Bee Colony Optimization (BCO)

Dušan Teodorović

University of Belgrade, Faculty of Transport and Traffic Engineering, Vojvode Stepe 305
11000 Belgrade, Serbia
dusan@sf.bg.ac.rs

Abstract. Swarm Intelligence is the part of Artificial Intelligence based on study of actions of individuals in various decentralized systems. The Bee Colony Optimization (BCO) metaheuristic has been introduced fairly recently as a new direction in the field of Swarm Intelligence. Artificial bees represent agents, which collaboratively solve complex combinatorial optimization problem. The chapter presents a classification and analysis of the results achieved using Bee Colony Optimization (BCO) to model complex engineering and management processes. The primary goal of this chapter is to acquaint readers with the basic principles of Bee Colony Optimization, as well as to indicate potential BCO applications in engineering and management.

1 Introduction

Many species in the nature are characterized by swarm behavior. Fish schools, flocks of birds, and herds of land animals are formed as a result of *biological needs* to stay together. Individuals in herd, fish school, or flock of birds has a higher probability to stay alive, since predator usually assault only one individual. A collective movement characterizes flocks of birds, herds of animals, and fish schools. Herds of animals respond quickly to changes in the direction and speed of their neighbors. Swarm behavior is also one of the main characteristics of social insects (bees, wasps, ants, termites). Communication between individual insects in a colony of social insects has been well known. The communication systems between individual insects contribute to the configuration of the ‘‘collective intelligence” of the social insect colonies. The term ‘‘Swarm intelligence”, that denotes this ‘‘collective intelligence” has come into use [1], [2], [3], [4].

Swarm Intelligence [4] is the part of Artificial Intelligence based on study of actions of individuals in various decentralized systems. These decentralized systems (Multi Agent Systems) are composed of physical individuals (robots, for example) or “virtual” (artificial) ones that communicate among themselves, cooperate, collaborate, exchange information and knowledge and perform some tasks in their environment.

The Bee Colony Optimization (BCO) metaheuristic [5], [6], [7], [8], [9] has been introduced fairly recently by Lučić and Teodorović as a new direction in the field of Swarm Intelligence. The BCO has been successfully applied to various engineering and management problems by Teodorović and coauthors ([10], [11], [12], [13], [14], [15], [16], [17]). The BCO approach is a "bottom-up" approach to modeling where special kinds of artificial agents are created by analogy with bees. Artificial bees represent agents, which collaboratively solve complex combinatorial optimization problem. The chapter presents a classification and analysis of the results achieved using BCO to model complex engineering and management processes. The primary goal of this paper is to acquaint readers with the basic principles of Bee Colony Optimization, as well as to indicate potential BCO applications in engineering and management.

2 Algorithms Inspired by Bees' Behavior in the Nature

The BCO is inspired by bees' behavior in the nature. The basic idea behind the BCO is to create the multi agent system (colony of artificial bees) capable to successfully solve difficult combinatorial optimization problems. The artificial bee colony behaves partially alike, and partially differently from bee colonies in nature. We will first describe the behavior of bees' in nature, as well as other algorithms inspired by bee s behavior. Then, we will describe a general Bee Colony Optimization algorithm and afterwards BCO applications in various engineering and management problems.

In spite of the existence of a large number of different social insect species, and variation in their behavioral patterns, it is possible to describe individual insects' as capable of performing a variety of complex tasks [18]. The best example is the collection and processing of nectar, the practice of which is highly organized. Each bee decides to reach the nectar source by following a nestmate who has already discovered a patch of flowers. Each hive has a so-called dance floor area in which the bees that have discovered nectar sources dance, in that way trying to convince their nestmates to follow them. If a bee decides to leave the hive to get nectar, she follows one of the bee dancers to one of the nectar areas. Upon arrival, the foraging bee takes a load of nectar and returns to the hive relinquishing the nectar to a food storer bee. After she relinquishes the food, the bee can (a) abandon the food source and become again uncommitted follower, (b) continue to forage at the food source without recruiting the nestmates, or (c) dance and thus recruit the nestmates before the return to the food source. The bee opts for one of the above alternatives with a certain probability. Within the dance area, the bee dancers "advertise" different food areas. The mechanisms by which the bee decides to follow a specific dancer are not well understood, but it is considered that "the recruitment among bees is always a function of the quality of the food source" [18].

Few algorithms inspired by bees' behavior appeared during the last decade (Bee System, BCO algorithm, ABC algorithm, MBO, Bees Algorithm, HBMO algorithm, BeeHive, Artificial Bee Colony, VBA algorithm). The year of publication, the names of the authors, the names of the algorithm, and the problems studied are shown in the Table 1. In a subsequent section we describe basic principles of these algorithms and we show their potential applications.

Table 1. The algorithms inspired by bees’ behavior

Year	Authors	Algorithm	Problem studied
1996	Yonezawa and Kikuchi	Ecological algorithm	Description of the collective intelligence based on bees’ behavior
1997	Sato and Hagiwara	Bee System (BS)	Genetic Algorithm Improvement
2001	Lučić and Teodorović	BCO	Traveling salesman problem
2001	Abbas	MBO	Propositional satisfiability problems
2002	Lučić and Teodorović	BCO	Traveling salesman problem
2003	Lučić and Teodorović	BCO	Vehicle routing problem in the case of uncertain demand
2003	Lučić and Teodorović	BCO	Traveling salesman problem
2004	Wedde, Farooq, and Zhang	BeeHive	Routing protocols
2005	Teodorović, and Dell’ Orco	BCO	Ride-matching problem
2005	Karaboga	ABC	Numerical optimization
2005	Drias, Sadeg, and Yahi	BSO	Maximum Weighted Satisfiability Problem
2005	Yang	Virtual Bee Algorithm (VBA)	Function optimizations with the application in engineering problems
2005	Benatchba, Admane, and Koudil	MBO	Max-Sat problem
2006	Teodorović, Lučić, Marković, and Dell’ Orco	BCO	Traveling salesman problem and a routing problems in networks
2006	Chong, Low, Sivakumar, and Gay	Honey Bee Colony Algorithms	Job shop scheduling problem
2006	Pham, Soroka, Ghanbarzadeh, and Koc	Bees Algorithm	Optimization of neural networks for wood defect detection
2006	Basturk and Karaboga	ABC	Numeric function optimization
2006	Navrat	Bee Hive Model	Web search
2006	Wedde, Timm, and Farooq	BeeHiveAIS	Routing protocols
2007	Yang, Chen, and Tu	MBO	Improvement of the MBO algorithm
2007	Koudil, Benatchba, Tarabetand, and El Batoul	MBO	Partitioning and scheduling problems

