

# A COMPARATIVE ANALYSIS OF METAHEURISTIC APPROACHES FOR SENSORS DEPLOYMENT PROBLEM ON TRANSPORT NETWORKS

Ivana Jovanović<sup>1</sup>, Miloš Nikolić<sup>2</sup>, Milica Šelmić<sup>3</sup>

<sup>1,2,3</sup> University of Belgrade, Faculty of Transport and Traffic Engineering, Vojvode Stepe 305, 11000 Belgrade, Serbia

Received 1 December 2020; accepted 20 January 2021

**Abstract:** The need for traffic flow data is essential for proper traffic management and control. Travel time estimation and early response to possible traffic incidents can be achieved with deployment of appropriate number of detectors, and placing them on optimal locations on traffic network. With more detectors located the level of accuracy of the data obtained increases, while at the same time requires more investment and maintenance costs. The detectors should be deployed in such a way to appropriately sample traffic conditions, and also provide travel time estimation with the lowest possible error. On the other hand, traffic authorities have a tendency to reduce the number of detectors located on the network in order to achieve investment savings. The proposed model provides the most suitable detector locations on a road section, while minimizing travel time estimation error with limited available funds which are considered in the model constraints. The Bee Colony Optimization metaheuristic was used to solve the sensor deployment problem, and its variant based on solution improvement, *BCOi*. The results obtained using *BCOi* metaheuristics were compared with the results obtained using the Simulated Annealing (*SA*) metaheuristics. In terms of the *CPU* time, *BCOi* outperformed the *SA* algorithm, while in comparable operating time the *BCOi* algorithm achieved better solutions to larger scale problems. The applications of both algorithms were tested on real case study data on a section of the E-763 road in the Republic of Serbia.

**Keywords:** Bee Colony Optimization, Simulated Annealing, sensors locations.

## 1. Introduction

The problem considered in this paper is the determination of the most suitable sensor locations on the section of the main road for the needs of estimating travel time, as one of the basic traffic flow parameters. The number of installed sensors affects the accuracy of the obtained data. The more sensors installed the more accurate the travel time estimation data will be.

On the other hand, the necessary funds to be invested in the procurement and maintenance of installed devices are limited, so it is in the interest of the road manager to invest as low financial resources as possible, i.e. to install as few sensors as possible. A mathematical model proposed by Edara *et al.* (2008) finds the best sensor locations by minimizing the error in estimating travel time, taking into account the limitations of available capital.

<sup>1</sup> Corresponding author: [ivana.jovanovic@sf.bg.ac.rs](mailto:ivana.jovanovic@sf.bg.ac.rs)

This problem has already been considered by the same group of authors (Jovanović *et al.*, 2019) and (Teodorović *et al.*, 2017). The problem was solved by applying the metaheuristic Bee Colony Optimization (BCO), a version based on the improvement of a complete solution, *BCOi*. In order to validate the solution obtained by *BCOi* metaheuristics, the Simulated Annealing (SA) metaheuristics was applied in this paper, as one of the well-known techniques from the group of Computational Intelligence methods.

The paper is organized as follows. After introductory considerations, a brief overview of the relevant literature is presented. The third chapter contains mathematical formulation of the problem, followed by a review of the metaheuristics of *BCOi* and SA. The method of data collection is described in the fifth chapter. The comparison of the achieved results obtained by applying two metaheuristics is presented in the sixth chapter. Concluding remarks and directions of future work make the seventh, last, chapter.

## 2. Literature Review

Bianco *et al.* (2006) determined the minimum number of sensors for counting traffic at network nodes, in order to determine the flow at all network links. The authors proved that the problem of locating sensors on network is a NP complete problem. In that sense, the use of metaheuristic algorithms for solving the mentioned problem is completely reasonable.

Based on the literature review in which the problem of locating sensors on the network has been solved, it can be concluded that numerous papers deal with the problem of

finding optimal sensor locations on transport networks. The authors applied different techniques to solve this combinatorial problem. Also, it can be noticed that the use of metaheuristics is not widespread, except for genetic algorithms (Edara *et al.*, 2008; Liu and Danczyk, 2009; Edara *et al.*, 2011; Kim *et al.*, 2011) and constructive version of BCO (Teodorović *et al.*, 2010). The constructive version of the BCO algorithm was used in the paper of Teodorović *et al.* (2010), while in the papers (Jovanović *et al.*, 2019) and (Teodorović *et al.*, 2017) a version of the BCO algorithm based on solution improvement - *BCOi* was proposed.

