

# **A Bi-objective Turning Restriction Design Problem in Urban Road Networks**

by

## **Jiancheng Long**

Professor  
School of Transportation Engineering  
Hefei University of Technology  
Hefei 230009, China  
Tel: +86-551-63831101  
E-mail: [jianchenglong@hfut.edu.cn](mailto:jianchenglong@hfut.edu.cn)

## **W.Y. Szeto\***

Assistant Professor  
Department of Civil Engineering  
The University of Hong Kong  
Pokfulam Road  
Hong Kong  
Tel: +852-28578552  
Email: [ceszeto@hku.hk](mailto:ceszeto@hku.hk)

## **Hai-Jun Huang**

Professor  
School of Economics and Management  
Beijing University of Aeronautics and Astronautics  
Beijing 100191, China  
Tel: 86-10-8231-6608  
E-mail: [haijunhuang@buaa.edu.cn](mailto:haijunhuang@buaa.edu.cn)

December 23, 2013

Paper revised and resubmitted to European Journal of Operational Research

---

\* Corresponding author

## **Abstract**

This paper introduces a bi-objective turning restriction design problem (BOTRDP), which aims to simultaneously improve traffic network efficiency and reduce environmental pollution by implementing turning restrictions at selected intersections. A bi-level programming model is proposed to formulate the BOTRDP. The upper level problem aims to minimize both the total system travel time (TSTT) and the cost of total vehicle emissions (CTVE) from the viewpoint of traffic managers, and the lower level problem depicts travelers' route choice behavior based on stochastic user equilibrium (SUE) theory. The modified artificial bee colony (ABC) heuristic is developed to find Pareto optimal turning restriction strategies. Different from the traditional ABC heuristic, crossover operators are captured to enhance the performance of the heuristic. The computational experiments show that incorporating crossover operators into the ABC heuristic can indeed improve its performance and that the proposed heuristic significantly outperforms the non-dominated sorting genetic algorithm (NSGA) even if different operators are randomly chosen and used in the NSGA as in our proposed heuristic. The results also illustrate that a Pareto optimal turning restriction strategy can obviously reduce the TSTT and the CTVE when compared with those without implementing the strategy, and that the number of Pareto optimal turning restriction designs is smaller when the network is more congested but greater network efficiency and air quality improvement can be achieved. The results also demonstrate that traffic information provision does have an impact on the number of Pareto optimal turning restriction designs. These results should have important implications on traffic management.

**Keywords:** Transportation; bi-objective turning restriction design problem; bi-level programming; stochastic user equilibrium; artificial bee colony; crossover operator; vehicle emissions

## 1. Introduction

Congestion is one of the most representative aspects of urban road traffic problems, and has become one of the most important factors that influence people's mobility and travel cost in daily life. Meanwhile, road traffic leads to some intractable problems that attract wide attention from the whole society, such as road safety, environmental pollution, and energy consumption. Traditionally, traffic control, such as signal control, lane allocation, turning restriction, and road pricing, mainly aim to improve traffic efficiency and alleviate traffic congestion. Nowadays, especially in developing countries, the environmental pollution problem caused by vehicle emissions attract more and more attention in the transportation engineering field and become a hot issue for studies (see Mayeres et al., 1996; Orubu, 2004; Pandian et al., 2009; Demir et al. 2012, 2014; Gokhale, 2012; Chung et al., 2013; Ng and Lo, 2013; Zhu et al., 2013 for example).

Vehicle emissions, such as carbon monoxide (CO), nitrogen oxides (NO<sub>x</sub>), volatile organic compounds (VOC), etc., have been shown to have a variety of negative effects on public health and the natural environment (Szeto et al., 2012). Compared to the emissions from industry, power plants, space heating, refuse disposal, etc., vehicle emissions play the most important role in contributing air pollutants in typical urban centers (Bach, 1972). Empirical studies show that transportation is identified as a highly significant contributor, accounting for well over 50% of total air pollutants in some countries (Fu et al., 2001; Orubu, 2004; Goyal et al., 2006). In some developing countries, this percentage is expected to increase quickly in the future because the vehicle growth has not been properly regulated by the government authorities. For example, the motor vehicle population continued to increase at an annual rate of approximately 10% in mega cities in China (Chan and Yao, 2008). This calls for the needs to control the situation. In fact, there are many means that have been successfully adopted to reduce vehicle emissions, such as tightening vehicle emission standards, reducing travel demand (e.g., pricing) and vehicle ownership (e.g., vehicle quota systems), improving traffic management and control (e.g., signal control), etc.

Turning restriction is one of the commonest traffic management techniques and an effective, low cost traffic improvement strategy for urban road networks. Turning restrictions are applied to a group of intersections where travelers are prohibited to drive into restricted downstream links. Long et al. (2010) proposed a turning restriction design problem (TRDP), which is the problem of determining a set of intersections to implement turning restrictions to minimize the total system travel time (TSTT). It has been demonstrated that implementing turning restrictions can effectively improve traffic efficiency by reducing the TSTT. However, the influence of turning restrictions on environmental pollution has not been examined yet. In fact, when a turning restriction is implemented, some drivers need to travel at a longer distance to reach their destinations as their original, shortest routes are not permitted to use. The vehicle emissions and the corresponding monetary values (i.e., the cost of vehicle emissions) are usually more as a result. Although

the overall TSTT is reduced, there may be a tradeoff between minimizing the cost of total vehicle emissions (CTVE) and TSTT. Therefore, this paper investigates a bi-objective TRDP (BOTRDP), which aims to reduce not only the TSTT, but also the CTVE by implementing turning restrictions.

Turning restrictions are implemented by traffic managers, but the implementation should take travelers' route choice behavior into account. A new turning restriction in road networks directly influences some travelers' route choice, because their original routes include the restricted turning and are infeasible after implementing the turning restriction. As a result, the original traffic equilibrium state is disrupted and some travelers adjust their routes. A new equilibrium is realized again only when all travelers cannot be better off by unilaterally changing their routes. Similar to the original TRDP (Long et al., 2010) and the traditional network design problems (see Farahani et al., 2013 for a comprehensive review), the BOTRDP can be mathematically formulated as a bi-level problem. The upper level problem (ULP) of the BOTRDP is a bi-objective optimization problem that minimizes both the TSTT and the CTVE. The lower level problem (LLP) formulates travelers' route choice behavior as the SUE assignment problem, which is more realistic than the deterministic user equilibrium (DUE) assignment problem (Chen and Alfa, 1991b).

Whether a turning restriction is implemented or not can be represented by a binary variable. Hence the ULP of the BOTRDP is a discrete optimization problem, and the BOTRDP is a bi-objective nonlinear bi-level mixed integer programming problem, which has been recognized as one of the most difficult and challenging problems in the transportation research area due to its computational difficulty. In general, all Pareto solutions of the BOTRDP cannot be obtained by using the exact solution methods when the number of binary variables is large. As a result, meta-heuristic methods are always used to find good, but not necessarily Pareto optimal solutions within a reasonable amount of computing time. During the past several decades, many meta-heuristic approaches were developed to tackle the multi-objective optimization problems (see Jones et al., 2002 for a comprehensive review). The most popular meta-heuristics for multi-objective optimization problems include genetic algorithms (GAs) (Yin, 2000, 2002; Shepherd and Sumalee, 2004; Cantarella and Vitetta, 2006; Chen et al., 2006, 2010; Teklu et al., 2007; Gen et al., 2008; Sharma et al., 2009; Xu et al., 2009; Sohn, 2011; Konak, 2012; Miandoabchi et al., 2012a,b; 2013; Szeto et al., 2013b), evolution strategies (Deb and Sinha, 2010; Zheng et al., 2011), simulated annealing (Friesz et al., 1993; Suppapitnarm et al., 2000; Xu et al., 2009), tabu search (Armentano and Claudio, 2004; Jaeggi et al., 2008; Uno and Katagiri, 2008), etc.

The artificial bee colony (ABC) algorithm is a swarm-based meta-heuristic algorithm that was introduced by Karaboga (2005) for solving numerical optimization problems. This algorithm is motivated by the foraging behavior of honey bees. Compared to existing evolutionary algorithms such as GA, one merit of the ABC algorithm is that the ABC algorithm has a better local search mechanism to improve the solution quality. Indeed, the ABC algorithm has been applied to solve other single objective optimization problems

with great success (Karaboga and Basturk, 2008; Karaboga, 2009, 2010; Singh, 2009; Szeto et al., 2011b, Szeto and Jiang, 2012; Szeto et al., 2013a, 2013c). Recently, the ABC algorithm has also been applied to solve the multi-objective optimization problems (Akbari et al., 2012; Omkar et al., 2011; Zhang et al., 2012). It is worthwhile to evaluate the performance of the ABC algorithm for solving the BOTRDP.

In this paper, a bi-level programming model for the BOTRDP is introduced, and the ABC algorithm is developed to solve the BOTRDP. Rather than directly applying the ABC algorithm proposed by Karaboga (2005), this paper improves the ABC algorithm by enhancing its global search ability. For this purpose, crossover operators that are normally used in GA are incorporated in the ABC algorithm. The resulting ABC algorithm has both the strength of the original ABC algorithm and GA. The computational experiments show that the inclusion of the crossover operators can indeed improve the performance of the ABC heuristic, and that the proposed heuristic significantly outperforms the non-dominated sorting genetic algorithm (NSGA) even if various operators are randomly selected and adopted in the NSGA as in our proposed heuristic. The results also demonstrate that the Pareto optimal turning restriction strategy can obviously reduce both TSTT and the cost of total vehicle emissions. It is also found that the number of Pareto optimal turning restriction designs is smaller when the network is more congested but greater network efficiency and air quality improvement can be achieved. The results also illustrate that traffic information provision does have an impact on the number of Pareto optimal turning restriction designs.

The contributions of this paper include the following. (1) This paper proposes a new bi-objective, bi-level problem, which is NP-hard. (2) A modified ABC heuristic is proposed for solving the proposed problem, and the heuristic outperforms the NSGA. (3) The tradeoff between the TSTT and the CTVE under the implementation of turning restrictions is pointed out and illustrated. (4) The effects of demand levels and traffic information provision on the number of Pareto-optimal solutions are demonstrated and discussed.

The rest of the paper is organized as follows: In the next section, a BOTRDP is formulated as bi-level programming model. The modified ABC algorithm is presented in Section 3 to solve the proposed bi-objective problem. In Section 4, numerical examples are given. Finally, conclusions are drawn in Section 5.

## **2. A bi-level programming model for the BOTRDP**

### *2.1. Notations*

Consider a multi-destination and connected network  $G$ .  $N$  denotes the set of nodes whereas  $A$  denotes the set of arcs (links).  $R$  and  $S$  denote the sets of origin and destination nodes, respectively. The following notations are adopted throughout this paper:

- $r$             origin node,  $r \in R$ ;
- $s$             destination node,  $s \in S$ ;

$\Pi'$	set of traffic movements in the whole network;
$\Pi$	set of indices for the traffic movements in the whole network;
$\Pi_a$	set of traffic movements related to link $a$ , and $\Pi_a \subset \Pi$ ;
$x_{ab}$	flow on link $a$ going to link $b$ or the movement flow from link $a$ to link $b$ ; $[a,b] \in \Pi_a$ ;
$\mathbf{x}$	vector of movement flows, $\mathbf{x} = [x_{ab}]$ ;
$\delta_{a,k}^{rs}$	0-1 indicator variable; it equals 1 if link $a$ is on path $k$ connecting $r$ and $s$ , and 0 otherwise;
$\delta_{ab,k}^{rs}$	0-1 indicator variable; it equals 1 if both links $a$ and $b$ are on path $k$ connecting $r$ and $s$ , and 0 otherwise;
$P_k^{rs}$	route choice probability of path $k$ connecting $r$ and $s$ ;
$c_k^{rs}$	travel cost of using path $k$ connecting $r$ and $s$ ;
$q^{rs}$	travel demand from origin $r$ to destination $s$ ;
$t_a(\mathbf{x}, \mathbf{y})$	travel cost function of link $a$ ;
$\mu$	an index representing the traffic movement from upstream link $u_\mu$ to downstream link $d_\mu$ , $u_\mu, d_\mu \in A$ , $[u_\mu, d_\mu] \in \Pi'$ , $\mu \in \Pi$ ;
$\Gamma$	set of indices for turning restrictions (or equivalently, set of candidate traffic movements to be prohibited), and $\Gamma \subset \Pi$ ;
$y_\mu$	0-1 decision variable; if traffic movement $\mu$ is prohibited, then $y_\mu = 1$ ; otherwise $y_\mu = 0$ ;
$\mathbf{y}$	vector of turning restriction decision variables, or turning restriction strategy $\mathbf{y} = [y_\mu]$ ;
$\varphi_{\mu k}^{rs}$	0-1 indicator variable; if both links $u_\mu$ and $d_\mu$ are on path $k$ connecting $r$ and $s$ , then $\varphi_{\mu k}^{rs} = 1$ ; otherwise, $\varphi_{\mu k}^{rs} = 0$ .

