International Journal of Transportation
Vol.2, No.1 (2014), pp.77-94
http://dx.doi.org/10.14257/ijt.2014.2.1.05

An Evaluation of Heuristic Methods for Determining Optimal Link Capacity Expansions on Road Network

Ozgur Baskan

Department of Civil Engineering, Pamukkale University, Denizli, 20070, Turkey
obaskan@pau.edu.tr

Abstract

Since the bilevel formulation of the Continuous Network Design Problem (CNDP) has the characteristic of a non-convex optimization problem, heuristic methods are usually the preferred option for solving it. On the other hand, the computation time is crucial importance for solving the CNDP because the algorithms implemented on real sized networks require solving traffic assignment model many times, in which algorithms spend more time in comparison with the other parts of the solution process. Therefore, solving the CNDP with less number of traffic assignments can be assessed as one of the most important topics in the transportation field. Thus, the paper deals with analyzing the performance of recently developed heuristic methods in solving the CNDP. In this study, the capability of Harmony Search (HS), Artificial Bee Colony (ABC) and Differential Evolution (DE) algorithms for solving the CNDP is evaluated, and numerical calculations are performed on example test networks. The results obtained through the HS, ABC and DE algorithms on 18-link network are compared with those generated by two different heuristic methods, which are available in a previous study. Additionally, the performance of the proposed heuristics is compared on Sioux Falls city network with other major algorithms available in the literature. Numerical examples have clearly indicated that the DE shows good performance in comparison with the proposed algorithms in terms of both objective function value and required computational effort.

Keywords: *Differential evolution; harmony search; artificial bee colony; continuous network design problem.*

1. Introduction

The Continuous Network Design Problem (CNDP) has been comprehensively studied in the literature for more than three decades. The CNDP deals with determining the optimal link capacity expansions by way of minimizing the network travel cost considering related budget. Since multiple objectives exist in the solution process of the CNDP, it can be formulated as bilevel programming model. It is clear that finding the global or near global optimum solution is a crucial importance in the CNDP because of the non-convex structure of the bilevel formulation. On the other hand, bilevel formulation of the CNDP is difficult to solve since the determining of the upper level objective function requires solving the lower level problem for each feasible set of upper level decision variables. In the CNDP, upper level objective function can be solved by minimizing the sum of network travel cost and investment cost of selected link capacity expansions. In the lower level, traffic assignment model is formulated, which can be static or dynamic.

As for the evolution of solution methods to the CNDP, the first attempt to solve the CNDP has been performed by [1] using Hooke-Jeeves (HJ) and Powell's method. In their study, the network design problem has been formulated as a nonlinear unconstrained optimization problem. In addition, the effect of type of investment function was also investigated. It has been found that the performance of two methods is approximately the

ISSN: 2287-7940 IJT

same for convex investment cost function whilst the HJ is better than the Powell's method in the case of using concave investment function. A new formulation to solve the CNDP by expressing the traffic equilibrium problem with differential constraints has been presented by [2]. Equilibrium Decomposition Optimization (EDO) algorithm for solving the CNDP was proposed by [3]. Results showed that the EDO is more efficient than the HJ algorithm. The efficiency of the proposed approach relies on decomposition of the problem into a set of corresponding sub optimization problems. The other advantage of the EDO was reported that the computational cost of the proposed model does not depend on the number of links which are being considered for capacity expansion. Following the above mentioned studies, efficient heuristic methods for solving the CNDP have been developed by [4]. It has been reported that the proposed heuristic methods are robust and do not require initial lower and upper bounds on the decision variables. In addition, sensitivity-based heuristic algorithms were proposed for the CNDP in different studies [5-8]. Furthermore, Simulated Annealing (SA) approach for solving the CNDP with variational inequality constraints was introduced by [9]. It has been found that the SA is more suitable for problems for which it is important to be found an exact global solution since the computational effort of the SA is intensive. Therefore, it was emphasized that its use by combining the projection method is preferable option for solving the CNDP. Unlike using classical lower level solution as in most studies, the Stochastic User Equilibrium (SUE) assignment procedure was embedded to the CNDP in [10]. In order to show the advantage for the use of SUE assignment, the generalized reduced gradient and sequential quadratic programming methods were combined with the SUE based on the logit model. The proposed solution procedure was tested on example test networks, and it has been found that the differentiable and tractable version of the CNDP could be created.

In order to avoid the disadvantages for the use of bilevel formulation, the CNDP was formulated as a single level problem by [11]. In addition to this improvement, a complicated implicit constraint was included into single level formulation. It has been shown that the value and the derivative of this constraint can be easily obtained, and this novelty encourages the application of existing nonlinear programming algorithms for the CNDP. The gradient based methods to solve the CNDP have been proposed by [12] and they were tested on several test networks. It has been found that the presented methods are able to produce better results in terms of the robustness to the initial values in comparison with the other methods and they also have computational efficiency for solving equilibrium assignment problems. Following the study made by [12], a new approach considering both symmetric and asymmetric user equilibria was developed by [13] in order to solve the CNDP. The original bilevel model was converted into a single level formulation by means of adding some constraints to the lower level problem and a relaxation scheme was proposed to solve it. The proposed solution algorithm was tested on different test networks and promising results were obtained. The SA and Genetic Algorithm (GA) methods to solve the CNDP were introduced by [14]. They emphasized that quality of the results are dependent on the demand level. Addition, it has been reported that the results obtained by [14] are different from those produced by [15] where the bilevel model was linear. For large scale application in solving the CNDP, GA was used by [16] because of its simplicity. The CNDP was solved on a large sized network under various budget scenarios. Results showed GA's ability to handle large network design problems. Afterwards, a mixed-integer linear program formulation was developed to determine the global optimal solution of the CNDP by [17]. Although the finding of the global optimum solution is guaranteed in the mentioned study, computational effort depends on the linearization of the link travel time function and partition scheme. To overcome the disadvantages of using the bilevel formulation in solving the CNDP, it was converted into a single level formulation and global optimization method was proposed for solving it by [18]. Recently, the CNDP has been solved using harmony search, cuckoo search and differential evolution based algorithms, respectively [19-21]. In addition, a comprehensive review of urban transportation network design problems was presented by [22]. This study provides an extensive comparison of the results of solution methods in various kinds of network design problems.

In the light of above mentioned literature, it is clear that most of the solution approaches developed up to now are based on the heuristic approaches, and ensure finding at least local optimal solutions. Although some exact algorithms for the CNDP are available in the literature, they may not be suitable especially for large scale networks. Therefore, a powerful algorithm, which is able to find better near global solutions to the CNDP and more suitable for large scale networks, is still needed. Additionally, it is also crucial to compare the efficiency of the recent heuristic methods, and to find the most efficient ones because the computation time for solving the CNDP is considerably large. Thus, this paper deals with comparing the performance of Harmony Search (HS), Artificial Bee Colony (ABC) and Differential Evolution (DE) for the CNDP. Although so far various types of population based heuristic algorithms were used in solving the CNDP, the performance of the HS, ABC and DE algorithms needs also to be investigated because of their ability to handle such complex problems. For this purpose, a bilevel model has been proposed, in which the lower level problem has been formulated as User Equilibrium (UE) traffic assignment model and Frank-Wolfe method [23] is used to solve it.