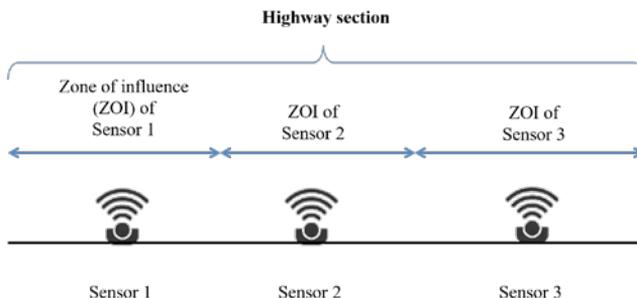
The constructive version of the BCO algorithm, as the first proposed BCO version, has been successfully applied to a number of combinatorial optimization problems. However, with the increasing dimensions of the problem, the constructive version failed to remain competitive with the best results achieved in the literature (Teodorović *et al.*, 2010; Davidović *et al.*, 2011). In recent years, the *BCOi* version of the algorithm has been compared with other metaheuristic approaches, such as: genetic algorithms, simulated annealing, particle swarm optimization, taboo search, and variable neighborhood search and showed very promising results.

## 3. Mathematical Formulation

The problem considered in this paper is determining the locations of point detectors for estimating travel time along a highway section. Edara *et al.* (2008) defined two types of travel time on a highway: Ground Truth Travel Time - *GTTT* and Estimated Travel Time - *ETT*. The difference between the time of exit and entry on the section for an individual floating, test, vehicle makes its

actual travel time  $GTTT$ . The Estimated Travel Time is calculated using the speed values measured by sensors distributed on the road section. The entire section is divided into Zones of Influence -  $ZOI$  of individual sensors. The  $ZOI$  of one sensor is

defined as half the distance to the adjacent detector on both sides along the traffic flow, as shown on Figure 1. Travel time for the entire section is estimated by calculating travel time for each defined Zone Of Influence ( $ZOI_i, i=1,2,\dots,n$ ).



**Fig. 1.**  
Highway Section and Sensors' Zones of Influence  
Source: (Jovanović, 2020)

We introduce the following notation:

- $n$  - Number of detectors on the highway section (= number of zones of influence);
- $i$  - Index for the  $i$ -th detector;
- $x_{PLi}$  - Position of the  $i$ -th detector, measured from the start of the highway section (decision variable in the objective function);
- $L$  - Length of the highway section;
- $K$  - Total number of test drives;
- $ZOI_i$  - Length of zone of influence of the  $i$ -th detector ( $\sum_{i=1}^n ZOI_i = L$ );
- $V_i$  - Speed reported by the  $i$ -th detector;
- $TT_i$  - Travel time for each  $ZOI_i$  ( $TT_i = \frac{ZOI_i}{V_i}$ );

- $ETT$  - Estimated travel time for the highway section;
- $GTTT$  - Ground truth travel time for the highway section ( $ETT = \sum_{i=1}^n TT_i$ );
- $\varepsilon$  - Travel time estimation error.

We can express  $ZOI_i$  in terms of the decision variable  $x_{PLi}$  as follows:

$$ZOI_i = \begin{cases} \frac{x_{PL1} + x_{PL2}}{2} & i = 1 \\ \frac{x_{PLi+1} - x_{PLi-1}}{2} & i = 2, 3, \dots, n - 1 \\ L - \frac{x_{PLn} + x_{PLn-1}}{2} & i = n \end{cases} \quad (1)$$

The goal is to find a placement of point detectors on highways in such a way to minimize the travel time estimation error.

$$\varepsilon = |ETT - GTTT| = |\sum_{i=1}^n TT_i - GTTT| \quad (2)$$

$$\varepsilon = \left| \sum_{i=1}^n \frac{ZOI_i}{V_i} - GTTT \right| \quad (3)$$

The relative error equals:

$$\varepsilon_r = \frac{\varepsilon}{GTTT} \quad (4)$$

Cumulative relative error CRE equals:

$\varepsilon_k$  - travel time estimation error for the  $k$ -th profile;  
 $GTTT_k$  - ground truth travel time for the  $k$ -th profile.

$$\varepsilon_{CRE} = \sum_{k=1}^K \frac{\varepsilon_k}{GTTT_k} \quad (5)$$

Where:

$k$  - index for the speed profile (i.e., GPS travel time run);

Upon substituting relations (1), (2), (3), and (4) into relation (5), the cumulative relative error CRE equals (Edara et al., 2008):

$$\varepsilon_{CRE} = \sum_{k=1}^K \left| \frac{\left( \frac{x_{PL_1} + x_{PL_2}}{2V_{1k}} \right) + \sum_{i=2}^{n-1} \left( \frac{x_{PL_{i+1}} - x_{PL_{i-1}}}{2V_{ik}} \right) + \left( \frac{L - \frac{x_{PL_n} + x_{PL_{n-1}}}{2}}{V_{nk}} \right) - GTTT_k}{GTTT_k} \right| \quad (6)$$

Subject to:

$$x_{PL_i} \geq 0 \quad \text{for all } i = 1, 2, \dots, n \quad (7)$$

Where,  $V_{ik}$  is the speed reported by the  $i$ -th sensor for the  $k$ -th profile. The cumulative relative error represents the objective function. In equation (6), the numerators of the first three terms are linear combination of two decision variables. The denominator has the speed term  $V_{ik}$ , which is a function of the distance  $x_{PL_i}$ . This means that the objective function is non-linear. The problem (6)-(7) is combinatorial by its nature (Edara et al., 2008)

## 4. Descriptions of used Metaheuristic Algorithms

### 4.1. BCOi

A BCO version based on solution improvement was first used in Davidović et al. (2011). In later literature, it was also used in the works of Nikolić and Teodorović (2013, 2014, 2015, and 2019), Todorović and Petrović (2013), Nikolić et al., (2015), Jovanović et al., (2017), Jovanović et al.,

(2019). Since the application of BCOi metaheuristics to solve the problem of locating sensors on transport networks is explained in detail in previous works (Jovanović et al., 2019) and (Teodorović et al., 2017), only the pseudocode of this version of BCO followed by the brief description of BCOi steps will be given in this paper.

The BCOi algorithm begins by determining the initial solution (steps 1 and 2 of Algorithm 1) which is saved in step 3 as the currently best known solution. Step 4 starts the iteration loop. In steps 5 and 6, the solutions are assigned to the bees, and the solution will be modified during the iteration. The forward pass is presented in steps from 8 to 11. Within the forward pass, each bee makes  $NC$  changes of its solution. When all the bees have made one modification of the solution, it is checked whether this solution is better than the best known one that has been found so far (steps 12 and 13 of Algorithm 1). The backward pass is presented in steps from 14 to 19. Within the backward pass, the loyalty of the bees is checked and, if the bee is not loyal, the bee to be followed is selected (Nikolić, 2015).

### Algorithm 1

Pseudocode of the BCOi Algorithm

BCOi procedure (input B, IT, NP, NC, output S)	
1:	Determine the initial solution.
2:	Evaluate the initial solution.
3:	$S \leftarrow$ Save the initial solution as the currently best known.
4:	<b>for</b> $j = 1$ <b>to</b> $IT$ <b>do</b>
5:	<b>for</b> $i = 1$ <b>to</b> $B$ <b>do</b>
6:	bee $i \leftarrow$ assign the initial solution to bee $i$ .
7:	<b>for</b> $k = 1$ <b>to</b> $NP$ <b>do</b>
8:	<b>for</b> $r = 1$ <b>to</b> $NC$ <b>do</b>
9:	<b>for</b> $i = 1$ <b>to</b> $B$ <b>do</b>
10:	Make one modification of the solution of bee $i$ .
11:	Evaluate the solution of bee $i$ .
12:	<b>if</b> the best solution of all bees is <b>better than</b> the solution $S$ <b>then</b>
13:	$S \leftarrow$ Save the best solution of all bees as the currently best known.
14:	Normalization of the objective function values of all bees.
15:	<b>for</b> $i = 1$ <b>to</b> $B$ <b>do</b>
16:	Determine if the bee $i$ is loyal.
17:	<b>for</b> $i = 1$ <b>to</b> $B$ <b>do</b>
18:	<b>if</b> the bee $i$ is not loyal <b>then</b>
19:	Determine one of the loyal bees that the bee $i$ will follow.

