XI Balkan Conference on Operational Research
BALCOR 2013
Belgrade, Zlatibor, 07-10 September 2013

UNIVERSITY OF BELGRADE
FACULTY OF ORGANISATIONAL SCIENCES

Parameter Calibration in the Bee Colony Optimization Algorithm

Petar Maksimović[1], Tatjana Davidović[1]

[1] Mathematical Institute of the Serbian Academy of Sciences and Arts, [petarmax,tanjad]@mi.sanu.ac.rs

Abstract: *We present a case study in which we examine in detail the behavior of the Bee Colony Optimization (BCO) metaheuristic when applied to the p-center problem. On a number of benchmark problems, we perform a fine-tuning of several parameters of the BCO algorithm, yielding insights into the behavior of the algorithm, its robustness and efficiency, and reach six new best-known solutions.*

Keywords: *Meta-heuristics, Nature-inspired algorithms, Combinatorial optimization, Location analysis, p-center problem*

1. Introduction

Bee Colony Optimization (BCO) is a biologically-inspired, population-based, stochastic random-search technique. It is based on an analogy between the foraging behavior of bees and the manner in which algorithms for combinatorial optimization attempt to find the optimum of a given problem. BCO was first presented in [Lučić and Teodorović, 2001, Lučić and Teodorović, 2002, Lučić and Teodorović, 2003a], where it was considered for the Traveling Salesman Problem, and has been, since then, successfully applied to a variety of real-life optimization problems, such as, but not limited to, the Vehicle Routing Problem [Lučić and Teodorović, 2003b], the Routing and Wavelength Assignment in All-Optical Networks [Marković et al., 2007], the Ride-Matching Problem [Teodorović and Dell'Orco, 2008], the Traffic Sensors Locations Problem on Highways [Edara et al., 2008], Static Scheduling of Independent Tasks on Homogeneous Multiprocessor Systems [Davidović et al., 2009, Davidović et al., 2012], and the p-center problem [Davidović et al., 2011].

The p-center problem is the well-known location problem of establishing p facilities on a network of n vertices, so that the maximal distance between a vertex and its associated facility, when considering all vertices, is minimized. It addresses the location of emergency-type facilities (such as ambulances, fire brigades, and police stations) in transportation networks, contexts in which critical situations could be life-threatening, and optimal reachability is essential. Hence, the p-center problem is considered to be of high practical importance and, as such, it has been extensively studied in the relevant literature [Hakimi, 1964, Hakimi, 1965, Minieka, 1977, Drezner, 1984, Beasley, 1985, Hassin et al., 2003, Mladenović et al., 2003, Caruso et al., 2003, Pacheco and Casado, 2005, ReVelle and Eiselt, 2005, Chen and Chen, 2009].

In this paper, we proceed along the path presented in [Davidović et al., 2011], where a novel version of the BCO algorithm was developed to target the p-center problem. The novelty of that approach amounts to the fact that the standard pure constructive concept of BCO was substituted with a problem-specific improvement strategy. Here, we re-implement the solution of [Davidović et al., 2011] considerably more efficiently and fine-tune the parameters of the BCO algorithm, using benchmark test examples, originally designed for testing the p-median problem (OR-Lib Test Problems [Beasley, 1985], available on-line at `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html`). Not only do the obtained results match those currently known in literature in terms of quality and speed, but we have also generated six new best known solutions.

The remainder of this paper is organized as follows: in Section 2. we summarize the BCO algorithm and elaborate on how it has been applied to the p-center problem. Next, in Section 3. we discuss the setup for the experiments and the methodology of the evaluation. In Section 4. we present the results of the experiments related to the influence of BCO parameters on solution quality and algorithm speed. Finally, in Section 5. we give conclusions and outline directions for future work.

2. Tackling the p-center problem with BCO

In this Section, we show how the BCO meta-heuristic has been applied to the p-center problem. After a short description of the BCO algorithm, we present the Solution Improvement Strategy specific to this setting and the data pre-processing required for its efficient implementation.

2.1. A Brief Overview of the BCO algorithm

As we have mentioned before, the BCO algorithm is inspired by the foraging behavior of honey bees in nature. Here, due to lack of space, we present only the outline of the algorithm, while we kindly refer the reader to [Lučić and Teodorović, 2001, Lučić and Teodorović, 2002, Lučić and Teodorović, 2003a, Lučić and Teodorović, 2003b, Marković et al., 2007, Teodorović and Dell'Orco, 2008, Edara et al., 2008, Davidović et al., 2011, Davidović et al., 2012] for a much more detailed background.