	Sahraoui		
2007	Quijano and Passino	Honey Bee Social Foraging Algorithm	Solving optimal resource allocation problems
2007	Marković, Teodorović, and Aćimović-Raspopović	BCO	Routing and wavelength assignment in all-optical networks
2007	Wedde, Lehnhoff, B.van Bonn, Bay, Becker, Böttcher, Brunner, Büscher, Fürst, Lazarescu, Rotaru, Senge, Steinbach, Yilmaz, and Zimmermann	BeeHive	Highway traffic congestion mitigation
2007	Karaboga and Basturk	ABC	Testing ABC algorithm on a set of multi-dimensional numerical optimization problems
2007	Karaboga, Akay and Ozturk	ABC	Feed-forward neural networks training
2007	Afshar, Bozorg Haddada, Marin, Adams	Honey-bee mating optimization (HBMO) algorithm	Single reservoir operation optimization problems
2007	Baykasoglu, Özbakýr, and Tapkan	Artificial Bee Colony	Generalized Assignment Problem
2007	Teodorović and Šelmić	BCO	*p*-Median Problem
2008	Karaboga and Basturk	ABC	Comparison performances of ABC algorithm with the performances of other population-based techniques
2008	Fathian, Amiri, and Maroosi	Honeybee mating optimization algorithm	Cluster analysis
2008	Teodorović	BCO	Comparison performances of BCO algorithm with the performances of other Swarm Intelligence-based techniques
2009	Pham, Haj Darwish, Eldukhr	Bees Algorithm	Tuning the parameters of a fuzzy logic controller
2009	Davidović, Šelmić and Teodorović	BCO	Static scheduling of independent tasks on homogeneous multiprocessor systems

Yonezawa and Kikuchi described collective intelligence based on bees' behavior [19]. Sato and Hagiwara [20] proposed an improved genetic algorithm named *Bee System.* The proposed *Bee System* employs new operations - *concentrated crossover* and *Pseudo-Simplex Method.* By computer simulations the authors showed that the *Bee System* has better performance than the conventional genetic algorithm. The *Bee System* proposed by Sato and Hagiwara [20] can rather be categorized as Genetic Algorithm than Swarm Intelligence algorithm.

Abbass [21] developed the *MBO* model that is based on the marriage process in honeybees. The model simulates the evolution of honeybees. The author started with a solitary colony (single queen without a family) to the emergence of an eusocial colony (one or more queens with a family). The model is applied to a fifty propositional satisfiability problems (SAT) with 50 variables and 215 constraints. The proposed MBO approach was very successful on a group of fifty hard 3-SAT problems.

Wedde et al [22] developed the *BeeHive* algorithm that is also based on honeybee behavior. The authors introduced the concept of foraging regions. Each foraging region has one representative node. There are two types of agents within the BeeHive algorithm: short distance bee agents and long distance bee agents. Short distance bee agents collect and disseminate information in the neighborhood, while long distance bee agents collect and disseminate information to typically all nodes of a network.

Karaboga [23] developed the *Artificial Bee Colony* (*ABC*) algorithm. Karaboga and Basturk, and Karaboga et. [24], [25], [26] further improved and applied the *ABC* algorithm to various problems. The authors created colony of artificial bees composed of the following agents: *employed bees* (a bee flying to the food source), *onlookers* (a bee waiting on the dance area for making decision to choose a food source) and *scouts* (a bee performing random search). In the *ABC* algorithm, half of the colony consists of employed bees. The second part of the colony is composed of onlookers. Every food source could be occupied by only one employed bee. The employed bee without food source becomes a scout. The *ABC* algorithm performs search in cycles. Each cycle consists of the following three steps: (a) Employed bees fly to the food sources, collect the nectar and return to the hive. In the hive we measure their nectar amounts; (b) Information on collected nectar amounts are on a disposal to all artificial bees. Based on this information, the onlookers select the food sources; (c) Chosen bees that become scout bees fly to the possible food sources. In the *ABC* algorithm, the initial population of the solutions is generated randomly. In the subsequent cycles, the employed bees, and the onlooker bees probabilistically create a modifications on the initial solutions. Karaboga and Basturk [24] compared the performances of the *ABC* algorithm with the performances of the PSO, PS-EA and GA. Karaboga and Basturk [24] concluded, "that the proposed algorithm has the ability to get out of a local minimum and can be efficiently used for multivariable, multimodal function optimization". Karaboga et al. [25] also used the ABC algorithm to train feed-forward artificial neural networks. The authors compared performances of the ABC algorithm with the back propagation algorithm and the genetic algorithm. Performed experiments showed that the *ABC* algorithm could be good addition to the existing algorithms for feed-forward neural networks training.

Drias et al. [27] studied Maximum Weighted Satisfiability Problem. They proposed the *Bees Swarm Optimization* (*BSO*) algorithm. The authors tested their approach on the well-known benchmark problems. The *BSO* outperformed other evolutionary algorithms especially AC-SAT, an ant colony algorithm for SAT.

Yang et al. [28] developed the *Virtual Bee Algorithm* (*VBA*) to solve the function optimizations with the application in engineering problems. The simulations of the optimization of De Jong's test function and Keane's multi-peaked bumpy function showed that the *VBA* is usually as effective as genetic algorithms.

Benatchba et al. [29] applied the *MBO* algorithm to the Max-Sat problem.

Chong et al. [30] applied honey bees foraging model to the job shop scheduling problem. The authors presented experimental results comparing the proposed honeybee colony approach with existing approaches such as ant colony and tabu search. The experimental results showed that the performance of the algorithm is equivalent to ant colony algorithms,

Pham et al. [31], [32] proposed population-based search algorithm called the *Bees Algorithm* (*BA*). This algorithm also mimics the food foraging behavior of honeybees. The algorithm performs a neighborhood search combined with random search.

Navrat [33] presented a new approach to web search, based on a beehive metaphor. The author proposed a modified model of a beehive. The proposed model is simple, and it describes some of the processes that take place in web search.

Wedde et al. [34] developed a novel security framework, which is inspired by the principles of Artificial Immune Systems (AIS), for Nature inspired routing protocols.

Yang et al. [35] proposed a faster Marriage in Honey Bees Optimization (FMBO) algorithm with global convergence. By the proposed approach, the computation process becomes easier and faster. The global convergence characteristic of FMBO is also proved by using the Markov Chain theory.

Koudil et al. [36] studied partitioning and scheduling in the design of embedded systems. The authors applied Marriage in honey-Bees Optimization algorithm (MBO).

Quijano and Passino [37], [38] developed the *Honey Bee Social Foraging Algorithm*. The proposed algorithm was successfully applied to the optimal resource allocation problems.

Wedde et al. [39] proposed decentralized multi-agent approach (termed *BeeJamA)* on multiple layers for car routing. The proposed approach is based on the *BeeHive* algorithm.

Afshar et al. [40] applied Honey-bee mating optimization (*HBMO*) algorithm to the single reservoir operation optimization problems.