Source: (Nikolić, 2015)

#### 4.1.1. Evaluation of the Solution and Determination of Bee Loyalty

After the solution is modified, the forward pass is completed, and each bee has one complete solution. For each complete solution, evaluation of the solution quality is done based on relation (6). When the quality of each complete solution is determined, at the start of the backward pass, it is necessary to determine which bee will stay loyal to its solution, and which bee will reject its solution and accept one of the other loyal bees' solutions. Loyalty is determined based on normalized values for quality of the solution. If  $C_b$  ( $b = 1, 2, \dots, B$ ) is value for criterion function of complete solution of  $b$ -th bee (relation (6)), normalized value  $O_b$  in case of minimization is calculated using following relation:

$$O_b = \frac{C_{max} - C_b}{C_{max} - C_{min}}, b = 1, 2, \dots, B \quad (8)$$

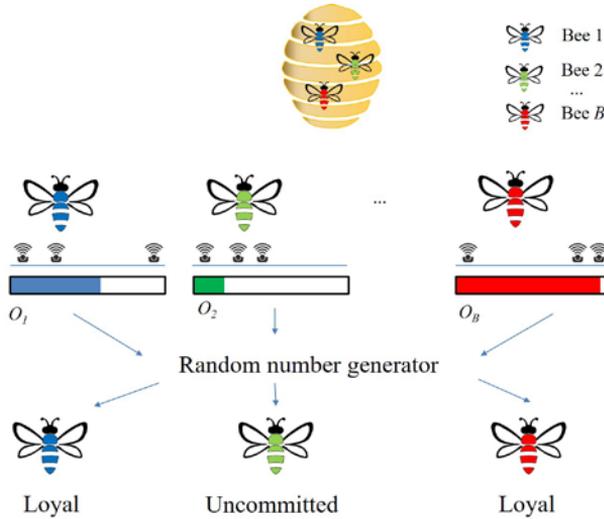
Where:

$C_{min}$  and  $C_{max}$  are complete solutions which respectively correspond to minimal and maximal value of objective function obtained from all the bees. Probability that the  $b$ -th bee will, at the beginning of the next forward pass, be loyal to its solution discovered so far is calculated as follows:

$$p_b = e^{-(O_{max} - O_b)}, b = 1, 2, \dots, B \quad (9)$$

Where:

$O_b$  - normalized value of objective function of complete solution for the  $b$ -th bee; and  $O_{max}$  - maximum of all normalized solution values which are compared. An example of loyalty testing is shown on figure 2.



**Fig. 2.**  
Loyalty Testing  
Source: (Jovanović et al., 2019)

#### 4.1.2. Recruitment of Uncommitted Bees

If, at the beginning of next forward pass, a bee does not want to expand/improve its previous complete solution, it then has to select and follow the solution from another bee. We assume here that the probability the recruiter's partial solution will be chosen by any uncommitted bee equals:

$$p_b = \frac{o_b}{\sum_{k=1}^R o_k}, b = 1, 2, \dots, R, \quad (10)$$

Where:

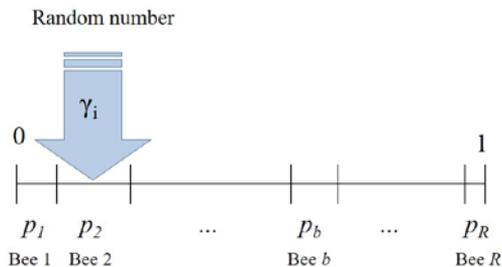
$O_k$  - objective function value of the  $k$ -th advertised solution;

$R$  - number of recruiters.

Using a random number generator, every uncommitted bee joins one recruiter. The recruitment process is shown on figure 3.

Finally, the iteration is completed after the entire forward and backward passes are completed  $NP$  times. The global best solution is accepted and all of the remaining data structures are deleted. Using  $BCO_i$  algorithm, iterations are repeated until one of the stopping criteria is fulfilled.

Stopping criteria can be different: maximum number of forward/backward passes, maximum number of forward/backward passes without improving the value of the objective function (without correcting the currently achieved best solution), number of iterations, maximum allowed  $CPU$  time, etc.



**Fig. 3.**  
 Recruitment Process  
 Source: (Jovanović, 2020)

### 4.2. Simulated Annealing

The Simulated Annealing (SA) metaheuristic algorithm is based on an analogy with the annealing process of metals (e.g., steel) in metallurgy. This process involves heating the metal to a liquid state (melting) and its gradual cooling (annealing). Gradual, controlled cooling achieves that the crystal structure of the metal reaches a minimum energy, which contributes to achieving higher metal strength.