In the BCO algorithm, there are B agents (we will refer to these agents as 'bees'), who collaboratively search for an optimal solution of a given problem. One *iteration* of the algorithm consists of NC *steps*, while each of these steps comprises two phases: a *forward pass* and a *backward pass*. Within the first forward pass, each of the bees constructs its initial solution, while in all of the other forward passes it attempts to improve it. The construction and improvement mechanisms are problem-specific. In each backward pass, the bees collectively assess their current solutions, and based on a loyalty criterion LC, each of the bees decides to either stay loyal to its own solution and continue improving it, or to discard it and become an *uncommitted follower*. Then, each uncommitted follower has to adopt the solution of one of the loyal bees as its own, and use it in the next step. In the final step, the best among all B obtained solutions is identified after the forward pass, and no backward pass is performed. This solution is then used to update the global best solution, all B solutions are deleted, and a new iteration can begin. BCO runs iteration-by-iteration, until the fulfillment of a given stopping condition. Stopping condition can include a maximum number of iterations, maximum number of iterations without improvement of the current global best solution, maximum allowed CPU time, etc.

2.2. Notation

Let us assume to have n distinct locations, and let us assume that we know the distances between each two of them. Our task is to, out of all of these locations, choose $p \leq n$ to act as centers, so that the maximal distance between a location and its closest center, when considering all locations, is minimized.

For a given set of n locations, we will denote the locations themselves by l_i, their corresponding centers by c_i, and the distance between l_i and c_i by d_i, with $1 \leq i \leq n$. We define the *critical pair* (l_c, c_c) as the pair consisting of a location and its corresponding center, for which it holds that $d_c = \max_{j\in\{1,\ldots n\}} d_i$. Also, we refer to d_c as *critical distance*, and it is precisely d_c that we wish to minimize. In the rest of this Section, we outline the BCO implementation from [Davidović et al., 2011], while pointing out several of our observations and improvements.

2.3. Pre-processing

Let the matrix $D = [d_{ij}]_{n\times n}$ be the full distance matrix between locations, obtained by applying the Floyd-Warshall algorithm to the input data. So as to gain significant computational speed, we can perform several pre-processing tasks, which we will illustrate on a running example of the following distance matrix

$$D = \begin{bmatrix} 0 & 0.4 & 0.9 & 0.2 & 0.3 \\ 0.4 & 0 & 1.2 & 0.6 & 0.7 \\ 0.9 & 1.2 & 0 & 0.1 & 0.2 \\ 0.2 & 0.6 & 0.1 & 0 & 0.4 \\ 0.3 & 0.7 & 0.2 & 0.4 & 0 \end{bmatrix}.$$

1. We form two auxiliary matrices, D_s and D_c. D_s is obtained from D by sorting all of its rows in non-descending order, while D_c contains the ordering of the corresponding indices:

$$D_c = \begin{bmatrix} 0 & 0.2 & 0.3 & 0.4 & 0.9 \\ 0 & 0.4 & 0.6 & 0.7 & 1.2 \\ 0 & 0.1 & 0.2 & 0.9 & 1.2 \\ 0 & 0.1 & 0.2 & 0.4 & 0.6 \\ 0 & 0.2 & 0.3 & 0.4 & 0.7 \end{bmatrix}, \qquad D_s = \begin{bmatrix} 1 & 4 & 5 & 2 & 3 \\ 2 & 1 & 4 & 5 & 3 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 3 & 1 & 5 & 2 \\ 5 & 3 & 1 & 4 & 2 \end{bmatrix}.$$

By using these two matrices we are able to obtain, for a given location l_i, its k-th closest node $(1 \leq k \leq n)$ and the corresponding distance with complexity $O(1)$.

2. Further, we construct a matrix D_r, where the element in the i-th row and the j-th column represents the total number of locations which are closer to l_i than l_j is. Therefore, in our running example, we have that:

$$D_r = \begin{bmatrix} 0 & 3 & 4 & 1 & 2 \\ 1 & 0 & 4 & 2 & 3 \\ 3 & 4 & 0 & 1 & 2 \\ 2 & 4 & 1 & 0 & 3 \\ 2 & 4 & 1 & 3 & 0 \end{bmatrix}.$$

   Using this matrix, we can obtain, again with complexity $O(1)$, how many locations are closer to location l_i than location l_j is, and from there, using the matrices D_s and D_c, we can obtain which are the locations in question and what their distances are. This will play a crucial role in the construction and modification of the solutions.

2.4. Generating the initial solution

In the first forward pass of every iteration, each of the bees needs to generate its own initial solution, i.e. it needs to choose some p locations to be the initial centers. The first center is chosen at random, and the critical pair (l_c, c_c) is determined. Then, we consider the set of locations that are closer to l_c than c_c is, and one element from this set is chosen at random to be the next center. The next critical pair is determined, and this procedure continues until p centers have been chosen. By choosing centers in this way, we ensure that the critical pair is different each time, thus attempting to directly reduce the critical distance, i.e. improve the quality of the solution. Here, choosing the next center at random is favored over a greedy selection because we would like different bees to end up with different initial solutions.