## 2.2. The lower level stochastic user equilibrium assignment problem

Since the implementation of turning restrictions make some routes become infeasible, it induces changes in traffic flow over the road network. The BOTRDP must take into an account the travelers' route choice behavior to depict the resultant traffic flow pattern after the implementation. Capturing their route choice behavior into the problem is therefore essential. In the literature, travelers' route choice is generally assumed to follow either the DUE principle (e.g., Friesz et al., 1993; Szeto and Lo, 2005, 2008; Cantarella et al., 2006; Lo and Szeto, 2009; Xu et al., 2009; Sohn, 2011) or the SUE principle (e.g., Chen and Alfa, 1991a; Huang, 1995; Akamatsu, 1996; Maher, 1998; Meng et al., 2004; Bekhor and Toledo, 2005; Chen et al., 2010; Meng and Liu, 2011). The DUE principle and SUE principle assume that travelers select their routes to minimize their actual travel times and perceived travel times, respectively. In reality, it is difficult for travelers to have perfect information on the network condition. Hence, the SUE problem is better than the DUE problem for describing the route choice behavior of travelers. Following Long et al. (2010), a logit-based SUE model is

adopted to formulate the LLP of the BOTRDP, which can be characterized by the following nonlinear equations:

$$x_{ab} = \sum_{rs} \sum_k q^{rs} P_k^{rs} \delta_{ab,k}^{rs}, \quad (1)$$

where

$$P_k^{rs} = \frac{\exp(-\theta c_k^{rs})}{\sum_l \exp(-\theta c_l^{rs})}, \quad (2)$$

and  $\theta$  is the dispersion parameter of the logit model, representing the travel time variation of travelers (Lo and Szeto, 2002). It could be interpreted as the information qualities available to these travelers. A larger value implies a smaller travel time perception variation among travelers and, therefore, better information available to travelers. This parameter can be used to capture the effect of providing better traffic information.

Opposing flows through intersections significantly contributes to the turning delays and travel times of the traffic concerned, and thus affect the route choice of travelers in urban road networks. This paper employs the following generalized BRP function (Long et al., 2010) to capture the effects of opposing flows on link travel time:

$$t_a(\mathbf{x}, \mathbf{y}) = t_a^0 \cdot \left( 1 + \chi_1 \left( \frac{x_{a,TH} + \phi_{RT} x_{a,RT} + \phi_{LT} x_{a,LT} + \phi_{opp} x_{a,opp}}{Q_a} \right)^{\chi_2} \right), \quad (3)$$

where  $t_a^0$  is the free-flow travel time on link  $a$ ;  $x_{a,TH}$ ,  $x_{a,LT}$ ,  $x_{a,RT}$ , and  $x_{a,opp}$  are the through, left turn, right turn, and opposing flows of link  $a$ , respectively;  $Q_a$  is the capacity of link  $a$ ;  $\phi_{LT}$ ,  $\phi_{RT}$ , and  $\phi_{opp}$  are parameters related to the influences of left turn, right turn, and opposing flows on the through flow, respectively;  $\chi_1$  and  $\chi_2$  are the BPR parameters. Opposing flow is a function of turning restriction decision variables because turning restrictions can remove the conflict between traffic flows moving in the concerned and opposing directions. The opposing flow of link  $a$  is formulated as follows:

$$x_{a,opp} = \sum_{\mu \in \Pi} \delta_{\mu a, opp} x_{\mu} (1 - y_{\mu}), \quad (4)$$

where  $\delta_{\mu a, opp}$  is an indicator variable; it is equal to 1 if traffic movement  $\mu$  is an opposing traffic movement of link  $a$ , and  $\delta_{\mu a, opp}$  is equal to zero otherwise.  $x_{\mu}$  denotes the flow of traffic movement  $\mu$ . According to (4), the flow  $x_{\mu}$  of a particular traffic movement  $\mu$  will only be contributed to the opposing flow  $x_{a,opp}$  of link  $a$  only if the movement conflicts with the flow of link  $a$  ( $\delta_{\mu a, opp} = 1$ ) and there is no turning restriction for that movement ( $y_{\mu} = 0$  or  $\mu \notin \Gamma$ ).

The BPR function (3) is the generalized version of the link cost function in Horowitz (1997) and can capture the effect of turning flows on the link travel time in addition to the opposing flow. Using link travel time function (3), we can obtain route travel times by Eq. (5):

$$c_k^{rs} = \sum_a t_a(\mathbf{x}, \mathbf{y}) \delta_{a,k}^{rs} + \sum_{\mu} y_{\mu} M \varphi_{\mu k}^{rs}, \quad (5)$$

where  $M$  is a positive constant, which can be considered as the punishment cost for a traveler taking a restricted turning. The value of  $M$  should be far greater than the maximum travel time of all paths in the choice set so that very few travelers take restricted turnings. The first term on the right hand side of Eq. (5) is the actual route travel time and is equal to the sum of the travel times of the links on the route. The second term is introduced to avoid restricted turnings to be used.

### 2.3. The upper level bi-objective optimization problem

The ULP of the TRDP is an optimization problem with two objectives. The first objective is to improve network traffic efficiency or mobility, i.e., minimizing the TSTT. The second objective is to reduce total vehicle emissions. These two objectives are set from the viewpoint of traffic managers and are related to traffic network performance. In fact, the choice of objectives depends on applications. If the leader of the upper level problem is concerned with user benefit, then one of the objectives of the problem should be maximizing the user benefit.

#### 2.3.1. The total system travel time

The TSTT was traditionally used as a performance measure of traffic efficiency in a traffic network. The TSTT is defined as the sum of travel times of all travelers traveling from their origins to their destinations, and also equals the sum of the products of link flows and the corresponding link travel times. The general expression of the TSTT can be formulated as follows:

$$T = \sum_a x_a(\mathbf{y}) t_a(\mathbf{x}(\mathbf{y}), \mathbf{y}), \quad (6)$$

where  $T$  denotes the TSTT and  $T = T(\mathbf{x}(\mathbf{y}), \mathbf{y})$  or simply  $T = T(\mathbf{y})$ ;  $x_a(\mathbf{y})$  is the flow on link  $a$  and  $x_a(\mathbf{y}) = \sum_b x_{ab}$ ;  $x_a(\mathbf{y})$  is an implicit function of  $\mathbf{y}$ , which can be obtained by solving the LLP.

#### 2.3.2. The cost of total vehicle emissions

This cost relies on the assumption that the emission rate of a link is a function of the average speed of vehicles on it. The average speed can be found by the following:

$$s_a = \frac{L_a}{t_a(\mathbf{x}(\mathbf{y}), \mathbf{y})}, \quad (7)$$

where  $s_a$  is the average speed of link  $a$  and in ft/s, and  $L_a$  is the length of link  $a$  in ft. The average speed is used in the following emission rate function (Mayeres et al., 1996):

$$\varphi_a^m(\mathbf{x}(\mathbf{y}), \mathbf{y}) = \frac{A_m \exp\{B_m s_a\}}{C_m s_a}, \quad (8)$$

where  $A_m$ ,  $B_m$  and  $C_m$  are constants with respect to pollutant  $m$ , and  $\varphi_a^m(\mathbf{x}(\mathbf{y}), \mathbf{y})$  is in grams per vehicle second.



Using the emission rate function (8), link flows, and the unit emission cost of each pollutant, we can obtain the total emissions of the whole network, which can be further translated into a monetary measure, i.e., the CTVE. The CTVE can be calculated by:

$$E = \sum_m \sum_a \eta_m x_a(\mathbf{y}) \varphi_a^m(\mathbf{x}(\mathbf{y}), \mathbf{y}) t_a(\mathbf{x}(\mathbf{y}), \mathbf{y}), \quad (9)$$

where  $\eta_m$  is the emission cost of unit pollutant  $m$ ;  $E$  denotes the CTVE and  $E = E(\mathbf{x}(\mathbf{y}), \mathbf{y}) = E(\mathbf{y})$ .

### 2.3.3. The bi-objective optimization problem

Whether a traffic movement is restricted or not is depicted by a 0-1 variable. Therefore, the BOTRDP is an integer programming problem. The problem aims to improve the network performance by reducing network-wide congestion and vehicle emissions simultaneously. They can be measured by the TSTT and the CTVE. Hence, the ULP of the BOTRDP is formulated as the following optimization problem:

$$\min \quad [T, E] = \left[ \sum_a x_a(\mathbf{y}) t_a(\mathbf{x}(\mathbf{y}), \mathbf{y}), \sum_m \sum_a \eta_m x_a(\mathbf{y}) \varphi_a^m(\mathbf{x}(\mathbf{y}), \mathbf{y}) t_a(\mathbf{x}(\mathbf{y}), \mathbf{y}) \right], \quad (10)$$

$$\text{subject to } y_\mu = \{0,1\} \quad \forall \mu \in \Gamma, \quad (11)$$

where the objective (10) is to minimize the TSTT and the CTVE simultaneously. Constraint (11) is the definitional constraint for every turning restriction decision variable. Note that the objective function is a function of the lower level decision variable.

## 3. An artificial bee colony algorithm for the BOTRDP

### 3.1. Overview of artificial bee colony algorithm

The ABC algorithm, firstly proposed by Karaboga (2005), is one of the most recently evolutionary algorithms motivated by the intelligent behavior of honey bees to search for foods near their hives. Similar to particle swarm optimization (PSO) and differential evolution (DE) algorithms, the ABC algorithm provides a population-based search procedure. In the ABC algorithm, the position of a food source represents a feasible solution to the problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The bees' aim is to discover the places of food sources with a large amount of nectar and finally the one with the largest nectar (Karaboga, 2010). This corresponds to finding good solutions in the solution space.

There are three groups of bees in the ABC algorithm: (i) employed bees, (ii) onlookers, and (iii) scouts (Szeto et al., 2011b). Both employed bees and onlookers fly around in a multidimensional search space and choose food sources depending on their own experience and their nestmates' experience. Each of them only searches for a new food source in the neighborhood of the food source in their memories and only memorizes the positions of good food sources. If the nectar amount of a new source is larger than that of the previous one in their memory, then each of them memorizes the new position and forgets the previous one.

Effectively, employed bees and onlookers carry out local searches in the solution space. The major difference between employed bees and onlookers is that each employed bee memorizes one food source that is not identical to the food sources memorized by other employed bees, whereas two or more onlookers can memorize the same food source before the start of finding a new food source in each iteration to have a more intensive local search in the region near that food source. The number of onlookers memorizing a particular food source depends on its relative amount of nectar. More onlookers will memorize a food source if the food source is better. Onlookers watch the dance of employed bees within the hive after one search to have the information of the nectar amount in each food source. Each of the onlookers then chooses a food source to memorize based on the relative amount of nectar informed by employed bees. Afterwards, each of the onlookers with the same food source in their memories searches for a new food source in the neighborhood of the old food source in their memories and memorizes the better one among the new and old food sources. Then, the food sources memorized by these onlookers are compared and only the best one is informed and memorized by the employed bee. (i.e., the old food source is replaced by the best one if the best one is better than the old one.) The number of employed bees is equal to the number of food sources (solutions) since each employed bee is associated with one and only one food source. The number of onlookers is usually equal to that of employed bees so as to reduce the number of parameters. Different from employed bees and onlookers, scouts fly around and choose their food sources randomly without using their experience, corresponding to carrying out global searches in the solution space. Scouts are related to employed bees because the employed bee becomes a scout if no better neighbor food source can be found in *limit* (a predetermined number) successive iterations. After finding a new food source, the scout becomes an employed bee again. Very often, there is no restriction on the number of scouts, but we can set a maximum number of scouts in each iteration.

Similar to most of meta-heuristic methods, the ABC algorithm is an iterative algorithm, which initially generates random solutions as food sources and assigns each employed bee to a unique food source. Thereafter, the nectar of food sources is exploited by firstly employed bees and then onlookers in each iteration. This continual exploitation can ultimately cause the food sources to become exhausted in certain iteration. Hence, after the local search of onlookers, the ABC heuristic checks whether any current food source should be abandoned in each iteration. A current food source will be abandoned by an employed bee in a particular iteration if no better neighbor food source can be found in *limit* successive iterations. Then, the employed bee becomes a scout in search for another new food source randomly. After that, the scout becomes an employed bee again. If no food sources are abandoned in an iteration, then there are no scouts in that iteration. A new iteration starts again by performing new local searches by first employed bees and then onlookers until a maximum number of iterations is reached.

