"Mircea cel Batran" Naval Academy Scientific Bulletin, Volume XVIII – 2015 – Issue 2
Published by "Mircea cel Batran" Naval Academy Press, Constanta, Romania // The journal is indexed in:
PROQUEST SciTech Journals, PROQUEST Engineering Journals, PROQUEST Illustrata: Technology, PROQUEST Technology Journals, PROQUEST Military Collection PROQUEST Advanced Technologies & Aerospace

BEE COLONY OPTIMIZATION: RECENT DEVELOPMENTS AND APPLICATIONS

T. DAVIDOVIĆ[1]

[1] Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia

Abstract: *An extensive survey of the Bee Colony Optimization (BCO) algorithm, proposed for the first time in 2001 by Lučić and Teodorović, will be presented. BCO and its numerous variants belong to a class of nature-inspired meta–heuristic methods, based on the foraging habits of honeybees. It is a simple, easily understandable and implementable technique that has been successfully applied to many optimization problems. A detailed description of the BCO algorithm and its modifications, including the strategies for BCO parallelization and hybridization will be provided. The preliminary results regarding its convergence will also be discussed. In the second part of the talk, the successful applications of BCO to various hard combinatorial optimization problems, mostly in transportation, location and scheduling fields will be summarized, together with some recent applications in the continuous optimization field. This paper is an extension of two survey papers, co-authored by Professor Dušan Teodorović and Dr. Milica Šelmić, recently published in Yugolsav Journal of Operations Research (YUJOR).*

Keywords: *Meta-heuristic methods, nature-inspired computing, optimization problems, swarm intelligence.*

I. INTRODUCTION

Natural processes have been the inspiration for designing optimization algorithms for almost forty years. Starting with simulated annealing (SA) [1:pp.1-39], genetic and other evolutionary algorithms (GA, EA) [1: pp.109-139,2], through ant colony optimization (ACO) [1:pp.227-263], particle swarm optimization (PSO) [3] and other Swarm Intelligence (SI) [4, 5] methods, till the artificial immune systems (AIS) [1:pp.421-448] and many more, we now have thousands of various nature-inspired computational intelligence techniques [6-10].

The Bee Colony Optimization (BCO) is a meta-heuristic inspired by foraging behavior of honeybees. It is one of the first algorithms that uses basic principles of collective bee intelligence in solving combinatorial optimization problems. BCO was initially proposed for dealing with the well known hard combinatorial optimization problems: travelling salesman [11-13] and vehicle routing [14]. BCO is a stochastic, random-search population-based technique. It was motivated by the analogy found between the natural behavior of bees searching for food and the behavior of optimization algorithms searching for an optimum in combinatorial optimization problems. The main idea is to build the multi agent system (a colony of artificial bees) able to efficiently solve hard optimization problems. Artificial bees investigate through the search space looking for feasible solutions. In order to increase the quality of produced solutions, autonomous artificial bees collaborate and exchange information. Sharing the available information and using collective knowledge, artificial bees concentrate on more promising areas, and slowly abandon solutions from those less promising. Step by step, artificial bees collectively generate and/or modify their solutions. BCO performs its search in iterations until some predefined stopping criterion is satisfied.

This paper is an extension of recently published survey papers [15,16]. It presents in details the BCO algorithm, its variations and modifications, as well as the classification and analysis of its recent applications. In the recent literature BCO is successfully used to model complex science and engineering optimization problems. Several PhD thesis [17-20] were defended considering the development and/or applications of BCO. Besides the successful applications reported by Teodorović and co-authors [11-14,21-32], some other researchers also used BCO meta-heuristic [33-42]. An extensive bibliography is provided and discussed later.

The paper is organized as follows. Section 2 contains a brief description of the basic BCO algorithm. Various modifications and recent developments are described in Section 3. Section 4 is devoted to an application survey, while the last section contains some conclusions and directions for further exploration of the BCO meta-heuristic.

II. OVERVIEW OF THE BCO ALGORITHM

A. Biological Background

Swarm behaviour (fish schools, flocks of birds, herds of land animals, insects' communities, etc.) is based on the *biological needs* of individuals to stay together. In such a way, individuals increase the probability to stay alive, since predators usually attack only isolated individuals. Flocks of birds, herds of animals, and fish schools are characterized by collective movement. Colonies of various social insects (bees, wasps, ants, termites) are also characterized by swarm behaviour. Swarm behaviour is primarily characterized by autonomy, distributed functioning and self-organizing. The communication systems between individual insects contribute to the *collective intelligence* pattern named ''Swarm Intelligence" in [4,5].

Swarm Intelligence represents the branch of the Artificial Intelligence that investigates individuals' actions in different decentralized systems. These decentralized systems (Multi Agent Systems) are composed of physical individuals (robots, for example) or "virtual" (artificial) ones that communicate, cooperate, collaborate, exchange information and knowledge and perform some tasks in their environment. When designing Swarm Intelligence models, researchers use some principles of the natural swarm intelligence. The development of artificial systems does not usually involve the entire imitation of natural systems, but explores and adapts them while searching for ideas and models.

Bees in nature [43] look for a food by exploring the fields in the neighborhood of their hive. They collect and accumulate food for later use by other bees. Typically, in the initial step, some scouts search the region. Completing the search, scout bees return to the hive and inform their hive-mates about the locations, quantity and quality of the available food sources in the areas they have examined. In the case they have discovered nectar in the previously investigated locations, scout bees dance in the so-called "dance floor area" of the hive, in an attempt to "advertise" food locations and encourage the remaining members of the colony to follow their lead. The information about the food quantity is presented using a ritual called a "waggle dance". If a bee decides to leave the hive and collect the nectar, it will follow one of the dancing scout bees to the previously discovered patch of flowers. Upon arrival, the foraging bee takes a load of nectar and returns to the hive relinquishing the nectar to a food store. Several scenarios are then possible for a foraging bee: (1) it can abandon the food location and return to its role of an uncommitted follower; (2) it can continue with the foraging behavior at the discovered nectar source, without recruiting the rest of the colony; (3) it can try to recruit its hive-mates with the dance ritual before returning to the food location. The bee opts for one of the above alternatives. As several bees may be attempting to recruit their hive-mates on the dance floor area at the same time, it is unclear how an uncommitted bee decides which recruiter to follow. The only obvious fact is that "the loyalty and recruitment among bees are always a function of the quantity and quality of the food source" [43]. The described process continues repeatedly, while the bees at a hive accumulate nectar and explore new areas with a potential food sources.