Baykasoglu et al. [41] made an excellent survey of the algorithms inspired by bees' behavior in the nature. The authors described the Artificial Bee Colony algorithm, and presented an artificial bee colony algorithm to solve Generalized Assignment Problem GAP.

Fathian et al. [42] applied algorithm inspired by bees' behavior in cluster analysis. The authors proposed a two-stage method. They used self-organizing feature maps (SOM) neural network to determine the number of clusters. In the second step, the authors used honeybee mating optimization algorithm based on K- means algorithm to find the final solution.

Pham et al. [43] used the *Bees Algorithm* to tune the parameters of a fuzzy logic controller. The controller was developed to stabilize and balance an under-actuated two-link acrobatic robot (ACROBOT) in the upright position. Simulation results showed that using the *Bees Algorithm* to optimize the membership functions of the fuzzy logic system enhanced the controller performance.

3 Bee Colony Optimization (BCO) Algorithm

Lučić and Teodorović [5], [6], [7], [8] were among first who used basic principles of collective bee intelligence in solving combinatorial optimization problems. The BCO is a population-based algorithm. Population of *artificial bees* searches for the optimal solution. Artificial bees represent agents, which collaboratively solve complex combinatorial optimization problems. Every artificial bee generates one solution to the problem. The algorithm consists of two alternating phases: *forward pass* and *backward pass*. In each forward pass, every artificial bee is exploring the search space. It applies a predefined number of moves, which construct and/or improve the solution, yielding to a new solution. Having obtained new partial solutions, the bees go again to the nest and start the second phase, the so-called backward pass. In the backward pass, all artificial bees share information about their solutions.

Let us consider Traveling Salesman Problem as an example. When solving the TSP problem by the BCO algorithm, we decompose the TSP problem into stages. In each stage, a bee chooses a new node to be added to the partial Traveling Salesman tour created so far (Figure 1).

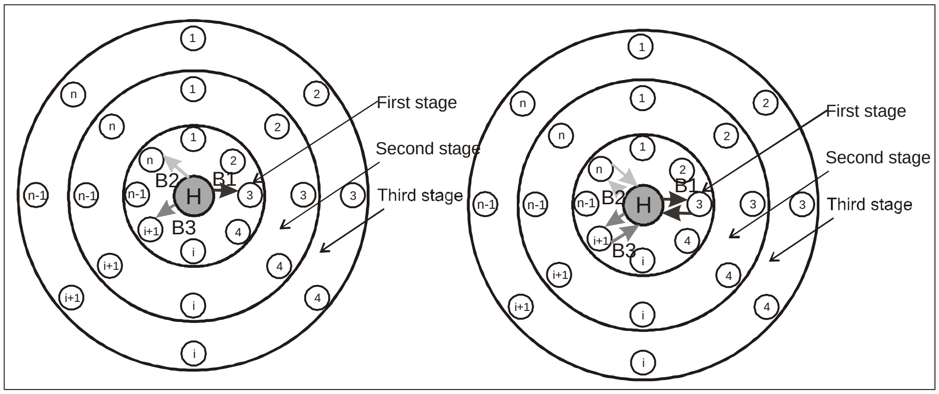


Fig. 1. First forward pass and the first backward pass.

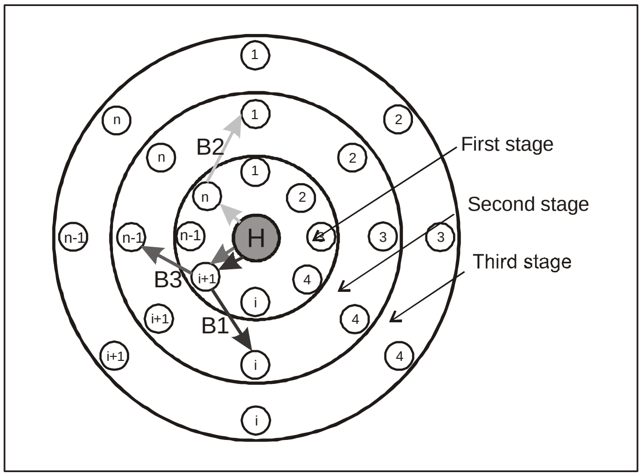


Fig. 2. Second forward pass

In nature, bees would perform a dancing ceremony, which would notify other bees about the quantity of food they have collected, and the closeness of the patch to the nest. In the BCO search algorithm, the artificial bees publicize the quality of the solution, i.e. the objective function value. During the backward pass, every bee decides with a certain probability whether to abandon the created partial solution and become again uncommitted follower, or dance and thus recruit the nestmates before returning to the created partial solution (bees with higher objective function value have greater chance to continue its own exploration). Every follower, choose a new solution from recruiters (Figure 3) by the roulette wheel (better solutions have higher probability of being chosen for exploration).

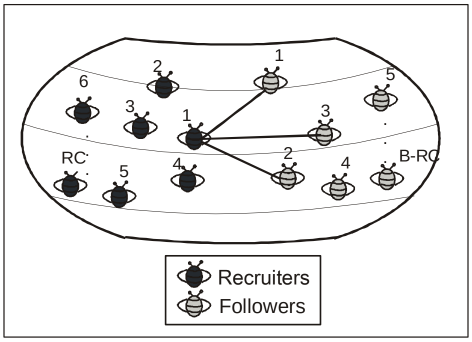


Fig. 3. Recruiting of uncommitted followers

During the second forward pass (Figure 2), bees expand previously created partial solutions, by a predefined number of nodes, and after that perform again the backward pass and return to the hive. In the hive, bees again participate in a decision making process, make a decision, perform third forward pass, etc. The two phases of the search algorithm, forward and backward pass, are performed *iteratively*, until a stopping condition is met. The possible stopping conditions could be, for example, the maximum total number of forward/backward passes, the maximum total number of forward/backward passes without the improvement of the objective function, etc.

The algorithm parameters whose values need to be set prior the algorithm execution are as follows:

B - The number of bees in the hive

NC - The number of constructive moves during one forward pass

In the beginning of the search, all the bees are in the hive. The following is the pseudocode of the BCO algorithm:

```
1. Initialization: every bee is set to an empty solution;
2. For every bee do the forward pass:
      a) Set $k = 1$;  //counter for constructive moves in the forward pass;
      b) Evaluate all possible constructive moves;
      c) According to evaluation, choose one move using the roulette wheel;
      d) $k = k + 1$; If $k \leq NC$  Go To step b.
3. All bees are back to the hive; // backward pass starts;
4. Sort the bees by their objective function value;
5. Every bee decides randomly whether to continue its own exploration and become
   a recruiter, or to become a follower (bees with higher objective function value
   have greater chance to continue its own exploration);
6. For every follower, choose a new solution from recruiters by the roulette wheel;
7. If the stopping condition is not met Go To step 2;
8. Output the best result.
```

3.1 Constructive and Improving BCO variants

A combinatorial optimization algorithm could be of constructive or improving type. Constructive approaches start from scratch. Within these approaches the analyst construct a solution step by step. When doing this, we usually apply some problem specific heuristics. On the other hand, improving approaches begin from a complete solution. The complete solution (possible a feasible one) is typically generated randomly or by some heuristics. By perturbing that solution, we try to improve it. The examples of such techniques are Simulated Annealing, or Tabu Search. Until now, the BCO algorithms in the literature have been constructive. Todorović et al. [16] developed a bee colony approach for the nurse rostering problem. Their approach is the first one that allows both constructive and improving steps to be applied and combined together.