### 3.2. Application to the BOTRDP

#### 3.2.1. Solution representation

We directly use a straightforward solution representation scheme. Fig. 1 illustrates a representation of a turning restriction scheme. The length equals the maximum number of turning restrictions that can be applied to the network. Each position in the solution vector represents a candidate traffic movement to be prohibited at a selected junction. A value of one in a position means that a turning restriction prohibits the candidate traffic movement and a value of zero means that the traffic movement is not prohibited. In Fig. 1, six turning restrictions out of thirteen potential turning restrictions are implemented.

#### 3.2.2. Initial solution

An initial solution is constructed by a binary vector. Each element of the vector takes a value of one with a predetermined probability  $P_0$ . i.e., each turning restriction is adopted with the probability  $P_0$ . This procedure is repeated until a total of  $m$  initial solutions are generated where  $m$  is the number of employed bees. In this paper, we set  $P_0 = 0.5$ .

#### 3.2.3. Neighborhood operators

In the ABC heuristic, a neighborhood operator is used to obtain a new solution  $\tilde{\mathbf{y}}$  from the current solution  $\mathbf{y}$ . This paper develops two categories of neighborhood operators for the BOTRDP: mutation operators and crossover operators. The mutation operators perform local searches in the neighborhood of current solutions. The crossover operators perform global searches and are basic operators in GAs. The crossover operators in this paper require two parents to produce an offspring. The offspring consist of the genetic information of both parents. At least one of the parents is required to be a *non-dominated solution* since non-dominated solutions contain more useful information and a crossover of a food source with a non-dominated solution can quickly produce a good solution. A good combination of the two categories of operators can strike a balance between exploration and exploitation processes. The developed operators include the following:

##### (1) Point mutation

This operator randomly selects one position  $\mu$  (or traffic movement  $\mu$ ) in the solution vector  $\mathbf{y}$  and then set  $y_\mu = 1 - y_\mu$  to obtain  $\tilde{\mathbf{y}}$ . As shown in Fig. 2, the seventh position is randomly selected (i.e.,  $\mu = 7$ ) among the 13 positions of the solution. Then, the operator changes  $y_7$  from 1 to 0 at the seventh position and does not change the values of  $y_\mu$  at other positions to obtain a new solution.

##### (2) Random mutation

This operator changes the value of every position  $\mu$  of  $\mathbf{y}$  from  $y_\mu$  to  $1 - y_\mu$  with a probability of  $P_m$ . An example is given in Fig. 3. The third, seventh, eighth, and twelfth positions are selected for

flipping their values from one to zero or vice versa because the random numbers generated are not greater than  $P_m$ . The random numbers generated at other positions are greater than  $P_m$ .

(3) Section mutation

This operator randomly selects two positions  $\mu_1$  and  $\mu_2$  in  $\mathbf{y}$  with  $\mu_1 \leq \mu_2$  without loss of generality. Afterwards, this operator changes the value of every position  $\mu$  between the two selected positions inclusively (i.e.,  $\mu_1 \leq \mu \leq \mu_2$ ) from  $y_\mu$  to  $1 - y_\mu$  to obtain  $\tilde{\mathbf{y}}$ . Fig. 4 illustrates an example, where  $\mu_1 = 6$  and  $\mu_2 = 9$ . All the values in the positions between the sixth and ninth positions are changed from zero to one or vice versa. All other values remain unchanged.

(4) Point crossover with a non-dominated solution

Let  $\Phi$  be the current set of non-dominated solutions. Then, this operator randomly selects a non-dominated solution  $j$ , denoted as  $\mathbf{y}_j \in \Phi$ , and randomly selects a position  $\mu$  in such  $\mathbf{y}$  that  $\mathbf{y}$  and  $\mathbf{y}_j$  have a different value at this position. Replace the value in position  $\mu$  in  $\mathbf{y}$  by that in  $\mathbf{y}_j$  to get  $\tilde{\mathbf{y}}$ . An example is shown in Fig. 5. The first row gives the current non-dominated solution of parent 1 whereas the second row gives the solution of parent 2. They have a different value at positions 1, 4, 5, 7, 8, 9, 12, and 13. Position 7 is randomly selected out of all these positions and the value of this position of parent 2 is replaced by that of parent 1 to form an offspring shown in the last row.

(5) Random crossover with a non-dominated solution

This operator randomly selects a non-dominated solution  $\mathbf{y}_j \in \Phi$  and replace the value in every position in  $\mathbf{y}$  by that in  $\mathbf{y}_j$  with a probability of  $P_m$ . An example is given in Fig. 6. As in Fig. 5, the first row gives the current non-dominated solution of parent 1 whereas the second row gives the solution of parent 2. By comparing the random numbers generated for each position and the probability  $P_m$ , it is found that positions 4, 8 and 11 of parent 2 are selected and their values are replaced by those values in the corresponding position of parent 1 to form an offspring.

(6) Section crossover with a non-dominated solution

This operator randomly selects a non-dominated solution  $\mathbf{y}_j \in \Phi$ , and two positions  $\mu_1$  and  $\mu_2$  ( $\mu_1 \leq \mu_2$ ). Replace the sequence formed by the values in the positions between  $\mu_1$  and  $\mu_2$  in  $\mathbf{y}$  by the sequence formed by the values in the corresponding positions in  $\mathbf{y}_j$  to obtain  $\tilde{\mathbf{y}}$ . An example is illustrated by Fig. 7 with  $\mu_1 = 6$  and  $\mu_2 = 9$ . The crossover section is formed by the values in positions 6-9 and this section in parent 2 is replaced by that in parent 1 to form a new solution.

The neighborhood operator in the ABC heuristic can be formed by either a single pre-selected operator or a combination of some of above operators. The latter randomly chooses one neighborhood operator from a predetermined set of neighborhood operators at a time to find a new food source near a selected food source. In this paper, we also consider an alternative combination of operators: At a time, we firstly choose either the category of crossover operators with a predetermined probability  $P_c$  or the category of mutation operators

with a predetermined probability  $1 - P_c$ , and then randomly select one neighborhood operator from the selected category of operators. The results in Section 4 show that too many crossover operators selected may not lead to the best result. The probability  $P_c$  can control the frequency of crossover operators used and determine the number of Pareto solutions obtained.

#### 3.2.4. Fitness evaluation

Before evaluating the fitness of a food source, we need to obtain the two objective values of the ULP, which requires the equilibrium link flows, i.e., the solution of the associated LLP. There are many solution algorithms to solve the lower level SUE problem, such as the method of successive averages (MSA) (Powell and Sheffi, 1982; Sheffi and Powell, 1982; Huang and Bell, 1998; Huang and Li, 2007), the optimal step length algorithm (OSLA) (Chen and Alfa, 1991a; Huang, 1995; Akamatsu, 1996; Maher, 1998), the Gradient Projection (GP) algorithm (Chen et al., 2002; Bekhor and Toledo, 2005), etc. Recently, Liu et al. (2009) improved the MSA method and proposed the self-regulated averaging method (SRAM) (see Appendix A for the algorithmic steps and the convergence requirements) to solve the SUE problem. The MSA relies on a predetermined step size for guaranteeing convergence, and its convergence speed is very slow due to improper step sizes (too large or too small) at some iterations. The SRAM can deal with this slow convergence problem and has been successfully applied to solve the SUE problem (Liu et al. 2009; Long et al., 2010) and a multi-class doubly stochastic dynamic user equilibrium problem (Szeto et al., 2011a). Ideally, the SRAM could be used to solve the lower level SUE problem for general networks. Indeed, in all our numerical studies, the SRAM converged to the equilibrium flow patterns. However, the link cost vector functions in our lower level problem do not have a symmetric continuous Jacobian with respect to the link flow vector. Hence, the convergence requirement of the SRAM is not met. Therefore, for general applications, we provide another version of the SRAM that can guarantee convergence when solving our lower level problem. The details of another version of the SRAM and its convergence requirements are given in Appendix B.

Fitness evaluation is a basic step in the ABC heuristic for solving multi-objective optimization problems. Since 1980s, several fitness assignment mechanisms have been proposed and applied in multiobjective optimization problems. Gen et al. (2008) classified the fitness assignment mechanisms into three categories: vector evaluation approach (e.g., Schaffer, 1985; Omkar et al., 2011), Pareto ranking with diversity approach (e.g., Goldberg 1989; Fonseca and Fleming, 1993; Srinivas and Deb, 1994; Deb et al., 2002; Zhang et al., 2011), and weighted sum with elitist preserving approach (e.g., Ishibuchi and Murata, 1998; Zitzler and Thiele, 1999; Akbari et al., 2011). This paper will use the second approach. The merit of this approach is that the fitness evaluation does not require specifying any parameters in advance. The Pareto ranking based fitness assignment method was first proposed by Goldberg (1989). The ranking procedure is as follows (Gen

et al., 2008): giving rank 1 to the non-dominated food sources and removing them from contention, then finding the next set of non-dominated food sources and assigning rank 2 to them and then removing them for contention. The process continues until the entire population is ranked. Besides Pareto ranking, we also use the distance of a food source to the current Pareto frontier to evaluate the quality of the food sources. The closer the food source to the Pareto frontier, the higher quality the food source has. The distance of a food source to the current Pareto frontier can be formulated as follows:

$$d(\mathbf{y}_i) = \min_{\mathbf{y}_j \in \Phi} \left\{ \sqrt{[T(\mathbf{y}_i) - T(\mathbf{y}_j)]^2 + [E(\mathbf{y}_i) - E(\mathbf{y}_j)]^2} \right\}, \quad (12)$$

where  $\Phi$  is the set of current Pareto solutions.  $d(\mathbf{y}_i)$  is the distance of food source  $i$  to the closest, existing Pareto solution.  $T(\mathbf{y}_j)$  and  $E(\mathbf{y}_j)$  are the TSTT and CTVE of current Pareto solution  $j$ , respectively.

The food source with a higher ranking will have a higher fitness, and the fitness of a food source is inversely related to its distance to the Pareto frontier. Therefore, the following fitness function is adopted to evaluate the quality of each food source:

$$f(\mathbf{y}_i) = -r(\mathbf{y}_i)D - d(\mathbf{y}_i), \quad (13)$$

where  $r(\mathbf{y}_i)$  is the Pareto ranking of food source  $i$  and  $D$  is a very large positive constant, for example  $D = \max\{d(\mathbf{y}_i)\}$ . By definition, the ranking of all the food sources on the Pareto frontier is 1 and their distance to the Pareto frontier is 0. Hence, they have the largest fitness  $-D$ .

### 3.2.5. Selection of food sources

On the onlooker phase, each onlooker randomly selects a food source. Both the roulette wheel selection method and the tournament selection method can be used to drive the selection process towards better food sources. This paper applies the latter since its implementation is simpler than that of the former. For each onlooker, two food sources  $\mathbf{y}_i$  and  $\mathbf{y}_j$  are selected randomly. If  $f(\mathbf{y}_i) > f(\mathbf{y}_j)$ , then the onlooker selects food source  $\mathbf{y}_i$ ; if  $f(\mathbf{y}_i) < f(\mathbf{y}_j)$ , then the onlooker selects food source  $\mathbf{y}_j$ ; otherwise, the onlooker randomly selects one of the two food sources with the same probability.

### 3.2.6. The ABC algorithm

The steps of the ABC algorithm for solving the BOTRDP are summarized as follows:

Step 0. *Initialization.* Randomly generate a set of solutions as initial food sources  $\mathbf{y}_i, i = 1, 2, \dots, m$ , where  $m$  is the number of employed bees. Assign each employed bee to a food source, and set the index  $\kappa = 0$ , the set of non-dominated solutions  $\Phi = \emptyset$  and the trial number  $l_1 = l_2 = \dots = l_m = 0$ . Set the values of *limit* and *MaxIterations*.

Step 1. *Pareto ranking.* Combine the set of non-dominated solutions  $\Phi$  with all food sources  $\mathbf{y}_i, i = 1,$

2, . . . ,  $m$  and implement the Pareto ranking. Reset  $\Phi$  to be the set of all the food sources on the current Pareto frontier.

Step 2. *Fitness update.* Using Eq. (13) to evaluate the fitness  $f(\mathbf{y}_i)$  for each of the food sources  $\mathbf{y}_i, i = 1, 2, \dots, m$ .

Step 3. *Working bee phase.*

Step 3.1. *Employed bee phase.* For each food source  $\mathbf{y}_i$ , apply a neighborhood operator on  $\mathbf{y}_i \rightarrow \tilde{\mathbf{y}}_i$ . If  $f(\tilde{\mathbf{y}}_i) > f(\mathbf{y}_i)$ , then replace  $\mathbf{y}_i$  with  $\tilde{\mathbf{y}}_i$  and set  $l_i = 0$ ; otherwise, set  $l_i = l_i + 1$ .