The metaheuristics of SA for solving complex combinatorial optimization problems was proposed by Kirkpatrick *et al.* (1983) and Černý (1985), independently of each other.

The idea of the SA technique is to explore the solution space in the neighborhood of the current solution, with small changes being made in a random manner. The achieved energy difference  $\Delta E$  between the previous and the new solution is calculated. If a better solution is reached ( $\Delta E < 0$ ), a new solution is adopted. Otherwise, if  $\Delta E > 0$ , the achieved worse solution is not rejected automatically, but the so-called “energy jumps” are permitted, that also occur in thermodynamic processes. These jumps were described by Metropolis *et al.* (1953) using

the Boltzmann distribution to calculate the probability of this energy jump (Teodorović, 2007). This allowing “jumps” avoids the solution being trapped in the local minimum.

The probability  $P$  that the energy  $\Delta E$  will increase if the system is at temperature  $T$  is calculated (Teodorović 2007):

$$P = e^{-\frac{\Delta E}{T}} \tag{11}$$

Introducing the analogy, the particles configuration of the crystal structure implies one feasible solution of the combinatorial optimization problem, and the energy of the physical system can be considered as the value of the objective function. The change in energy  $\Delta E$  then refers to the change in the value of the target function. The probability that a new solution whose objective function is worse will still be accepted as the new current best solution, is then calculated as follows:

$$P = e^{-\frac{f(j)-f(i)}{T}} \tag{12}$$

Where  $f(j)$  and  $f(i)$  are the objective function values for the old  $i$  and new solution  $j$ , and  $T$  is a control parameter that is equivalent to the temperature at which the system is located.

This probability  $P$  is now compared with the generated random number  $R \in [0,1]$  and, if  $R > P$ , the new solution is adopted as the new best, while otherwise it is rejected (Teodorović 2007). The process of lowering the energy at a given temperature is repeated until the so-called thermal equilibrium is reached, which means that after a large number ( $N(t)$ ) of random perturbations of the solution, no reduction in energy is obtained. Then, the temperature drops and the search for a new thermal equilibrium continues. Lowering the temperature is performed as in the relation:

$$T(t) = \alpha T(t - 1) \tag{13}$$

Where  $T(t)$  is the new temperature,  $T(t - 1)$  is the previous temperature, and  $\alpha$  is the temperature lowering parameter. The temperature lowering parameter  $\alpha$

is from the interval  $[0,1]$  and, if its value is small, it means that the temperature drop is fast, which can lead to a drop to the local minimum, which is analogous to metal cracking in physical systems. If the parameter is too large, it leads to too slow temperature decreasing; temperatures are high for a long time, which leads to too often worse solutions being accepted as the best, due to the probability depending on the temperature in equation (12).

The whole procedure is repeated until the stopping criterion is met, which can be the maximum number of iterations, the maximum number of repetitions of the best solution, the total execution time of the algorithm etc.

The SA algorithm can be represented by a pseudocode given in Algorithm 2.

**Algorithm 2**

*Pseudocode of the Simulated Annealing Algorithm*

<b>Simulated annealing procedure</b>	
1:	Setting the initial solution.
2:	Set the initial temperature $T$ .
3:	Set the temperature counter $t = 0$ .
4:	<b>repeat</b>
5:	Set the repetition counter $n = 0$ .
6:	<b>repeat</b>
7:	Generate state $j$ , in the neighborhood of the previous state $i$ .
8:	Calculate $\Delta E = f(j) - f(i)$
9:	<b>if</b> $\Delta E < 0$ <b>then</b> $i = j$
10:	<b>else if</b> random (0,1) < exp ( $-\Delta E / T$ ) <b>then</b> $i = j$
11:	<b>until</b> $n = N(t)$
12:	$t = t+1$
13:	$T = T(t)$
14:	<b>until</b> satisfied stopping criterion.