3.2 The Artificial Bees and Approximate Reasoning

Artificial Bees confront few decision-making problems while searching for the optimal solution. The next are bees' choice dilemmas: (a) What is the next solution component to be added to the partial solution? (b) Should the partial solution be discarded or not? The greater part of the choice models in the literature, are based on random utility modeling concepts. These approaches are highly rational. They are based on assumptions that decision makers have perfect information processing capabilities and always act in a rational way (trying to maximize utility). In order to present an alternative modeling approach, researchers started to use less normative theories. The basic concepts of Fuzzy Set Theory, linguistic variables, approximate reasoning, and computing with words have more sympathy for uncertainty, imprecision, and linguistically expressed observations. Following these ideas, Teodorović and Dell'Orco [10], [14] started from the assumption that the quantities perceived by artificial bees are "fuzzy". In other words, artificial bees could also use approximate reasoning and rules of fuzzy logic in their communication and acting. When adding the solution component to the current partial solution during the forward pass, a specific bee could perceive a specific solution component as 'less attractive', 'attractive', or 'very attractive'. We also assume that an artificial bee can perceive a specific attribute as 'short', 'medium' or 'long' (Figure 4), 'cheap', 'medium', or 'expensive', etc. The approximate reasoning algorithm for calculating the solution component attractiveness consists of the rules of the following type:

If the attributes of the solution component are VERY GOOD
Then the considered solution component is VERY ATTRACTIVE

The main advantage of using the approximate reasoning algorithm for calculating the solution component attractiveness is that it is possible to calculate solution component attractiveness even if some of the input data were only approximately known.

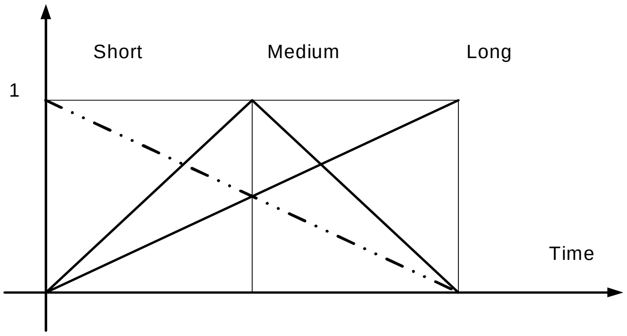


Fig. 4. Fuzzy sets describing time

4 BCO Applications

4.1 Solving the Traveling Salesman Problem by BCO

The main goal of Lučić and Teodorović [5], [6], [7], [8] research was not to develop a new heuristic algorithm for the traveling salesman problem but to explore possible applications of Swarm Intelligence (particularly collective bee intelligence) in solving complex engineering and control problems. The traveling salesman problem is only an illustrative example, which shows the characteristics of the proposed concept. Lučić and Teodorović [5], [6], [7], [8] tested the Bee Colony Optimization approach on a large number of numerical examples. The benchmark problems were taken from the following Internet address: http://www.iwr.uni- heidelberg.de/iwr/comopt/software/TSPLIB95/tsp/. The following problems were considered: Eil51.tsp, Berlin52.tsp, St70.tsp, Pr76.tsp, Kroa100.tsp and a280.tsp. All tests were run on an IBM compatible PC with PIII processor (533MHz). The results obtained are given in Table 2.

Table 2. TSP benchmark problems: The results obtained by the BCO algorithm

Problem name	Optimal value (O)	The best value obtained by the BCO (B)	$\frac{(B-O)}{O}$ (%)	CPU (sec)
Eil51	428.87	428.87	0	29
Berlin52	7544.366	7544.366	0	0
St70	677.11	677.11	0	7
Pr76	108159	108159	0	2
Kroa100	21285.4	21285.4	0	10
Eil101	640.21	640.21	0	61
Tsp225	3859	3899.9	1.06%	11651
A280	2586.77	2608.33	0.83%	6270
Pcb442	50783.55	51366.04	1.15%	4384
Pr1002	259066.6	267340.7	3.19%	28101

The solution of the benchmark problem Tsp.225 obtained by the BCO algorithm is shown in Figure 5.

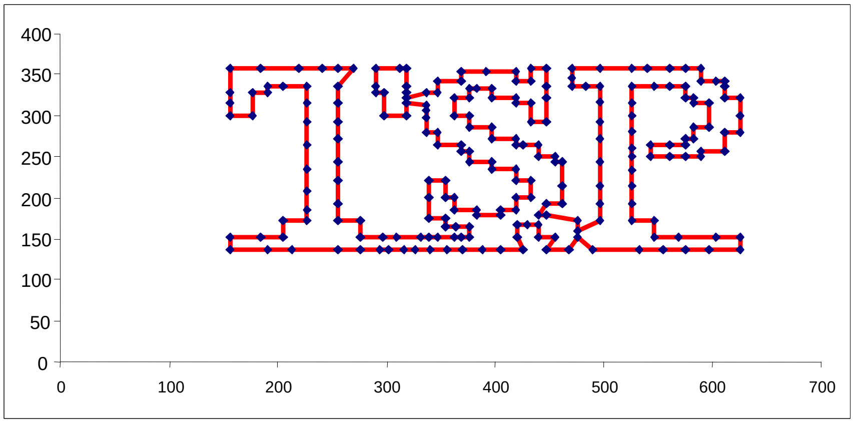


Fig. 5. Solution of the benchmark problem Tsp.225 obtained by the BCO algorithm

We can see from the Table 2 that the proposed BCO produced results of a very high quality. The BCO was able to obtain the objective function values that are very close to the optimal values of the objective function. The times required to find the best solutions by the BCO are very low. In other words, the BCO was able to produce “very good” solutions in a “reasonable amount” of computer time.