This paper is organized as follows. In Section 2, problem formulation for the CNDP is given. In the next section, the HS, ABC and DE approaches and their solution procedures on the CNDP are presented. In Section 4, numerical applications are conducted on example test networks. Finally, concluding remarks are given.

2. Problem Formulation

In the CNDP, the optimal link capacity expansions are determined by minimizing the total network travel cost considering related budget. The CNDP can be formulated as follows:

$$\min_{\mathbf{y}} \quad Z(\mathbf{x},\mathbf{y}) = \sum_{a\in A}(t_a(x_a, y_a)x_a + \rho g_a(y_a)) \tag{1}$$

$$\text{s.t.} \quad 0 \le y_a \le u_a \ , \quad \forall a \in A \tag{1a}$$

where Z is the upper level objective function; $\mathbf{x}$ is the vector of UE link flows which can be derived from the lower level problem; t_a is the travel time on link $a \in A$, and described as a function of link flow x_a and capacity expansion y_a ; $g_a(y_a)$ is the investment function of link $a \in A$; A is the set of links in the network; u_a is the upper bound of capacity expansion of link $a \in A$; ρ the conversion factor from investment cost to travel times. Note that the Eq. (1a) ensures that the positive-valued investment cost of link $a \in A$ will not exceed the related budget.

In the CNDP models, the users' route choice behavior is generally characterized by the UE assignment. The UE assignment problem can be described to find the link flows, $\mathbf{x}$, which satisfies the user-equilibrium criterion when Origin-Destination (O-D) demands have been assigned. On the other hand, network-wide congestion may be increased by way of applying capacity expansions or adding of new links to the network when the response of road users to those applications is not considered. Therefore, prediction of traffic flows is crucial in solving the CNDP. The UE assignment problem can be formulated as follows:

$$\min_{x} z = \sum_{a \in A} \int_{0}^{x_a} t_a(w, y_a) dw \tag{2}$$

$$\text{s.t.} \quad \sum_{k \in K} f_k^{rs} = D_{rs} \qquad \forall r \in R,\ s \in S,\ k \in K_{rs} \tag{2a}$$

$$x_a = \sum_{rs} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \qquad \forall r \in R,\ s \in S,\ a \in A, k \in K_{rs} \tag{2b}$$

$$f_k^{rs} \geq 0 \qquad \forall r \in R,\ s \in S,\ k \in K_{rs} \tag{2c}$$

where z is the lower level objective function; R is the set of origins; S is the set of destinations; K_{rs} is the set of paths between O-D pair $rs \quad \forall r \in R,\ s \in S$; f_k^{rs} is the flow on path k between O-D pair $rs \ \forall r \in R,\ s \in S$; D_{rs} is the demand of O-D pair $rs \quad \forall r \in R, \quad s \in S$; $\delta_{a,k}^{rs}$ is the variable of link/path incidence matrix. The expressions, which are given in Eqs. (2a)-(2c), represent definitional, conservation of the flow and non-negativity constraints.

3. Heuristic Methods

3.1. Harmony Search Algorithm

The HS algorithm is a meta-heuristic method developed by [24] and based on the musical process of searching for a perfect state of harmony. During the recent years, the HS has been successfully applied to solve complex optimization problems. In the HS, four algorithm parameters are used to manage the solution process. The first one of them is Harmony Memory Size (HMS) which defines the number of existed solution vectors in the Harmony Memory (HM). The second one is Harmony Memory Considering Rate (HMCR) that defines the rate of selecting the values from the HM. The other one is Pitch Adjusting Rate (PAR) that sets the rate of adjustment for the pitch chosen from the HM and the last one is number of improvisations (NI) that defines the number of iterations [25]. The phases of the HS algorithm for solving the CNDP can be given as:

Initialization: The HM is filled with randomly generated link capacity expansions as shown in Eq. (3).

$$\begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_{N-1}^1 & y_N^1 \\ y_1^2 & y_2^2 & \cdots & y_{N-1}^2 & y_N^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_1^{\text{HMS-1}} & y_2^{\text{HMS-1}} & \cdots & y_{N-1}^{\text{HMS-1}} & y_N^{\text{HMS-1}} \\ y_1^{\text{HMS}} & y_2^{\text{HMS}} & \cdots & y_{N-1}^{\text{HMS}} & y_N^{\text{HMS}} \end{bmatrix} \tag{3}$$

where y_a is capacity expansion on link $a \in A$; N is the number of candidate links for capacity expansion in a given road network.

Improvisation: In this step, a new solution vector $y' = (y_1', y_2', \ldots, y_N')$ is created considering the HS operators which are the memory consideration, pitch adjustment and random selection. In the memory consideration, the value of the first decision variable (y_1') for the new solution vector is randomly selected from any value in the HM range ($y_1^1 - y_1^{\mathrm{HMS}}$). Values of the other decision variables $(y_2',....,y_N')$ are selected in the same way. The value of (1-HMCR) represents the rate of selecting a value according to the upper and lower bounds of link capacity expansions as shown in Eq. (4).

$$y_i' \leftarrow \begin{cases} y_i' \in \{y_i^1, y_i^2, \ldots.. y_i^{\mathrm{HMS}}\} & \text{with probability of HMCR} \\ y_i' \in \text{rand}\,[0, u_i] & \text{with probability of (1 - HMCR)} \end{cases} \quad (4)$$

After improvisation step, the pitch adjustment operator is applied to the new solution vector considering the parameter of PAR, which varies between 0-1 as follows:

$$y_i' = \begin{cases} y_i' \pm \text{rand}\,[0,1] \times bw & \text{with probability of PAR} \\ y_i' & \text{with probability of (1 - PAR)} \end{cases} \quad (5)$$

The parameter of *bw* is defined to find possible neighborhood solutions. The pitch adjusting process is carried out only after a value has been chosen from the HM. The value (1-PAR) represents the rate of doing nothing. Using parameter values ranged between 0.70 and 0.95 for HMCR, 0.20 and 0.50 for PAR, and 10 and 50 for HMS was recommended by [26]. Hence, in this study, HMCR, PAR and HMS are selected for all experiments as 0.90, 0.40 and 10, respectively.

Update: After determining the objective function values in HM using Eq. (1), the solution vector which gives the worst objective function value is removed from the HM. Finally, solution processes of the HS are repeated until a predetermined stopping criterion is met or maximum number of improvisations is reached. The steps of the HS on the CNDP are summarized as follows:

Step 0: Input the HS control parameters.
Step 1: Generate the HM with capacity expansions y_a for each link $a \in A$ by giving the upper and lower bounds.
Step 2: A new vector is created by way of memory consideration, pitch adjustment and random selection.
Step 3: Solve the UE assignment problem by using Frank-Wolfe method using populated capacity expansions in HM.
Step 4: Find the value of the upper level objective functions in HM for resulting capacity expansions at Steps 1-2 and the equilibrium link flows obtained from Step 3.
Step 5: The new solution vector is compared with the vector giving the worst objective function value.
Step 6: Check the termination criterion. If the relative error between the average and best objective function values in HM is less than a predetermined value, the algorithm is terminated. Else go to Step 7.
Step 7: Terminate the algorithm if maximal number of improvisations is reached. Else go to Step 2.