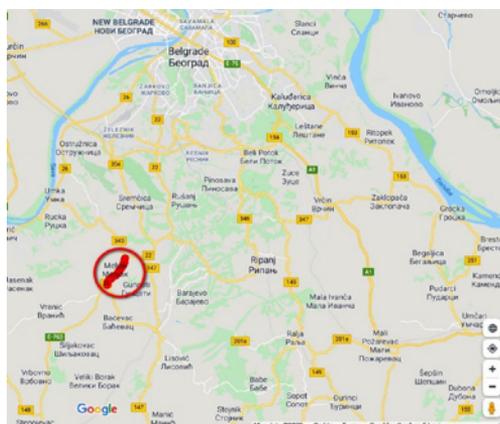
Source: (Eglese, 1990; Jovanović, 2017)

## 5. Data Collection

A test vehicle equipped with a *Garmin GPSMAP 64s* GPS device was used to measure speeds within the research. The floating vehicle method was used. A floating vehicle moves through the traffic flow, imitating other vehicles in the traffic flow. In that sense, the driver tries to overtake the same number of vehicles with the number of vehicles that preceded him. The vehicle is entering in the traffic flow approximately 400m before the start of the section where the recording is performed, in order to ensure that the vehicle reaches the speed of the traffic flow before the start of the recording. During the drive, data on the position and speed of the vehicle are recorded every second, until the end of the section. Recording was performed in both directions during peak and off-peak traffic loads. From March to April 2015, a total of 200 rides were performed, 100 in each direction. In order

to ensure similar accuracy of *GPS* devices, recordings were made on sunny days, so that clouds would not affect the reduced accuracy of measurement results. The data used in the paper were collected for the purposes of preparing the doctoral dissertation Vidas (2017).

Data were collected on the section of the Ibar highway in the length of 3000 m. The Ibar highway is a road of IB order, and in the reference system of roads in Republic of Serbia it stands the number 22. The Ibar highway connects Belgrade with Čačak, and further across road 23 with the Republic of Montenegro, it is part of the European road E-763. The section of interest for testing is the section that passes through Meljak, between the intersections Velika Moštanica and Bačevac. The section is located in the suburban Belgrade municipality of Barajevo, and is about 20 km away from the center of Serbian capital Belgrade (Figure 4).



**Fig. 4.**  
Section of Highway E-763 in Serbia (Map data © Google)  
Source: (Jovanović, 2020)

## 6. Comparative Analysis of BCOi and SA

For the purpose of comparing the results obtained by solving the problem of locating the sensors by applying the two metaheuristics *BCOi* and *SA*, it was necessary to bring the algorithms to a comparable level. The algorithms solved the same problems, starting from the same initial solution. The problems of locating 2 to 20 sensors on the observed section of the road were solved, in two periods: the peak with 38 drives, and off-peak with 62 drives.

All problems were solved on a laptop with the following features: Intel (R) Core (TM) i3-4000M, 2.40GHz, and 4.00GB of RAM (Jovanović *et al.*, 2019).

After a detailed parametric analysis, a combination of parameters was selected for the application of *BCOi* metaheuristics: 15 bees (*B*), 15 forward passes (*NP*), one solution modification (*NC*) in each forward pass. The maximum number of iterations (*IT*) was implemented for the stopping criterion, specifically  $IT = 1000$ . For the application of *SA* metaheuristic the following combination of parameters was chosen: initial temperature  $T = 250$  degrees, number of repetitions at one temperature  $N(t) = 20000$  and temperature lowering parameter  $\alpha = 0.98$ .

The most significant difference between these two algorithms is the speed of

obtaining an acceptably good solution. The obtained results of computer time consumption indicate that the *BCOi* algorithm is very efficient, considering that for a maximum of about 1.8 s the *CPU* time found the solution of the most complex example, locating 20 sensors in the off-peak period, where the data consists of 62 test drives. The operating time of the *SA* algorithm in the parameter analysis was defined at a much higher value, it was 2 minutes. In order to make the operating times comparable, the results achieved by the *SA* algorithm during the operation of 0.2 minutes (12 s) and 0.03 minutes (1.8 s, which time was achieved by the *BCOi* algorithm) were compared.

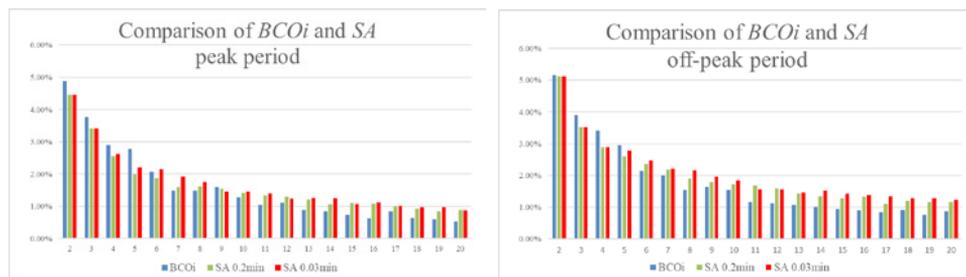
Table 1 and Figure 5 show the results of a comparative analysis of the two algorithms. The values of the cumulative relative error of the estimated travel time per drive, expressed as a percentage, were compared.