Let us briefly discuss the advantages of using the pre-processed matrices with respect to the presented algorithm. The number of locations closer to l_c than c_c can be obtained instantly as $D_r(l_c, c_c)$. Furthermore, the centers in question can be found at $D_c(l_c, i)$, where $1 \leq i \leq D_r(l_c, c_c)$. Finally, the random choice of the new center amounts to calculating $D_c(l_c, \text{rand}(1, D_r(l_c, c_c)))$, which is of complexity $O(1)$. Had we not performed the pre-processing, this selection would have had the complexity of $O(n^2)$.

2.5. Solution Improvement Strategy

In each subsequent forward pass, each of the bees attempts to improve on its current solution in the following manner:

1. It chooses a random number q between 1 and $\min(p, n/10)$.
2. It adds another q centers to its solution in the same manner as when constructing the initial solution, by finding the critical pair and attempting to reduce the critical distance. This step results in having a non-feasible solution, containing $p + q$ centers.
3. It removes q centers, one by one, in a greedy fashion, i.e. in each step it removes the center whose removal causes the smallest increase in critical distance.

Here, it has to be noted that, even though this algorithm has yielded very good results, both in [Davidović et al., 2011] and in this paper, contrary to the claims of [Davidović et al., 2011], it does not *always* lead to a solution of at least the same quality than the one the bee had originally started from. This can be approached in two ways:

1. We allow the modification regardless of the outcome. This can be justified by the fact that the selection process is inherently random, and a worse result at one point could lead to an even better one in the next step. This is effectively the approach implemented in [Davidović et al., 2011].
2. We do not allow the modification if it worsens the solution. Having conducted a number of preliminary experiments, we have noticed that the time to obtain the solution is invariably longer when using the first approach, and, therefore, opted to use this one instead.

2.6. Deciding on Loyalty

In each backward pass, each bee b, with $1 \leq b \leq B$, decides whether it will stay loyal to its current solution or not. To this end, the obtained objective function values s_b, are first normalized:

$$N_b = \begin{cases} \dfrac{s_{\max} - s_b}{s_{\max} - s_{\min}}, & \text{if } s_{\max} \neq s_{\min}, \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

where $s_{\min}$ and $s_{\max}$ respectively denote the minimum and the maximum of the obtained solutions. Next, the loyalty probability P_b is calculated using a loyalty criterion LC, described in detail in the next section. Next, for each bee, a random number r_b is chosen from the real unit interval, and if $r_b \leq P_b$, the b-th bee is declared to be loyal to its solution. Otherwise, it becomes an uncommitted follower. In the end, each of the uncommitted followers adopts a solution of one of the loyal bees as its own, using the roulette wheel approach.

3. Experimental Setup and Evaluation Methodology

The parameters which we will be fine-tuning, and on which the quality of the experimental results depends significantly, are the number of bees B, the number of steps per iteration NC, and the loyalty criterion LC. We denote by $N_{\max}$ the maximum of the normalized values obtained by (1). So far, the loyalty criterion has been a general one, whereas in this paper we investigate four alternatives:

1. LC_1 $(\mathrm{Exp_{Lin}}) : P_b^{1,u+1} = e^{-\frac{N_{\max}-N_b}{u}}$.

   This is the standard methodology for determining the loyalty of a bee to its solution in the BCO algorithm, and is present in both the first literature on BCO [Lučić and Teodorović, 2001, Lučić and Teodorović, 2002, Lučić and Teodorović, 2003a], as well as in the subsequent research endeavors [Lučić and Teodorović, 2003b, Marković et al., 2007, Teodorović and Dell'Orco, 2008, Edara et al., 2008, Davidović et al., 2011, Davidović et al., 2012]. It stems from the original constructive approach, where the bee is more loyal to its solution the longer it is being constructed, i.e. as the number of performed forward passes increases.

2. LC_2 $(\mathrm{Exp_{Sqrt}}) : P_b^{2,u+1} = e^{-\frac{N_{\max}-N_b}{\sqrt{u}}}$.

   We have noticed that, for small values of B and NC, loyalty probabilities from the first method very quickly become very close to one, thus almost certainly making the bee permanently loyal to its solution. In order to weaken this dependency on the number of forward/backward passes, we have applied the square root to the denominator.

3. LC_3 $(\mathrm{Exp_{Solo}}) : P_b^3 = e^{-(N_{\max}-N_b)}$.

   We have also investigated how the BCO algorithm behaves when the loyalty of the bee is not dependent on the number of forward/backward passes at all.

4. LC_4 $(\mathrm{N_V}) : P_b^4 = N_b$.

   Finally, we have investigated the possibility of dropping the exponentiation altogether, and directly treating normalized values as loyalty probabilities.

As for the number of bees and the number of steps per iteration, extensive preliminary experiments have shown that, for each test example and each loyalty criterion, it is sufficient to consider $B \in \{1, \ldots, 10\}$ and $NC \in \{10, 15, \ldots, 100\}$.