3.2. Artificial Bee Colony Algorithm

The ABC algorithm is a population-based metaheuristic proposed by [27]. It is inspired by the foraging behavior of honeybee swarm. The foraging behaviors of honeybees have recently been one of the most interesting research areas in swarm intelligence. The foraging bees consist of three categories employed, onlookers and scouts. In the ABC, the position of a food source describes a possible solution to the given optimization problem while the nectar amount of a food source represents the quality of the solution [28]. To adapt this concept to the CNDP, the ABC algorithm randomly creates link capacity expansions as number of *SN*, where *SN* denotes the number of solution vectors. Each initial solution vector, **y**, is *N*-dimensional vector where *N* is the number of links in a given road network. At the first step, employed bees share their information about nectar sources (i.e. the quality of objective function values) with onlooker bees waiting on the dance area. Onlooker bees are responsible for making decision about the choice of a food source. At the second step, an onlooker chooses a food source area depending on the nectar information. It is clear that the higher nectar amount of a food source increases the probability of choice of that food source. The new food source is determined by the bees by way of visual comparison of positions of food sources. At the last step, when a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and is replaced with the abandoned one. Solution steps are repeated until a predetermined stopping criterion is met or maximum cycle number (MCN) is reached.

In order to find a candidate food source, which denotes the new value of capacity expansion y_a on link $a \in A$ for *j*th solution vector in memory, the following equation is used.

$$v_i^j = y_i^j + \phi_i^j (y_i^j - y_i^k) \tag{6}$$

where v_i^j is the candidate food source, $j \in \{1,2,....,SN\}$, which can replace the old one in the memory. ϕ_i^j is a random number generated within the range [-1,1] where $i \in \{1,2,....,N\}$ and the value of $k \in \{1,2,...,SN\}$ is randomly chosen index. Note that *k* must be different from *j* to avoid that old and new locations coincide, in order to find food sources having more nectar amount than the old one. The parameter ϕ_i^j controls the production of neighbour food sources around y_i^j, and represents the visual comparison of two food positions carried out by a bee. If a parameter value determined using Eq. (6) exceeds the constraint of the capacity expansions, the parameter is set to its upper or lower bounds depending on which constraint has been exceeded. After each candidate food source v_i^j is generated, its performance is compared with that of the old one in terms of the nectar amount (i.e. the quality of objective function values). If the new food source produces better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is saved in the memory. In other words, a greedy selection mechanism is carried out for selection between the old and the candidate location.

For making decision about the choice of a food source, an onlooker bee decides taking the probability value, p^j, for *j*th solution vector into account as follows:

$$p^j = \frac{Z^j}{\sum_{m=1}^{SN} Z^m} \tag{7}$$

where *Z* is the value of upper level objective function on the CNDP. In this way, the employed bees exchange their information with the onlookers. In order to share the information of nectar amount of the food sources, the employed bees use a proportional selection method known as “roulette wheel selection” [29].

As mentioned above, the food source abandoned by the bees is replaced with a new food source by the scouts at the third step of the cycle. This is simulated by generating a random location and replacing the abandoned one with it. If a location cannot be further improved in a predetermined number of cycles, then that food source is assumed to be abandoned. This operation can be performed using Eq. (8).

$$y_i^j = y_{\min}^j + \mathrm{rand}[0,1](y_{\max}^j - y_{\min}^j) \tag{8}$$

The value of predetermined number of cycles, called ‘‘limit”, is an important control parameter of the algorithm, and it can be determined as *N**SN* [28]. The steps of the ABC algorithm on the CNDP can be summarized as follows:

Step 0: Input the ABC parameters.

Step 1: Initialize the capacity expansion y_a for each link $a \in A$ by giving the upper and lower bounds.

Step 2: Generate new solutions for the employed bees by using Eq. (6).

Step 3: Solve the UE assignment problem using capacity expansions obtained at Steps 1-2.

Step 4: Find the value of the upper level objective functions for resulting capacity expansions at Steps 1-2 and link flows in Step 3. Apply greedy selection process for the employed bees.

Step 5: Calculate the probability values for the employed bees by using Eq. (7).

Step 6: Generate new solutions for the onlookers from the solutions at Step 5 depending on probability values.

Step 7: Solve the UE assignment problem using capacity expansions obtained at Step 6.

Step 8: Find the value of the upper level objective functions for resulting capacity expansions at Step 6 and link flows at Step 7. Apply greedy selection process for the onlookers.

Step 9: Determine the abandoned solution for the scout bee, if exists, and replace it with a new randomly produced solution by using Eq. (8).

Step 9.1: Solve the UE traffic assignment problem using capacity expansions obtained at Step 9.

Step 9.2: Find the value of the upper level objective functions for resulting capacity expansions at Step 9 and link flows at Step 9.1.

Step 10: Memorize the best solution vector of capacity expansions achieved so far.

Step 11: Check the termination criterion. If the relative error between the average and best objective function values in the memory is less than a predetermined value, the algorithm is terminated. Else go to Step 12.

Step 12: Terminate the algorithm if maximum cycle number is reached. Else go to Step 2.

3.3. Differential Evolution Algorithm

The DE algorithm is a relatively simple, fast and powerful heuristic algorithm, which is proposed by [30] to solve complex optimization problems. In the DE, mutation, crossover and selection steps are used to control the solution process. In the mutation process, new solution vectors are generated by adding the weighted difference of two randomly chosen vectors to a third vector [31]. Then, crossover operator is applied to the mutant vector, and then a new solution vector is obtained. In the selection step, the objective function value of a newly created solution vector is compared with that produced by the parent vector. If the new solution vector generates better functional value than the parent, it is replaced with the parent vector. This process is applied to the all solution vectors in the population, and then a new population is created for the next generation. Three algorithm parameters are used to control the optimization process of the DE. The first parameter is the number of populations (*NP*) that represents the number of solution vectors used in the solution process. The mutation factor (*F*), which is the second parameter and recommended to set between 0.5-1 by [30]. The third one is the crossover rate (*CR*) that is the probability of mixing the variables of the mutant and target vectors. The recommended range of the crossover rate is [0.8, 1] by [30]. In this study, parameters *F* and *CR* are selected as 0.8 for all numerical applications. The solution procedure of the DE on the CNDP can be explained as below:

Initialization: At generation *t*, the initial solution vector, $\mathbf{y}^t = \left[y_i^j \right]$, where $i \in \{1,2,\ldots,N\}$ and $j \in \{1,2,\ldots,NP\}$, is filled with randomly generated capacity expansions for a set of selected links in a given road network by considering upper and lower bounds. Then their corresponding objective function values are calculated using Eq. (1). Note that *N* represents the number of links in road network.