4.2. Solving the Ride-Matching Problem by the BCO

Urban road networks in many countries are severely congested, resulting in increased travel times, increased number of stops, unexpected delays, greater travel cost, inconvenience to drivers and passengers, increased air pollution, noise level and number of traffic accidents. Increasing traffic network capacities by building more roads is enormously costly as well as environmentally destructive. More efficient usage of the existing supply is vital in order to sustain the growing travel demand. Ridesharing is one of the widely used Travel Demand Management (TDM) techniques. Within this concept, two or more persons share vehicle when traveling from few origins to few destinations. The operator of the system must posses the following information regarding trips planned for the next week: (a) Vehicle capacity (2, 3, or 4 persons); (b) Days in the week when person is ready to participate in ride-sharing; (c) Trip origin for every day in a week; (d) Trip destination for every day in a week; (e) Desired departure and/or arrival time for every day in a week. The ride-matching problem considered by Teodorović and Dell’Orco [10], [14] could be defined in the following way: Make routing and scheduling of the vehicles and passengers for the whole week in such a way to minimize the total distance traveled by all participants. Teodorović and Dell’Orco [10], [14] developed BCO based model for the ride-matching problem. The authors tested the proposed model in the case of ridesharing demand from Trani, a small city in the southeastern Italy. They collected the data regarding 97 travelers demanding for ridesharing, and assumed, for sake of simplicity, that the capacity is 4 passengers for all their cars. Changes of the best discovered objective function values are shown in Figure 6.

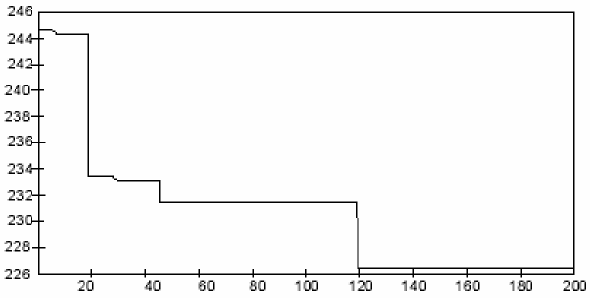


Fig. 6. Changes of the best-discovered objective function values.

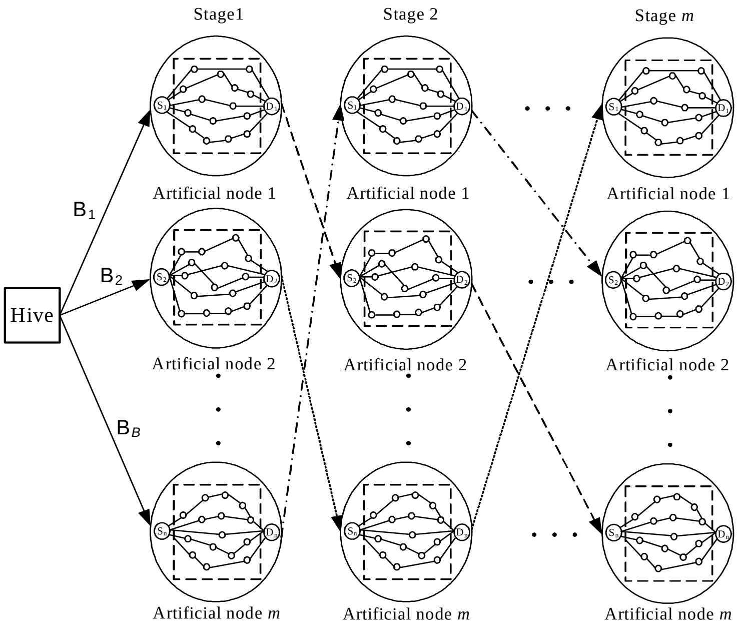
4.3 Routing and wavelength assignment in all-optical networks based on the BCO

The BCO metaheuristic has been successfully tested [12] in the case of the Routing and Wavelength Assignment (RWA) in All-Optical Networks. This problem is, by its nature similar to the traffic assignment problem. The results achieved, as well as experience gained when solving the RWA problem could be used in the future research of the traffic assignment problem.

Let us briefly describe the RWA problem. Every pair of nodes in optical networks is characterized by a number of requested connections. The total number of established connections in the network depends on the routing and wavelength assignment procedure.

Routing and wavelength assignment (RWA) problem in all-optical networks could be defined in the following way: Assign a path through the network and a wavelength on that path for each considered connection between a pair of nodes in such a way to maximize the total number of established connections in the network.

Marković et al. [12] proposed the BCO heuristic algorithm tailored for the RWA problem. They called the proposed algorithm the BCO-RWA algorithm. The authors created the artificial network shown in the Figure 7. The node depicted by the square in the Figure 7 represents hive. At the beginning of the search process all artificial agents are located in the hive. Bees depart from the hive and fly through the artificial network from the left to the right. Bee's trip is divided into stages. Bee chooses to visit one artificial node at every stage. Each stage represents the collection of all considered origin-destination pairs. Each artificial node is comprised of an origin and destination linked by a number of routes. Lightpath is a route chosen by bee agent. Bee agent's entire flight is collection of established lightpaths. The authors determined in advance the number of bees *B* and the number of iterations *I*.



m-total number of requested lightpaths

Fig. 7. Artificial network

During forward pass every bee visits *n* stages (bee tries to establish *n* new lightpaths). In every stage a bee chooses one of the previously not visited artificial nodes. Sequence of the *n* visited artificial nodes generated by the bee represents one partial solution of the problem considered. Bee is not always successful in establishing lightpath when visiting artificial node. Bee's success depends on the wavelengths' availability on the specific links. In this way, generated partial solutions differ among themselves according to the total number of established lightpaths.

After forward pass, bees perform backward pass, i.e. they return to the hive. The number of nodes *n* to be visited during one forward pass is prescribed by the analyst at the beginning of the search process, such that $n<<m$, where *m* is the total number of requested lightpaths.

Probability *p* that specific unvisited artificial node will be chosen by the bee equals $1/n_{tot}$, where n_{tot} is the total number of unvisited artificial nodes. By visiting specific artificial node in the network shown in Figure 7 bees attempt to establish the requested lightpath between one real source-destination node pair in optical network. Let us assume that the specific bee decided to consider the lightpath request between the source node *s* and the destination node *d*. In the next step, it is necessary to choose the route and to assign an available wavelength along the route between these two real nodes. In this paper, we defined for every node pair (*s*, *d*), the subset R^{sd} of allowed routes that could be used when establishing the lightpath. We defined these subsets by using the *k* shortest path algorithm. We calculated for every of the *k* alternative routes the bee's utility when choosing the considered route. The shorter the chosen route and the higher the number of available wavelengths along the route, the higher the bee's utilities are. We define the bee's utilities V_r^{sd} when choosing the route *r* between the node pair (*s*, *d*) in the following way:

$$V_r^{s,d} = \left\{ a \frac{1}{h_r - h_{r\min} + 1} + (1-a) \frac{W_r}{W_{\max}} \right. \quad (1)$$

where:

r – the route ordinary number for a node pair, *r* =1, 2,..., *k*, $r \in \{R^{sd}\}$

h_r – the route length expressed in the number of physical hops,

$h_{r\min}$ – the length of the shortest route *r*,

W_r – the number of available wavelengths along the route *r*,

$W_{\max} = \max\limits_{r \in R^{sd}} \{W_r\}$ – the maximum number of available wavelengths among all the routes $r \in R^{sd}$

a – weight (importance of the criteria), $0 \le a \le 1$

Bees decide to choose a physical route in optical network in a random manner. Inspired by the Logit model, Marković et al. [12] assumed that the probability p_r^{sd} of choosing route *r* in the case of origin-destination pair (*s*, *d*) equals:

$$p_r^{sd} = \begin{cases} \dfrac{eV_r^{sd}}{\sum\limits_{i=1}^{|R^{sd}|} eV_i^{sd}} & \forall r \in R^{sd} \quad and \quad W_r \rangle\, 0 \\ 0 & \forall r \in R^{sd} \quad and \quad W_r = 0 \end{cases} \quad (2)$$

where $|R^{sd}|$ is the total number of available routes between pair of nodes (*s*, *d*). The route *r* is availlable if there is at least one available wavelength on all links that belong to the route *r*.