Comparing the data in Table 1, it can be concluded that the *SA* algorithm achieves better results than *BCOi* for smaller problem dimensions, for values of the number of located sensors up to 6 for peak and up to 7 for off-peak period, during computer operation of 12 s. For comparable computer run time, the *BCOi* algorithm achieves better results than the *SA* algorithm, starting with 6 located sensors for both peak and off-peak traffic periods.

**Table 1**

Cumulative Relative Error per Drive for Algorithms BCOi and SA

Number of Sensors	Peek Period (38 Drives)			Off-peek Period (62 Drives)		
	BCOi	SA 0,2 min	SA 0,03 min	BCOi	SA 0,2 min	SA 0,03 min
2	4.88%	4.47%	4.47%	5.16%	5.12%	5.12%
3	3.77%	3.42%	3.42%	3.91%	3.52%	3.52%
4	2.90%	2.55%	2.62%	3.42%	2.90%	2.90%
5	2.77%	1.99%	2.20%	2.95%	2.61%	2.77%
6	2.08%	1.87%	2.14%	2.15%	2.36%	2.47%
7	1.49%	1.59%	1.92%	2.00%	2.19%	2.21%
8	1.49%	1.62%	1.76%	1.56%	1.89%	2.16%
9	1.61%	1.54%	1.46%	1.64%	1.80%	1.96%
10	1.28%	1.41%	1.46%	1.56%	1.72%	1.84%
11	1.04%	1.34%	1.40%	1.16%	1.67%	1.57%
12	1.11%	1.29%	1.23%	1.12%	1.59%	1.57%
13	0.89%	1.21%	1.27%	1.08%	1.43%	1.47%
14	0.84%	1.05%	1.25%	1.01%	1.35%	1.52%
15	0.74%	1.09%	1.07%	0.95%	1.29%	1.41%
16	0.62%	1.09%	1.13%	0.90%	1.34%	1.37%
17	0.85%	1.00%	1.02%	0.84%	1.10%	1.35%
18	0.64%	0.93%	0.97%	0.91%	1.20%	1.29%
19	0.60%	0.85%	0.97%	0.77%	1.16%	1.29%
20	0.52%	0.89%	0.88%	0.87%	1.15%	1.24%



**Fig. 5.**

Comparison of BCOi and SA (Relative Cumulative Error per Drive)

## 7. Conclusions

The proposed model of determining the most suitable sensor locations can be used primarily for planning purposes as an assistance tool for local and state transportation authorities. The model enables fast comparison of a large number of different solutions for detector placement

and observes performances such as number of detectors used and travel time estimation error.

The performed analysis showed that the proper placement of the detectors has high influence on the travel time estimate, as well as total costs of detector deployment and maintenance. Results show that 20

sensors for peak period and 19 for off-peak period result in the least value of travel time cumulative relative error per drive (Jovanović *et al.*, 2019).

The problem of determining the most suitable sensor locations was solved by applying two known metaheuristics on real data collected on the section of the road E-763 in the Republic of Serbia. The results obtained using *BCOi* metaheuristics in the previous work of the same authors were compared with the results obtained using Simulated Annealing metaheuristic.

Comparing the two metaheuristics, *BCOi* proved to be competitive in comparison with SA. In terms of execution speed, the *BCOi* algorithm outperformed the SA algorithm. In comparable algorithm run times, *BCOi* has achieved better solutions to larger problems.

In the future work, it would be interesting to conduct a sensitivity analysis of certain parameters used in the proposed model solved by applying *BCOi*. Changing the way of obtaining the initial solution, varying the number of modifications of the solution, applying different stopping criteria could possibly contribute to better obtained solutions or their faster achievement.

## Acknowledgements

This research was partially supported by the Ministry of Education, Science and Technological Development, Government of the Republic of Serbia, through the project TR36002 for the period 2011-2020.