Step 3.2. *Onlooker phase.* For each onlooker, select a food source  $\mathbf{y}_i$  using the tournament selection method, and apply a neighborhood operator on  $\mathbf{y}_i \rightarrow \tilde{\mathbf{y}}_i$ . If  $f(\tilde{\mathbf{y}}_i) > f(\mathbf{y}_i)$ , then replace  $\mathbf{y}_i$  with  $\tilde{\mathbf{y}}_i$  and set  $l_i = 0$ ; otherwise, set  $l_i = l_i + 1$ .

Step 3.3. *Scout phase.* For each food source  $\mathbf{y}_i$ , if  $l_i = \text{limit}$ , then replace  $\mathbf{y}_i$  with a randomly generated solution.

Step 4. *Convergence test.* Set  $\kappa = \kappa + 1$ . If  $\kappa = \text{MaxIterations}$ , then stop; otherwise, return to step 1.

#### 4. Numerical studies

We adopted the Sioux Falls network (Leblanc et al., 1975) as shown in Fig. 8 to illustrate the performance of the proposed ABC algorithm and the properties of the proposed problem. The Sioux Falls network consists of 24 nodes, 76 links, and 528 OD pairs. The feasible turning restrictions are listed in Table 1. The two BPR parameters are  $\chi_1 = 0.15$  and  $\chi_2 = 4$  (Leblanc et al., 1975). Following Horowitz (1997), we use a  $\phi_{opp}$ -value of 0.4. Both  $\phi_{LT}$  and  $\phi_{RT}$  were set to 1 for simplicity. We set the punishment cost  $M = 50.0$ , which is believed to be large enough in this study because the maximum route travel time is less than 2.0 if there are no turning restrictions in the network. Following Long et al. (2010), the path set was generated by a combination of the link elimination method (see Bekhor et al., 2001 for details) and Dial's (1971) STOCH method (see, Dial, 1971; Sheffi, 1985 for details), where the STOCH method can be used to generate the basic path set and the link elimination method can generate some shorter paths omitted by the STOCH method. The average number of paths per OD pair is 3.6 and the maximum number of routes between an OD pair is 15. The values of the parameters for the solution algorithm of the lower level SUE problem are as follows:  $\eta = 1.5$ ,  $\gamma = 0.1$ , and  $\varepsilon = 1.0 \times 10^{-5}$ . Parameters of emission rate functions and the emission costs of unit pollutant are given in Table 2. Since right turns have the lowest influence on other traffic movements for the case of drive-to-the-right as in China, this study does not consider any right turning restrictions. Since the most CPU time was spent on solving the lower level SUE problem, we compared the performance of the algorithms under the same number of SUE traffic assignment problems solved. The maximum number of SUE traffic assignment problems solved is 50000.

We used the number of obtained Pareto solutions to evaluate the performance of the proposed algorithm.

To do this, all the Pareto solutions ought to be found firstly. However, the set of Pareto solutions was unknown beforehand. We ran the ABC algorithm 20 times for each test scenario in this paper using the above setting and combined the set of non-dominated solutions of each test scenario to estimate the set of Pareto solutions. Specially, a total of 62 Pareto solutions was obtained when  $\theta = 1.0$ , which was used to calibrate the parameters of the proposed ABC algorithm.

#### 4.1. ABC algorithm calibration

##### 4.1.1. Calibrating the number of food sources

Following Karaboga and Basturk (2008), the number of employed bees and the number of onlookers were set to be equal to the number of food sources memorized by all the employed bees at any iteration to reduce the number of parameters. The point mutation operator was used in the ABC algorithm to calibrate the number of food sources. We ran the algorithm 20 times for each predetermined number of food sources considered initially. The average numbers of Pareto solutions obtained by the ABC algorithm with different initial number of food sources are shown in Fig. 9. It can be seen that the average number of Pareto solutions has an increasing tendency firstly and drops later as the number of food sources grows up. The best number of food sources is 80, which was used in the following experiments.

##### 4.1.2. Calibrating limit

As mentioned earlier, a food source  $\mathbf{y}_i$  will be abandoned if no better neighbor solutions  $\tilde{\mathbf{y}}_i$  can be found in the neighborhood of  $\mathbf{y}_i$  in consecutive *limit* iterations. Karaboga (2009) has shown that this parameter is important regarding the performance of the ABC algorithm on solving function optimization problems. Hence, we calibrated this parameter. We ran the algorithm 20 times for each predetermined value of *limit* and determined the average numbers of Pareto solutions obtained by the ABC algorithm with different values of *limit*. The results are presented in Fig. 10, showing that the average number of Pareto solutions has an increasing tendency when *limit* is less than 150, and then fluctuates around a constant value when *limit* is greater than 150. In the following experiments, we set *limit* to be equal to 200.

##### 4.1.3. Calibrating the mutation probability

Calibrating the mutation probability is required for incorporating the random mutation operator in the ABC algorithm. Therefore, we only used the random mutation operator as the neighborhood operator in the ABC algorithm in this numerical experiment and repeated the experiment 20 times for each predetermined value of mutation probability to determine the average number of Pareto solutions for each case. The results are shown in Fig. 11. From this figure, we can observe that the average number of Pareto solutions has an increasing tendency firstly and drops later as the mutation probability grows up. The best value of mutation



probability is 0.01, which was used in the following experiments.

#### 4.1.4. Comparison of neighborhood operators

Numerical experiments were set up to test the performance of the six neighborhood operators. Hence, six variants of ABC heuristics were developed, each with only one operator. Each of the six variants was run 20 times and the results are reported in Table 3. The results illustrate that different neighborhood operators performed differently. In particular, the ABC heuristic with the operator of point mutation or random mutation achieved the best average number of Pareto solutions while the ABC heuristic with point crossover or random crossover operator did not yield promising results. Moreover, we can observe that the ABC heuristic with section mutation and section crossover operators did not yield any Pareto solutions. This is because the solutions were changed a lot after carrying out section mutation or section crossover, and hence the heuristic could not perform a good local search.

In the ABC heuristic, all the neighborhood operators can be used randomly. Since including either the section mutation operator or the section crossover operator in the ABC algorithm did not yield any Pareto solutions, these two operators were omitted and the rest of the four operators formed the available set of neighborhood operators for being randomly chosen. This combined approach was tested and the results are reported in the last column of Table 3. One can observe from the table that the combined approach outperformed all the approaches of using single operator in terms of obtaining the average number and standard deviation of Pareto solutions obtained in 20 runs. Moreover, the maximum number of Pareto solutions obtained by the combined approach in 20 runs is very promising and is very close to the best known number of Pareto solutions (i.e., 62). An explanation for this result is that the crossover operator is useful for the ABC heuristic as it can improve the global search ability of the algorithm.

#### 4.1.5. Calibrating the probability of crossover

This numerical experiment aims to test the effect of the probability of crossover  $P_c$  in an alternative combined approach on the performance of the algorithm. The idea of this alternative combined approach is as follows: At a time, we firstly choose either the category of crossover operators with a predetermined probability  $P_c$  or the category of mutation operators with a predetermined probability  $1 - P_c$ , and then randomly select one neighborhood operator from the selected category of operators. We considered two combinations of operators: Combined approach (i) and combined approach (ii). Both allow the random and point mutation operators as well as the random and point crossover operators to be chosen. However, the first approach did allow the section crossover operator to be chosen and the second approach did not. Hence, there are two variants of ABC algorithm. We varied the crossover probability  $P_c$  from 0.00 to 0.20 with an increment of 0.01, and for each value of  $P_c$ , we solved for solutions using the two variant 20 times. The

average numbers of Pareto solutions obtained by the two variants are shown in Fig. 12. From the figure, one can observe that the average number of Pareto solutions increases firstly and decreases later, and combined approach (i) outperforms combined approach (ii). The best crossover probability for combined approach (i) is within 0.05 to 0.15. In the following experiments, the section crossover operator was included, and the crossover probability took a value of 0.10.

Fig. 13 shows the plot of the average number of Pareto solutions found by the ABC heuristic using the point mutation operator only, combined approach (i) with  $P_c = 0.5$ , and combined approach (i) with  $P_c = 0.1$ . The results presented in Fig. 13 show that both combined approaches outperformed the approach of only using the operator of point mutation, which confirm the results presented in Table 3 and Fig. 12 that incorporating crossover operators in the ABC heuristic indeed improves the performance of the ABC heuristic. In the first combined approach (i.e., combined approach (i) with  $P_c = 0.5$ ), the probability of using all crossover operators is much higher than that in the second combined approach (i.e., combined approach (i)  $P_c = 0.1$ ). One can observe from Fig. 13 that the first combined approach outperformed the second combined approach during the initial iterations, but was inferior to the second combined approach when the number of iterations grew up.

#### 4.1.6. Comparison with genetic algorithms

To further illustrate the merits of the proposed method, we compared our proposed ABC heuristic with GAs. One of the GAs considered is the non-dominated sorting genetic algorithm (NSGA). It is an extension of GA (Srinivas and Deb, 1994; Deb et al., 2002), which has been successfully applied to a wide range of multiobjective optimization problems (e.g., Vilcot and Billaut, 2008; Whittaker et al., 2009; Fernandez et al., 2011). Herein, we applied the NSGA to solve the proposed bi-objective TRDP. Previous results show that incorporating all the operators of crossover in this paper into the ABC heuristic can improve the performance of the heuristic. Hence, we also incorporated all those operators into the NSGA and allowed these operators to be selected randomly during the computation, leading to a modified NSGA (MNSGA). The parameters for the NSGA and MNSGA were set as follows: population size = 65, crossover probability = 0.4 and mutation probability = 0.03. The experimental results of the NSGA, the MNSGA and the ABC heuristic are presented in Table 4. One can observe that the MNSGA slightly outperforms the NSGA. This implies that incorporating all the operators of crossover into the NSGA can improve its performance. One can also observe from Table 4 that the ABC heuristic remarkably outperforms the two GA-based methods.

#### 4.2. The effect of the value of the dispersion parameter

We changed the value of the dispersion parameter and applied the proposed ABC algorithm to solve the BOTRDP. The Pareto frontiers under different values of the dispersion parameter are graphically displayed in Fig. 14. We can clearly observe a trade off between improving traffic efficiency (i.e., lowering the TSTT)

and mitigating traffic-induced environment pollution (i.e., lowering the CTVE). We can also observe that the Pareto frontier moves towards the origin when the value of the dispersion parameter  $\theta$  grows up. The result implies that both the TSTT and CTVE decrease as the value of the dispersion parameter  $\theta$  grows up or as better traffic information is provided to travelers.

The numbers of Pareto solutions under different values of the dispersion parameter  $\theta$  are given in Fig. 15. We can observe that the number of solutions on the Pareto frontier increases as the value of the dispersion parameter  $\theta$  grows up. This result is consistent with that in Fig. 14, where the number of Pareto solutions with respect to  $\theta=1.5$  is greater than that with respect to  $\theta=1.0$ , and the number of Pareto solutions with respect to  $\theta=1.0$  is greater than that with respect to  $\theta=0.5$ . This implies that the network has more alternative Pareto improvements when  $\theta$  is larger and vice versa. It may be because when the perception error of network status is large (i.e., the value of  $\theta$  is small), the drivers tends to use all the available routes, not concentrating on a few shortest routes. Hence, fewer effective alternatives can be found.

There are two special solutions for the BOTRDP: only optimizing the TSTT and only optimizing the CTVE. The objective values of the two solutions under various values of the dispersion parameter are given in Fig. 16. We can observe that both the two solutions can improve the network performance in terms of reducing the TSTT. Moreover, according to Fig. 16 (a), when the objective is to minimize the TSTT, the better the traffic information perceived by travelers, the lower the TSTT obtained. This is quite reasonable as fewer travelers travel on non-shortest routes. Furthermore, the TSTT obtained must be smaller than that under the objective of minimizing the CTVE. This also makes sense because by definition, the global minimum of the TSTT must be smaller than other TSTTs obtained by other feasible solutions, including the turning restriction scheme obtained under minimizing the CTVE. However, as the perception of travelers on the traffic network status is improved (i.e., as the dispersion value increases), the TSTT does not decrease monotonically under the objective of minimizing the CTVE. This is due to the tradeoff between the two objectives. Hence, there are sudden increases in TSTT as the dispersion value increases.

Similarly, we observe from Fig. 16 (b) that when the objective is to minimize the CTVE, the better the traffic information perceived by travelers, the lower the minimum CTVE obtained. The minimum CTVE must also be smaller than the CTVE obtained under the objective of minimizing the TSTT. More importantly, Fig. 16 (b) shows that the network performance can be worsened in terms of increasing the CTVE when we only consider optimizing the TSTT and the traffic information availability to travelers is good. Therefore, traffic managers should consider the congestion issue when they implement traffic controls to improve the CTVE.