Mutation: The mutation is performed by adding the weighted difference vector between two solution vectors to a third vector. Combining three different randomly chosen solution vectors to create a mutant vector, $\mathbf{m}^t = \left[m_i^j \right]$, can be defined as given in Eq. (9).

$$m_i^{j,t} = y_i^{1,t} + F(y_i^{2,t} - y_i^{3,t}) \tag{9}$$

where $y_i^{1,t}$, $y_i^{2,t}$ and $y_i^{3,t}$ are randomly selected capacity expansions for *j*th solution vector within the range [0,*NP*] at generation *t*, and $y_i^{1,t} \neq y_i^{2,t} \neq y_i^{3,t}$.

Crossover: The search process of the DE is completed with the crossover operator. At this step, each member of the trial vector, $r_i^{j,t}$, is chosen from the mutant vector with the probability of *CR* or from the target vector with the probability of (1-*CR*) as given in Eq. (10).

$$r_i^{j,t} = \begin{cases} m_i^{j,t}, & \text{if rand } [0,1] \leq CR \text{ or } j = j_{rand} \\ y_i^{j,t}, & \text{otherwise} \end{cases} \tag{10}$$

As can be seen from Eq. (10), *CR* is compared with the output of a uniform random number generator, $\text{rand}\,[0,1]$, to determine either mutant vector or target vector will provide the member of the trial vector. If the random generated number within the range [0,1] is less than or equal to *CR* at generation *t*, the trial parameter is chosen from the mutant vector, $\mathbf{m}^{j,t}$; otherwise the parameter is chosen from the target vector, $\mathbf{y}^{j,t}$. Additionally, the constraint, $j = j_{\text{rand}}$, where j_{rand} is the uniformly distributed random number in the range [1,*NP*], ensures that at least one member of the trial vector is inherited from the mutant vector.

Selection: At this step, the trial vector, $\mathbf{r}^t$, is compared with the parent individual $\mathbf{y}^t$ by way of determining the objective function values and the best one enters to the generation *t*+1.

$$\mathbf{y}^{t+1} = \begin{cases} \mathbf{r}^t, & \text{if } f(\mathbf{r}^t) \leq f(\mathbf{y}^t) \\ \mathbf{y}^t, & \text{otherwise} \end{cases} \tag{11}$$

Termination: Mutation, crossover and selection steps of the DE are repeated until a predetermined stopping criterion is met or maximum number of generations is reached. The procedure of the DE algorithm on the CNDP can be summarized as:

Step 0: Input the DE parameters.

Step 1: Initialize the capacity expansion y_a for each link $a \in A$ by giving the upper and lower bounds.

Step 2: Solve the UE assignment problem using capacity expansions generated at Step 1.

Step 3: Find the value of the upper level objective functions using Eq. (1) for resulting capacity expansions at Step 1 and link flows at Step 2.

Step 4: Find a mutant vector using Eq. (9) by considering capacity expansion y_a for each link $a \in A$.

Step 5: Apply crossover operator to obtain the trial vector by using Eq. (10).
Step 6: Solve the UE assignment problem using capacity expansions (trial vector) created at Step 5.
Step 7: Find the value of the upper level objective functions using Eq. (1) for resulting capacity expansions at Step 5 and link flows at Step 6.
Step 8: Apply selection process. In this step, trial vector is compared with the parent vector in terms of the objective function value obtained at Step 7. The best one enters the next generation.
Step 9: Check the termination criterion. If the relative error between the average and best objective function values in the memory is less than a predetermined value, the algorithm is terminated. Else go to Step 10.
Step 10: Terminate the algorithm if maximum number of generations is reached. Else go to Step 4.

4. Numerical Applications

4.1. 18-link Network

Before applying the proposed heuristics to Sioux Falls city network, 18-link network is considered in order to show their ability in solving the CNDP. The chosen test network is adopted from [14] as shown in Figure 1. The travel demand for this network includes three cases and can be seen in Table 1.

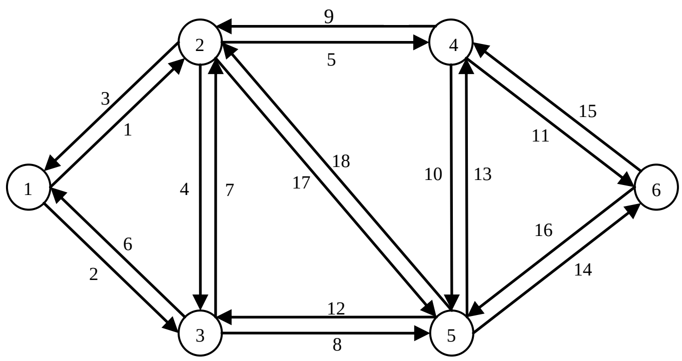


Figure 1. Test Network

Table 1. Travel Demands

Case	q_{16}	q_{61}	Total demand
1	5	10	15
2	10	20	30
3	15	25	40

The link travel time function is defined as shown in Eq. (12) and its corresponding parameters for the 18-link network are taken from [14].

$$t_a(x_a, y_a) = \alpha_a + \beta_a (\frac{x_a}{\theta_a + y_a})^4 \qquad (12)$$

where α, β are the parameters and θ_a is the capacity of link $a \in A$. The upper level objective function for the 18-link network is defined as:

$$\min_{\mathbf{y}} \quad Z(\mathbf{x},\mathbf{y}) = \sum_{a \in A} (t_a(x_a, y_a) x_a + d_a y_a) \qquad (13)$$

$$\text{s.t.} \quad 0 \le y_a \le u_a \ , \quad \forall a \in A \tag{13a}$$

where d_a is the cost coefficient and u_a is upper bound for capacity expansion of link $a \in A$, and is set to 20. The parameters used in the proposed methods are given in Table 2.

Table 2. Parameters for HS, ABC and DE on all Example Test Networks

Parameters of HS	*Value*
HMCR	0.90
PAR	0.40
HMS	10
Maximum number of improvisations	20000
Parameters of ABC	
SN	10
MCN	500
Parameters of DE	
F	0.80
CR	0.80
NP	10
Maximum number of generations	250
Common parameters for HS, ABC and DE	
Stopping criterion	$\frac{\lvert Z_{best} - Z_{average} \rvert}{Z_{best}} \le 10^{-3}$

The HS, ABC and DE algorithms for finding optimal link capacity expansions are tested on 18-link network, where the upper level optimization problem is solved through the proposed heuristics, and UE traffic assignment is performed by way of the Frank-Wolfe method at the lower-level. The comparison of computation times for the proposed heuristics is conducted in terms of the number of UE assignments (NUE) since the process of UE assignment is the most time consuming part of the algorithms. In order to demonstrate stochastic behavior of the proposed heuristics, they are run for 20 times. The average number of UE assignments and average objective function values are also reported. Results are compared with those produced by SA and GA algorithms taken from [14] on the same network, and they are given in Table 3. As it seen from Table 3, the link capacity expansions produced by the algorithms are different from each other although the resulting values of *Z* generated by some algorithms are very close. The reason is that each method leads to a different solution to the CNDP since it has multiple local optima due to the non-convexity of the bilevel formulation. Among the compared algorithms, the DE shows steady convergence towards the optimum or near optimum for all demand cases and achieved good solutions in terms of the objective function value and the number of UE assignments. For case 1, the convergence graph of the HS, ABC and DE algorithms can be seen on a logarithmic scaled *x* axis in Figure 2.