Let us assume to have a given test example i, number of bees B, number of steps per iteration NC, and loyalty criterion LC_j. Then, one *algorithm run* for the given parameters consists of running the BCO algorithm with the test example i as input, and with parameters B, NC, and LC_j. One *experiment run*, for the given parameters, consists of repeatedly running the corresponding *algorithm run*, until the currently best known solution in literature is reached. Finally, to obtain statistical significance, one *experiment*, for the given parameters, consists of executing 1000 corresponding experiment runs.

For each experiment run, we measure the time needed to find the solution. For each experiment, we report the average over the times obtained in the 1000 experiment runs, as well as the corresponding standard deviation. We have opted for this methodology because we feel that the results obtained in this way provide better insights into the behavior of the BCO algorithm than, for instance, imposing a time-limit on the experiment runs. We also take into account, for each B, NC and LC, the sum of average best times obtained across all experiments, which we take as the overall quality assessment criterion.

All of the experiments were performed on a single core of a machine equipped with an Intel Core i7-2600 processor, with 8Gb of DDR3 RAM running at 1333MHz, and a 1Tb SATA III hard drive.

4. Experimental Results

Thus far, out of the 40 test examples [Beasley, 1985], we have been able to run the full experiments on test examples from 1 to 12, 14 to 17, 21, 26, 27, 31, 35 and 38, with the best obtained results per experiment presented in Table 1, and the best overall obtained results presented in Table 2.

Table 1. Best results obtained for the fully performed experiments

Example	n	p	Best-known	LC	B	NC	Avg. time (s)	St. dev (s)
1	100	5	127	4	1	15	0.000595406	0.000585464
2	100	10	98	4	2	15	0.030971790	0.032620809
3	100	10	93	4	2	30	0.367946010	0.367294711
4	100	20	74	3	3	15	0.012104291	0.011663945
5	100	33	48	3	1	10	0.000454851	0.000387997
6	200	5	84	2	1	55	0.048833944	0.047556002
7	200	10	64	3	1	65	0.008490737	0.008497499
8	200	20	55	4	4	35	0.111264099	0.113067074
9	200	40	37	3	4	20	0.020121393	0.018888634
10	200	67	20	1	1	10	0.025953049	0.026987125
11	300	5	59	4	2	55	0.036946726	0.036267730
12	300	10	51	1	5	85	0.145222598	0.139168805
14	300	60	26	3	3	100	0.360333774	0.361781028
15	300	100	18	3	1	10	0.004810655	0.004638123
16	400	5	47	4	2	25	0.002725710	0.002426084
17	400	10	39	3	2	95	0.021034696	0.018498344
21	500	5	40	4	1	100	0.007731410	0.006425355
26	600	5	38	1	1	90	0.024119023	0.022054725
27	600	10	32	2	1	100	0.041140207	0.035992213
31	700	5	30	3	1	85	0.012955429	0.010569439
35	800	5	30	4	1	90	0.114130097	0.113215234
38	900	5	29	2	2	30	0.014014507	0.013029383

Table 2. Best overall results for each LC across all fully performed experiments

LC	B	NC	Time (s)
1	1	80	1.718853077
2	1	75	1.693554724
3	2	65	1.677097522
4	2	50	1.657951070

As can be seen from Table 1, for each of the individual examples, the number of bees required to obtain the best known solution in the shortest possible time was very small (did not exceed 5), and there were even experiments where one bee was sufficiently powerful (thus reducing BCO to a single-solution-based algorithm). Another thing which we can notice is that the standard deviation is consistently almost equal to the average time obtained, suggesting an exponential-like distribution of the time required to find the best solution. Also, we can notice that the loyalty criterion which was most frequently the best performing were $\mathrm{Exp}_{\mathrm{Solo}}$ and N_{V}, indicating that, locally, the best loyalty criterion for the BCO algorithm in this setting is not $\mathrm{Exp}_{\mathrm{Lin}}$, as was originally considered. Finally, we can notice that the overall best performing loyalty criterion is N_{V}, which is again indicative of a need for a more thorough examination of the manner in which loyalty is calculated in future applications of BCO, even though the times obtained here also depend on the pre-processing and the improvement strategy. More detailed insights are presented in Section 4.2.

4.1. New Best Known Solutions

Apart from these results, in the preliminary phase of our experiments we have obtained six new best-known solutions for the problems listed in Table 3. Since the new best-known results were found sparsely, the running of full experiments on these problems was extremely demanding in terms of time and had to be relegated to future work.

Table 3. New best-known solutions found

Test example	n	p	Previous best-known	New best-known
20	400	133	14	13
23	500	50	23	22
24	500	100	16	15
28	600	60	19	18
30	600	200	10	9
37	800	80	16	15

These results show the power of the BCO combined with the Solution Improvement Strategy when it comes to problems with a large number of centers (from 10% to 33% of the overall number of locations). For all of the six experiments, the previously best known results have been consistently reached in fractions of a second, while the new best known results have appeared intermittently. In particular, we would like to single out test example 30, in which an improvement has been made in [Davidović et al., 2011] (from 11 to 10), and again now (from 10 to 9), with the use of a similar, but more efficiently implemented method.