In the hive every bee makes the decision about abandoning the created partial solution or expanding it in the next forward pass. The authors assumed that every bee can obtain the information about partial solution quality created by every other bee. They calculated the probability that the bee *b* will at beginning of the *u* + 1 forward pass use the same partial tour that is defined in forward pass *u* in the following way:

$$p_b = e^{-\frac{C_{\max} - C_b}{u}} \tag{3}$$

where:

C_b - the total number of established lightpaths from the beginning of the search process by the *b*-th bee

C_{max} - the maximal number of established lightpaths from the beginning of the search process by any bee

u - ordinary number of forward pass, *u = 1,2,...,U*

Marković et al. [12] calculated the probability p_P that the *P*-th advertised partial solution will be chosen by any of the uncomitted follower using the following relation:

$$p_P = \frac{e^{C_P}}{\sum_{p=1}^{P} e^{C_p}} \tag{4}$$

where C_P is the total number of the established lightpaths in the case of the *P*-th advertised partial solution.

The BCO-RWA algorithm was tested on a few numerical examples. The authors formulated corresponding Integer Linear Program (ILP) and discovered optimal solutions for the considered examples. In the next step, they compared the BCO-RWA results with the optimal solution. The comparison for the considered network is shown in the Table 3.

Table 3. The results comparison

Total number of requested light-paths	Number of wave-lengths	Number of established lightpaths		CPU time [s]		Relative error [%]
		ILP	BCO-RWA	ILP	BCO-RWA	
28	1	14	14	4	4.33	0
	2	23	23	94	4.58	0
	3	27	27	251	4.68	0
	4	28	28	313	4.66	0
31	1	15	14	4	4.73	6.67
	2	25	25	83	5.00	0
	3	30	30	235	5.19	0

	4	31	31	1410	5.21	0
34	1	15	14	14	5.19	6.67
	2	27	26	148	5.50	3.70
	3	33	33	216	5.64	0
	4	34	34	906	5.64	0
36	1	16	15	23	5.64	6.25
	2	27	26	325	6.09	3.70
	3	34	34	788	6.11	0
	4	36	36	1484	6.13	0
38	1	17	16	16	5.67	5.88
	2	28	27	247	6.09	3.57
	3	35	35	261	6.23	0
	4	38	38	1773	6.33	0
40	1	17	16	31	6.00	5.88
	2	28	27	491	6.28	3.57
	3	35	35	429	6.61	0
	4	40	40	1346	6.67	0

We can see from the Table 3 that the proposed BCO-RWA algorithm has been able to produce optimal, or a near-optimal solutions in a reasonable amount of computer time.

4.4 Scheduling Independent Tasks by the BCO

Davidović et al. [17] studied the problem of static scheduling of independent tasks on homogeneous multiprocessor systems. The studied problem is solved by the BCO. The authors considered the following problem. Let $T = \{1,2,...,n\}$ be a given set of independent tasks, and $P = \{1,2,...m\}$ set of identical processors. The processing time of task i (i = 1,2,…,n) is denoted by l_i. All tasks are mutually independent and each task can be scheduled to any processor. All given tasks should be executed. Task should be scheduled to exactly one processor and processors can execute one task at a time. The goal is to find scheduling of tasks to processors in such a way as to minimize the completion time of all tasks (the so called *makespan*).

The considered scheduling problem could be graphically represented by Gantt diagram (Figure 8).

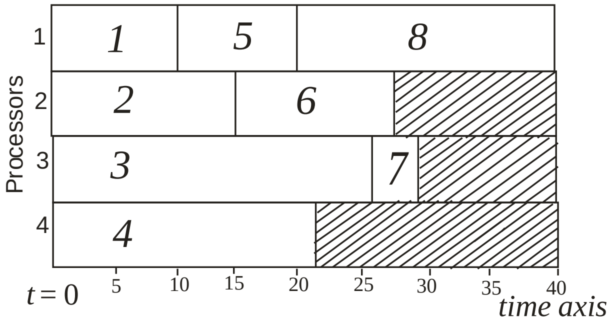


Fig. 8. Gantt diagram: Scheduled tasks to processors

The horizontal axis in the diagram represents time. The rectangulars in the Gantt diagram represent tasks. The starting time of a task is determined by the completion times of all tasks already scheduled to the same processor. The total completion time for the schedule shown in the Figure 8 equals 40 time units (the completion time of task 8 scheduled to processor 1). Any schedule that has completion time less than 40 time units is considered better. The goal is to discover the schedule of tasks to processors that has shortest completion time.

Let us briefly describe the main results achieved by Davidović et al. [17]. The authors decomposed considered problem in stages. The first task to be chosen represents the first stage, the second task to be chosen represents the second stage, the third task represents the third stage, etc. They denoted by p_i the probability that specific bee chooses task *i*. The probability of choosing task *i* equals:

$$p_i = \frac{l_i}{\sum_{k=1}^{K} l_k} \qquad i=1,2,\ldots,n \tag{5}$$

where:

l_i - processing time of the *i*-th task;

K - the number of "free" tasks (not previously chosen).

Obviously, tasks with a higher processing time have a higher chance to be chosen. Using relation (6) and a random number generator, we schedule task to bee. Let us denote by p_j the probability that specific bee chooses processor *j*. Davidović et al.[17] assumed that the probability of choosing processor *j* equals:

$$p_j = \frac{V_j}{\sum_{k=1}^{m} V_k} \qquad j=1,2,\ldots,m \tag{6}$$

where:

$$V_j = \frac{\max F - F_j}{\max F - \min F} \qquad j=1,2,\ldots,m \tag{7}$$

F_j - running time of processor j based on tasks already scheduled to it;

max F- maximum over all processors running times;

min F - minimum over all processors running times.

