



Title	A turning restriction design problem in urban road networks
Author(s)	Long, J; Gao, Z; Zhang, H; Szeto, WY
Citation	European Journal Of Operational Research, 2010, v. 206 n. 3, p. 569-578
Issued Date	2010
URL	http://hdl.handle.net/10722/91223
Rights	Creative Commons: Attribution 3.0 Hong Kong License

A Turning Restriction Design Problem in Urban Road Networks

by

Jiancheng Long

Postdoctoral Research Fellow
School of Economics and Management
Beijing University of Aeronautics and Astronautics
Beijing 100083, China
Tel: 86-10-8231-6608
E-mail: 04114182@bjtu.edu.cn

Ziyu Gao

Professor
State Key Laboratory of Rail Traffic Control and Safety
Beijing Jiaotong University
Beijing 100044, China
Tel: 86-10-5168-8193
Fax: 86-10-5168-7127
E-mail: gaoziyou@jtys.bjtu.edu.cn

Haozhi Zhang

Senior Engineer
China Urban Sustainable Transport Research Centre
China Academy of Transport Sciences,
Beijing, 100029, China
Tel: +86-10-5827-8510
Email: haozhizhang7@126.com

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

Abstract

Turning restriction is one of the commonest traffic management techniques and an effective low cost traffic improvement strategy in urban road networks. However, the literature has not paid much attention to the turning restriction design problem (TRDP), which aims to determine a set of intersections where turning restrictions should be implemented. In this paper, a bi-level programming model is proposed to formulate the TRDP. The upper level problem is to minimize the total travel cost from the viewpoint of traffic managers, and the lower level problem is to depict travelers' route choice behavior based on stochastic user equilibrium (SUE) theory. We propose a branch and bound method (BBM), based on the sensitivity analysis algorithm (SAA), to find the optimal turning restriction strategy. A branch strategy and a bound strategy are applied to accelerate the solution process of the TRDP. The computational experiments give promising results, showing that the optimal turning restriction strategy can obviously reduce system congestion and are robust to the variations of both the dispersion parameter of the SUE problem and the level of demand.

Keywords: Turning restriction design problem; bi-level programming; stochastic user equilibrium; sensitivity analysis; branch and bound method

1. Introduction

Urban traffic control is one of the hottest topics in the domain of traffic studies. The familiar control strategies include signal control, lane allocation, turning restriction, road pricing, and so on. In general, the approaches to traffic control can be classified into two categories: local control and system control. The local control approach focuses on a part of networks (e.g., a single road or a single intersection), and improves traffic mobility and other performance measures by optimizing the use of local traffic control resources. The crucial disadvantage of this kind of control approaches is that the system performance can be deteriorated, although the local performance is improved. The control methods studied by Troutbeck and Kako (1999), Harwood et al. (1999), and Al-Madani (2003) can be classified into this category. On the contrary, the system control approach emphasizes on improving the network performance from the perspective of global optimization (e.g., to minimize the total system travel cost). Sometimes the improvement of the network performance may sacrifice the performance of local traffic systems. The methods studied by Gao and Song (2002), Cipriani and Fusco (2004), and Ying et al. (2007) can be classified into this category.

This paper studies turning restriction using a system control approach. Turning restrictions are applied to a group of intersections where travelers are prohibited to drive into restricted downstream links. Herein, we propose the turning restriction design problem (TRDP), which is the problem of determining a set of intersections to implement turning restrictions to maximize the performance of the overall traffic system. At the microscopic level, implementing turning restrictions at intersections can potentially relieve vehicle interaction (e.g., between turning and ahead vehicles), reduce traffic incidents (e.g., by avoiding side collisions between turning and ahead vehicles), enhance traffic capacity, and alleviate travel delay by improving journey velocity (Chen and Luo, 2006). More importantly, by considering the interactions of multi-intersections, implementing turning restrictions can improve traffic efficiency not only locally but also at the network level. Compared with some other system control methods, such as road capacity enhancement, congestion pricing, etc., turning restriction is low cost and can be implemented quickly.

The objective of the TRDP is to optimize a given system performance measure such as to minimize total system travel cost, while accounting for the route choice behavior of network users. A new turning restriction in networks influences some travelers' route choice, because their original routes are infeasible after implementing the turning restriction. Consequently, the original traffic equilibrium state is unstable and some travelers switch their routes to reduce their travel costs. A

new equilibrium is realized again only when all travelers cannot change their routes to reduce their travel costs. Therefore, taking travelers' route choice behavior into account is one of the most important considerations to solve the TRDP.

The turning restriction decision affects the route choice behavior of network users, which is normally described through a network user equilibrium model. Mathematically, the TRDP can be formulated as a bi-level problem. The upper level problem (ULP) is to minimize the total system cost. The lower level problem (LLP) is the stochastic user equilibrium (SUE) assignment problem, which is more realistic than the deterministic user equilibrium (DUE) assignment problem (Chen and Alfa, 1991b). The resultant bi-level problem is formulated as a mathematical program with equilibrium constraints.

