Bee Colony Algorithm Articles summaries

William Bezuidenhout

Tuesday, 11 June 2010

1 A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem

```
@article{Pan2010,
title = "A discrete artificial bee colony algorithm for the lot-streaming flow sho
journal = "Information Sciences",
volume = "In Press, Corrected Proof",
number = "",
pages = " - ",
year = "2010",
note = "",
issn = "0020-0255",
doi = "DOI: 10.1016/j.ins.2009.12.025",
url = "http://www.sciencedirect.com/science/article/B6V0C-4Y34SV5-1/2/202d791c7dd2
author = "Quan-Ke Pan and M. Fatih Tasgetiren and P.N. Suganthan and T.J. Chua",
keywords = "Flow shop scheduling",
keywords = "Lot-streaming",
keywords = "Artificial bee colony algorithm",
keywords = "Weighted earliness and tardiness criterion"
}
```

1.1 Introduction

In general, swarm intelligence is based on collective behavior of self-organized systems. As a typical example of swarm intelligence, the bee swarming around her hive has received significant interest from researchers. Recently, by modeling the specific intelligent behaviors of honey bee swarms, an artificial bee colony (ABC) algorithm is developed by Karaboga to optimize

multi-variable and multi-modal continuous functions. Numerical comparisons demonstrated that the performance of the ABC algorithm is competitive to other population-based algorithm with an advantage of employing fewer control parameters. Due to its simplicity and ease of implementation, the ABC algorithm has captured much attention and has been applied to solve many practical optimization problems. Since there is no published work to deal with the lot-streaming flow shop problem by using the ABC algorithm, we present a novel discrete ABC (DABC) algorithm for solving the lot-streaming flow shop problem with a criterion of total weighted earliness and tardiness penalties

1.2 The basic bee colony algorithm

The artificial bee colony (ABC) algorithm is a new swarm intelligence based optimizer proposed by Karaboga for multi-variable and multi-modal continuous function optimization. Inspired by the intelligent foraging behavior of honeybee swarm, the ABC algorithm classifies the foraging artificial bees into three groups, namely, employed bees, onlookers and scouts. A bee that is currently exploiting a food source is called an employed bee. A bee waiting in the hive for making decision to choose a food source is named as an onlooker. A bee carrying out a random search for a new food source is called a scout. In the ABC algorithm, each solution to the problem under consideration is called a food source and represented by an n-dimension real-valued vector, whereas the fitness of the solution is corresponded to the nectar amount of the associated food resource. Similar to the other swarm intelligence based approaches, the ABC algorithm is an iterative process. It starts with a population of randomly generated solutions or food sources. Then the following three steps are repeated until a termination criterion is met:

1. Send the employed bees onto the food sources and then measure their nectar amounts.

2. Select the food sources by the onlookers after share the information of employed bees and determine the nectar amount of the food sources.

3. Determine the scout bees and send them onto possible food sources.

1.2.1 Initialization of Parameters

The parameters of the basic ABC algorithm are the number of food sources (SN) which is equal to the number of the employed bees or on looker bees,

the number of trials after which a food source is assumed to be abandoned (limit), and a termination criterion. In the basic ABC algorithm, the number of employed bees or the onlookers is set equal to the number of food sources in the population. In other words, for every food source, there is only one employed bee.

1.2.2 Initialization of Population

The initial population of solutions is filled with SN number of randomly generated n-dimensional real-valued vectors (i.e.,food sources). Let $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ represent the ith food source in the population and then each food source is generated as follows:

$$x_{ij} = LB_j + (UB_j - LBj) \times r \text{for } j = 1, 2, \ldots, n \text{ and } i = 1, 2, \ldots, SN \quad (1)$$

where r is a uniform random number in the range $[0, 1]$; and LB_j and UB_j are the lower and upper bounds for the dimension j,respectively. These food sources are randomly assigned to PS number of employed bees and their finesses are evaluated.

1.2.3 Initialization of the bee phase

At this stage, each emploed bee x_i generates a new food source x_{new} in the neighborhood of its present position as follows.

$$x_{new(j)} = x_{ij} + (x_{ij} - x_{kj}) \times r, \quad (2)$$

where $k \in \{1, 2, \ldots, PS\} \wedge k \neq i$ and $j \in 1, 2, ..., n$ are randomly chosen indexes. r is a uniformly distributed real number in $[-1,1].

Once x_{new} is obtained, it will be evaluated and compared to x_i. If the fitness of x_{new} is eqaul to or better than that of x_i, x_{new} will replace x_i and become a new member of the population; otherwise x_i is retained. In other words, a greedy selection mechanism is employed between the old and candidate solutions.