Processors with a lower value of the running times have a higher chance to be chosen. Using relation (6) and a random number generator, we schedul processor to previously chosen task. In total, B bees choose $B*NC$ task-processor pairs after the first forward pass. After scheduling tasks to processors we update processors' running times.

All bees return to the hive after generating the partial solutions. All these solutions are then evaluated by all bees. (The latest time point of finishing the last task at any processor characterizes every generated partial solution). Let us denote by C_b (b=1, 2,..., B) the latest time point of finishing the last task at any processor in the partial solution generated by the b-th bee. We denote by O_b normalized value of the time point C_b, i.e.:

$$O_b = \frac{C_{\max} - C_b}{C_{\max} - C_{\min}}, \quad b = 1,2,..., B \tag{8}$$

where C_{min} and C_{max} are respectively the smallest and the largest time point among all time points produced by all bees. The probability that b-th bee (at the beginning of the new forward pass) is loyal to the previously discovered partial solution is calculated in this paper in the following way:

$$p_b^{u+1} = e^{-\frac{O_{\max} - O_b}{u}} \qquad b = 1,2,...B \tag{9}$$

where u is the ordinary number of the forward pass.

Within the dance area the bee-dancers (recruiters) "advertise" different partial solutions. We have assumed in this paper that the probability the recruiter b's partial solution will be chosen by any uncommitted bee equals:

$$p_b = \frac{O_b}{\sum_{k=1}^{R} O_k} \qquad (10)$$

where:

O_k - objective function value of the *k*-th advertised solution;

R - the number of recruiters.

Using relation (10) and a random number generator, every uncommitted follower join one bee dancer (recruiter). Recruiters fly together with a recruted nestmates in the next forward pass along the path discovered by the recruiter. At the end of this path all bees are free to independently search the solution space.

The proposed algorithm was tested on a various test problems. We denote respectively by *NT*, and *NP* the number of tasks and the number of processors. The problem parameters range from instances with *NT* = 10 up to the instances with *NT* = 50. In all cases we set *NP* = 4. The algorithm parameters whose values need to be set prior the algorithm execution are as follows: The total number of bees *B* engaged in the search process was equal to 10; The number of moves (generated task-processor pairs) *NC* during one forward pass was equal to 1; The number of iterations *I* within one run was equal to 100.

The authors compared the obtained BCO results with the optimal solution. The comparison results are shown in the Table 1. Within the table, BCO represents objective function value obtained by the BCO algorithm; OPT denotes the optimal makespan obtained by using ILOG AMPL and CPLEX 11.2 optimization software; CPU time shows the time required by BCO algorithm to obtain the optimal solution; I stands for the number of iteration until optimal solution was reached.

Table 4. The comparison of the BCO results with objective function optimal values for medium problems (*NT*=50, *NP*=4)

Test problem	BCO	OPT	BCO time (sec)	I
It50 70	212	212	1.0776	5
It50 80	196	196	0.5637	1
It50 80_1	234	234	1.1848	4
It50 80_2	337	337	1.7368	8
It50 80_3	216	216	0.8077	3
It50 80_4	276	276	0.7472	2
It50 80_5	128	128	0.8814	4
It50 80_6	167	167	1.8514	8

The BCO algorithm was able to obtain the optimal value of objective function in all test problems. The CPU times required to find the best solutions by the BCO are acceptable. All the tests were performed on AMD Sempron (tm) Processor with 1.60 GHz and 512 MB of RAM under Windows OS.

5. Conclusion

The Bee Colony Optimization is the youngest Swarm Intelligence technique. It is a meta-heuristic motivated by foraging behavior of honeybees. It represents general algorithmic framework that can be applied to various optimization problems in management, engineering, and control. This general algorithmic framework should be always *tailored* for a specific problem. The BCO approach is based on the concept of *cooperation*. Cooperation enables artificial bees to be more efficient and to achieve goals they could not achieve individually. The BCO has the capability, through the information exchange and recruiting process, to intensify the search in the promising regions of the solution space. When it is necessary, the BCO can also diversify the search. The BCO has not been widely used for solving real-life problems. There are no theoretical results in this moment that could support BCO concepts. Usually, development of various Swarm Intelligence approaches was based on experimental work in initial stage. Good experimental results should motivate researchers to try to produce some theoretical results in future research. Preliminary results have shown that the development of new models based on BCO principles (autonomy, distributed functioning, self-organizing) could probably contribute significantly to solving complex engineering, management, and control problems. In years to come, one could expect more BCO based models.

The most important aspect of future research is the mathematical justification of the BCO technique. The other interesting aspects of the future research could be bees' homogeneity (homogenous vs. heterogeneous artificial bees), various information sharing mechanisms, and various collaboration mechanisms.

Acknowledgment

This research is partially supported by the Ministry of Science of Serbia.

References

1. Beni, G.: The concept of cellular robotic system. In: Proceedings of the 1988 IEEE International Symposium on Intelligent Control. IEEE Computer Society Press, Los Alamitos, CA (1988) 57–62

2. Beni, G., Wang, J.: Swarm intelligence. In: Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan. RSJ Press, Tokyo (1989) 425–428

3. Beni, G., Hackwood, S.: Stationary waves in cyclic swarms. In: Proceedings of the 1992 International Symposium on Intelligent Control. IEEE Computer Society Press, Los Alamitos, CA (1992) 234–242

4. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence. Oxford University Press, Oxford (1997)

5. Lučić, P., Teodorović, D.: Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, Portugal (2001) 441–445

6. Lučić, P., Teodorović, D.: Transportation modeling: an artificial life approach. In: Proceedings of the 14th IEEE ‘‘International Conference on Tools with Artificial Intelligence”, Washington, DC (2002). 216–223

7. Lučić, P., Teodorović, D.: Computing with bees: attacking complex transportation engineering problems. Int. J. Artif. Intell. T. 12 (2003a) 375–394

8. Lučić, P., Teodorović, D.: Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach. In: Verdegay, J.L. (Ed.): Fuzzy Sets in Optimization. Springer-Verlag, Heidelberg, Berlin (2003b) 67–82

9. Teodorović, D.: Transport Modeling by Multi-Agent Systems: A Swarm Intelligence Approach. Transport. Plan. Techn. 26 (2003b) 289–312

10. Teodorović, D., Dell’Orco, M.: Bee colony optimization – a cooperative learning approach to complex transportation problems. In: Advanced OR and AI Methods in Transportation. Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland (2005) 51–60

11. Teodorović, D., Lučić, P., Marković, G., Dell’ Orco, M.: Bee colony optimization: principles and applications. In: Reljin, B., Stanković , S. (Eds.): Proceedings of the Eight Seminar on Neural Network Applications in Electrical Engineering – NEUREL 2006, University of Belgrade, Belgrade (2006) 151–156

12. Marković, G., Teodorović, D., Aćimovic´ Raspopović, V.: Routing and wavelength assignment in all-optical networks based on the bee colony optimization. AI Commun. 20 (2007) 273–285

13. Teodorović, D., Šelmić, M.: The BCO Algorithm For The *p* Median Problem. In: Proceedings of the XXXIV Serbian Operations Research Conferece. Zlatibor, Serbia (2007) - (in Serbian).