B. ***The BCO Algorithm***

BCO is a population based algorithm [15]: population of artificial bees searches for the optimal solution of a given combinatorial or continuous optimization problem. Every artificial bee generates one solution to the problem. The algorithm consists of two alternating phases: *forward pass* and *backward pass*. During each forward pass, all bees are exploring the search space. Each bee applies a predefined number of moves, which construct and/or improve the solution, yielding a new solution.

Having obtained new partial/complete solutions, the bees start executing a second phase, the so-called backward pass. During the backward pass, all bees share information about their solutions. In nature, bees would perform a dancing ritual, which would inform other bees about the amount of food they have found, and the proximity of the food source to the hive. In the search algorithm, the quality of each solution is defined as the current value of the objective function.

Having all solutions evaluated, each bee decides with a certain probability whether it will stay *loyal* to its solution or not. This decision depends on the quality of its own solution related to all other existing solutions. The probability that *b*-th bee (at the beginning of the new forward pass) is loyal to its previously generated partial/complete solution is expressed as follows:

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

where:

O_b - denotes the normalized value for the objective function of partial/complete solution created by the *b*-th bee;

O_{max}- represents the maximum over all normalized values of partial/complete solutions to be compared (it actually corresponds to the best solution discovered by bees in the current forward pass);

u - counts the forward passes, takes values 1,2,..., *NC*.

The normalization is performed in two ways, depending on whether a minimization or maximization of the objective function is required. If C_b (*b* = 1, 2,..., *B*) denotes the objective function value of *b*-th bee partial/complete solution, normalized value of the C_b in the case of minimization is calculated as follows:

$$O_b = \frac{C_{\max} - C_b}{C_{\max} - C_{\min}}, \quad b = 1,2,...,B \qquad (2)$$

where C_{min} and C_{max} are the values of partial/complete solutions related to minimal and maximal objective function value, respectively, obtained by all engaged bees. From equation (2) it could be seen that if *b*-th bee partial/complete solution is closer to maximal value of all obtained solutions, C_{max}, than its normalized value, O_b, is smaller and vice versa.

In the case of maximization criterion, normalized value of C_b is calculated as follows:

$$O_b = \frac{C_b - C_{\min}}{C_{\max} - C_{\min}}, \quad b = 1,2,...,B. \qquad (3)$$

From above equation (3) it is obvious that if the value of the partial/complete solution, C_b, is higher, then its normalized value, O_b, is larger, and vice versa.
Using equation (1) and a random number generator, each artificial bee decides whether to become uncommitted follower or to continue exploring its own solution. If chosen random number is smaller than the calculated probability, then the bee stays loyal to its own solution. Otherwise, if the random number is greater than the probability, p_b^{u+1}, the bee becomes uncommitted.
Some other probability functions are examined in [20, 44] and they will be discussed later.
It is obvious that bees with better solutions have more chances to keep and advertise their solutions. Contrary to the bees in nature, artificial bees that are loyal to their partial/complete solutions are at the same time the *recruiters*, i.e., their solutions are advertised and would be considered by other bees. Once the solution is abandoned, the corresponding bee becomes *uncommitted* and has to select one of the advertised solutions. This selection is taken with a probability, such that better advertised solutions have greater opportunities to be chosen for further exploration.
For each uncommitted bee it is decided which recruiter it will follow, taking into account the quality of all advertised solutions. The probability that b's partial/complete solution would be chosen by any uncommitted bee equals:

$$p_b = \frac{O_b}{\sum_{k=1}^{R} O_k} \qquad b = 1,2,...,R \qquad (4)$$

where O_k represents the normalized value for the objective function of the k-th advertised solution and R denotes the number of recruiters. Using equation (4) and a random number generator, each uncommitted follower joins one recruiter through a roulette wheel.
The roulette wheel is a well-known model of choice. The main inspiration for its development came from a game-gambling roulette. Any solution can be chosen, and the probability of its selection (the size of a particular slot on the roulette wheel) depends on the quality of that solution, i.e., the value of the objective function.
In the practice, the size of the slot on the roulette wheel associated to each solution is determined by the ratio of the corresponding normalized objective function value and the sum of the normalized objective function values for all advertised solutions. On one hand, a solution with better objective function value has a higher chance to be selected. On the other hand, there is still a possibility that it will be eliminated from further search process.
The two phases of the search algorithm, namely the forward and backward pass (Fig. 1) alternate *NC* times, i.e., until each bee completes the generation of its solution or performs *NC* solution modifications. Parameter *NC* is used to define the frequency of information exchange between bees. When *NC* steps are completed, the best among all *B* solutions is determined. It is then used to update global best solution, and an iteration of BCO is accomplished. At this point, all *B* solutions are deleted and the new iteration can start. The BCO algorithm runs iteration by iteration until a stopping condition is met. The possible stopping condition could be, for example, the maximum number of iterations, the maximum number of iterations without the improvement of the objective function value, maximum allowed CPU time, etc. At the end, the best found solution (the so called global best) is reported as the final one.
The BCO algorithm parameters, whose values need to be set prior the algorithm execution, are:
B— the number of bees involved in the search and
NC— the number of forward (backward) passes constituting a single BCO iteration.

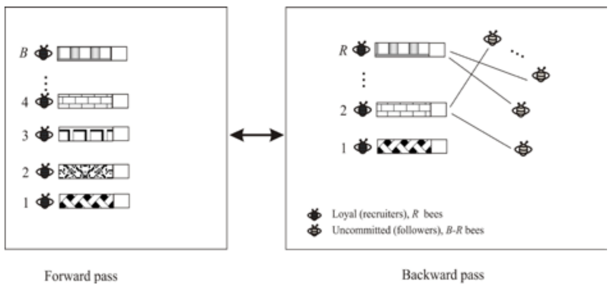

Figure 1. Main steps of the BCO algorithm

The pseudo-code of the BCO algorithm could be described in the following way:

Do
1. Initialization: a(n) (empty) solution is assigned to each bee.
2. For (i = 0; i < *NC*; i ++)

//forward pass
(a) For (b = 0; b < B; b++)
i) Evaluate possible moves;
ii) Choose one move using the roulette wheel.