1.2.4 Onlooker bee phase

An onlooker bee evaluates the nectar information taken from all the employed bees and selects a food source x_i depending on its probability value p_i calculated by the follow expression.

$$p_i = \frac{f_i}{\sum_{i=1}^{SN} f_i} \quad (3)$$

where f_i is the nectar amount (i.e, the fitness value) of the ith food source x_i. Obviously, the higher the f_i is, the ability that the ith food source is selected.

Once the onlooker has selected her food source x_i, she produces a modification on x_i by using equation 2. As in the case of employed bees, if the modified food source has a btter or equal nectar amount than x_i the modified food source will replace x_i and become a new member in the population.

1.2.5 Scout bee phase

If a food source x_i cannot be futher improved through a predetermined number of trail *limit*, the food source is assumed to be abandoned, and the corresponding employed bee becomes a scout. The scout produces a food source randomly as follows:

$$x_{ij} = LB_j + (UB_j - LB_j) \times r \text{ for } j = 1, 2, \ldots, n. \tag{4}$$

where r is a uniform random number in the range $[0, 1]$. In the basic ABC algorithm, at each cycle at most one count goes outside for searching a new food source.

1.2.6 Main steps of the ABC algorithm

Based on the above explanation of initializing the algorithm parameters and population, the employed bee phase, the onlooker bee phase, and the scout bee phase, the main steps of the basic ABC algorithm can be summarized as follows:

Step 1 Initialization.

Step 2 Place the employed bees on their food sources.

Step 3 Place the onlooker bees on the food sources depending on their nectar amounts.

Step 4 Send the scouts to the search area for discovering new food sources.

Step 5 Memorize the best food source found so far.

Step 6 If a termination criterion is not satisfied, go to step 2; otherwise stop the procedure and output the best food source found so far..

2 A bee colony optimization algorithm to job shop scheduling

```
Jabref reference = BeeJobShop
```

2.1 HONEY BEE COLONY

Colonies of social insects such as ants and bees have instinct ability known as swarm intelligence (Nakrani and Tovey 2004, Teodorovic and Dell'orco, 2005). This highly organized behavior enables the colonies of insects to solve problems beyond capability of individual members by functioning collectively and interacting primitively amongst members of the group. In a honey bee colony for example, this behavior allows honey bees to explore the environment in search of flower patches (food sources) and then indicate the food source to the other bees of the colony when they return to the hive. Such a colony is characterized by self-organization, adaptiveness and robustness. Seeley (1995) proposed a behavioral model of self-organization for a colony of honey bees. In the model, foraging bees visiting flower patches return to the hive with nectar as well as a profitability rating of respective patches. The collected nectar provides feedback on the current status of nectar flow into the hive. The profitability rating is a function of nectar quality, nectar bounty and distance from the hive. The feedback sets a response threshold for an enlisting signal which is known as waggle dance, the length of which is dependent on both the response threshold and the profitability rating. The waggle dance is performed on the dance floor where individual foragers can observe. The foragers can randomly select a dance to observe and follow from which they can learn the location of the flower patch and leave the hive to forage. This self-organized model enables proportionate feedback on goodness of food sources.

2.2 Honey Bee Algorithms

This section details algorithms to perform job shop scheduling inspired by the behavior of honey bee colony. The challenge is to adapt the self-organization behavior of the colony for solving job shop scheduling problems. There are two major characteristics of the bee colony in searching for food sources: waggle dance and forage (or nectar exploration). We will discuss in separate sub-sections on how we map these characteristics of a bee colony to job shop scheduling.

2.2.1 Waggle Dance

A forager f_i on return to the hive from nectar exploration will attempt with probability p to perform waggle dance on the dance floor with duration $D = d_i A$, where d_i changes with profitability rating while A denotes waggle dance scaling factor. Further, it will also attempt with probability r_i to observe and follow a randomly selected dance. The probability r_i is dynamic and also changes with profitability rating. If a forager chooses to follow a selected dance, it will use the path taken by the forager performing the dance to guide its direction for flower patches. We term the path as preferred path. The path for a forager is a series of landmarks from a source (hive) to a destination (nectar). For job shop scheduling, the profitability rating should be related to the objective function, which in our case, is makespan. Let Pf_i denote the profitability rating for a forager, it is given by:

$$Pf_i = \frac{1}{C^i_{max}} \tag{5}$$

where C^i_{max} is the makespan of the schedule generated by a forager f_i. The bee colonies average profitability rating Pf_{colony} is given by:

$$Pf_{colony} = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{C^j_{max}} \tag{6}$$

where n is the number of waggle dances at time t (we only consider those bees that dance when computing profitability rating). C^j_{max} is the makespan of the schedule generated by forage f_i performing waggle dance. The dance duration, d_i is given by:

$$d_i = \frac{Pf_i}{Pf_colony} \tag{7}$$

The probability r_i of following a path is adjusted according the profitability ratings of a forager and the colony based on the lookup table 1 (adopted from Nakrani and Tovey 2004). Essentially, a forager is more likely to randomly observe and follow a waggle dance on the dance floor if its profitability rating is low as compared to the colonys.

2.2.2 Forage (Nectar Exploration)

For foraging algorithm, a population of l foragers is defined in the colony. These foragers cyclically construct solutions to the job shop scheduling problems. The foragers move along branches from one node to another node in

the disjunctive graph and so construct paths representing solutions. A forager must visit every node once and only once in the graph, starting from initial node (i.e. source) and finishing at final node (i.e. sink), so as to construct a complete solution. When a forager is at a specific node, it can only move to next node that is defined in a list of presently allowed nodes, imposed by precedence constraints of operations. A forager chooses the next node from the list according to the state transition rule:

$$P_{ij} = \frac{[p_{ij}(t)]^{\alpha}.[\frac{1}{d_{ij}}]^{\beta}}{\sum_{j \in allowednodes}[p_{ij}(t)]^{\alpha}.[\frac{1}{d_{ij}}]^{\beta}} \tag{8}$$

where p_ij is he rating of the edge between node i and node j. d_{ij} is the heuristic distance between node i and node j. P_{ij} is the probability to branch from node i and node j.

The rating p_{ij} of the edge (directed) between node i and node j is given by:

$$p_{ij} = \begin{cases} \alpha \\ \frac{1-m\alpha}{k-m} \end{cases} \tag{9}$$

where α is the value assigned to the preffered path, $\alpha < 1.0$. k is the number of allowed nodes. m is the number of preffered paths, $m = 1$ or 0.

Based on the expression, it should be noted that for the first nectar exploration expidition by the foragers, p_{ij} will be assigned the same value for all allowed nodes (since $m = 0$).

The parameters α and β tune the relative importance in probability of the weight in edges found in the preferred path versus the heuristic distance. According to this rule, edges that are found in the preferred path and that are shorter will have a higher probability to be chosen for the solution. The heuristic distance is the processing time of the operation associated with nodej. When a forager completes a full path, the edges it has traveled and the makespan of the resulting solution will be kept for the waggle dance when it returns to the hive.

2.2.3 Results and Conclusion

From the results, it is obvious that tabu search outper forms other two heuristics. Tabu search records the closest results to the best known solutions and has the most number of best solutions. Further, it also manages to achieve best results in the shortest execution time. These spectacular

results are attributed to the efficient critical block neighborhoods. Moreover, a tabu list that keeps track of the most recent tabu moves prevents the search algorithm to be locked in local minimums.

Comparing the performance of peers, bee colony and ant colony heuristics, bee algorithm performs slightly better than ant algorithm. Bee algorithm achieves better mean and maximum percentages as well as higher number of best solutions in comparison to ant algorithm.

3 A comparative study of Artificial Bee Colony algorithm

```
@article{Karaboga2009108,
title = "A comparative study of Artificial Bee Colony algorithm",
journal = "Applied Mathematics and Computation",
volume = "214",
number = "1",
pages = "108 - 132",
year = "2009",
note = "",
issn = "0096-3003",
doi = "DOI: 10.1016/j.amc.2009.03.090",
url = "http://www.sciencedirect.com/science/article/B6TY8-4W211Y3-1/2/f92b8a34ede7
author = "Dervis Karaboga and Bahriye Akay",
keywords = "Swarm intelligence",
keywords = "Evolution strategies",
keywords = "Genetic algorithms",
keywords = "Differential evolution",
keywords = "Particle swarm optimization",
keywords = "Artificial Bee Colony algorithm",
keywords = "Unconstrained optimization"
}
```

3.1 Behavior of real honey bees

Tereshko developed a model of foraging behaviour of a honeybee colony based on reactiondiffusion equations. This model that leads to the emergence of collective intelligence of honeybee swarms consists of three essential components: food sources, employed foragers, and unemployed foragers, and defines two leading modes of the honeybee colony behaviour: recruitment

to a food source and abandonment of a source. Tereshko explains the main components of his model as below:

1. Food Sources: In order to select a food source, a forager bee evaluates several properties related with the food source such as its closeness to the hive, richness of the energy, taste of its nectar, and the ease or difficulty of extracting this energy. For the simplicity, the quality of a food source can be represented by only one quantity although it depends on various parameters mentioned above.