4.2. A More Detailed Look into the Obtained Results

Here, we will provide more detailed insights into the behavior of the BCO algorithm, with respect to the performed experiments. So far, we have observed in the results several different patterns which we present in the form of graphs, showing the nature of the four loyalty criteria and the sensitivity of the solution to the choice of B and NC.

1. The first type of graphs is are like the one shown in Figure 1. This type of graph was obtained for test problems 1, 5 and 10, for all four loyalty criteria, with the average finding time of the best known solution being $9.3 \cdot 10^{-3}s$. The best known solution is found immediately for only one bee and very small NC, and the more B and NC grow, the more time it takes to find it. The increase in time is more evident with the increase of NC than with the increase of B. This is a consequence of the fact that, in these experiments, the Solution Improvement Strategy is sufficiently powerful on its own to handle the p-center problem, rendering multiple bees unnecessary and a high NC time-consuming.

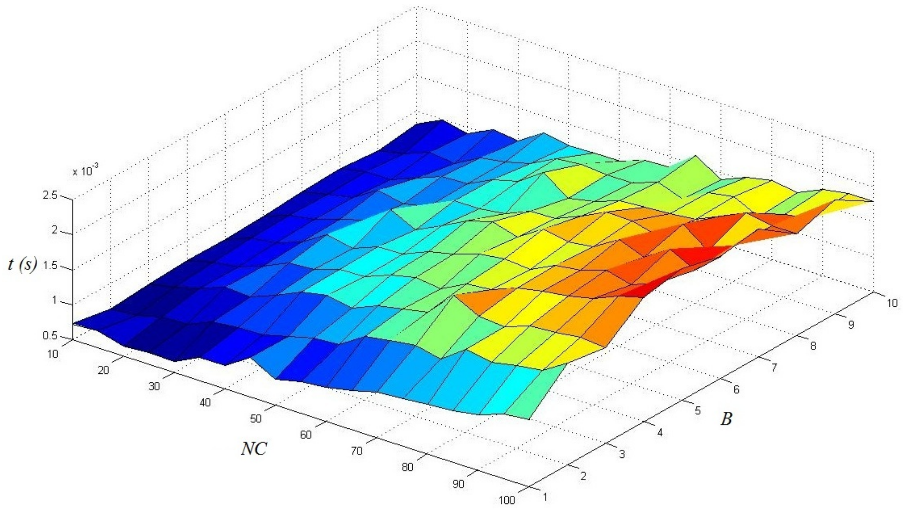


Figure 1. $\mathrm{Exp}_{\mathrm{Lin}}$ strategy for pmed01

2. The second type of graphs is like the one shown in Figure 2. This type of graph was obtained for test problems 2, 3, 4, 6, and 9, with the average finding time of the best known solution being $1.2 \cdot 10^{-2}s$. Here, between one and four bees are required, while NC is somewhat higher than for the first graphs, taking the average value of 27. The nature of the loyalty criterion $\mathrm{Exp}_{\mathrm{Lin}}$ becomes apparent as, for a fixed B, as we increase NC from 10 to 100 the obtained values first drop significantly and then rise steadily. This behavior will be even more emphasized in the fifth type, shown further below. In this case, the collaborative nature of the BCO algorithm obviously contributes to the quality of the solution.

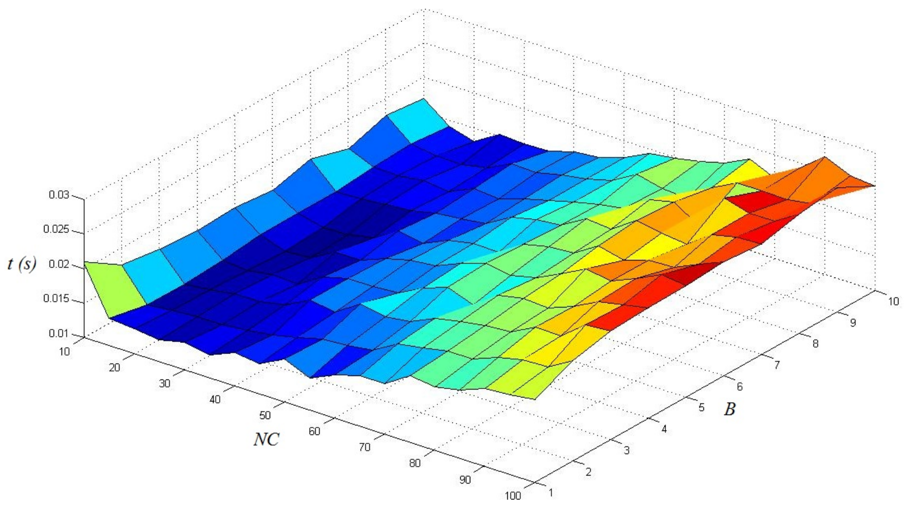


Figure 2. $\mathrm{Exp}_{\mathrm{Lin}}$ strategy for pmed04

3. The third type of graphs are like the one shown in Figure 3. This type has been obtained only for test problem 15, and for all four loyalty criteria. Here, the Solution Improvement Strategy is sufficient on its own, the choice of NC is not relevant, and we can notice a linear increase in time as the number of bees increases.