## References

- Bianco, L.; Confessore, G.; Gentili, M. 2006. Combinatorial aspects of the sensor location problem, *Annals of Operations Research* 144: 201–234.
- Černý, V. 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications* 45: 41–51.
- Davidović, T.; Ramljak, D.; Šelmić, M.; Teodorović, D. 2011. Bee colony optimization for the p-center problem, *Computers and Operations Research* 38(10): 1367–1376.
- Edara, P.; Guo, J.; Smith, B. L.; McGhee, C. 2008. Optimal placement of point detectors on Virginia's highways: case studies of northern Virginia and Richmond. Report VTRC08-CR3. Virginia Transportation Research Council, Richmond, VA.
- Edara, P.; Smith, B.; Guo, J.; Babiceanu, S.; McGhee, C. 2011. Methodology to identify optimal placement of point detectors for travel time estimation, *Journal of Transportation Engineering* 137(3): 155–173.
- Eglese, R.W. 1990. Simulated Annealing: A tool for Operational Research, *European Journal of Operational Research* 46(3): 271–281.
- Jovanović, A. 2017. *Choice of signal timing for traffic control by bee colony optimization*. Doctoral dissertation, Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade. 131p. (in Serbian).
- Jovanović, I. 2020. *Sensors selection and deployment on transport networks using operations research methods*. Doctoral dissertation, Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade. 169p. (in Serbian).

- Jovanović, A.; Nikolić, M.; Teodorović, D. 2017. Area-wide urban traffic control: a bee colony optimization approach, *Transportation Research Part C: Emerging Technologies* 77: 329–350.
- Jovanović, I.; Šelmić, M.; Nikolić, M. 2019. Metaheuristic approach to optimize placement of detectors in transport networks – case study of Serbia, *Canadian Journal of Civil Engineering* 46(3): 176–187.
- Kim, J.; Park, B.; Lee, J.; Won, J. 2011. Determining optimal sensor locations in freeway using genetic algorithm-based optimization, *Engineering Applications of Artificial Intelligence* 24: 318–324.
- Kirkpatrick, S.; Gelatt Jr., C.D.; Vecchi, M.P. 1983. Optimization by Simulated Annealing, *Science* 220(4598): 671–680.
- Liu, H.; Danczyk, A. 2009. Optimal sensor locations for freeway bottleneck identification, *Computer-Aided Civil and Infrastructure Engineering* 24 (8): 535–550.
- Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A. 1953. Equation of state calculations by fast computing machines, *Journal of Chemical Physics* 21: 1087–1092.
- Nikolić, M.; Teodorović, D. 2013. Transit network design by bee colony optimization, *Expert Systems with Applications* 40(15): 5945–5955.
- Nikolić, M.; Teodorović, D. 2014. A simultaneous transit network design and frequency setting: Computing with bees, *Expert Systems with Applications*, 41 (16): 7200–7209.
- Nikolić, M.; Teodorović, D. 2015. Vehicle rerouting in the case of unexpectedly high demand in distribution systems, *Transportation Research Part C: Emerging Technologies* 55: 535–545.
- Nikolić, M.; Teodorović, D. 2019. Mitigation of disruptions in public transit by Bee Colony Optimization, *Transportation Planning and Technology* 42(6): 573–586.
- Nikolić, M.; Teodorović, D.; Vukadinović, K. 2015. Disruption management in public transit: the bee colony optimization approach, *Transportation Planning and Technology* 38(2): 162–180.
- Nikolić, M. 2015. *Disruption management in transportation by the Bee Colony Optimization metaheuristic*. Doctoral dissertation, Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade. 159p. (in Serbian).
- Teodorović, D. 2007. *Transportation networks*. Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade. 429 p. (in Serbian).
- Teodorović, D.; Šelmić, M.; Praveen, E. 2010. Bee colony optimization approach to optimize placement of traffic sensors on highways. In *Proceedings of the 13th International Conference on Traffic Science*, 27.-28. May 2010, Portorož, Slovenia.
- Teodorović, D.; Šelmić, M.; Nikolić, M.; Jovanović, I.; Vidas, M. 2017. Metaheuristic approach for detector locations in transport networks. In *Proceedings of SYM-OP-IS 2017: XLIV Symposium on Operational Research*, Zlatibor, 723–728. (in Serbian).
- Todorović, N.; Petrović, S. 2013. Bee colony optimization algorithm for nurse rostering, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43(2): 467–473.
- Vidas, M. 2017. *Impact of access control on capacity and level of service of two-lane highways*. Doctoral dissertation, Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade. 203p. (in Serbian).