2. Employed foragers: An employed forager is employed at a specific food source which she is currently exploiting. She carries information about this specific source and shares it with other bees waiting in the hive. The information includes the distance, the direction and the profitability of the food source.

3. Unemployed foragers: A forager bee that looks for a food source to exploit is called unemployed.It can be either a scout who searches the environment randomly or an onlooker who tries to find a food source by means of the information given by the employed bee. The mean number of scouts is about 510

The exchange of information among bees is the most important occurrence in the formation of collective knowledge. While examining the entire hive it is possible to distinguish some parts that 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 waggle dance. Since information about all the current rich sources is available to an onlooker on the dance floor, probably she could watch numerous dances and chooses to employ herself at the most profitable source. There is a greater probability of onlookers choosing more profitable sources since more information is circulating about the more profitable sources. Employed foragers share their information with a probability which is proportional to the profitability of the food source, and the sharing of this information through waggle dancing is longer in duration. Hence, the recruitment is proportional to profitability of a food source.

In order to better understand the basic behaviour characteristics of foragers, let us examine Fig. 1. Assume that there are two discovered food sources: A and B. At the very beginning, a potential forager will start as unemployed forager. That forager bee will have no knowledge about the food sources around the nest. There are two possible options for such a bee:

1. It can be a scout and starts searching around the nest spontaneously for food due to some internal motivation or possible external clue (S on Fig. 1).

2. It can be a recruit after watching the waggle dances and starts searching for a food source (R on Fig. 1).

After finding the food source, the bee utilizes its own capability to memorize the location and then immediately starts exploiting it. Hence, the bee will become an employed forager. The foraging bee takes a load of nectar from the source and returns to the hive, unloading the nectar to a food store. After unloading the food, the bee has the following options:

1. It might become an uncommitted follower after abandoning the food source (UF).

2. It might dance and then recruit nest mates before returning to the same food source (EF1).

3. It might continue to forage at the food source without recruiting bees (EF2).

It is important to note that not all bees start foraging simultaneously. The experiments confirmed that new bees begin foraging at a rate proportional to the difference between the eventual total number of bees and the number of bees presently foraging.

3.1.1 The Artificial Bee Colony (ABC) Algorithm

In ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. At the first step, the ABC generates a randomly distributed initial population $P(C = 0)$ of SN solutions (food source positions), where SN denotes the size of employed bees or onlooker bees. Each solution $x_i (i = 1, 2, \ldots, SN)$ is a D-dimensional vector. Here, D is the number of optimization parameters. After initialization, the population of the positions (solutions) is subject to repeated cycles, $C = 1, 2, \ldots, MCN$, of the search processes of the employed bees, the onlooker bees and the scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and

tests the nectar amount (fitness value) of the new source (new solution). If the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process, they share the nectar information of the food sources and their position information with the onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As in the case of the employed bee, she produces a modification on the position in her memory and checks the nectar amount of the candidate source. If the nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The main steps of the algorithm are as below:

1. Initialize Population
2. **repeat**
3. Place the employed bees on their food sources
4. Place the onlooker bees on the food sources depending on their nectar amounts
5. Send the scouts to the search area fpr discovering new food sources
6. Memorize the best source found so far
7. **until** requirements are met

In ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto their food sources and evaluating their nectar amounts; after sharing the nectar information of food sources, the selection of food source regions by the onlookers and evaluating the nectar amount of the food sources; determining the scout bees and then sending them randomly onto possible new food sources. At the initialization stage, a set of food sources is randomly selected by the bees and their nectar amounts are determined. At the first step of the cycle, these bees come into the hive and share the nectar information of the sources with the bees waiting on the dance area. A bee waiting on the dance area for making decision to choose a food source is called onlooker and the bee going to the food source visited by herself just before is named as employed bee. After sharing their information with onlookers, every employed bee goes to the food source area visited

by herself at the previous cycle since that food source exists in her memory, and then chooses a new food source by means of visual information in the neighbourhood of the one in her memory and evaluates its nectar amount. At the second step, an onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area. As the nectar amount of a food source increases, the probability of that food source chosen also increases. After arriving at the selected area, she chooses a new food source in the neighbourhood of the one in the memory depending on visual information as in the case of employed bees. The determination of the new food source is carried out by the bees based on the comparison process of food source positions visually. At the third step of the cycle, when the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one. In our model, at each cycle at most one scout goes outside for searching a new food source, and the number of employed and onlooker bees is selected to be equal to each other. These three steps are repeated through a predetermined number of cycles called Maximum Cycle Number (MCN) or until a termination criterion is satisfied.

