for searching new food sources.
5. Memorize the best food source found so far.
6.Repeat steps 2-5 until the stopping criterion is met.

In the first step of the algorithm, x_i(i=1,….,SN) solutions are randomly produced in the range of parameters where SN is the number of food sources. In the second step of the algorithm, for each employed bee, whose total number equals to the half of the number of the food sources, a new source is produced by (1)

$v_{ij}= x_{ij}+ø_{ij}(x_{ij}-x_{kj})$ ---------------- (1)

where $ø_{ij}$ is a uniformly distributed real random number within the range [-1,1]
k is the index of the solution chosen randomly from the colony (k=int(rand*SN)+1)
j=1,…….,D and D is the dimension of the problem
After producing v_i, this new solution is compared to x_i and the employed bee exploits the better source. In the third step of the algorithm, an on looker bee chooses a food source with probability (2) and produces a new source in the selected food source site by (1). As for employed bees, the better source is decided to be exploited.

$p_i =\frac{fit_i}{\sum_{j=1}^{SN} fit_j}$ --------- (2)

where fit_i is the fitness of the solution x_i.
After all onlookers are distributed to the sources, sources are checked whether they are to be abandoned. If a number of cycles that a source cannot be improved is greater than the predefined limit, the source is considered to be exhausted. The employed bees associated with the exhausted source become a scout and makes a random search in the problem domain by (3)

$x_{ij}=x_j^{min}+(x_j^{max}-x_j^{min})*rand$ --------- (3)

IV. CONCLUSION

Literature has proved the individual performance of ABC and PSO while solving various optimization problems. However, as PSO searches the solution by updating the particles and the ABC searches by bees' wandering behavior, there are drawbacks persist in the individual performance. This paper provides a brief review on PSO and ABC. In recent part, non-traditional population-based stochastic search and optimization methods, inspired by natural evolution have shown capabilities of approximating optimal solutions to complex real-world problems within reasonable computational time. In this regard, the Evolutionary Computation (EC) and Swarm Intelligence (SI) techniques have been receiving increasing attention in view of their potential as global optimizers.

REFERENCES

[1] Jorge A. Ruiz-Vanoye, OcotlánDíaz-Parra, Felipe Cocón, Andrés Soto, "Meta-Heuristics Algorithms based on the Grouping of Animals by Social Behavior for the Traveling Salesman Problem", International Journal of Combinatorial Optimization Problems and Informatics, Vol. 3, No. 3, Sep.-Dec. 2012.

[2] Shivakumar B L, Amudha T, "A Novel Nature-inspired Algorithm to solve Complex Generalized Assignment Problems", International Journal of Research and Innovation in Computer Engineering , Vol 2, No. 3, pp. 280-284, June 2012.

[3] Yamille del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez, and Ronald G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", IEEE Transactions on Evolutionary Computation, Vol. 12, No. 2, pp. 171-195, April 2008.

[4] R. Umarani, V. Selvi, "Particle Swarm Optimization-Evolution, Overview and Applications", International Journal of Engineering Science and Technology, Vol. 2, No.7, pp. 2802-2806, 2010.

[5] SujathaBalaraman, N. kamaraj, "Application of Differential Evolution for Congestion Management in Power System", Modern Applied Science, Vol. 4, No. 8, pp. 33-42, August 2010.

[6] Rhythm S. Wadhwa, Terje Lien, "Electromagnet Shape Optimization using Improved Discrete Particle Swarm Optimization", Excerpt from the Proceedings of COMSOL Conference in Stuttgart, pp. 1-6, 2011

[7] G. K. Venayagamoorthy, and R. G. Harley, "Swarm Intelligence for Transmission System Control", IEEE Power Engineering Society General Meeting, pp. 1-4, 2007.

[8] NadezdaStanarevic, Milan Tuba, and NebojsaBacanin, "Modified artificial bee colony algorithm for constrained problems optimization", International Journal of Mathematical Models and Methods In Applied Sciences, Vol, 5, No. 3, pp. 641-655, 2011.

[9] Mustafa ServetKiran and MesutGunduz, "A Novel Artificial Bee colony based algorithm for solving the number optimization problem", International Journal of Innovative Computing, Information and Control, Vol. 8, No. 9, Sep. 2012.

[10] Arun Kumar, Rajeshwar Singh, "Mobile Ad Hoc Networks Routing Optimization Techniques Using Swarm Intelligence", International Journal of Research in IT & Management, Vol. 1, No. 4, August, 2011.

[11] Milos Subotic, Milan Tuba and NadezdaStanarevic, "Different approaches in parallelization of the artificial bee colony algorithm", International Journal of Mathematical Models And Methods in Applied Sciences, Vol. 5, No. 4, pp. 755-762, 2011.

[12] O.A. Mohamed Jafar and R. Sivakumar, "Ant-based Clustering Algorithms: A Brief Survey", International Journal of Computer Theory and Engineering, Vol. 2, No. 5, 1793-8201, October, 2010

[13] S.M. ELseuofi, "Quality Of Service Using Pso Algorithm", International Journal of Computer Science & Information Technology, Vol 4, No 1, pp. 165-175, Feb 2012.

[14] V. Selvi, R.Umarani, "Comparative Study of Swarm Intelligence Techniques", International Journal of Research in Engineering Design, Vol 01, No. 01, pp. 37-41, April - July 2012.

[15] PinfaBoonrong, BoonsermKaewkamnerdpong, "Canonical PSO based Nanorobot Control for Blood Vessel Repair", World Academy of Science, Engineering and Technology, Vol. 58, 2011.

[16] BhartiSuri, Snehlata, "Review of Artificial Bee Colony Algorithm to Software Testing", International Journal of Research and Reviews in Computer Science, Vol. 2, No. 3, pp. 706-711, June 2011.