#### 4.3. The effect of the traffic demand level

It is also interesting to test the effect of traffic demand levels on the solutions of the BOTRDP. The demand level reflects the degree of network congestion. The demand matrix of the Sioux Falls network was

uniformly varied by multiplying by a constant demand scaling factor, where the factor represents the demand level relative to the base demand matrix. Since the Sioux Falls network in the base scenario (i.e., when demand factor equals 1) is quite congested, we only varied the factor from 0.5 to 1.5, with an increment of 0.1. The numbers of Pareto solutions under different traffic demand levels are shown in Fig. 17. The results show that the number of Pareto solutions decreases as the travel demand level grows up. This implies the number of alternatives of implementing turning restrictions is smaller when traffic network becomes more congested. Perhaps, it is because when the network is congested, it is not so easy to apply turning restrictions (abandon traffic movements) that do not result in an increase in total travel time without increasing the total emissions or vice versa. The travel time increase due to detouring is larger when the network is more congested because the travel time is highly affected by the congestion level. Hence, as the network is more congested, the increase in travel time due to detouring is much easier to be greater than the reduction of the travel time on a link due to lowering opposing flow. Moreover, the increase in vehicle emissions due to detouring is larger when the network is more congested because the speed is lower and is farther away from the optimal speed. Hence, fewer effective alternatives exist when the demand level is higher.

We also considered the two special solutions for the BOTRDP, i.e., only optimizing the TSTT or only optimizing the CTVE, when traffic demand level varies. Fig. 18 presents the relative network performance improvement after implementing the turning restrictions according to the two special solutions. The results show that the TSTT can be reduced when the BOTRDP only considers optimizing the TSTT. However, the TSTT can be increased after implementing the turning restrictions according to the optimal solution that only minimizes the CTVE. As shown in Fig. 18 (a), the TSTT after implementing the optimal turning restrictions in terms of minimizing the CTVE only is greater than that of the case of no turning restriction when the traffic demand level is less than 0.9. Similar results can be observed in Fig. 18 (b), which shows that the CTVE increases after implementing the optimal turning restrictions in terms of optimizing the TSTT only when the traffic demand level is less than 0.9. The results presented in Fig. 18 also illustrates that the network performance can be improved in terms of both reducing the TSTT and reducing the CTVE when the traffic demand level is high. Moreover, the more congested the traffic network is, the more improvement the network performance is in terms of both objectives. Therefore, the implementation of turning restrictions is very useful for improving the performance of congested traffic networks. Since the number of alternatives of implementing turning restrictions decreases as traffic demand grows up as observed in previous result, it is relatively easier to select Pareto improved turning restrictions for more congested networks.

## 5. Conclusions

This paper presents a bi-level programming model and proposes an ABC heuristic for solving the BOTRDP. Three mutation neighborhood operators, three crossover neighborhood operators (crossover

between current non-dominated food sources) and their combinations with mutation operators are introduced to find a new food source near a selected food source. A Pareto ranking based fitness assignment method is adopted to evaluate the quality of each food source. The proposed ABC heuristic was applied to solve the BOTRDP for the Sioux Falls network. The test results show that some combinations of neighborhood operators including crossover operators can indeed improve the performance of the ABC heuristic, and that the proposed heuristic outperforms the NSGA even when different operators are randomly chosen and utilized in the NSGA as in our proposed heuristic. Numerical results also show that implementing turning restrictions can obviously reduce the TSTT and the CTVE individually, especially when the traffic network is very congested. However, we find that if we only consider optimizing one respect of the network performance, the other respect of the network performance can become inferior when the traffic network is not so congested. Therefore, we should consider the trade off between traffic efficiency and traffic induced environmental pollution when implementing a turning restriction strategy. We also find that the number of turning restriction designs is smaller when the network is more congested but greater network improvement can be achieved. The results also show that the effect of traffic information provision does have the impact on the number of Pareto optimal turning restriction designs. These results should have important implications on traffic management including special event traffic management.

This study proposed a modified ABC algorithm for solving the BOTRDP. In fact, there are many other meta-heuristic algorithms that can be developed to solve the BOTRDP. In the future, we will test the performance of other meta-heuristic algorithms for solving the BOTRDP.

### Acknowledgements

This work is jointly supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (HKU 716312E), a grant from the Central Policy Unit of the Government of the Hong Kong Special Administrative Region and the Research Grants Council of the Hong Kong Special Administrative Region, China (HKU7026-PPR-12), the National Basic Research Program of China (2012CB725401), and the National Natural Science Foundation of China (71001001, 71001005, 71271183). The authors are grateful to the two referees for their constructive comments.

### Appendix A: The SRAM

A simple descent direction, denoted by  $\mathbf{d} = [d_{ab}]$ , can be applied to solve the problem (1) for a given  $\mathbf{y}$ , given by:

$$d_{ab} = \sum_{rs} \sum_k q^{rs} P_k^{rs} \delta_{ab,k}^{rs} - x_{ab}. \quad (14)$$

Note that  $P_k^{rs}$  is a function of  $x_{ab}$ .

The SRAM for solving the SUE problem is outlined as follows (Long et al., 2010):

Step 1: *Initialization*. Calculate the route choice probability using free flow travel cost. Obtain the initial movement flow  $\mathbf{x}^1$  by Eqs. (1) and (2). Set  $\kappa=1$ ,  $\eta>1$ ,  $0<\gamma<1$ ,  $\beta^0=1$ , and the convergence tolerance  $\varepsilon>0$ .

Step 2: *Descent direction finding*. Compute the descent direction  $\mathbf{d}^\kappa$  by Eq. (14).

Step 3: *Step size determination*. Obtain the step size  $\lambda^\kappa=1/\beta^\kappa$ , where

$$\beta^\kappa = \begin{cases} \beta^{\kappa-1} + \eta, & \text{if } \|\mathbf{d}^\kappa\| \geq \|\mathbf{d}^{\kappa-1}\|, \\ \beta^{\kappa-1} + \gamma, & \text{otherwise.} \end{cases}$$

Step 4: *Link flow update*. Let  $\mathbf{x}^{\kappa+1} = \mathbf{x}^\kappa + \lambda^\kappa \mathbf{d}^\kappa$ .

Step 5: *Convergence test*. If  $\|\mathbf{d}^\kappa\| \leq \varepsilon$ , stop; otherwise, let  $\kappa = \kappa + 1$ , and go to Step 2.

The SRAM is similar to the MSA (which is also called the flow averaging method) except the choice of the step size. Hence, the proof of the convergence for the SRAM basically follows that for the MSA. The MSA was originally introduced by Robbins and Monro (1951) for solving the fixed point problem  $\mathbf{x} = \mathbf{F}(\mathbf{x})$ . The MSA adopts the following to solve the fixed point problem:

$$\mathbf{x}^{\kappa+1} = \mathbf{x}^\kappa + \lambda^\kappa (\mathbf{F}(\mathbf{x}^\kappa) - \mathbf{x}^\kappa), \quad (15)$$

where  $\lambda^\kappa$  is the step size in the  $\kappa$ th iteration.

Robbins and Monro (1951) and Blum (1954) state that one of the convergence requirements for the procedure (15) is that the sequence of step sizes  $\{\lambda^\kappa\}$  should satisfy  $\sum_\kappa \lambda^\kappa = \infty$  and  $\sum_\kappa (\lambda^\kappa)^2 < \infty$ . The sequence of step sizes  $\{\lambda^\kappa\}$  generated by the SRAM satisfies  $\sum_\kappa \lambda^\kappa = \infty$  and  $\lim_{\kappa \rightarrow \infty} \lambda^\kappa = 0$  in which the latter is a more general condition than  $\sum_\kappa (\lambda^\kappa)^2 < \infty$  (Liu et al., 2009). Hence, the SRAM also satisfies the convergence requirement. The other convergence requirements for the SRAM are the same as those for the MSA because the two methods differs only in terms of step size determination. For the proof of convergence of the MSA, the readers can refer to Daganzo (1983) or Theorem 3 in Cantarella (1997). Note that one convergence condition required for the MSA is that the link cost vector functions have a symmetric continuous Jacobian with respect to the link flow vector (Daganzo, 1983; Cantarella, 1997). This condition is not satisfied in our problem. Yet, in our computation tests, the SRAM still converged to SUE flow patterns. Nevertheless, for general applications, we provide another version of the SRAM in Appendix B that guarantees convergence as if the cost averaging algorithm proposed by Cantarella (1997) and that avoids the slow convergence of the flow or cost averaging method.

## Appendix B: The cost averaging version of the SRAM

We define an equivalent link flow  $\hat{x}_a$  for each link  $a$  as follows:

$$\hat{x}_a = x_{a,TH} + \phi_{RT} x_{a,RT} + \phi_{LT} x_{a,LT} + \phi_{opp} x_{a,opp}. \quad (16)$$

Then, by combining Eqs. (1), (2), (4), (5), and (16), we can express the equivalent link flow vector  $\hat{\mathbf{x}} = [\hat{x}_a]$  as a function of the link travel time vector  $\mathbf{t} = [t_a]$  for a given  $\mathbf{y}$ , expressed as follows:

$$\hat{\mathbf{x}} = \mathbf{X}(\mathbf{t}), \quad (17)$$

where  $\mathbf{X}$  is the composite map. Moreover, the link travel time can be equivalently reformulated as follows:

$$t_a = t_a^0 \cdot \left( 1 + \chi_1 \left( \frac{\hat{x}_a}{Q_a} \right)^{\chi_2} \right). \quad (18)$$

According to the BPR function (18), the link travel time vector  $\mathbf{t}$  is also a function of the equivalent link flow vector  $\hat{\mathbf{x}}$ , defined as follows:

$$\mathbf{t} = \mathbf{C}(\hat{\mathbf{x}}), \quad (19)$$

where  $\mathbf{C}$  is the cost-flow map.

Substituting Eq. (17) into Eq. (19), we can obtain a fixed-point problem in terms of link times for the SUE problem addressed in this paper:

$$\mathbf{t} = \mathbf{C}(\mathbf{X}(\mathbf{t})). \quad (20)$$

A simple descent direction denoted by  $\mathbf{h} = [h_a]$  can then be applied to solve the problem (20) for a given  $\mathbf{y}$ , given by:

$$\mathbf{h} = \mathbf{C}(\mathbf{X}(\mathbf{t})) - \mathbf{t}. \quad (21)$$

An extension of the SRAM in terms of link times for solving the SUE problem is outlined as follows:

Step 1: *Initialization*. Set the initial link travel time  $\mathbf{t}^1$  equal to the travel time derived from a feasible flow pattern. Set  $\kappa = 1$ ,  $\eta > 1$ ,  $0 < \gamma < 1$ ,  $\beta^0 = 1$ , and the convergence tolerance  $\varepsilon > 0$ .

Step 2: *Descent direction finding*. Compute the link flow  $\hat{\mathbf{x}}^\kappa = \mathbf{X}(\mathbf{t}^\kappa)$  by Eqs. (1), (2), (4), (5), and (16).

Compute the link travel time  $\tilde{\mathbf{t}}^\kappa = \mathbf{C}(\hat{\mathbf{x}}^\kappa)$  by the BPR function (3), and obtain the descent direction  $\mathbf{h}^\kappa = \tilde{\mathbf{t}}^\kappa - \mathbf{t}^\kappa$ .

Step 3: *Step size determination*. Obtain the step size  $\lambda^\kappa = 1 / \beta^\kappa$ , where

$$\beta^\kappa = \begin{cases} \beta^{\kappa-1} + \eta, & \text{if } \|\mathbf{h}^\kappa\| \geq \|\mathbf{h}^{\kappa-1}\|, \\ \beta^{\kappa-1} + \gamma, & \text{otherwise.} \end{cases}$$

Step 4: *Link travel time update*. Let  $\mathbf{t}^{\kappa+1} = \mathbf{t}^\kappa + \lambda^\kappa \mathbf{h}^\kappa$ .

Step 5: *Convergence test*. If  $\|\mathbf{h}^\kappa\| \leq \varepsilon$ , stop; otherwise, let  $\kappa = \kappa + 1$ , and go to Step 2.

The proof of the convergence for the cost-averaging version of the SRAM is basically the same as for the cost averaging algorithm. The only difference between the two methods is the choice of step size and hence the only difference between the two proofs is on checking the convergence of the step size sequence. The choice of the step size in the cost-averaging version of the SRAM is basically the same as that for the original version of the SRAM, and therefore the sequence of step sizes  $\{\lambda^\kappa\}$  generated by the cost-averaging version also satisfies  $\sum_{\kappa} \lambda^\kappa = \infty$  and  $\lim_{\kappa \rightarrow \infty} \lambda^\kappa = 0$ . For the proof of the convergence of the

cost-averaging algorithm proposed by Cantarella (1997), the readers can refer to Theorem 4 in his paper. This theorem states that if 1) the existence and uniqueness conditions mentioned in theorems 1 and 2 in his paper hold, and 2) all the choice maps are continuous with continuous first derivatives additive probabilistic, then the cost averaging algorithm converges to an equilibrium link cost vector. These two conditions are satisfied by our proposed lower level problem.