//backward pass
(b) For (b = 0; b < B ; b++)
Evaluate the partial/complete solution for bee b;
(c) For (b = 0; b < B; b++)

Loyalty decision using the roulette wheel for bee b;
(d) For (b = 0; $b < B$; b++)
If (b is uncommitted),
choose a recruiter by the roulette wheel.
3. Evaluate all solutions and find the best one.
while stopping criteria is not satisfied.

Steps (a) and (b) are problem dependent and should be resolved in each particular implementation of the BCO algorithm. On the other hand, there are formulae specifying steps (c), loyalty decision, and (d), recruiting process that are given by the equations (1) and (4), respectively.

C. BCO Modifications

The extensive application of BCO to various and difficult optimization problems requires its adaptation to problem characteristics. Therefore, each implementation represents actually the development and modification of the original BCO algorithm. The first versions of BCO [11-14] were constructive and had more similarities with the behavior of bees in the nature, than the recent variants of the algorithm. The main difference between these versions is in the fact that hive had an important role in the first version. The hive had specified location (for example the initial node of the search process or the first selected component) although it could also change its position during the search process. Contrary to the newer variants, in the original BCO not all the bees were engaged at the beginning of the search process. The scout bees started the search, and at each stage new bees were join in by the recruiting process. In this initial BCO version, the authors proposed Logit-based model [45] for calculating the probability of choosing next node to be visited, while in recent versions roulette wheel is used for this purpose.

The most remarkable modification of the BCO algorithm is certainly the introduction of solution transformation (improvement) process. During numerous applications of the BCO algorithm, it was observed that the constructive version cannot successfully solve some optimization problems. Therefore, an improvement variant, based on the transformation of complete solutions assigned to bees was proposed. It was used for the first time in solving p-center problem [21], and later on it became the dominant variant used in the literature [25,26,38].

In most of the early applications, the BCO algorithm was constructive [11-14,22-24,27-29,39-42]. For each bee a solution was constructed from scratch, step by step, applying some stochastic, problem specific, heuristic rules. Randomness induced by these stochastic construction processes assured diversity of the search. In each forward pass a number of components was included in the current partial solution by the bees. Decision about loyalty and recruitment were made based on the evaluation of partial solutions and estimation for the quality of potential final solutions. It is obvious that the precision of the estimation process cannot be always controlled well enough. In addition, after the recruitment, a group of bees is assigned the same partial solution and therefore, the diversity of final solutions is reduced.

Within each iteration B solutions were generated and the best of them was used for updating the current global best solution. Sometimes, global knowledge was used to direct the construction towards high quality solutions (possibly better than the current global best solution constructed by the bees so far). Each iteration begun with B empty solutions and resulted in B new solutions among which we searched for the new global best one.

In contrast to the constructive version, a new concept based on improving the complete solution held by each bee was developed. In [21] it was named BCOi. The BCOi algorithm can be described as follows. At the beginning of each iteration, bees are assigned initial complete solutions, and have to transform (modify) them through the forward passes. Initial solutions can be generated randomly [21,25] or selected among already existing solutions (global best known solution [46], final solutions from previous iteration [26,47], good solutions stored in a pool for further exploration [48], etc.)

The BCOi algorithm implementation proposed in [21] consists of the following five steps. The first step, called pre-processing is performed ''off-line''. In this step the input data is transformed in order to reduce the time required for all computations performed online. The second step represents generation of the initial complete solutions. In the third, the most significant step, bees transform current solutions through NC forward passes within the single iteration. This step is the key factor that enables reaching the best possible solution quality. Its main role is to assure different treatment of the same solutions held by different bees. Solution transformation has to be stochastic and without local search embedded in the process. Having in mind that BCOi is a population-based method, local search procedure would be too expensive (time consuming) transformation and that could deteriorate the performance of BCOi search. Randomness is required to ensure that each bee performs different transformation and produce variety of new and potentially better solutions. Steps 4 (solutions comparison mechanism) and 5 (recruitment) are identical to the corresponding steps from the constructive BCO. The last four steps are all executed on-line and have to be implemented efficiently.

The idea of improving complete solutions could be developed in many different ways, and this approach certainly may be very useful for solving difficult optimization problems. It has already been explored in the recent literature for solving some hard optimization problems like network design

[25], vehicle routing problem with time windows [46] and satisfiability problem in the logic with approximate conditional probability [47], while for the berth allocation problem [33] hybrid variant was designed.

New developments of BCO always assume that all the bees are involved in search process. However, contrary to the earlier variants where they perform the same task (construction or improvement), recent implementations show tendency to assign different roles to the bees [25,38,46,48-50]. Therefore, we can distinguish variants with homogeneous and heterogeneous bees. To be heterogeneous, bees are divided into groups, each of them performing different tasks. In some of the implementations, two groups can be distinguished, the one that constructs initial solutions and the other trying to improve them [38]. Some of the variants are designed in such a way that different groups of bees perform different types of transformations [25,46,48-50]. However, the loyalty decision and the recruitment are uniform for all bees.

Asynchronous communication, which appears in nature, has not been explored yet. This would assume that each bee decides whether it will participate in the backward pass or not. It may continue to transform its current solution many times before allowing other bees to "see" it and include in loyalty decision and recruitment processes. In that case, the number of evaluated and compared solutions will vary from one backward pass to another.

Considering loyalty decision, some researchers investigated the probability that a bee stays loyal to its current solution and proposed alternative formulations [20,44]. The initial calculation based on (1) is proposed for constructive variant with two aims. The first is allowing higher probability for bee staying loyal to the better generated partial solution. Since greater O_b value corresponds to a better generated solution, it provides higher probability of the bee staying loyal to the previously discovered solution. The second aim is increasing the influence of already discovered partial solutions as the number of forward passes increases. In other words, at the beginning of the search process bees are ''braver'' when searching the solution space. The more forward passes are made, the less courage they have: as we approach to the end of the search process, the bees are more focused on already known solutions. This is expressed by the term u in the denominator of the exponent in (1).