An artificial onlooker bee chooses a food source depending on the probability value associated with that food source, p_i ,calculated by the following expression (6):

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{10}$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees or onlooker bees. In order to produce a candidate food position from the old one in memory, the ABC uses the following expression (7):

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{11}$$

where $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. ϕ_{ij} is a random between $[-1, 1]$. It controls the production of neighbour food sources around xi;j and represents the comparison of two food positions visually by a bee. As can be seen from (7), as the difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} gets decreased, too. Thus, as the search approaches the optimum solution in the search space, the step length is adaptively reduced.

If a parameter value produced by this operation exceeds its predetermined limit, the parameter can be set to an acceptable value. In this work, the value of the parameter exceeding its limit is set to its limit value.

The food source of which the nectar is abandoned by the bees is replaced with a new food source by the scouts. In ABC, this is simulated by producing a position randomly and replacing it with the abandoned one. In ABC, if a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called limit for abandonment. Assume that the abandoned source is x_i and $j \in \{1, 2, \ldots, Dg\}$, then the scout discovers a new food source to be replaced with xi . This operation can be defined as in (12)

$$x_i^j = x_{min}^j + rand[0, 1](x_{max}^i - x_{min}^j) \tag{12}$$

After each candidate source position v i;j is produced and then evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has an equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. In other words, a greedy selection mechanism is employed as the selection operation between the old and the candidate one.

Totally, ABC algorithm employs four different selection processes: (1) a global probabilistic selection process, in which the probability value is calculated by (6) used by the onlooker bees for discovering promising regions, (2) a local probabilistic selection process carried out in a region by the employed bees and the onlookers depending on the visual information such as the color, shape and fragrance of the flowers (sources) (bees will not be able to identify the type of nectar source until they arrive at the right location and discriminate among sources growing there based on their scent) for determining a food source around the source in the memory in a way described by (7), (3) a local selection called greedy selection process carried out by onlooker and employed bees in that if the nectar amount of the candidate source is better than that of the present one, the bee forgets the present one and memorizes the candidate source produced by (7). Otherwise, the bee keeps the present one in the memory. (4) a random selection process carried out by scouts as defined in (8). It is clear from the above explanation that there are three control parameters in the basic ABC: The number of food sources which is equal to the number of employed or onlooker bees (SN), the value of limit and the maximum cycle number (MCN).

In the case of honeybees, the recruitment rate represents a measure of how quickly the bee colony finds and exploits a newly discovered food source. Artificial recruiting could similarly represent the measurement of the speed with which the feasible solutions or the good quality solutions of the difficult optimization problems can be discovered. The survival and progress of the bee colony are dependent upon the rapid discovery and efficient utilization of the best food resources. Similarly, the successful solution of difficult engineering problems is connected to the relatively fast discovery of good solutions especially for the problems that need to be solved in real time. In a robust search process, exploration and exploitation processes must be carried out together. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. Detailed pseudo-code of the ABC algorithm is given below:

1. Initialize the population of solutions $x_i, i = 1, \ldots, SN$
2. Evaluate the population
3. cycle = 1
4. **repeat**
5. Produce new solutions v_i for the employed bees by using 11 and evaluate them
6. Apply the greedy selection process for the employed bees
7. Calculate the probability values P_i for the solutions x_i selected depending on P_i and evaluate them
8. Apply the greedy selection process for the onlookers
9. Determine the abandoned solution for the scout, if exists, and replace it with new randomly produced solution x_i by 12.
10. Memorize the best solution achieved so far
11. cycle = cycle + 1
12. **until** cycle = MCN

3.1.2 Conclusion

In this work, the performance of ABC algorithm was compared with those of GA, PSO, DE and ES optimization algorithms. Although there are several improved versions of GA, DE and PSO in the literature, their standard versions were used since the ABC algorithm presented in this work is its first version i.e. its standard version. Although the performance of ABC algorithm can be improved by integrating useful heuristics, in this work our purpose was to compare the performance of standard version of ABC with those of other well-known population-based algorithms. In Experiments 1, we used the same population number and the maximum evaluation number for all problems although it is a known fact that these control parameters affect the performance of algorithms significantly. However, in most comparative studies these parameter values are varied with respect to the dimension of the problems or to their other characteristics. The reason is that we assumed the users of an algorithm do not know much about the recommended values of these parameters for their problems to be optimized.