## References

- Akamatsu, T., 1996. Cyclic flows, Markov process and stochastic traffic assignment. *Transportation Research B* 30 (5), 369-386.
- Akbari, R., Hedayatzadeh, R., Ziarati, K., Hassanzadeh, B., 2012. A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation* 2, 39-52.
- Armentano, V.A., Claudio, J.E., 2004. An application of a multi-objective tabu search algorithm to a bicriteria flowshop problem. *Journal of Heuristics* 10 (5), 463-481.
- Bach, W., 1972. *Atmospheric pollution*. USA: McGraw-Hill Book Company.
- Bekhor, S., Ben-Akiva, M.E., Ramming, M.S., 2001. Route choice: choice set generation and probabilistic choice models. *Proceedings of the 4th TRISTAN Conference, Azores, Portugal*.
- Bekhor, S., Toledo, B., 2005. Investigating path-based solution algorithms to the stochastic user equilibrium problem. *Transportation Research Part B* 39 (3), 279-295.
- Blum, J.R., 1954. Multidimensional stochastic approximation methods. *Annals of Mathematical Statistics* 25 (4), 737-744.
- Cantarella, G.E., 1997. A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand. *Transportation Science* 31 (2), 107-128.
- Cantarella, G.E., Vitetta, A., 2006. The multi-criteria road network design problem in an urban area. *Transportation* 33, 567-588.
- Cantarella, G.E., Pavone, G., Vitetta, A., 2006. Heuristics for urban road network design: Lane layout and signal settings. *European Journal of Operational Research* 175, 1682-1695.
- Chan, C.K., Yao, X.H., 2008. Air pollution in mega cities in China. *Atmospheric Environment* 42 (1), 1-42.
- Chen, A., Kim, J., Lee, S., Kim, Y., 2010. Stochastic multi-objective models for network design problem. *Expert Systems with Applications* 37 (2), 1608-1619.
- Chen, A., Lee, D., Jayakrishnan, R., 2002. Computational study of state-of-the-art path-based traffic assignment algorithms. *Mathematics and Computers in Simulation* 59 (6), 509-518.
- Chen, A., Subprasom, K., Ji, Z., 2006. A simulation-based multi-objective genetic algorithm (SMOGA) procedure for BOT network design problem. *Optimization and Engineering* 7 (3), 225-247.
- Chen, M.Y., Alfa, A.S., 1991a. Algorithms for solving Fisk's stochastic traffic assignment model. *Transportation Research Part B* 25 (6), 405-412.
- Chen, M.Y., Alfa, A.S., 1991b. A network design algorithm using a stochastic incremental traffic assignment approach. *Transportation Science* 25 (3), 215-224.
- Chung, S.H., Weaver, R.D., Friesz, T.L., 2013. Strategic response to pollution taxes in supply chain networks:



- Dynamic, spatial, and organizational dimensions. *European Journal of Operational Research*, 231 (2), 314-327.
- Daganzo, C.F., 1983. Stochastic network equilibrium with multiple vehicle types and asymmetric, indefinite link cost Jacobians. *Transportation Science* 17 (3), 282–300.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2), 182–197.
- Deb, K., Sinha, A., 2010. An efficient and accurate solution methodology for bilevel multiobjective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary Computation Journal* 18 (3), 403-449.
- Demir, E., Bektaş, T., Laporte, G., 2014. The bi-objective pollution-routing problem. *European Journal of Operational Research* 232 (3), 464-478.
- Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research* 223 (2), 346–359.
- Dial, R.B., 1971. A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research* 5 (2), 83-111.
- Farahani, R.Z., Miandoabchi, E., Szeto, W.Y., Rashidi, H., 2013. A review of urban transportation network design problems. *European Journal of Operational Research* 229 (2), 281-302.
- Fernandez, E., Lopez, E., Lopez, F., Coello, C.A., 2011. Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: the extended NOSGA method. *Information Sciences* 181 (1), 44–56.
- Fonseca, C., Fleming, P., 1993. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of the 5th International Conference on GAs*, pp. 416-423.
- Friesz, T.L., Anandalingham, A., Mehta, N.J., Nam, K., Shah, S.J., Tobin, R.L., 1993. The multiobjective equilibrium network design problem revisited: a simulated annealing approach. *European Journal of Operational Research* 65 (1), 44-57.
- Fu, L., Hao, J., He, D., He, K., 2001. Assessment of vehicular pollution in China. *Journal of the Air and Waste Management Association* 51 (5), 658-668.
- Gen, M., Cheng, R., Lin, L., 2008. *Network models and optimization: multiobjective genetic algorithm approach*. Springer, Heidelberg.
- Gokhale, S., 2012. Impacts of traffic-flows on vehicular-exhaust emissions at traffic junctions. *Transportation Research Part D* 17 (1), 21-27.
- Goldberg, D., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison-Wesley.
- Goyal, S.K., Ghatge, S.V., Nema, P., Tamhane, S.M., 2006. Understanding urban vehicular pollution problem vis-a-vis ambient air quality -- case study of megacity (Delhi, India). *Environmental monitoring and assessment* 119 (1-3), 557-569.
- Horowitz, A.J., 1997. Intersection delay in region wide traffic assignment: implications of 1994 update of the highway capacity manual. *Transportation Research Record* 1572, 1-8.
- Huang, H.J., 1995. A combined algorithm for solving and calibrating the stochastic traffic assignment model. *Journal of Operational Research Society* 46 (8), 977-987.
- Huang, H.J., Bell, M.G.H., 1998. A study on logit assignment which excludes all cyclic flows.

- Transportation Research B 32 (6), 401-412.
- Huang, H.J., Li, Z.C., 2007. A multiclass, multicriteria logit-based traffic equilibrium assignment model under ATIS. *European Journal of Operational Research* 176 (3), 1464-1477.
- Ishibuchi, H., Murata, T., 1998. A multiobjective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics* 28 (3), 392-403.
- Jaeggi, D.M., Parks, G.T., Kipouros, T., Clarkson, P.J., 2008. The development of a multi-objective Tabu search algorithm for continuous optimisation problems. *European Journal of Operational Research* 185 (3), 1192-1212.
- Jones, D.F., Mirrazavi, S.K., Tamiz, M., 2002. Multi-objective meta-heuristics: an overview of the current state-of-the-art. *European Journal of Operational Research* 137 (1), 1-9.
- Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Karaboga, D., 2010. Artificial bee colony algorithm. *Scholarpedia* 5 (3), 6915.
- Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing* 8 (1), 687-697.
- Karaboga, N., 2009. A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute* 346 (4), 328-348.
- Konak, A., 2012. Network design problem with relays: A genetic algorithm with a path-based crossover and a set covering formulation. *European Journal of Operational Research* 218 (1), 829-837.
- Leblanc, L.J., Morlok, E.K., Pierskalla, W.P., 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research* 9 (5), 309-318.
- Liu, H., He, X., He, B., 2009. Method of successive weighted averages (MSWA) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Networks and Spatial Economics* 9 (4), 485-503.
- Lo, H.K., Szeto, W.Y., 2002. A methodology for sustainable traveller information services. *Transportation Research Part B* 36(2), 113-130.
- Lo, H.K., Szeto, W.Y. 2005. Road pricing modeling for hyper-congestion. *Transportation Research Part A*, 39(7-9), 705-722.
- Lo, H.K., Szeto, W.Y., 2009. Time-dependent transport network design under cost-recovery. *Transportation Research Part B* 43(1), 142-158.
- Long, J.C., Gao, Z.Y., Zhang, H.Z., Szeto, W.Y., 2010. A turning restriction design problem in urban road networks. *European Journal of Operational Research* 206(3), 569-578.
- Maher, M., 1998. Algorithms for logit-based stochastic user equilibrium assignment. *Transportation Research Part B* 32 (8), 539-549.
- Mayeres, I., Ochelen, S., Proost, S., 1996. The marginal external costs of urban transport. *Transportation Research Part D* 1 (2), 111-130.
- Meng, Q., Lee, D.H., Cheu, R.L., Yang, H., 2004. Logit based stochastic user equilibrium problem for entry exit toll schemes. *Journal of Transportation Engineering ASCE* 130 (6), 787-795.
- Meng, Q., Liu, Z., 2011. Trial-and-error method for congestion pricing scheme under side-constrained probit-based stochastic user equilibrium conditions. *Transportation* 38 (11), 819-843.

- Miandoabchi, E., Zanjirani Farahani, R., Szeto, W.Y., 2012a. Bi-objective bimodal urban road network design using hybrid metaheuristics. *Central European Journal of Operations Research* 20(4), 583-621.
- Miandoabchi, E., Zanjirani Farahani, R., Dullaert, W., Szeto, W.Y., 2012b. Hybrid evolutionary metaheuristics for concurrent multi-objective design of urban road and public transit networks. *Networks and Spatial Economics*, 12 (3), 441-480.
- Miandoabchi, E., Daneshzand, F., Szeto, W.Y., Farahani, R.Z., 2013. Multi-objective discrete urban road network design. *Computers & Operations Research* 40 (10), 2429-2449.
- Ng, M.W., Lo H.K., 2013. Regional air quality conformity in transportation networks with stochastic dependencies: A theoretical copula-based model. *Networks and Spatial Economics*. 10.1007/s11607-0139185-7.
- Omkar, S.N., Senthilnath, J., Khandelwal, R., Naik, G.N., Gopalakrishnan, S., 2011. Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures. *Applied Soft Computing* 11 (1), 489-499.
- Orubu, C.O., 2004. Using transportation control measures and economic instruments to reduce air pollution due to automobile emissions. *Journal of Social Science* 8 (3), 227-236.
- Pandian, S., Gokhale, S., Ghoshal, A.K., 2009. Evaluating effects of traffic and vehicle characteristics on vehicular emissions near traffic intersections. *Transportation Research Part D* 14 (3), 180-196.
- Powell, W.B., Sheffi, Y., 1982. The convergence of equilibrium algorithms with predetermined step sizes. *Transportation Science* 16 (1), 45-55.
- Robbins, H., Monro, S., 1951. A stochastic approximation method. *Annals of Mathematical Statistics* 22 (3), 400-407.
- Schaffer, J.D., 1985. Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the 1st International Conference on GAs*, pp. 93-100.
- Sharma, S., Ukkusuri, S.V., Mathew, T.V., 2009. Pareto optimal multiobjective optimization for robust transportation network design problem. *Transportation Research Record* 2090, 95-104.
- Sheffi, Y., 1985. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.
- Sheffi, Y., Powell, W.B., 1982. An algorithm for the equilibrium assignment problem with random link times. *Networks* 12 (2), 191-207.
- Shepherd, S.P., Sumalee A., 2004. A genetic algorithm based approach to optimal toll level and location problem. *Networks and Spatial Economics* 4, 161-179.
- Singh, A., 2009. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing* 9 (2), 625-631.
- Sohn, K., 2011. Multi-objective optimization of a road diet network design. *Transportation Research Part A* 45(6), 499-511.
- Srinivas, N., Deb, K., 1994. Multiobjective optimization using nondominated sorting genetic algorithms. *Evolutionary Computation* 2 (3), 221-248.
- Suppaitnarm, A., Seffen, K.A., Parks, G.T., Clarkson, P.J., 2000. A simulated annealing algorithm for multiobjective optimization. *Engineering Optimization* 33 (1), 59-85.
- Szeto, W.Y., Lo, H.K., 2008. Time-dependent transport network improvement and tolling strategies. *Transportation Research Part A* 42(2), 376-391.