In [44] three additional formulae for calculating probability that the bee will stay loyal to its current solution are proposed.

$$p2_b^{u+1} = e^{-\frac{O_{\max} - O_b}{\sqrt{u}}}, \qquad b = 1,2,...,B \qquad (5)$$

$$p3_b^{u+1} = e^{-(O_{\max} - O_b)}, \qquad b = 1,2,...,B \qquad (6)$$

$$p4_b^{u+1} = O_b, \qquad b = 1,2,...,B \qquad (7)$$

The experimental evaluation of all four possibilities is performed on the BCOi for *p*-center problem with the conclusion that the simplest one (7) performs the best. The main reasons are the improvement variant of BCO that does not distinguish between the phases of the search and powerful enough solution transformation process that does not need any sophisticated loyalty decision function. The evaluation of other BCO implementations is still in progress.

Various loyalty decision functions were examined also in [20]. In total, ten different functions are proposed and evaluated for the constructive variant of BCO on the scheduling independent tasks to homogeneous processors. However, the most frequently used in the literature are (1) and (6).

*D. **BCO Paralellization***

Swarm intelligence algorithms are generally suitable for parallelization since they are population-based optimization algorithms. Moreover, they are created as the multi-agent systems that operate individually, yet collaboratively. Therefore they provide a good basis for the parallelization on different levels. High-level parallelization assumes a coarse granulation of tasks and can be applied to iterations. Smaller parts of the algorithms usually also contain a lot of independent executions, and are suitable for low-level parallelization. Parallelization strategies for BCO were proposed in [51, 52], and later, systematically reviewed in [53].

The authors of [51,52] addressed the parallelization of BCO for distributed memory multiprocessors systems. They considered the coarse granulation strategy in both synchronous and asynchronous way. Fine-grained parallelization is not suitable for these multi-processors systems, as it was verified in [52]. Three different strategies for parallelization of BCO, two of them synchronous and one asynchronous, were proposed in [51] and are recalled briefly here.

Independent run of several BCO algorithms represents the simplest form of coarse-grained parallelization. All necessary computations are distributed among different processors. The main aim of this strategy was to speed up the search performed by BCO. In [52] this was realized by the reduction of the stopping criterion on each processor. For example, if the stopping criterion is defined as the allowed CPU time (given as a *runtime* value in seconds), the BCO could run in parallel on q processors for *runtime/q* seconds. Similar rule can be introduced in the case when maximum number of iterations is selected as the stopping criterion. In both cases, each processor has to perform independently sequential variant of

BCO, but with the reduced value of the stopping criterion. The BCO parameters (number of bees *B* and number of forward/backward passes *NC*) were the same for all BCO processes executing on different processors in order to assure a load balance between all processors. The BCO algorithms running on different processors were differing in the seeds values. This variant of parallelized BCO was named *distributed BCO* (DBCO). Another coarse grained parallelization strategy proposed in [52] reduces the number of bees on each processor. Namely, if the sequential execution uses *B* bees for the search, parallel variant executing on *q* processors would be using *B*/*q* bees only. Actually, on each processor, a sequential BCO is running with the reduced number of bees. This variant was referred to as BBCO since the bees were distributed among processors and again the BCO parameters had the same values for all BCO processes executing on different processors.

The third variant of the independent BCO execution proposed in [52] was referred to as MBCO and it involved varying values of the BCO parameters and changing stopping criterion in such a way to obtain load balancing between processors. In has been shown that this introduces more diversification in the search process.

Synchronous cooperation of various BCO algorithms is a more sophisticated way to realize coarse-grained parallelization. It assumes a cooperative work of several BCO processes and therefore was named *Cooperative BCO* (CBCO) in [52]. At certain predefined execution steps, all processes are stopped to exchange the relevant data (usually the current best solutions) that are used to guide further search. The data exchange moments are called *communication points* and were determined in two different ways [52]: fixed and processor dependent. In the first case, the best solution was exchanged 10 times during the parallel BCO execution regardless the number of processors engaged. In such a way processors were given more freedom to perform independent part of the search. In the second case, the authors of [52] tested how the increasing in communication frequency when adding new processors influence the search process. The idea was to broadcast the information about the improvement of the current global best solution to all the processors as soon as possible. Since the current global best solution represents the reference point for constructing new solutions, immediate broadcast should guarantee faster convergence of the resulting search. For the definition of communication points in this case in [52] was used the following rule: current global best solution was exchanged each $n_{it}/(10*q)$ iterations where n_{it} represented the maximum allowed number of iterations.

Asynchronous cooperation of BCO algorithms is the most sophisticated parallelization strategy. Its aim is to decrease the communication and synchronization overhead during the cooperative execution of different BCO algorithms. In [52] the asynchronous execution strategy was implemented in two different ways, but under the common name *General BCO* (GBCO). The first implementation involved a centrally coordinated knowledge exchange, while the second one utilized non-centralized parallelism. Each processor executes a particular sequential variant of BCO until some predefined communication condition is satisfied. It than informs others about its search status, collects the current global best, and continues its execution without affecting the execution of other processors. Each processor can individually decide when to send its results and collect the ones arrived from the others. Non-blocking message passing interface and the large enough mailbox buffer provided by the MPI communication library fully support the implementation of this strategy.

The first asynchronous approach proposed in [52] assumed the existence of a *central blackboard* (a kind of global memory) that could be accessed asynchronously by all processors. The communication condition was defined as *the improvement of the current best solution or the execution of 5 iterations without improvement*.

Non-centralized asynchronous parallel BCO execution assumed the existence of several blackboards, and that only a subset of (adjacent) processors may post and access information on the corresponding blackboard. In this case, the communication condition was the execution of a single iteration of the corresponding BCO. After that, a processor is addressing its associated blackboard. In the case that it managed to improve the current best solution from the blackboard, it would post that information on the blackboard and check if there are better solutions already posted there. The best posted solution would be adopted as the new reference point. If the improvement did not occur in the current iteration, the corresponding processor would simply check for a better solution on its associated blackboard.