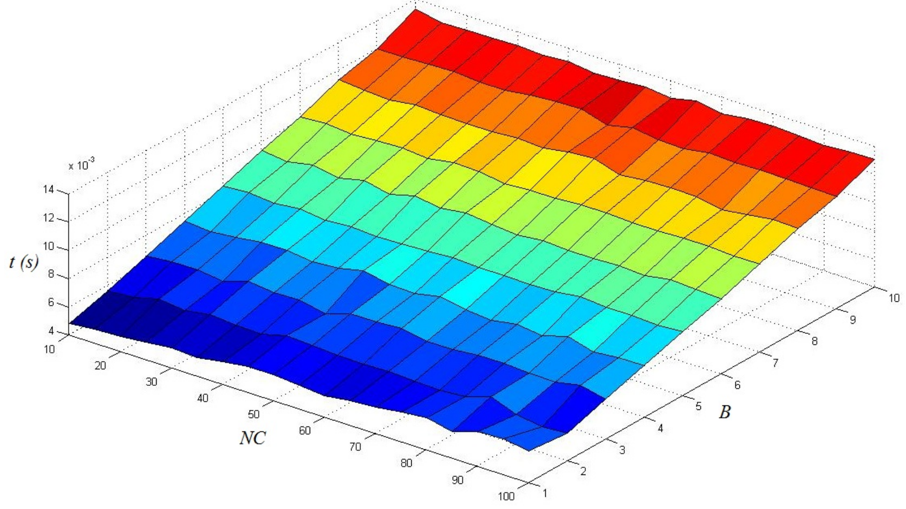


Figure 3. $\mathrm{Exp}_{\mathrm{Solo}}$ strategy for pmed15

4. The fourth type of graphs has been obtained for test problems 16 and 38, and is shown in Figure 4. It lends itself to a similar interpretation as the second type, with at most two bees and a relatively small NC required to obtain the best known solution in the least amount of time. However, this type is more sensitive to the increase of B for a fixed NC than the second type. The Solution Improvement Strategy again proves to be very powerful, but BCO does manage to provide a slight speedup in some of the experiments.

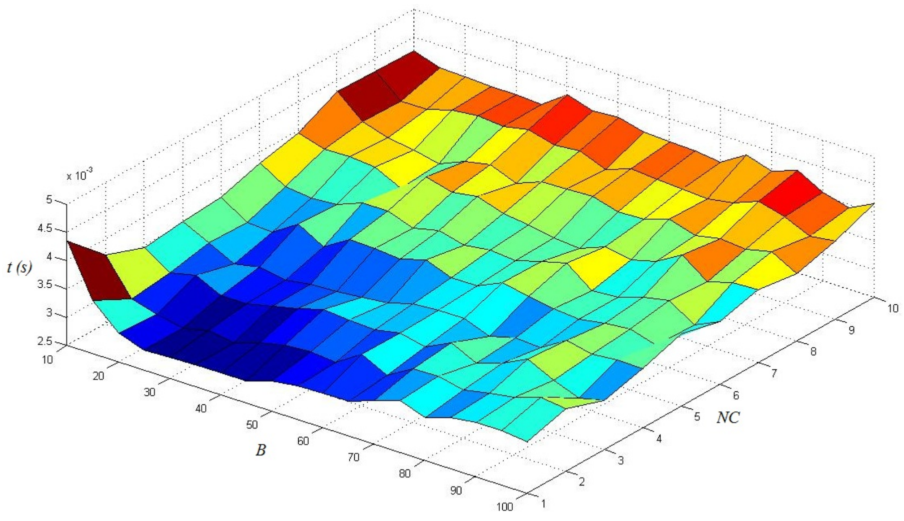
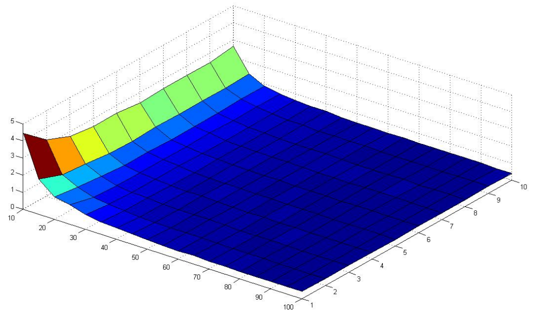
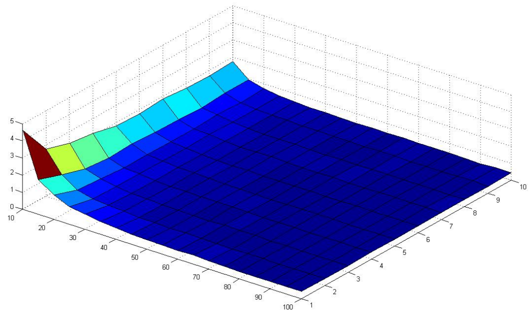