- Szeto, W.Y., Jiang, Y., Sumalee, A., 2011a. A cell-based model for multi-class doubly stochastic dynamic traffic assignment. *Computer-Aided Civil and Infrastructure Engineering* 26 (8), 595–611.
- Szeto, W.Y., Wu, Y., Ho, S.C., 2011b. An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research* 215 (1), 126-135.
- Szeto, W.Y., Jaber, X.Q., Wong, S.C. 2012. Road network equilibrium approaches to environmental sustainability. *Transport Reviews* 32, 491-518.
- Szeto, W.Y., Jiang, Y., 2012. Hybrid artificial bee colony algorithm for transit network design. *Transportation Research Record* 2284, 47-56.
- Szeto, W.Y., Jiang, Y., Wang, D.Z.W., Sumalee, A., 2013a. A sustainable road network design problem with land use transportation interaction over time. *Networks and Spatial Economics*. DOI: 10.1007/s11607-013-9191-9.
- Szeto, W.Y., Wang, Y., Wong, S.C., 2013b. The chemical reaction optimization approach to solving the environmentally sustainable network design problem. *Computer-Aided Civil and Infrastructure Engineering*, DOI: 10.1111/mice.12033.
- Szeto, W.Y., Jiang, Y., Zanjirani Farahani, R., 2013c. Bilevel transit network design: hybrid artificial bee colony algorithm approach. *Transportation Research Part B*, submitted.
- Teklu F., Sumalee A., Watling D.P., 2007. A genetic algorithm approach for optimising traffic control signals considering routing. *Computer-Aided Civil and Infrastructure Engineering* 22(1), 31-43.
- Uno, T., Katagiri, H., 2008. Single- and multi-objective defensive location problems on a network. *European Journal of Operational Research* 188 (1), 76-84.
- Vilcot, G., Billaut, J.C., 2008. A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. *European Journal of Operational Research* 190 (2), 398–411.
- Whittaker, G., Confesorjr, R., Griffith, S., Fare, R., Grosskopf, S., Steiner, J., Muellerrarrant, G., Banowetz, G.M., 2009. A hybrid genetic algorithm for multiobjective problems with activity analysis-based local search. *European Journal of Operational Research* 193 (1), 195–203.
- Xu, T., Wei, H., Wang, Z.D., 2009. Study on continuous network design problem using simulated annealing and genetic algorithm. *Expert Systems with Applications* 36 (2), 2735-2741.
- Yin, Y., 2000. Genetic-algorithm-based approach for bilevel programming models. *Journal of Transportation Engineering* 126 (2), 115-120.
- Yin, Y., 2002. Multiobjective bilevel optimization for transportation planning and management problems. *Journal of Advanced Transportation* 36 (1), 93-105.
- Zhang, H., Zhu, Y., Zou, W., Yan, X., 2012. A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production. *Applied Mathematical Modelling* 36 (6), 2578–2591.
- Zheng, Y., Wan, Z., Wang, G., 2011. A fuzzy interactive method for a class of bilevel multiobjective programming problem. *Expert Systems with Applications* 38 (8), 10384-10388.
- Zhu, F., Lo, H.K., Lin, H.Z. 2013. Delay and emissions modelling for signalised intersections. *Transportmetrica B: Transport Dynamics* 1 (2), 111-135.
- Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3 (4), 257-271.

**Table 1**

Turning restrictions

$\mu$	$u_\mu$	$d_\mu$	$\mu$	$u_\mu$	$d_\mu$	$\mu$	$u_\mu$	$d_\mu$
1	2	6	19	36	31	37	50	54
2	8	7	20	10	32	38	60	55
3	11	10	21	27	34	39	45	58
4	31	8	22	40	33	40	61	57
5	15	13	23	33	37	41	64	63
6	23	11	24	7	36	42	68	61
7	12	14	25	34	41	43	59	60
8	19	15	26	44	42	44	56	62
9	24	19	27	28	45	45	75	65
10	16	20	28	67	44	46	69	64
11	17	22	29	41	43	47	72	67
12	47	21	30	57	46	48	46	68
13	21	25	31	29	47	49	63	69
14	13	24	32	55	49	50	65	70
15	32	26	33	22	50	51	42	72
16	25	29	34	52	48	52	70	73
17	48	28	35	30	52	53	39	76
18	43	27	36	58	51	54	73	75

**Table 2**

Parameters of emission rate functions and monetary valuation of each type of pollutants (Mayeres et al., 1996).

Type of pollutant $m$	NO <sub>x</sub>	VOC	CO
$A_m$	1.5718	2.7843	3.3963
$B_m$	0.040732	0.015062	0.014561
$C_m$	10000	10000	1000
$\eta_m$ (in EURO/kg)	13.80	2.95	0.01

**Table 3**Experimental results by different operators ( $\theta = 1.0$ ).

Operator	Mutation			crossover			Combined <sup>e</sup>
	Point	Random	Section	Point	Random	Section	
Minimum <sup>a</sup>	38	40	0	20	20	0	49
Maximum <sup>b</sup>	54	54	0	47	48	0	58
Average <sup>c</sup>	47.5	48	0.00	32.25	34.75	0.00	52.5
Standard deviation <sup>d</sup>	4.44	3.91	0.00	8.43	7.40	0.00	2.31

<sup>a</sup> Minimum number of Pareto solutions obtained in 20 runs.<sup>b</sup> Maximum number of Pareto solutions obtained in 20 runs.<sup>c</sup> Average number of Pareto solutions of 20 runs.<sup>d</sup> Standard deviation of the numbers of Pareto solutions of 20 runs.<sup>e</sup> Except the section mutation operator and the section crossover operator, one of the rest four neighborhood operators was randomly applied to find a new food source near a selected food source at a time.**Table 4**

Comparison of the proposed ABC algorithm with genetic algorithms ( $\theta = 1.0$ ).

Algorithm	NSGA	MNSGA	Proposed ABC algorithm
Minimum	17	17	42
Maximum	52	61	60
Average	35.00	39.35	55.50
Standard deviation	10.76	11.36	5.06

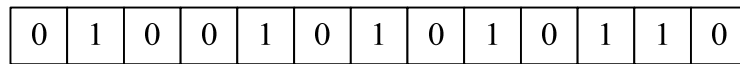


Fig. 1. Solution representation.

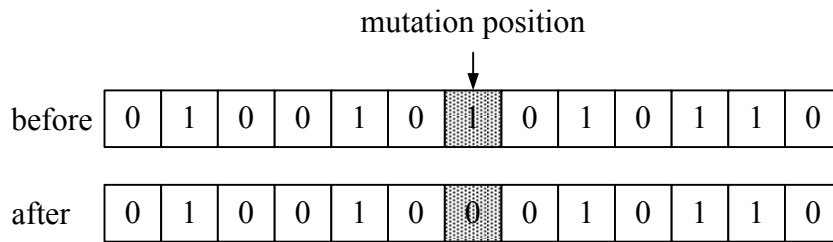


Fig. 2. Point mutation.

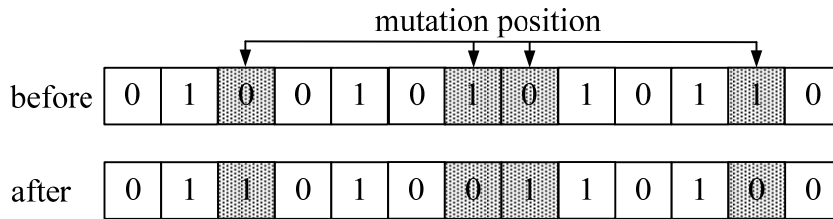


Fig. 3. Random mutation.

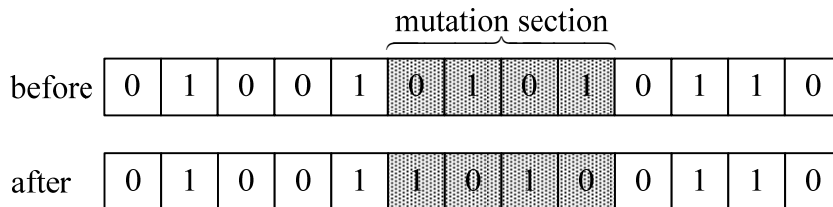


Fig. 4. Section mutation.

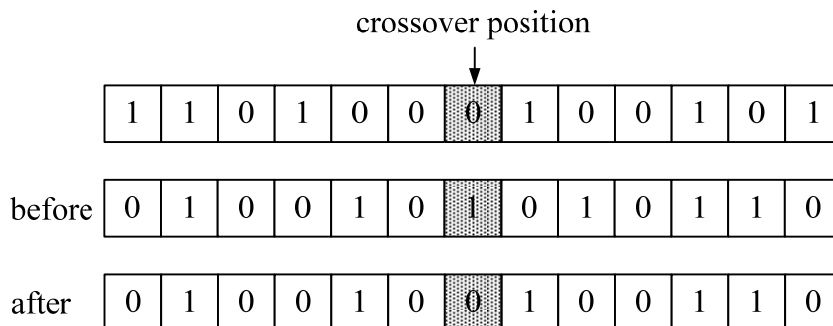


Fig. 5. Point crossover.

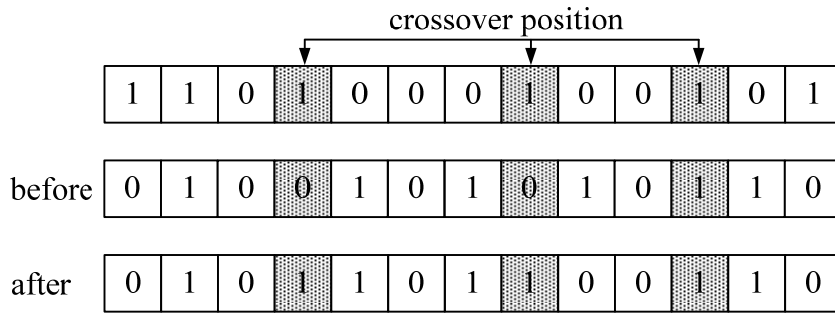


Fig. 6. Random crossover.

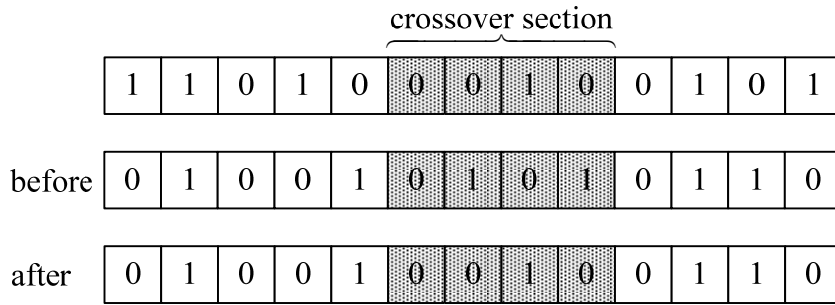


Fig. 7. Section crossover.

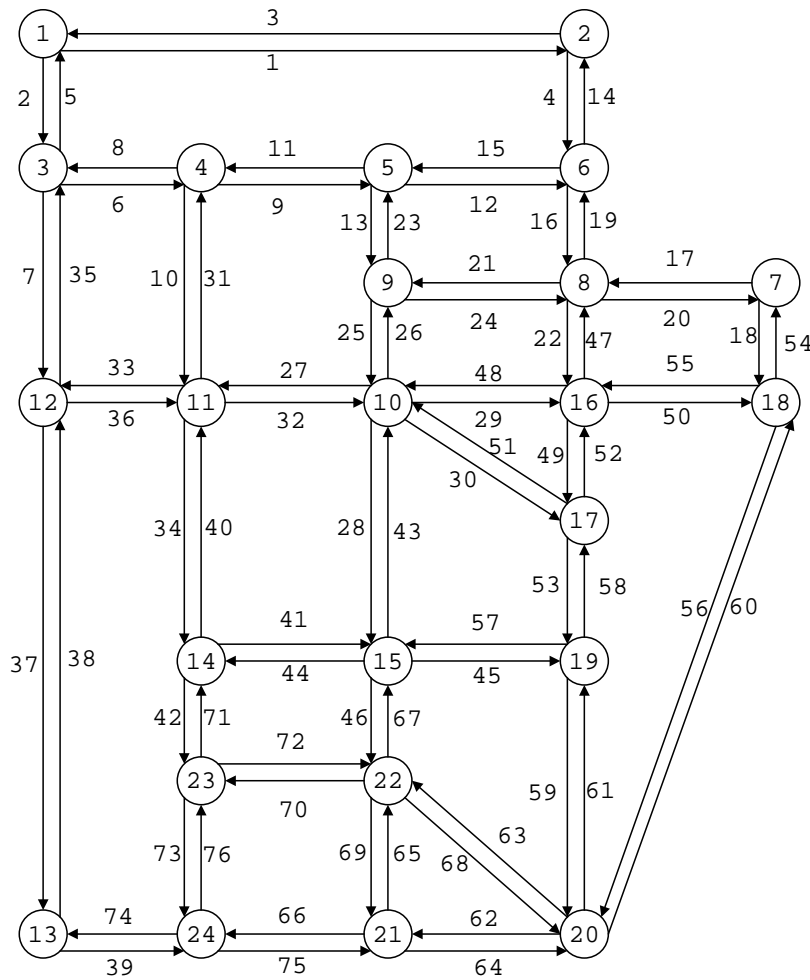


Fig. 8. The Sioux Falls network.