All the above described strategies were tested in [52] on the constructive BCO for scheduling independent tasks to identical processors. The obtained results show large benefits from, particularly asynchronous, parallel execution of BCO. The first strategy was also used in [19,47] to speedup experimental evaluation.

Shared memory parallelization, under the openMP paradigm, was implemented and analyzed in [20]. The same problem served as benchmark for testing and linear speedup was reported. The only drawbacks of this approach are hardware limitations: the number of processors that have the access to the common memory is usually small. Thus, the promising avenue for further exploration is certainly the hybridization of MPI and openMP.

*E. **Theoretical Evaluation of BCO***

A numerous successful applications of the BCO method illustrated its efficiency in an empirical way. Moreover, there are some recent papers dealing with the empirical evaluation [26] and parameter calibration [44] of BCO. However, given the final solution of the BCO execution, identified as the best solution found before the stopping criterion is fulfilled, we still cannot discuss its quality with respect to (the unknown) optimal solution. Is it the optimal one or, if not, how far is it from the desired optimum? The only thing we can do is to increase the maximum number of iterations and, possibly, obtain a better final solution.

Therefore, the theoretical analysis of meta-heuristics become a very popular research topic. The theoretical (mathematical) foundations of the constructive BCO algorithm are given in [54-56]. In [54] the necessary conditions assuring that an optimal solution can be generated by any bee when the number of iterations is sufficiently large were identified. Then, the so-called *best-so-far* convergence of the BCO algorithm was proven. It was shown that the current best solution converges to one of the optimal solutions, as the number of iterations increases, with the probability one. This type of convergence is quite common and holds even for some simple, single solution based, stochastic search techniques, like e.g., *random walk*. The more sophisticated the so-called *model convergence* of constructive BCO was considered in [55,56]. Model convergence assumes learning from the previous experience and therefore, can be considered only for the variants where iterations are dependent, i.e., there are some global knowledge exchange between iterations. Moreover, iteration-dependent probabilities for selecting components during the forward pass (step 2 (a)(i)) have to satisfy some additional properties that were identified in [57]. The theoretical analysis for the improvement variant of BCO was considered in [20].

*F. **BCO Hybridization***

In order to increase the efficiency of BCO or to apply it to stochastic or multi-criteria optimization problems, its hybridizations with the appropriate techniques are proposed in the literature. An example of this hybridization can be found in [28]: the BCO is combined with compromise programming and applied to multi-criteria optimization problem of locating the inspection stations. Two conflicting criteria should be optimized simultaneously: the number of the inspection stations should be minimized while the risk reduction has to be maximized.

Fuzzy sets and logics [58] are common tool to deal with uncertain data and therefore, are usually used to deal with stochastic optimization. The first combination of BCO and fuzzy logic is reported in [15]. The approximate reasoning based on fuzzy logic has been used to model uncertain demands in nodes when solving vehicle routing problem. It has also applied to model some uncertain quantities for solving Ride-Matching problem in [29,30].

Local search (LS) is not appropriate for systematic use in population based meta-heuristic due to its complexity and therefore, time consuming execution. However, it is commonly combined with these methods to for example, additionally improve the current best solution at the end of execution. The examples of using LS within BCO can be found in [34, 40, 42]. In [37], as a local search procedure Tabu Search (TS) was used.

III. Recent applications of BCO. This section summarizes the applications of the above described BCO method or some of its variations. The applications, summarized in the previous review paper [16], can be classified as follows:

- Routing: the traveling salesman problem [11, 40-42], vehicle routing problem [14], and the routing and wavelength assignment (RWA) in all-optical networks [24].
- Location: the *p*-median problem [31], traffic sensors locations problem on highways [27], inspection stations locations in transport networks [28], anti-covering location problem [23], *p*-center problem [21] location of distributed generation resources [39], and capacitated plant location problem [34].
- Scheduling: static scheduling of independent tasks on homogeneous multiprocessor systems [22], the ride-matching problem [29,30], job shop scheduling [37], task scheduling in computational grids [35], backup allocation problem [36], and berth allocation problem [33].
- Medicine with chemistry: cancer therapy [32]; chemical process optimization [38].
- Networks: network design [25].
- Continuous and mixed optimization problems: numerical function minimization [26]; the satisfiability problem in probabilistic logic [47].

The additional applications include transportation problems [46,48-50,59], scheduling problems [60,61], and feature selection problem [62].

Management of the access charges level for the use of railway infrastructure by bee colony optimization was reported in [48]. Initial solution to the first iteration is the real life data from the case study of Montenegro rail. The improvement variant of BCO is applied and a pool of good solution is generated and reported to the users. Bees are homogeneous, however there are 8 types of solution modifications depending on the current solution assigned to a bee. The significant improvements were obtained with respect to the real data for years 2011 to 2014.

The BCO meta-heuristic for vehicle routing problem with time windows was considered in [46]. The authors proposed improvement variant with heterogeneous bees divided into three groups to perform three types of solution modifications. At the beginning of the program execution, initial solution is generated using the simple insertion constructive heuristic. The initial solution for all other iterations is the current best solution. Each solution is modified in such a way that couple of routes are removed and the nodes in these routes became un-routed. After that, the insertion heuristic is used to make the new routes. Three insertion heuristics are used, differing in a way they calculate costs. Consecutively, there are three types of artificial bees. The proposed algorithm is applied on a various test problems. from the Solomon's benchmark instances. Preliminary computational results show that the BCO algorithm produced high quality solutions, with the gap with respect to the best known in range from 0 to 6.19% depending on instance's difficultness.

The authors of [49] considered the transit network design problem in a way that they simultaneously determine the links to be included in the transit network, assemble chosen links into bus routes, and determine bus frequency on each of the designed routes. Their approach is based on improvement BCO. The BCO implementation is similar to the one proposed in [46]. Initial solution for the first iteration is determined using the constructive heuristic, while for the other iterations the initial solution is selected randomly among *B*+1 solution from the previous iteration (B solutions held by the bees and the current global best solution). There are two types of artificial bees performing different solution modifications. The problem belong to the class of multi-criteria optimization (there are three objective functions) and the authors used lexicographical method to solve it. The comparison of the solutions generated by the BCO and other competitive approaches are reported showing that the BCO algorithms significantly improve initial solutions producing the final solution that is the best for users.