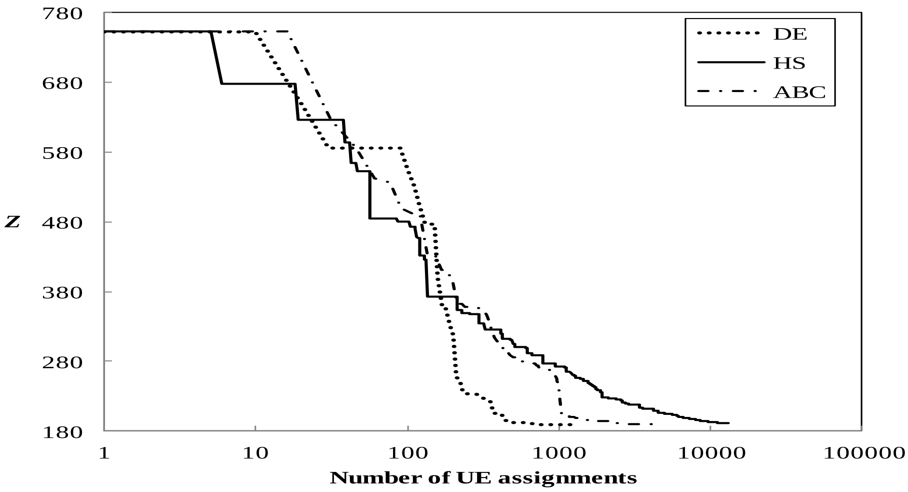


Figure 2. Convergence of the HS, ABC and DE for Case 1

In case 1, the DE algorithm converges to the value of 189.40 after 1250 UE assignments while the ABC achieves almost same near optimal value. However, the ABC needs much more UE assignments than the DE in order to reach that near optimal value as shown in Figure 2. The DE initially starts with a randomly generated memory considering upper and lower bounds of capacity expansions and picked up the best *Z* which is about 750. It easily locates the best *Z* after a few hundred UE assignments and reached near optimal value after only 750 UE assignments. Additionally, the HS algorithm reached to the value of 190.61 after 13087 UE assignments while SA and GA produced the value of 205.89 and 191.26 after 15000 and 50000 number of UE assignments, respectively [14]. In comparison with the results generated by the ABC and DE algorithms, it has been found that the HS, SA and GA are also capable of finding near optimal values of *Z*, but their computational effort are much more intensive. In case 2, the DE reached to the value of 487.80 after 2150 UE assignments as shown in Figure 3. Although the DE slightly outperformed than the ABC, the required number of UE assignments are nearly same for both algorithms. On the other hand, the HS reached to the value of 491.75 after 8185 UE assignments for case 2.

Although the values of *Z* produced by the HS, ABC and DE algorithms are quite close for all demand cases, the HS required much more UE assignments than the ABC and DE in order to reach to the near optimal value of *Z*. In addition, as it can be seen in Table 3, it is remarkable that the results produced by the SA and GA are not as good as those generated by the other compared algorithms in terms of both objective function value and required number of UE assignments.

To investigate the capability of the HS, ABC and DE algorithms in solving the CNDP under heavy demand condition, case 3 has been considered and the corresponding convergence histories of the proposed heuristics are given in Fig. 4.

Table 3. Results of the Heuristics on 18-link Network

	Case 1					Case 2					Case 3				
	HS	ABC	DE	SA	GA	HS	ABC	DE	SA	GA	HS	ABC	DE	SA	GA
y_1	0	0	0	0	0	0.96	0.60	0	0	0	1.39	3.87	1.43	0	0
y_2	0	0	0	0.47	0	2.07	1.97	2.15	1.73	2.20	8.22	7.18	8.80	9.12	11.98
y_3	0	0.03	0	0.65	0	9.50	9.96	10.58	11.77	10.61	16.15	13.29	16.28	18.12	16.24
y_4	0.03	0	0	0	0	0.03	0	0	0	0	0	0	0	0	0
y_5	0	0	0	0	0	0	0	0	0	0	0.03	0	0	0	0
y_6	5.06	5.15	5.14	6.53	4.47	8.86	8.47	6.82	4.75	6.68	9.49	13.93	10.24	4.98	5.40
y_7	0	0	0	0.80	0	0	0	0	0.14	0	0.02	0.05	0	0.11	0
y_8	0	0	0	0.25	0	0	0	0	0.78	0	3.31	2.69	3.41	1.58	6.04
y_9	0.02	0	0	0	0	0	0	0.02	0	0	0.09	0	0	0	0
y_{10}	0.14	0	0	0	0	0.13	0	0	0	0	0.11	0	0	0	0
y_{11}	0	0	0	0	0	0	0	0	0	0	0.24	0	0.06	0	0
y_{12}	0	0	0	0	0	0.15	0.16	0	0	0	0.14	2.69	0	0	0
y_{13}	0	0.03	0	0	0	0	0	0	0	0	0.43	0	0	0	0
y_{14}	0	0	0	0.84	0	1.57	1.28	1.33	5.94	1.22	13.04	12.15	11.24	11.66	12.28
y_{15}	0	0.02	0	0.14	0	0.34	0.07	0.11	1.51	6.30	0.32	0.15	0	2.97	0.82
y_{16}	7.82	7.38	7.32	7.34	7.54	19.41	19.88	19.88	18.45	11.93	19.91	20.00	20.00	19.71	19.99
y_{17}	0	0	0	0	0	0	0	0	0	0	0.25	0.03	0	0	0
y_{18}	0	0.02	0	0	0	0.04	0	0	0	0	0.10	0	0	0	0
Z_{best}	190.61	189.59	189.40	205.89	191.26	491.75	489.01	487.80	505.39	515.09	734.00	736.38	728.14	739.54	744.39
$Z_{average}$	191.11	190.78	190.56	NA*	NA	492.02	490.69	488.41	NA	NA	735.15	737.41	729.47	NA	NA
NUE_{best}	13087	4305	1250	15000	50000	8185	2175	2150	42500	50000	4356	1620	1240	22500	50000
$NUE_{average}$	12910	4231	1020	NA	NA	8090	2098	1810	NA	NA	4216	1591	1120	NA	NA

*NA: not available.