Figure 4. Exp_{Sqrt} strategy for pmed15

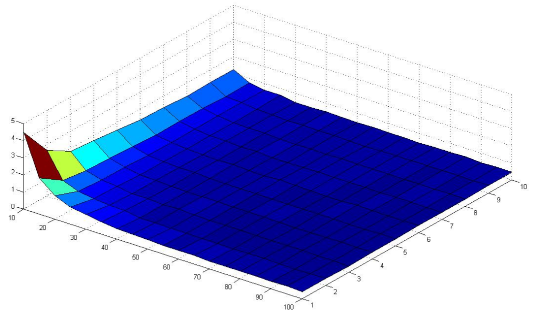
5. The fifth and final type of graphs has been the most dominant one, appearing for test problems 7, 8, 11, 12, 14, 17, 21, 26, 27, 31, and 35. Graphs for all four strategies have been shown in Figure 5, and there we can see both similarities and differences in the functioning of the loyalty criteria. The main difference between the loyalty criteria is exhibited for small values of NC ($10 \leq NC \leq 30$), as the times obtained in that region tend to be up to ten times slower than the best ones, depending on the selection of the loyalty criterion, with the Exp_{Lin} strategy being arguably the worst one, followed by Exp_{Sqrt}, Exp_{Solo}, and finally N_{V} as the best one. With the increase in NC, the obtained times drop significantly and tend to stabilize.



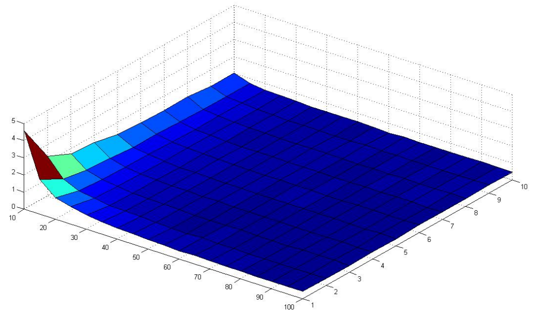
(a) Exp_{Lin} strategy for pmed14



(b) Exp_{Sqrt} strategy for pmed14



(c) Exp_{Solo} strategy for pmed14



(d) N_{V} strategy for pmed14

Figure 5. Graphs obtained for the *pmed*14 test problem

Also, as the value of NC increases, the time required to reach the best known solution decreases, regardless of the number of bees. In Table 4, we can see that the best results are obtained for only two or three bees.

	B_{min}	NC_{min}	$t_{avg}(s)$	σ	Iter
Exp_{Lin}	2	95	0.367088515	0.347066375	3.414
Exp_{Sqrt}	2	85	0.369687042	0.365915785	3.757
Exp_{Solo}	3	100	0.360333774	0.361781028	2.344
N_v	3	80	0.383342763	0.397298755	2.918

Table 4. Best obtained results for the $pmed14$ test problem

To better illustrate the difference that the number of bees makes for a fixed NC, we provide, in Figure 6, a zoomed portion of the graph (for $NC \geq 50$) for the $\mathrm{Exp}_{\mathrm{Solo}}$ strategy for the test problem 35. There, we can see that the best obtained times rise with the increase of B, but not as dramatically as for small values of NC. More generally, problems of this type require at most three bees, but a very high NC.