14.Teodorović, D., Dell’Orco, M.: Mitigating traffic congestion: solving the ride-matching problem by bee colony optimization. Transport. Plan. Techn. 31, (2008) 135–152

15. Teodorović, D.: Swarm Intelligence Systems for Transportation Engineering: Principles and Applications. Transp. Res. Pt. C-Emerg. Technol. 16 (2008) 651-782

16. Todorović, N. Petrović, S., Teodorović, D.: Bee Colony Optimization for Nurse Rostering (submitted).

17. Davidović, T., Šelmić, M., Teodoroivić, D.: Scheduling Independent Tasks: Bee Colony Optimization Approach (submitted)

18. Camazine S, Sneyd J.: A Model of Collective Nectar Source by Honey Bees: Self-organization Through Simple Rules. J. Theor. Biol. 149 (1991) 547- 571

19. Yonezawa, Y., Kikuchi, T.: Ecological algorithm for optimal ordering used by collective Honey bee behavior. In: Proceedings of the Seventh International Symposium on Micro Machine and Humane Science. Nagoya, Japan (1996) 249- 255

20. Sato, T., Hagiwara, M.: Bee System: Finding Solution by a Concentrated Search. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics "Computational Cybernetics and Simulation". Orlando, FL, USA (1997) 3954-3959

21. Abbass, H.A.: MBO: marriage in honey bees optimization-a Haplometrosis polygynous swarming approach. In: Proceedings of the Congress on Evolutionary Computation. Seoul, South Korea (2001) 207- 214

22. Wedde, H.F., Farooq, M., Zhang, Y. BeeHive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. In: Ant Colony Optimization and Swarm Intelligence. LNCS 3172, Springer Verlag, Berlin (2004) 83–94

23. Karaboga, D.: An idea based on honey bee swarm for numerical optimization (Technical Report-Tr06, October, 2005), Erciyes University, Engineering Faculty Computer Engineering Department Kayseri/Türkiye (2005).

24. Karaboga D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global. Optim. 39 (2007) 459-471

25. Karaboga, D., Basturk Akay, B., Ozturk, C.: Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. In: LNCS: Modeling Decisions for Artificial Intelligence, Springer-Verlag, Berlin Heidelberg (2007) 4617/2007, 318-319.

26. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft. Comput. 8 (2008) 687–697

27. Drias, H., Sadeg, S. and Yahi, S.: Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem. In : Computational Intelligence and Bioinspired Systems, Lecture Notes in Computer Science 3512, Springer Berin/Heilderberg (2005) 318-325

28. Yang, X-S.: Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms. In: J. Mira and J.R. Alvarez (Eds.): IWINAC 2005, Lecture Notes in Computer Science 3562, Springer-Verlag Berlin Heidelberg (2005) 317-323

29. Benatchba, K., Admane, L., Koudil, M. Using Bees to Solve a Data-Mining Problem Expressed as a Max-Sat One. In: J. Mira and J.R. ´Alvarez (Eds.): IWINAC 2005, LNCS, 3562, Springer-Verlag Berlin Heidelberg (2005) 212–220

30. Chong, C.S., Low, M.Y.H., Sivakumar, A.I., Gay, K.L.: A Bee Colony Optimization Algorithm to Job Shop Scheduling Simulation. In: L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, (Eds.): Proceedings of the Winter Conference, Washington, DC (2006) 1954 - 1961

31. Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Zaidi M.: The Bees Algorithm - A Novel Tool for Complex Optimisation Problems. In: Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006), Elsevier, Cardiff, (2006) 454-459
32. Pham, D.T, Soroka, A.J., Ghanbarzadeh, A., Koc, E.: Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm. In: Proceedings of the IEEE International Conference on Industrial Informatics, Singapore (2006) 1346-1351
33. Navrat, P.: Bee Hive Metaphor for Web Search. In: B. Rachev, A. Smrikarov (Eds.): Proceedings of the International Conference on Computer Systems and Technologies - CompSysTech 2006, , Veliko Turnovo, Bulgaria, (2006), IIIA.12- 1-7
34. Wedde, H.F., Timm, C., Farooq, M.: BeeHiveAIS: A Simple, Efficient, Scalable and Secure Routing Framework Inspired by Artificial Immune Systems. In: T.P. Runarsson et al. (Eds.): LNCS 4193, Springer-Verlag, Berlin Heidelberg (2006) 623–632
35. Yang, C., Chen, J., Tu, X.: Algorithm of Fast Marriage in Honey Bees Optimization and Convergence Analysis. In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China (2007) 1794-1799
36. Koudil, M., Benatchba, K., Tarabetand, A., El Batoul Sahraoui: Using artificial bees to solve partitioning and scheduling problems in codesign. Appl. Math. Comput. 186 (2007) 1710-1722
37. Quijano, N., Passino, K.M.: Honey Bee Social Foraging Algorithms for Resource Allocation, Part I: Algorithm and Theory. In: Proceedings of the 2007 American Control Conference, New York (2007a) 3383-3388.
38. Quijano, N., Passino, K.M.: Honey Bee Social Foraging Algorithms for Resource Allocation, Part II: Application. In: Proceedings of the 2007 American Control Conference, New York (2007b) 3389-3394
39. Wedde, H.F., Lehnhoff, S., van Bonn, B., Bay, Z., Becker, S., Böttcher, S., Brunner, C., Büscher, A., Fürst, T., Lazarescu, M., Rotaru, E., Senge, S., Steinbach, B. Yilmaz, F., Zimmermann, T.: A Novel Class of Multi-Agent Algorithms for Highly Dynamic Transport Planning Inspired by Honey Bee Behavior. In: Proceedings of the 12th IEEE International Conference on Factory Automation, Patras, Greece (2007) 1157-1164
40. Afshar, A., Bozorg Haddada, O, Marin, M.A., Adams, B.J.: Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation. J. Frank. Instit. 344 (2007) 452–462
41. Baykasoglu, A., Özbakýr, L., Tapkan, P.: Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem. In: Eds. Felix T. S. Chan and Manoj Kumar Tiwari: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Itech Education and Publishing, Vienna, Austria (2007) 113-143
42. Fathian, M., Amiri, B., Maroosi, B.: A honeybee-mating approach for cluster analysis. Int. J. Adv. Manuf. Technol. 38 (2008) 809–821
43. Pham, D.T., Haj Darwish, A., Eldukhr, E.E.: Optimisation of a fuzzy logic controller using the Bees Algorithm. Int. J., Comp. Aid. Eng. Tech. 1 (2009) 250 - 264