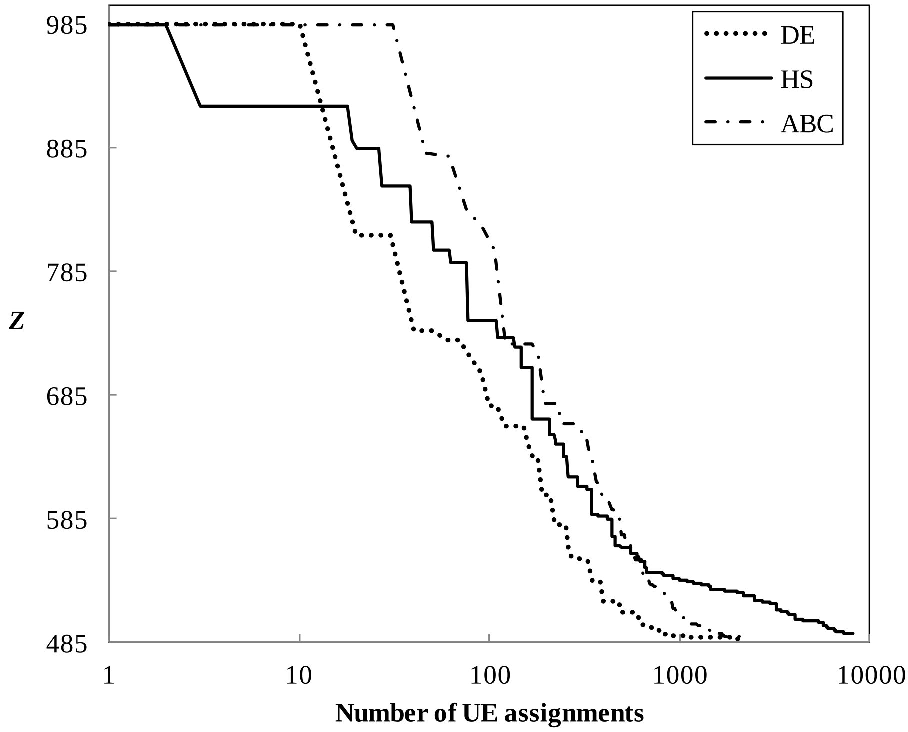


Figure 3. Convergence of the HS, ABC and DE for Case 2

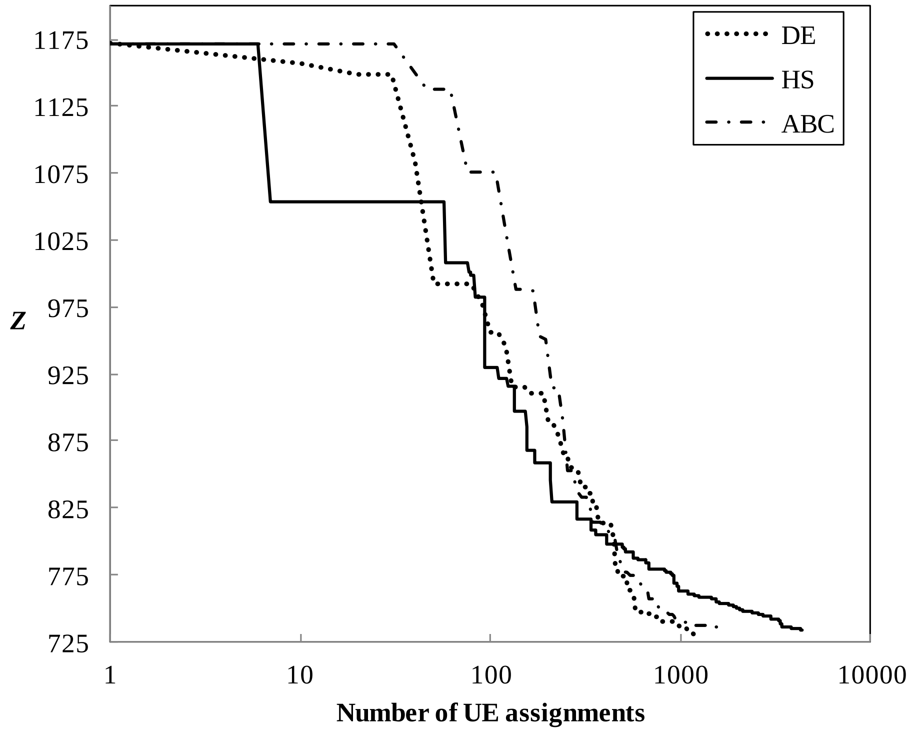


Figure 4. Convergence of the HS, ABC and DE for Case 3

The DE algorithm achieved good solution with less computational efforts especially with respect to the HS, SA and GA. The SA and GA need much more UE assignments, namely 22500 and 50000, than the other compared algorithms to reach to the near optimal value of *Z*. The DE reached to the value of 728.14 and 1240 UE assignments are needed to reach to this value, as shown in Figure 4. The ABC is also capable of finding near optimal value of *Z* but it needs little more number of UE assignments in comparison with the DE. Although the HS slightly outperformed than the ABC in terms of the objective function value for case 3, it needs much more UE assignments with respect to the ABC. The numerical experiments performed on the 18-link network show that the DE is much more efficient and effective method than other compared algorithms in terms of the objective function value and required number of UE assignments in solving the CNDP for all demand levels. On the other hand, it may be emphasized that the ABC algorithm also achieves acceptable solutions although it requires more UE assignments in comparison with the DE in all cases for the test network.

4.2. Sioux Falls city Network

In order to compare the performance of the HS, ABC and DE algorithms on the realistic test network, the city of Sioux Falls is chosen since it is probably the most used test network for the CNDP. The other reason for choosing this network is that readers may have opportunity to compare performance of the proposed heuristics with other existing methods. The Sioux Falls network consists of 24 nodes and 76 links. The link parameters of the network and the travel demands between the 552 O-D pairs are adopted from [3]. The dashed links 16, 17, 19, 20, 25, 26, 29, 39, 48 and 74 of the network are candidates for capacity expansion as shown in Figure 5. The upper level objective function for the Sioux Falls network is formulated as in Eq. (14). The results of the proposed heuristics are compared with those produced by major algorithms available in literature. The compared algorithms are given in Table 4 and the corresponding results are tabulated in Table 5-6. For this example network, the computational performance of the algorithms is compared by considering the number of Frank-Wolfe iterations since the literature supports such evaluation for fair comparison [16].

$$\min_{\mathbf{y}} \quad Z(\mathbf{x},\mathbf{y}) = \sum_{a \in A} (t_a(x_a, y_a)x_a + 0.001 d_a y_a^2) \tag{14}$$

$$\text{s.t.} \quad 0 \le y_a \le u_a \ , \qquad \forall a \in A \tag{14a}$$

Table 4. The Compared Algorithms on Sioux Falls Network

Methods	References
Hooke-Jeeves algorithm (HJ)	Abdulaal and LeBlanc [1]
Equilibrium Decomposed Optimization (EDO)	Suwansirikul et al. [3]
Genetic Algorithm (GA)	Mathew and Sarma [16]
Simulated Annealing algorithm (SA)	Friesz et al. [9]
Augmented Lagrangian algorithm (AL)	Meng et al. [11]
Gradient Projection method (GP)	Chiou [12]
Conjugate Gradient projection method (CG)	Chiou [12]
Quasi-NEWton projection method (QNEW)	Chiou [12]
ParaTan version of gradient projection method (PT)	Chiou [12]
Harmony Search (HS) algorithm	This paper
Artificial Bee Colony (ABC) algorithm	This paper
Differential Evolution (DE) algorithm	This paper