While GA and DE employ crossover operators to produce new or candidate solutions from the present ones, ABC algorithm does not. ABC algorithm produces the candidate solution from its parent by a simple operation based on taking the difference of randomly determined parts of the parent and a randomly chosen solution from the population. This process increases the convergence speed of search into a local minimum. In GA, DE and PSO the best solution found so far is always kept in the population and it can be used for producing new solutions in the case of DE and GA, new velocities in the case of PSO. However, in ABC, the best solution discovered so far is not always held in the population since it might be replaced with a randomly produced solution by a scout. Therefore, it might not contribute to the production of trial solutions. Both DE and ABC employ a greedy selection process between the candidate and the parent solutions. In ABC, on employed bees stage a trial solution is produced for each solution in the population as in the case of DE without depending on the quality of solutions. On onlooker bees stage, the solutions with the higher fitness value are used more often than those with less fitness values to produce trial solutions. It means that the promising regions of the search space are searched in shorter time and in detail. This selection process is similar to the natural selection or to the seeded selection employed in GA.

In GA or DE, mutation process creates a modification on a randomly selected part of a solution to provide required diversity in the population. In ABC, there are two types of mechanisms to control the diversity in the

population: (a) As in DE or GA, a randomly chosen part of a parent is modified with a magnitude determined randomly to obtain a trail solution. This modification is relatively small and useful for local search and fine tuning. (b) Rather than changing a part of a solution, a whole solution in the population is removed and then a new one produced randomly is inserted into the population by a scout. This mechanism provides the ABC algorithm a global search ability and prevents the search from premature convergence problem. This feature weakens the dependency of the algorithms performance on the population size, too. Hence, there is a good balance between the local search process carried out by artificial onlooker and employed bees and the global search process managed by artificial scouts. Therefore, the ABC algorithm produces better results on multimodal and multivariable problems than other algorithms considered in this paper.

Apart from the maximum evaluation number and population size, a standard GA has three more control parameters (crossover rate, mutation rate, generation gap), a standard DE has at least two control parameters (crossover rate, scaling factor) and a basic PSO has three control parameters (cognitive and social factors, inertia weight). Also, limit values for the velocities v_{max} have a significant effect on the performance of PSO. The ABC algorithm has only one control parameter (limit) apart from Colony Size and Maximum Cycle Number. In the present work, we described an expression for determining the value of limit depending on population (colony size) and dimension of problem. Therefore, now ABC has only two common control parameters: maximum cycle number (MCN) and colony size (SN). Consequently, ABC is as simple and flexible as DE and PSO; and also employs less control parameters.

ESs used in Experiments 2 and 3 employ recombination and mutation operators to produce offsprings (new individuals) while ABC uses only mutation operator. Although the basic version of ES is as simple as ABC, the improved versions used for comparison in this work are more complex than ABC. Moreover, all of them employ more control parameters than ABC.

This work compared the performance of ABC algorithm with those of GA, DE, PSO and ES algorithms on a large set of unconstrained test functions. From the results obtained in this work, it can be concluded that the performance of ABC algorithm is better than or similar to that of these algorithms although it uses less control parameters and it can be efficiently used for solving multimodal and multidimensional optimization problems.

4 An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem

```
@article{Singh2009625,
title = "An artificial bee colony algorithm for the leaf-constrained minimum spann
journal = "Applied Soft Computing",
volume = "9",
number = "2",
pages = "625 - 631",
year = "2009",
note = "",
issn = "1568-4946",
doi = "DOI: 10.1016/j.asoc.2008.09.001",
url = "http://www.sciencedirect.com/science/article/B6W86-4TDC093-1/2/c00d080ad03f
author = "Alok Singh",
keywords = "Artificial bee colony algorithm",
keywords = "Constrained optimization",
keywords = "Leaf-constrained minimum spanning tree",
keywords = "Swarm intelligence"
}
```