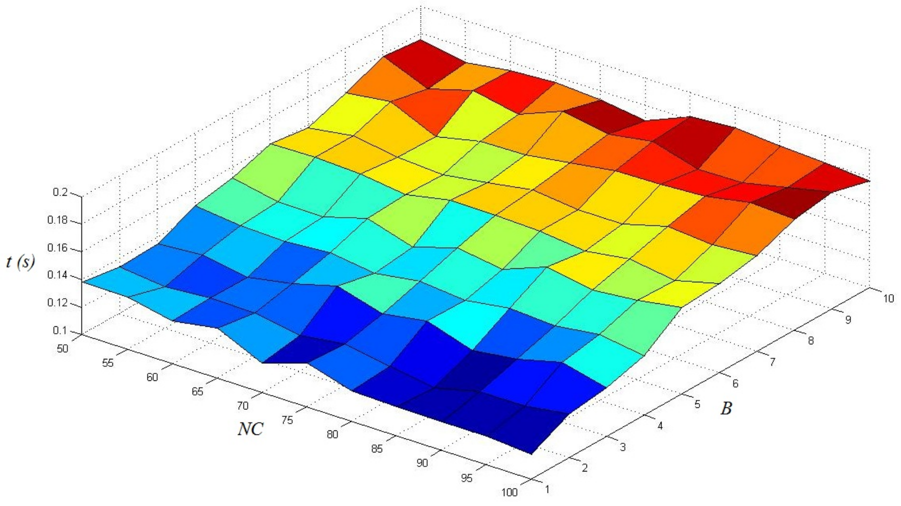


Figure 6. Zoom of the $\mathrm{Exp}_{\mathrm{Solo}}$ strategy graph for pmed35

5. Conclusions and Future Work

In this paper, we have presented an initial, yet detailed investigation of the behavior of the BCO algorithm with respect to its main parameters, on the case study of the p-center problem. We have offered three new criteria for determining the loyalty of the bees during the backward pass phase of the algorithm, and have shown that in this setting all of them outperform, both on a local and global level, the approach currently advocated in the literature. Also, we have lent further credence to the notion that the optimal number of bees is generally small, as was conjectured in [Lučić and Teodorović, 2001, Lučić and Teodorović, 2002, Lučić and Teodorović, 2003a]. Next, we have shown that the overall optimal value of the NC parameter is considerably greater than 10, contrary to the hypothesis of [Davidović et al., 2011]. In addition, we have discovered six new best-known solutions for the benchmark examples.

As for future work, we feel that two main directions exist: on the one side we plan to perform further analysis on the remaining benchmark examples for the p-center problem, with expectations of additional insights into the behavior of the BCO algorithm, while on the other side we anticipate similar analyses to be done for various different applications of BCO, such as the Vehicle Routing Problem or various scheduling problems.

Acknowledgement

This work was supported by the Serbian Ministry of Science, Education and Technological Development, projects III44006, ON174010, ON174026, and ON174033.

REFERENCES

Beasley, 1985 Beasley, J. E. (1985). A note on solving large p-median problems. *European Journal of Operational Research*, 21:270–273.

Caruso et al., 2003 Caruso, C., Colorni, A., and Aloi, L. (2003). Dominant, an algorithm for the p-center problem. *European Journal of Operational Research*, 149(1):53–64.

Chen and Chen, 2009 Chen, D. and Chen, R. (2009). New relaxation-based algorithms for the optimal solution of continuous and discrete p-center problems. *Comput. Oper. Res.*, 36(5):1646–1655.

Davidović et al., 2011 Davidović, T., Ramljak, D., Šelmić, M., and Teodorović, D. (2011). Bee colony optimization for the p-center problem. *Comput. Oper. Res.*, 38(10):1367–1376.

Davidović et al., 2009 Davidović, T., Šelmić, M., and Teodorović, D. (2009). Scheduling independent tasks: Bee colony optimization approach. In *Proc. 17th Mediterranean Conference on Control and Automation*, pages 1020–1025, Makedonia Palace, Thessaloniki, Greece.

Davidović et al., 2012 Davidović, T., Šelmić, M., Teodorović, D., and Ramljak, D. (2012). Bee colony optimization for scheduling independent tasks to identical processors. *J. Heur.*, 18(4):549–569.

Drezner, 1984 Drezner, Z. (1984). The planar two center and two median problems. *Transportation Science*, 18:451–461.

Edara et al., 2008 Edara, P., Šelmić, M., and Teodorović, D. (2008). Heuristic solution algorithms for a traffic sensor optimization problem. In *INFORMS 2008*, Washington D.C.

Hakimi, 1964 Hakimi, S. L. (1964). Optimal locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459.

Hakimi, 1965 Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475.

Hassin et al., 2003 Hassin, R., Levin, M., and Morad, D. (2003). Lexicographic local search and the p-center problem. *European Journal of Operational Research*, 151:265–279.

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

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

Lučić and Teodorović, 2003a Lučić, P. and Teodorović, D. (2003a). Computing with bees: attacking complex transportation engineering problems. *International Journal on Artificial Intelligence Tools*, 12:375–394.

Lučić and Teodorović, 2003b Lučić, P. and Teodorović, D. (2003b). Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach. In Verdegay, J. L., editor, *Fuzzy Sets based Heuristics for Optimization*, pages 67–82. Physica Verlag: Berlin Heidelberg.

Marković et al., 2007 Marković, G., Teodorović, D., and Aćimović-Raspopović, V. (2007). Routing and wavelength assignment in all-optical networks based on the bee colony optimization. *AI Commun.*, 20(4):273–285.

Minieka, 1977 Minieka, E. (1977). The centers and medians of a graph. *Operations Research*, 25(4):641–650.

Mladenović et al., 2003 Mladenović, N., Labbe, M., and Hansen, P. (2003). Solving the p-center problem with tabu search and variable neighborhood search. *Networks*, 42(1):48–64.

Pacheco and Casado, 2005 Pacheco, J. A. and Casado, S. (2005). Solving two location models with few facilities by using a hybrid heuristic: a real health resources case. *Comput. Oper. Res.*, 32(12):3075–3091.

ReVelle and Eiselt, 2005 ReVelle, C. and Eiselt, H. (2005). Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19.

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