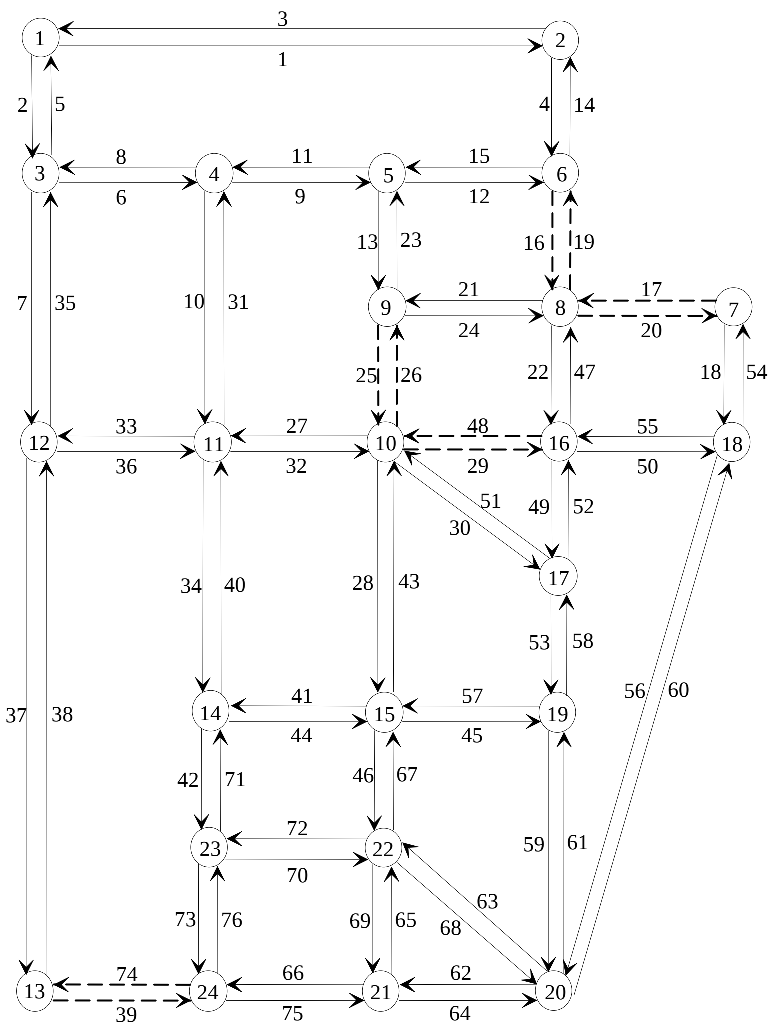


Figure 5. Sioux Falls Network

Table 5. Comparison of Algorithms on Sioux Falls Network

	HJ	EDO	SA	AL	GP	CG
Initial value of ya	1.0	12.5	6.25	12.5	12.5	12.5
y16	3.8	4.59	5.38	5.5728	4.8693	4.7691
y17	3.6	1.52	2.26	1.6343	4.8941	4.8605
y19	3.8	5.45	5.50	5.6228	1.8694	3.0706
y20	2.4	2.33	2.01	1.6443	1.5279	2.6836
y25	2.8	1.27	2.64	3.1437	2.7168	2.8397
y26	1.4	2.33	2.47	3.2837	2.7102	2.9754
y29	3.2	0.41	4.54	7.6519	6.2455	5.6823
y39	4.0	4.59	4.45	3.8035	5.0335	4.2726
y48	4.0	2.71	4.21	7.3820	3.7597	4.4026
y74	4.0	2.71	4.67	3.6935	3.5665	5.5183
Zbest	81.77	83.47	80.87	81.75	82.71	82.53
Zaverage	-	-	-	-	-	-
#	108	12	3900	2700	9	6

Note: The upper bound for *y* was set to 25 except HJ. # denotes the number of Frank-Wolfe iterations performed. $Z_{average}$ was obtained for 20 runs in HS, ABC and DE.

Table 6. Comparison of Algorithms on Sioux Falls Network (continued)

	QNEW	PT	GA	HS	ABC	DE
Initial value of y_a	6.25	12.5	-	-	-	-
y_{16}	4.9776	5.0237	5.17	5.8057	5.9129	5.1546
y_{17}	5.0287	5.2158	2.94	2.8516	1.9505	1.6531
y_{19}	1.9412	1.8298	4.72	4.4821	4.8668	5.8942
y_{20}	2.1617	1.5747	1.76	2.2265	1.7556	1.2921
y_{25}	2.6333	2.7947	2.39	4.5586	2.5498	2.5883
y_{26}	2.7923	2.6639	2.91	3.1673	2.9854	1.6994
y_{29}	5.7462	6.1879	2.92	3.1774	3.6906	3.3243
y_{39}	5.6519	4.9624	5.99	4.7492	3.7716	5.1140
y_{48}	4.5738	4.0674	3.63	2.6648	3.0216	3.2682
y_{74}	4.1747	3.9199	4.43	5.4052	4.9115	4.5044
Z_{best}	83.08	82.53	81.74	81.80	81.78	81.60
$Z_{average}$	-	-	-	81.97	82.02	81.76
#	5	7	77	27	32	23

Note: The upper bound for *y* was set to 25 except GA. # denotes the number of Frank-Wolfe iterations performed. $Z_{average}$ was obtained for 20 runs in HS, ABC and DE.

From Tables 5-6, it can be observed that the DE algorithm produces the best solution in terms of objective function value among the compared algorithms except SA. Besides the results generated by the DE and SA algorithms are quite close, the DE needs much less computational efforts in solving the traffic assignment problem in comparison with the SA. In addition, the difference of the best and average objective function values obtained by the DE is not significant. This result shows that the DE is indeed a powerful method in solving the CNDP, as well as robust. Although the HS algorithm is not as good as the DE in solving the CNDP, it produced better results than many of the compared algorithms. The HJ, SA, AL and GA algorithms produce slightly better results than the HS in terms of the objective function value, but they need more computational efforts in solving the traffic assignment problem in comparison with the HS. The objective function value achieved by the ABC algorithm is quite close to that generated by the HS with almost the computational effort. In comparison with other algorithms, the ABC is able to produce good results in solving the CNDP and performs better than many of the other compared algorithms such as EDO, GP, CG, QNEW and PT in terms of the objective function value.

5. Conclusions

The CNDP can be described as one of the most difficult problems in transportation science because of its non-convex structure. The methods developed so far to solve the CNDP are generally based on the heuristic approaches. Therefore, it is crucial importance to seek for the most efficient ones between the available heuristic methods in the literature in solving the CNDP. In this paper, the HS, ABC and DE algorithms were employed to solve the CNDP which was modeled as a bilevel programming model. The upper level objective function has been solved by minimizing the sum of the network travel cost and total investment costs of link capacity expansions while the lower level problem was formulated as user equilibrium traffic assignment model. The Frank-Wolfe method was used to solve the traffic assignment problem at the lower level. Numerical comparisons were conducted on 18-link and Sioux Falls test networks. Results showed that the DE produced better solutions than those generated by other compared algorithms in all demand cases for 18-link network, as well as with less number of UE assignments. In particular, for case 3, which represents heavy demand condition, the DE algorithm achieved substantially better results in comparison with the HS, ABC, SA and GA. For

Sioux Falls network, the DE algorithm performed better than other compared algorithms in terms of objective function value except SA. Although the SA produced slightly better results, it needs much more computational effort in solving the traffic assignment problem. The HS and ABC methods are also able to produce good results in solving the CNDP. All these experiments may imply that the DE algorithm has the potential for large network applications for the CNDP due to its simplicity and robustness.