5 The Artificial Bee Colony Algorithm

The artificial bee colony algorithm is a new population-based metaheuristic approach proposed by Karaboga and further developed by Karaboga and Basturk. This approach is inspired by the intelligent foraging behavior of honeybee swarm. The foraging bees are classified into three categoriesemployed, onlookers and scouts. All bees that are currently exploiting a food source are classified as employed. The employed bees bring loads of nectar from the food source to the hive and may share the information about food source with onlooker bees. Onlookers are those bees that are waiting in the hive for the information to be shared by the employed bees about their food sources and scouts are those bees which are currently searching for new food sources in the vicinity of the hive. Employed bees share information about food sources by dancing in a common area in the hive called dance area. The duration of a dance is proportional to the nectar content of the food source currently being exploited by the dancing bee. Onlooker bees which watch numerous dances before choosing a food source tend to choose a food source according to the probability proportional to the quality of that food

source. Therefore, the good food sources attract more bees than the bad ones. Whenever a bee, whether it is scout or onlooker, finds a food source it becomes employed. Whenever a food source is exploited fully, all the employed bees associated with it abandon it, and may again become scouts or onlookers. Scout bees can be visualized as performing the job of exploration, whereas employed and onlooker bees can be visualized as performing the job of exploitation.

Motivated by this foraging behavior of honeybees, Karaboga proposed the artificial bee colony algorithm. In the ABC algorithm, each food source represents a possible solution to the problem under consideration and the nectar amount of a food source represents the quality of the solution represented by that food source. In this algorithm also colony of artificial bees (bees for short) has same three types of beesemployed, onlookers and scouts. First half of the bee colony comprises employed bees, whereas the latter half contains the onlookers. The ABC algorithm assumes that there is only one employed bee for every food source, i.e., the number of food sources is same as the number of employed bees. The employed bee of an abandoned food source becomes a scout and as soon as it finds a new food source it again becomes employed. The ABC algorithm is an iterative algorithm. It starts by associating all employed bees with randomly generated food sources (solution). Then during each iteration, every employed bee determines a food source in the neighborhood of its currently associated food source and evaluates its nectar amount (fitness). If its nectar amount is better than that of its currently associated food source then that employed bee moves to this new food source leaving the old one, otherwise it retains its old food source. When all employed bees have finished this process, they share the nectar information of the food sources with the onlookers, each of whom selects a food source according to a probability proportional to the nectar amount of that food source. The probability pi of selecting a food source i is determined using the following expression:

$$p_i = \frac{f_i}{\sum_{j=1}^{m} f_j} \tag{13}$$

where f_i is the fitness of the solution represented by the food source i and m is the total number of food sources. Clearly, with this scheme good food source will get more onlookers that the bad ones. After all onlookers have selected their food sources, each of them determines a food source in the neighborhood of his chosen food source and computes its fitness. The best food source among all the neighboring food sources determined by the onlookers associated with a particular food source i and food source i itself,

will be the new location of the food source i. If a solution represented by a particular food source does not improve for a predetermined number of iterations then that food source is abandoned by its associated employed bee and it becomes a scout, i.e., it will search for a new food source randomly. This tantamounts to assigning a randomly generated food source (solution) to this scout and changing its status again from scout to employed. After the new location of each food source is determined, another iteration of ABC algorithm begins. The whole process is repeated again and again till the termination condition is satisfied.

The food source in the neighborhood of a particular food source is determined by altering the value of one randomly chosen solution parameter and keeping other parameters unchanged. This is done by adding to the current value of the chosen parameter the product of an uniform variate in and the difference in values of this parameter for this food source and some other randomly chosen food source. Formally, suppose each solution consists of d parameters and let $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ be a solution with parameter values $x_{i1}, x_{i2}, \ldots, x_{id}$. In order to determine a solution x_{0i} in the neighborhood of x_i, a solution parameter j and another solution $x_k = (x_{k1}, x_{k2}, ..., x_{kd})$ are selected randomly. Except for the value of the selected parameter j, all other parameter values of x_{0i} are same as x_i, i.e., $x'_i = (x_{i1}, x_{i2}, \ldots, x_{i(j-1)}, x_{ij}, x_{i(j+1)}, \ldots, x_{id})$. The value x'_i of the selected parameter j in x_i is determined using the following formula:

$$x'_{ij} = x_{ij} + u(x_{ij} - x_{kj}) \tag{14}$$

where u is an uniform variate in $[-1, 1]$. If the resulting value falls outside the acceptable range for parameter j, it is set to the corresponding extreme value in that range.