Vehicle rerouting in the case of unexpectedly high demand in distribution systems by BCO was reported in [50]. Mathematical formulation of the problem is provided as well as the BCO implementation quite similar to [46,49]. BCO was compared with CPLEX solver and the obtained results showed that the BCO meta-heuristic can find high-quality solutions in a reasonable amount of the CPU time.

Urban transit network design was considered in [59] and solved by the improvement BCO. The main goal is to minimize the passenger cost that consist of six components: access cost, waiting cost, in-vehicle cost, transfer cost, fare cost, operational cost. The BCO implementation involves three types of bees differing in the solution modification scheme. A the beginning of each forward pass, the bees decide with a predefined probability which modification to perform. At the beginning, the initial solution is determined heuristically, and algorithm performs a single iteration. As a case study, road system consisting of a grid network containing 12 nodes and 15 edges is used. The results show that the BCO approach can be successfully and efficiently used to solve the considered problem.

Constructive BCO for scheduling dependent tasks to homogeneous systems is developed in [60]. The proposed implementation is similar to the one presented in [22]. Tasks are selected randomly depending on the assigned priorities. The priority depends on task duration and the number of successors. The authors included short term memory connecting the two successive iterations. This memory contains average values of all partial solutions generated during the forward pass. All the bees having partial solutions worse that the average value saved in the memory automatically become uncommitted. The random task graphs are used to evaluate the proposed approach. BCO is compared against Ant colony system (ACS), Simulate Annealing (SA), and Tabu Search (TS). The BCO solutions are better in comparison with the ACS algorithm, similar compared with TS, and considerably close to SA that is state-of-art scheme.

An open shop scheduling problem is considered in [61]. A new bee colony optimization algorithm, with an idle-time-based filtering scheme, which can automatically stop searching a partial solution with insufficient profitability is proposed. At the same time, the scheduler is creating a new scheduling solution, and therefore, save on time–cost for the remaining partial solution. Bees can have different roles (scout, dancer, follower). The roles are assigned at the beginning of each forward pass depending on the results from the previous forward pass. Each iteration (round) contains only one forward and one backward pass. The BCO approach is compared with PSO and evaluated on known test instances from the literature. Reported results show that average BCO schedule length is slightly closer to the Best-Known Solution, than average PSO schedule length. In addition, the running time of BCO is significantly shorter that the time required by PSO.

In order to improve the exploitation power of constructive BCO, in [62] the authors introduced a novel algorithm, named weighted BCO (wBCO), that allows the bees to search in the solution space deliberately while considering policies to share the attained information about the solution sub-spaces (food sources) heuristically. The modification affects the loyalty decision and also recruiter selection. In the backward step for wBCO, where the bees measure how loyal they are to the partial solutions, the algorithm considers two weights for each food source. A global weight measures how popular a given food source is among the bees, A local weight, indicates the extent to which a selected food source can contribute to the category label of the classification problem. In the recruiter selection step, in order to preserve diversity the followers select their recruiters in a filtering stepwise process. Two filtering stages are applied; the first based on similarity, and the second depending on fitness values. In similarity filtering, for a given follower a set of recruiters is selected based on the traversal similarity and then the follower selects a recruiter which has the closest fitness value.

The proposed wBCO algorithm is applied to the task of feature selection, that is modelled as a discrete (or categorical) optimization problem. Comparison with ABC and some other methods from the literature shows that wBCO performs better on most of the instances.

CONCLUSION

The Bee Colony Optimization (BCO) algorithm, a meta-heuristic method inspired by the foraging behavior of honeybees, belongs to the class of Swarm Intelligence techniques. It represents a general algorithmic framework applicable to various optimization problems in combinatorial/continuous optimization and engineering. The BCO method is based on the concept of *cooperation*, which increases the efficiency of artificial bees and allows achieving goals that could not be reached by individual actions only. Through the information exchange and recruiting process, BCO has the capability to intensify the search in the promising regions of the solution space. BCO become very popular algorithm due to its simplicity: it is easy to understand and has a small number of parameters (number of bees and number of transformations during a single iteration).

This paper presents and overview of the recent developments and applications of the BCO algorithm to combinatorial and continuous optimization problems in order to promote this simple yet effective optimization method. The survey is certainly not exhaustive since the possibilities for new applications are endless. Moreover, the suitability for parallelization of BCO opens not only a new research direction but also some new potential applications. Based on the achieved results and gained experience, new models founded on BCO principles (autonomy, distributed functioning, self-organizing, information exchange, collaboration) are likely to significantly contribute to solving complex engineering, management, and control problems.

ACKNOWLEDGMENT

The author would like to thank the organizers of the BALCOR 2015 conference for the invitation to present this work and promote the BCO meta-heuristic among distinguished colleagues, operational researchers.

BIBLIOGRAPHY:

[1] M. Gendreau, and J-Y. Potvin, (eds.) *Handbook of Metaheuristics*, (second edition) Springer, New York Dordrecht Heidelberg London, 2010.
[2] T. Bäck, D.B. Fogel, and Z. Michalewicz, *Evolutionary computation 1: Basic algorithms and operators*, Taylor & Francis Group, LLC, CRC Press, 2000.
[3] E. Garcia-Gonzalo, and J.L. Fernandez-Martinez, " A brief historical review of particle swarm optimization (PSO) ", *Journal of Bioinformatics and Intelligent Control*, vol. 1, no. 1, pp. 3-16, 2012.
[4] G. Beni, "The concept of cellular robotic system", in Proc. 1988 *IEEE International Symposium on Intelligent Control*, IEEE Computer Society Press, Los Alamitos, CA, 1988, pp. 57–62.
[5] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence*, Oxford University Press, Oxford, 1997.
[6] R. Chiong, *Nature-inspired algorithms for optimization*, Springer, 2009.
[7] G. Rozenberg, T. Bäck, and J.N. Kok, (eds.) *Handbook of Natural Computing*, vol.I-IV, Springer, 2012.
[8] B. Xing, and W.-J. Gao, *Innovative computational intelligence: a rough guide to 134 clever algorithms*, Springer, 2014.
[9] X.-S. Yang, *Nature-inspired metaheuristic algorithms*, Luniver press, 2010.
[10] X.-S. Yang, *Nature-inspired optimization algorithms*, Elsevier, 2014.
[11] P. Lučić, and D. Teodorović, "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, pp.441–445.
[12] P. Lučić, and D. Teodorović, "Transportation modeling: an artificial life approach", In: Proceedings of the 14th *IEEE International Conference on Tools with Artificial Intelligence*, Washington, DC, 2002, pp. 216–223.
[13] P. Lučić, and D. Teodorović, "Computing with bees: attacking complex transportation engineering problems", *International Journal on Artificial Intelligence Tools*, vol. 12, 2003, pp. 375–394.
[14] P. Lučić, and D. Teodorović, "Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach", In: Verdegay JL (ed.) *Fuzzy Sets in Optimization*, Springer-Verlag, Heidelberg Berlin, 2003, pp 67–82.
[15] T. Davidović, M. Šelmić, and D. Teodorović, "Bee colony optimization part I: The algorithm owerview", *YUJOR*, vol. 25, no. 1, 2015, pp.33-56 .
[16] D. Teodorović, M. Šelmić, and T. Davidović, "Bee colony optimization part II: The applications survey", *YUJOR*, vol. 25, no. 2, 2015, pp.185-219.
[17] M. Šelmić, *Location problems on transport networks by computational intelligence methods*, PhD thesis, Faculty of Traffic and Transportation, University of Beograde, 2011.
[18] M. Nikolić, *Resolving the consequences of traffic disturbances by bee colony optimization*, PhD thesis, Faculty of Traffic and Transportation, University of Beograde, 2015.
[19] T. Stojanović, *The development and analisys of metaheuristics for satisfiability in probabilistic logics*, Faculty of Science, University of Kragujevac, 2015.

[20] T. Jakšić Krüger, *The development, parallelization and theoretical verification of bee colony optimization*, Faculty of Technical Sciences, University of Novi Sad, 2015.

[21] T. Davidović, D. Ramljak, M. Šelmić, and D. Teodorović, "Bee colony optimization for the *p*-center problem", *Computers and Operations Research*, vol. 38, no. 10, 2011, pp. 1367-1376.

[22] T. Davidović, M. Šelmić, D. Teodorović, and D. Ramljak, "Bee colony optimization for scheduling independent tasks to identical processors", *Journal of Heuristics*, vol. 18, 2012, pp. 549-569.

[23] B. Dimitrijević, D. Teodorović, V. Simić, and M. Šelmić, "A bee colony optimization approach to solving the anti-covering location problem", *Journal of Computing in Civil Engineering*, vol. 26, 2011, pp. 759–768.

[24] G. Marković, D. Teodorović, and V. Aćimović-Rspopović, "Routing and wavelength assignment in all-optical networks based on the bee colony optimization", *AI Commun.* vol. 20, 2007, pp. 273–285.

[25] M. Nikolić, and D. Teodorović, "Transit network design by bee colony optimization", *Expert Systems with Applications*, vol. 40, 2013, pp. 5945–5955.

[26] M. Nikolić, and D. Teodorović, "Empirical study of the bee colony optimization (BCO) algorithm", *Expert Systems with Applications*, vol. 40, pp. 4609–4620.

[27] M. Šelmić, P. Edara, and D. Teodorović, "Bee colony optimization approach to optimize locations of traffic sensors on highways", *Tehnika*, vol. 6, 2008, pp. 9-15 (in Serbian).

[28] M. Šelmić, D. Teodorović, and K. Vukadinović, "Locating inspection facilities in traffic networks: an artificial intelligence approach", *Transportation Planning and Technology*, vol. 33, 2010, pp. 481-493.

[29] D. Teodorović, and M. Dell'Orco, "Bee colony optimization – a cooperative learning approach to complex transportation problems", In: *Advanced OR and AI Methods in Transportation,* Proc. 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland, 2005, pp. 51–60.

[30] D. Teodorović, and M. Dell'Orco, "Mitigating traffic congestion: solving the ride-matching problem by bee colony optimization", *Transport. Plan. Techn.* vol. 31, 2008, pp. 135–152.

[31] D. Teodorović, and M. Šelmić, "The BCO algorithm for the *p* median problem", In: Proceedings of the XXXIV *Serbian Operations Research Conference*. Zlatibor, Serbia, 2007, pp. 417-420 (in Serbian).

[32] D. Teodorović, M. Šelmić, and Lj. Mijatović-Teodorović, "Combining case-based reasoning with bee colony optimization for dose planning in well differentiated thyroid cancer treatment", *Expert Systems with Applications*, vol. 40, 2013, pp. 2147-2155.

[33] N. Kovač, "Bee colony optimization algorithm for the minimum cost berth allocation problem", In: XI *Balkan Conference on Operational Research,* BALCOR, 2013, pp. 245-254.

[34] T.V. Levanova, and E.A. Tkachuk, "Development of a bee colony optimization algorithm for the capacitated plant location problem", Book of Abstracts of II *International conference, Optimization and applications* (OPTIMA-2011), Montenegro, 2011, pp. 153–156.

[35] Z. Mousavinasab, R. Entezari-Maleki, and A. Movaghar, "A bee colony task scheduling algorithm in computational grids", In: *Digital Information Processing and Communications International Conference*, ICDIPC, Vaclav Snasel Jan Platos Eyas El-Qawasmeh (Eds.), Czech Republic, 2011, pp. 200-210

[36] R. Nedeljković, S. Mitrović, and D. Drenovac, "Bee colony optimization problem to backup allocation problem", XXVII *PosTel*, Belgrade, 2009, pp. 432-442 (in Serbian).