References

[1] M. Abdulaal and L. LeBlanc, "Continuous equilibrium network design models", Transportation Research Part B, vol. 13, **(1979)**, pp. 19-32.

[2] H. N. Tan, S. B. Gershwin and M. Athans, "Hybrid Optimization in Urban Traffic Networks", Report No. DOT-TSCRSPA-79-7, Laboratory for Information and Decision System, MIT Press, Cambridge, MA, **(1979)**.

[3] C. Suwansirikul, T. L. Friesz and R. L. Tobin, "Equilibrium decomposed optimisation: a heuristic for the continuous equilibrium network design problem", Transportation Science, vol. 21, no. 4, **(1987)**, pp. 254-263.

[4] P. Marcotte and G. Marquis, "Efficient implementation of heuristics for the continuous network design problem", Annals of Operational Research, vol. 34, **(1992)**, pp. 163-176.

[5] H. J. Cho, "Sensitivity analysis of equilibrium network flows and its application to the development of solution methods for equilibrium network design problems", PhD Dissertation, University of Pennsylvania, Philadelphia, **(1988)**.

[6] T. L. Friesz, R. L. Tobin, H. J. Cho and N. J. Mehta, "Sensitivity analysis based algorithms for mathematical programs with variational inequality constraints", Mathematical Programming, vol. 48, **(1990)**, pp. 265-284.

[7] H. Yang, "Sensitivity analysis for queuing equilibrium network flow and its application to traffic control", Mathematical and Computer Modelling, vol. 22, no. 4-7, **(1995)**, pp. 247–258.

[8] H. Yang, "Sensitivity analysis for the network equilibrium problem with elastic demand", Transportation Research, vol. 31, **(1997)**, pp. 55-70.

[9] T. L. Friesz, H. J. Cho, N. J. Mehta, R. L. Tobin and G. Anandalingam, "A simulated annealing approach to the network design problem with variational inequality constraints", Transportation Science, vol. 26, no. 1, **(1992)**, pp. 18-26.

[10] G. A. Davis, "Exact local solution of the continuous network design problem via stochastic user equilibrium assignment", Transportation Research Part B, vol. 28, no. 1, **(1994)**, pp. 61-75.

[11] Q. Meng, H. Yang and M. G. H. Bell, "An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem", Transportation Research Part B, vol. 35, **(2001)**, pp. 83-105.

[12] S. W. Chiou, "Bilevel programming for the continuous transport network design problem", Transportation Research Part B, vol. 39, **(2005)**, pp. 361-383.

[13] X. G. Ban, H. X. Liu, J. G. Lu and M. C. Ferris, "Decomposition scheme for continuous network design problem with asymmetric user equilibria", Transportation Research Record (1964), **(2006)**, pp. 185-192.

[14] T. Xu, H. Wei and G. Hu, "Study on continuous network design problem using simulated annealing and genetic algorithm", Expert Systems with Applications, vol. 36, **(2009)**, pp. 1322-1328.

[15] A. Karoonsoontawong and S. T. Waller, "Dynamic continuous network design problem-Linear bilevel programming and metaheuristics approaches", Transportation Research Record (1964), **(2006)**, pp. 104-117.

[16] T. V. Mathew and S. Sarma, "Capacity expansion problem for large urban transportation networks", Journal of Transportation Engineering, vol. 135, no. 7, **(2009)**, pp. 406-415.

[17] D. Z. W. Wang and H. K. Lo, "Global optimum of the linearized network design problem with equilibrium flows", Transportation Research Part B, vol. 44, no. 4, **(2010)**, pp. 482-492.

[18] C. Li, H. Yang, D. Zhu and Q. Meng, "A global optimization method for continuous network design problems", Transportation Research Part B, vol. 46, **(2012)**, pp. 1144-1158.

[19] O. Baskan, "Harmony search algorithm for continuous network design problem with link capacity expansions", KSCE Journal of Civil Engineering, DOI: 10.1007/s12205-013-0122-6, **(2013)**, pp. 1–11.

[20] O. Baskan, "Determining optimal link capacity expansions in road networks using Cuckoo Search algorithm with Lévy Flights", Journal of Applied Mathematics, vol. 2013, Article ID 718015, 11 pages, **(2013)**.

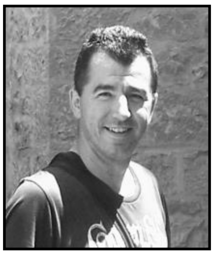
[21] O. Baskan and H. Ceylan, "Modified differential evolution algorithm for the continuous network design problem", 16th Euro Working Group on Transportation, Porto, Portugal, **(2013)** September 4-6.

[22] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto and H. Rashidi, "A review of urban transportation network design problems", European Journal of Operational Research, vol. 229, no. 2, **(2013)**, pp. 281-302.

[23] M. Frank and P. Wolfe, "An algorithm for quadratic programming", Naval Research Logistics Quaterly, vol. 3, **(1956)**, pp. 95-110.

[24] Z. W. Geem, J. H. Kim and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search", Simulation, vol. 76, no. 2, **(2001)**, pp. 60-68.
[25] H. Ceylan, H. Ceylan, S. Haldenbilen and O. Baskan, "Transport energy modeling with meta-heuristic harmony search algorithm, an application to Turkey", Energy Policy, vol. 36, **(2008)**, pp. 2527-2535.
[26] Z. W. Geem, "Optimal design of water distribution networks using Harmony Search, PhD Thesis, Korea University, Seoul, Korea, **(2000)**.
[27] D. Karaboga, "An idea based on honeybee swarm for numerical optimization", Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, **(2005)**.
[28] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm", Applied Mathematics and Computation, vol. 214, **(2009)**, pp. 108-132.
[29] O. Baskan and M. Dell'Orco, "Artificial bee colony algorithm for continuous network design problem with link capacity expansions", 10th International Congress on Advances in Civil Engineering, Ankara, Turkey, **(2012)** October 17-19.
[30] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces", ICSI, USA, Tech. Rep. TR-95-012, **(1995)**.
[31] H. Liu, Z. Cai and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization", Applied Soft Computing, vol. 10, **(2010)**, pp. 629-640.

Author



Ozgur Baskan, he was born in Bursa, Turkey, in 1976. He received the B.Sc. degree in Civil Engineering, the M.Sc. and the Ph.D. degrees in Transportation Engineering from Graduate School of Natural and Applied Sciences in Pamukkale University, in 1998, 2004, and 2009, respectively. Since 2002, he has been a research assistant with the Department of Civil Engineering, Faculty of Engineering, Pamukkale University, and has been involved in several scientific and technical projects. He studies in the fields of traffic and transportation planning, specifically network design, traffic assignment, optimization, and population-based heuristic methods. Dr. Baskan is the author of more than 25 papers in peer-reviewed international journals and conference proceedings.