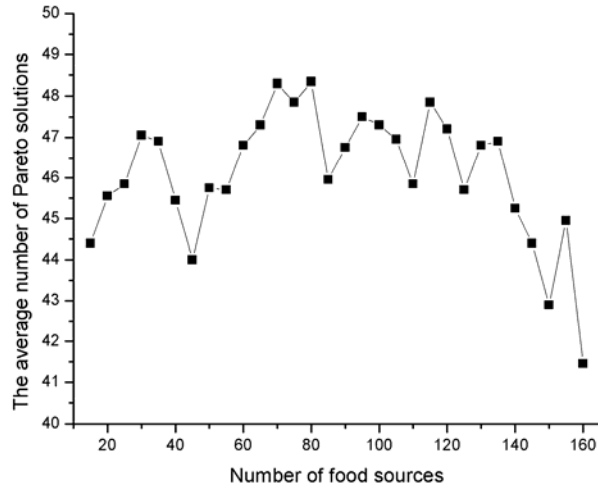


Fig. 9. The influence of the initial number of food sources being searched on the performance of the ABC heuristic.

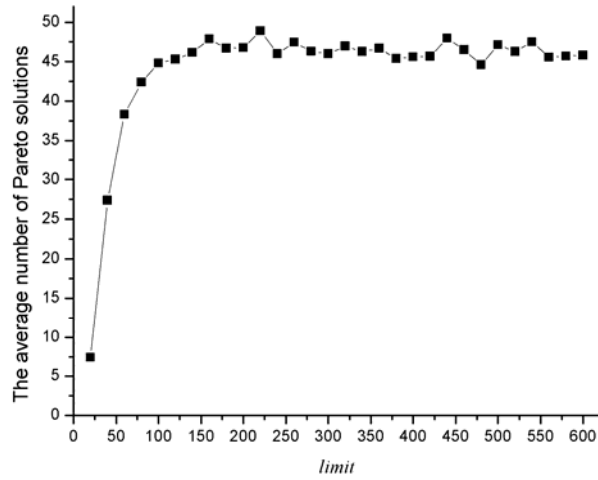


Fig. 10. The influence of *limit* on the performance of the ABC heuristic.

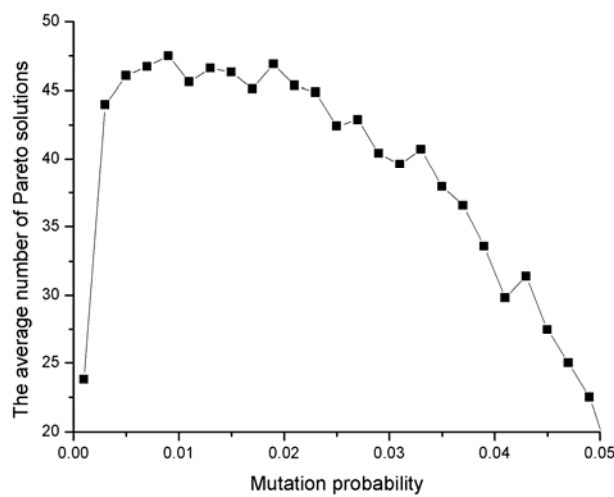


Fig. 11. The influence of the mutation probability on the performance of the ABC heuristic with only using the random mutation neighborhood operator.



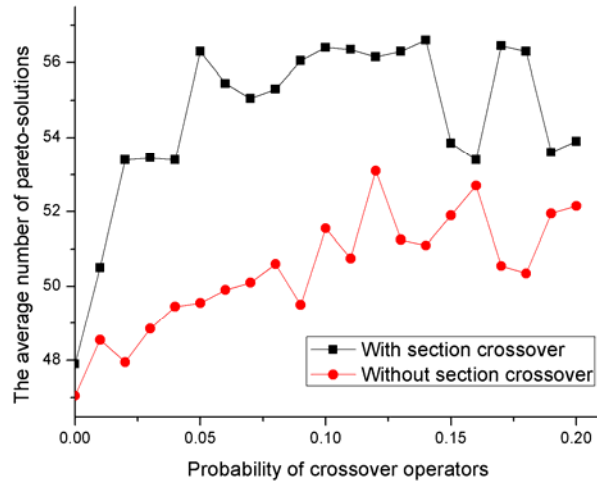


Fig. 12. The influence of the probability of crossover operators on the performance of the ABC heuristic.

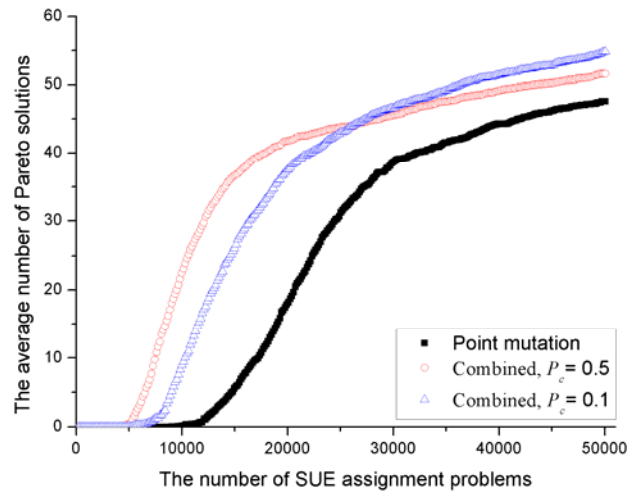


Fig. 13. The convergence processes of the ABC heuristics.

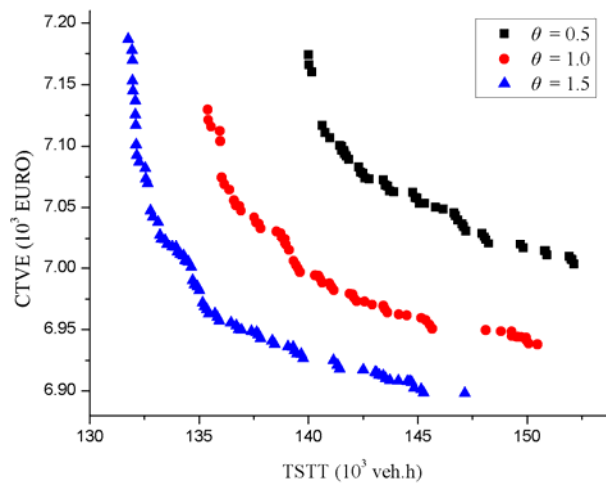


Fig. 14. The Pareto frontier of the BOTRDP.

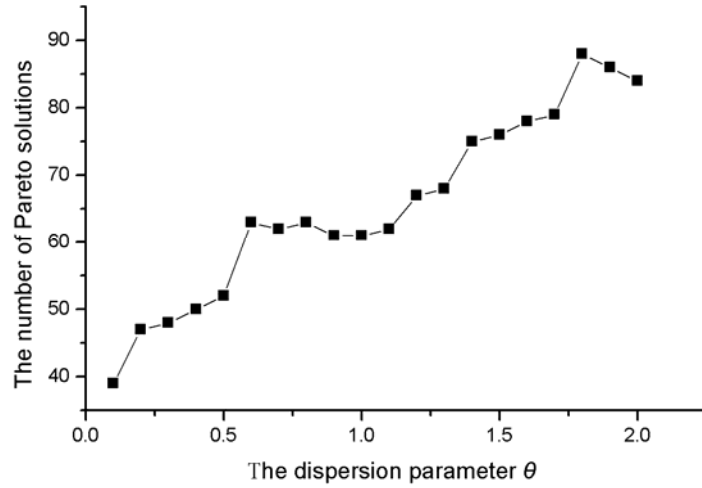


Fig. 15. The number of Pareto solutions against the dispersion parameter  $\theta$ .

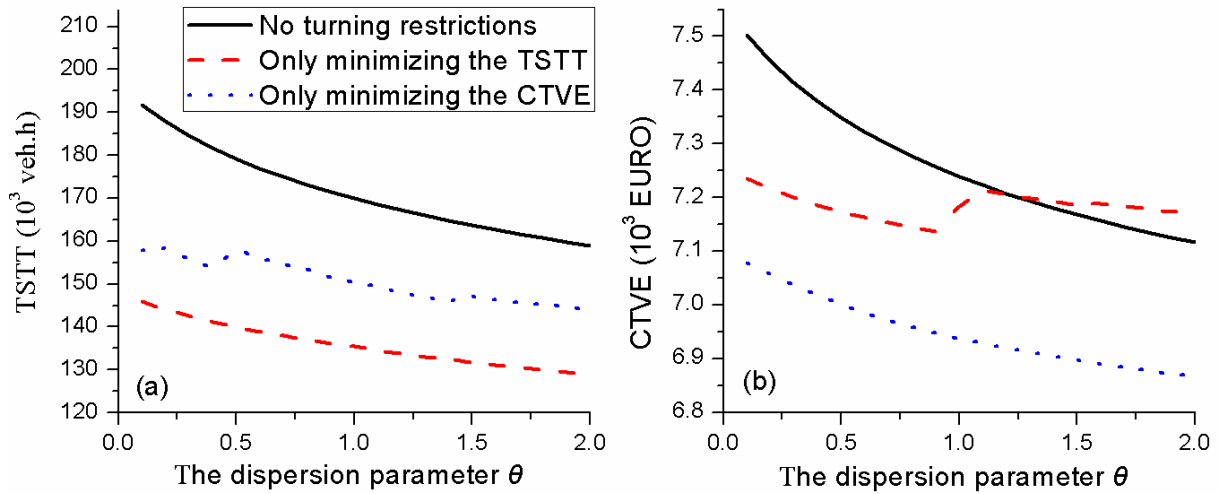


Fig. 16. The influence of the value of the dispersion parameter  $\theta$  on network performance with the consideration of single objective.

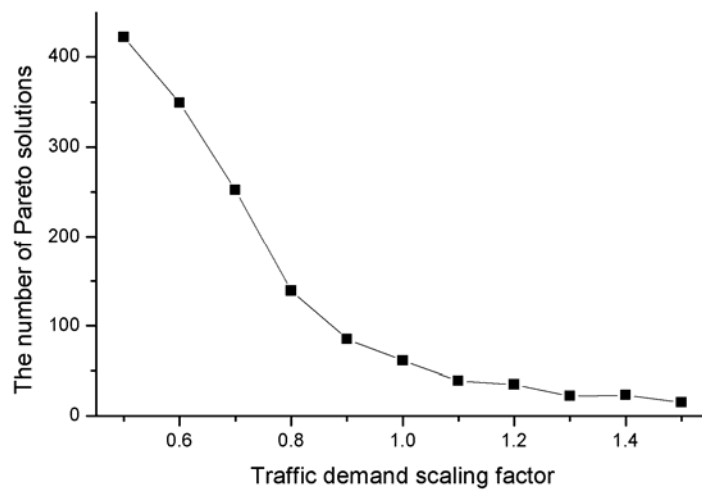


Fig. 17. The number of Pareto solutions against the traffic demand matrix scaling factor.

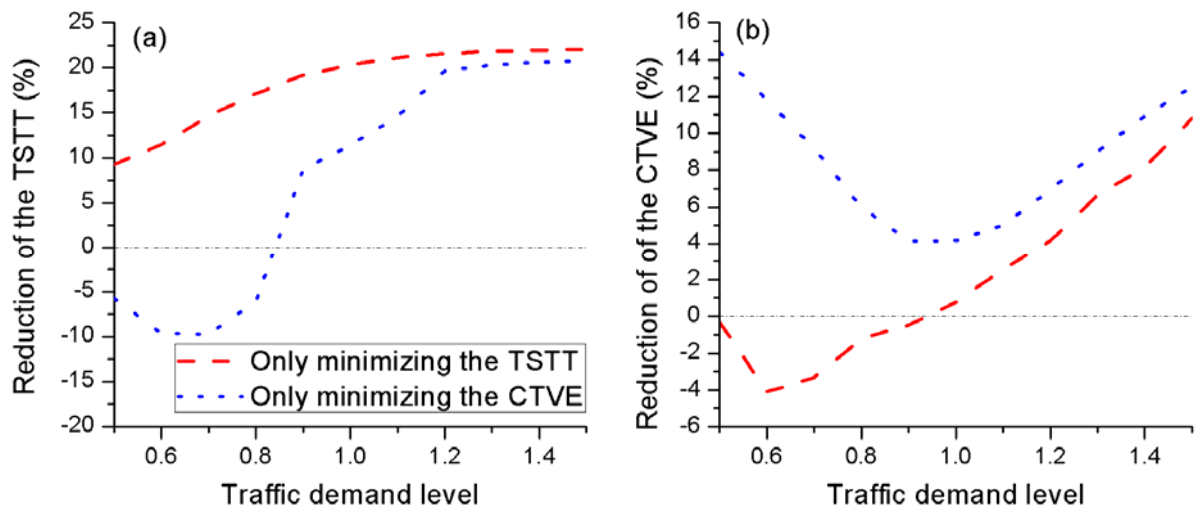


Fig. 18. The influence of the value of the dispersion parameter  $\theta$  on network performance improvement with the consideration of single objective.