5.0.3 Collisions

LCMST is a discrete optimization problem. Moreover, we have represented the solutions by the set of their interior nodes, i.e., there is no ordering among interior nodes. Therefore, the method described in the last paragraph of previous section cannot be applied to LCMST problem. Even selecting a random node from another randomly chosen solution for altering the solution in the aforementioned way at the best produces a random effect. It is desirable to utilize somehow the information gathered about the food source by another employed bee while generating a neighboring solution. This may help in producing a good neighboring solution. In the LCMST problem, if a node is an interior node in one good solution then it

is highly likely that it is an interior node in many other good solutions also. We have used this observation while generating the neighboring solutions. We randomly remove an interior node from the solution and in its place we insert an interior node selected from another randomly chosen solution. The node selected for insertion from another solution should be different from all other nodes present in the solution as well as from the node that has been removed. Ties are broken arbitrarily. If we cannot find any such node then that means that the two solutions are identical. We call such a situation a collision. If a collision occurs while generating a neighboring solution for an employed bee then original solution is abandoned and the concerned employed bee becomes scout, i.e., a new solution is generated randomly and the status of this scout bee is again changed to employed by associating it with that solution. This is done to eliminate one duplicate solution. If a collision occurs while generating neighboring solution for an onlooker bee then another solution is chosen randomly for selecting an interior node from it to insert in the original solution. This process is repeated till we find a solution that is different from the original solution. Here it is to be noted that for onlookers there is no point in generating a random solution in case of a collision because for survival this randomly generated solution has to compete with the original solution and with the solutions of other onlookers, which are also associated with the same original solution. Therefore, such randomly generated solutions seldom survive.

5.0.4 Other Features

If the solution associated with an employed bee does not improve for a predetermined number of iterations then it becomes a scout. In our ABC algorithm there is a second possibility in which an employed bee can become scout. As described previously an employed bee can become scout through collision also. We have not put any limit on the number of scouts in a single iteration like other ABC algorithms. In our ABC algorithm number of scouts depends on the above two conditions. There can be many scouts in an iteration if these two conditions are satisfied many times, or there can be no scout if these two conditions remain unsatisfied.

6 A novel approach to image edge enhancement using Artificial Bee Colony optimization algorithm for hybridized smoothening filters

```
@INPROCEEDINGS{5393866,
author={Benala, T.R. and Jampala, S.D. and Villa, S.H. and Konathala, B.},
booktitle={Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congres
year={2009},
month={9-11},
volume={},
number={},
pages={1071 -1076},
keywords={artificial bee colony optimization algorithm;feature extraction;hybridiz
doi={10.1109/NABIC.2009.5393866},
ISSN={},}
```

6.1 Artificial Bee Colony Algorithm

ABC is a new member of swarm intelligence (a branch of nature inspired algorithms focused on insect behavior) that tries to model natural behavior of real honey bees in food foraging. The foraging property of bees is used in problem modeling and solution, using several mechanisms of which waggle dance is a key to optimally locate food sources and to search for new ones. The dancing behavior of foraging bees while performing waggle dance is such that the direction of bees indicates the direction of the food source in relation to the sun, the intensity of the waggles indicates how far away it is and the duration of the dance indicates the amount of nectar on related food source. The bee system consists of Food Sources and Foragers.

6.1.1 Food Sources

The value of a food source depends on parameters like its proximity to the nest, richness of energy and ease of extracting this energy.

6.1.2 Foragers

There are three types of foragers – Unemployed, employed and experienced.

Unemployed — There are two possibilities for an unemployed forager.

1. Scout Bee: if the bee starts searching spontaneously without any knowledge. Their percentage varies from 5% to 30% according to the information into the nest.
2. Recruit Bee: if the bee starts searching by using the knowledge from waggle dance done by some other bee.

Employed — Employed foragers: when the recruit bee finds and exploits the food source it will raise to be an employed forager who memorizes the location of the food source, loads a portion of nectar and unloads the nectar to the food area in the hive (apiary).

1. If the nectar amount is exhausted the bee abandons the food source and becomes an unemployed bee
2. If there are still sufficient amount of nectar it continues to forage without sharing the food source information with the nest mates
3. Or it can go to the dance area to perform waggle dance for informing the nest mates about the same food source

Experienced — These types of foragers use their historical memories for the location and quality of food sources. They can be an inspector, a reactivated forager or a recruit bee.

7 A hybrid discrete Artificial Bee Colony - GRASP algorithm for clustering

```
@INPROCEEDINGS{5223810,
author={Marinakis, Y. and Marinaki, M. and Matsatsinis, N.},
booktitle={Computers Industrial Engineering, 2009. CIE 2009. International Confere
year={2009},
month={6-9},
volume={},
number={},
pages={548 -553},
keywords={GRASP algorithm;UCI Machine Learning Repository;ant colony optimization;
doi={10.1109/ICCIE.2009.5223810},
ISSN={},}
```

This algorithm looks like the Ant Colony Optimization algorithm but it does not use at all the concept of pheromone trails.