[37] A.P.P. Pertiwi, and P. Suyanto, "Globally evolved dynamic bee colony optimization", In: *Knowledge-Based and Intelligent Information and Engineering Systems*, 15th International Conference, KES 2011, A. König *et al.* (eds.), Germany, LNAI 6881, 2011, pp. 52-61.

[38] M. Saidi, N. Mostoufi, and R.S. Gharebagh, "Modelling and optimisation of continuous catalytic regeneration process using bee colony algorithm", *The Canadian Journal of Chemical Engineering*, 2013, pp. 1256–1269.

[39] M.F. Sohi, M. Shirdel, and A. Javidaneh, "Applying BCO algorithm to solve the optimal DG placement and sizing problem", The 5th *Int.Conf. Power Engineering and Optimization* (PEOCO2011), Malaysia, 2011, pp. 71-76.

[40] L.P. Wong, M.Y. Hean Low, and C.S. Chong, "Bee colony optimization with local search for traveling salesman problem", In 6-th IEEE *International Conference on Industrial Informatics*, 2008, pp. 1019–1025.

[41] L.P. Wong, M.Y. Hean Low, and C.S. Chong, "An efficient bee colony optimization algorithm for traveling salesman problem using frequency-based pruning", In 7-th IEEE *International Conference on Industrial Informatics*, 2009, pp. 775–782.

[42] L.P. Wong, M.Y. Hean Low, and C.S. Chong, "Bee colony optimization with local search for traveling salesman problem", *International Journal on Artificial Intelligence Tools*, vol. 19, no. 3, 2010, pp. 305-334.

[43] S. Camazine, and J. Sneyd, "A model of collective nectar source by honey bees: Self-organization through simple rules", *J. Theor. Biol.* vol. 149, 1991, pp. 547-571.

[44] P. Maksimović, and T. Davidović, "Parameter calibration in the bee colony optimization algorithm", Proc. XI *Balcan Conf. Oper. Res.*, BALCOR, Sept. 07-11, Beograd-Zlatibor, Serbia, 2013, pp. 255-264.

[45] D. McFadden, "Conditional logit analysis of quantitative choice behavior", In: P. Zaremmbka (ed.), *Frontier of Econometrics*, Academic Press, New York, 1973.

[46] M. Nikolić, D. Teodorović, and M. Šelmić, "Solving the vehicle routing problem with time windows by Bee Colony Optimization metaheuristic", Proc. 1st *Logistic International Conference*, Belgrade, Serbia, 2013, pp. 44-48.

[47] T. Stojanović, T. Davidović, and Z. Ognjanović, "Bee-colony optimization for the satisfiability problem in probabilistic logic", *Applied Soft Computing*, vol. 31, 2015, pp. 339-347.

[48] M. Bugarinović, T. Davidović, and B. Bošković, "Management of the access charges level for the use of railway infrastructure by bee colony optimization", *Book of Abs. 18th EWGT*, 2015, p. 184.

[49] M. Nikolić, and D. Teodorović, "A simultaneous transit network design and frequency setting: computing with bees", *Expert Syst. Appl.* vol. 41, no. 16, 2014, pp. 7200–7209.

[50] M. Nikolić, and D. Teodorović, "Vehicle rerouting in the case of unexpectedly high demand in distribution systems", *Transportation Research Part C: Emerging Technologies*, vol. 55 2015, pp. 535-545.

[51] T. Davidović, D. Ramljak, M. Šelmić, and D. Teodorović, "MPI parallelization of bee colony optimization", Proc. 1st *International Symposium* & 10th *Balkan Conference on Operational Research*, vol. 2, , Thessaloniki, Greece, 2011, pp. 193-200.

[52] T. Davidović, T. Jakšić, D. Ramljak, M. Šelmić, and D. Teodorović, "MPI parallelization strategies for bee colony optimization", *Optimization*, vol. 62, no. 8, 2013, pp. 1113-1142.

[53] T. Crainic, T. Davidović, and D. Ramljak, "Designing parallel meta-heuristic methods", In: *High Performance and Cloud Computing in Scinece and Education*, M. Despotović-Zrakić, V. Milutinović, and A. Belić, (eds.) IGI-Global, 2014, pp. 260-280.

[54] T. Jakšić, "On the convergence of the bee colony optimization meta-heuristic", In: *Proc. The rth Symposium Mathematics and Applications*, Belgrade, May 24-25, 2013, pp. 176-187, (in Serbian).

[55] T. Jakšić, T. Davidović, D. Teodorović, and M. Šelmić, "The BCO algorithm and its convergence", *Int. J. Bio-inspired computing* 2013, (accepted).

[56] T. Jakšić, T. Davidović, "Model convergence properties of the constructive bee colony optimization algorithm", *Proc. 41th Symp. on Operational Research, SYM-OP-IS 2014*, Divčibare, Sept. 16-19, 2014, pp. 340-345.

[57] W.J. Gutjahr, "Convergence analysis of metaheuristics", In: V. Maniezzo, T. Stutzle, S. Voss, (eds) *Matheuristics Hybridizing Metaheuristics and Mathematical Programming*, Springer, 2009, pp 159-187.

[58] L. Zaheh, "Fuzzy sets", *Information and Control,* vol. 8, 1965, pp. 338-353.

[59] H. Zhang, P. Zhao, J. Gao, C. Zhuge, and X. Yao, "An effective intelligent method for optimal urban transit network design", *Journal of Information & Computational Science*, vol. 12, no. 6, 2015, pp. 2177-2184.

[60] V. Arabnejad, A. Moeini, and N. Moghadam. "Using bee colony optimization to solve the task scheduling problem in homogenous systems." *IJCSI International Journal of Computer Science Issues*, vol. 8. iss. 5, no. 3, 2011, pp. 348-353.

[61] H. Yueh-Min, and L. Jin-Chen, "A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems", *Expert Systems with Applications*, vol. 38, no. 5, 2011, pp. 5438-5447.

[62] A. Moayedikia, R. Jensen, U.K. Wiil, and R. Forsati, "Weighted bee colony algorithm for discrete optimization problems with application to feature selection", *Engineering Applications of Artificial Intelligence*, vol. 44, 2015, pp. 153-167.