Since the turning restriction decision variables of the TRDP can be denoted by a group of 0-1 variables, the proposed bi-level problem is a nonlinear mixed integer programming problem. This proposed problem has a similar form to the discrete network design problem (DNDP) which deals with the selection of link additions to an existing road network. Because of the computational difficulty for solving the nonlinear bi-level mixed integer programming problem with a large number of 0-1 variables, the TRDP has been recognized as one of the most difficult yet challenging problems in the transportation research area. Reviewing the solution algorithms for the nonlinear bi-level mixed integer programming problem, the branch and bound methods and heuristic algorithms are the two primary approaches. The following paragraph gives some representative examples for the two approaches.

Leblanc (1975) presented a branch and bound algorithm to solve the DNDP, but the bounding step was based on the assumption that additional link improvements would always reduce total user cost. Edmunds and Bard (1992) provided an effective branch and bound algorithm for the mixed-integer nonlinear bi-level programming model. The algorithm requires a convex quadratic objective function in the LLP. Poorzahedy and Turnquist (1982) developed a bi-level programming formulation to describe the DNDP and applied a heuristic algorithm based on the branch and bound algorithm. Drezner and Wesolowsky (2003), Poorzahedy and Rouhani (2007), Zhang and Gao (2007), and Gallo et al. (2010) constructed some meta-heuristic algorithms to solve DNDPs or other network design problems.

In this paper, a bi-level programming model for the TRDP is introduced. A path-based algorithm is applied to solve the lower level SUE problem, by which we can obtain the path-based information (e.g., turning flow). Two sensitivity analysis algorithms are employed to solve the

relaxed TRDP, and then the branch and bound strategies are developed to solve the TRDP.

This paper is organized as follows: In the next section, a bi-level programming model for the TRDP is proposed. The path choice set generation method and the solution algorithm for the lower problem is discussed in Section 3. The branch and bound method based on sensitivity analysis algorithms is presented in Section 4 to solve the proposed problem. In Section 5, a numerical example is given. Finally, conclusions are drawn in Section 6.

2. A bi-level programming model for the TRDP

2.1 Notations

In a multi-destination and strongly 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, $s \in S$;
Π	set of flow directions of the whole network;
Π_a	set of flow directions related to link a , and $\Pi_a \subset \Pi$;
x_{ab}	flow on link a going to link b ; $[a,b] \in \Pi_a$;
\mathbf{x}	vector for link flow, $\mathbf{x} = [x_{ab}]$;
K^{rs}	set of paths between r and s ; $ K^{rs} $ is the number of paths;
f_k^{rs}	flow on path k connecting r and s ;
δ_{ak}^{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 on 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 ;
Γ	set of turning restrictions, and $\Gamma \subset \Pi$;
$\mu = [\mu_i, \mu_h]$	turning restriction that prohibits vehicles on link μ_i flowing into downstream link μ_h , where $\mu_i, \mu_h \in A$; $\mu \in \Gamma$;
y_μ	0-1 decision variable; if the turning restriction μ is applied, then $y_\mu = 1$; otherwise $y_\mu = 0$;
\mathbf{y}	vector of turning restriction decision variables, or turning restriction strategy $\mathbf{y} = [y_\mu]$;
$\phi_{\mu k}^{rs}$	0-1 indicator variable; if both links μ_i and μ_h are on path k connecting r and s , then $\phi_{\mu k}^{rs} = 1$; otherwise, $\phi_{\mu k}^{rs} = 0$.

2.2 The lower level stochastic user equilibrium assignment problem

It is worth emphasizing that the TRDP must take into account the travelers' route choice behavior, which is similar to the network design problem (NDP). In general, the implementation of turning restrictions definitely induces changes in traffic flow over the network. Improper turning restrictions may aggravate the network congestion and bring inconvenience to travelers by extending their travel distances. Therefore, an accurate prediction of traffic patterns via a behaviorally sound model is essential to the turning restriction design process.

The DUE conditions are adopted to represent the route choice behavior in early NDP formulations. It assumes that all travelers have perfect information about the network condition and they select routes with the lowest travel cost. In fact, travelers make their route choices according to the lowest perceived travel cost because travelers have imperfect information on the network condition. Therefore, the SUE problem is better than the DUE problem to describe the route choice behavior of travelers. Herein, the SUE traffic assignment model is employed to formulate the LLP of the TRDP.

Following Ying et al. (2007), the stochastic user equilibrium on the traffic network is then characterized by the following nonlinear equations:

$$F_{ab}(\mathbf{x}, \mathbf{y}) = x_{ab} - \sum_{rs} \sum_k q^{rs} \frac{\partial S^{rs}}{\partial c_k^{rs}} \delta_{ab,k}^{rs} = 0, \quad (1)$$

where S^{rs} is the satisfaction function, defined as the expected minimum perceived travel cost from r to s :

$$S^{rs} = E[\min_{k \in K^{rs}} \{c_k^{rs}\}]. \quad (2)$$

If a turning restriction is implemented, it will reduce link travel cost directly by eliminating the influence of opposing flow on turning flows. Furthermore, the route travel cost cannot be simply formulated by the sum of the link travel costs. If a traveler takes a restricted turning, he or she will be punished by a punishment cost. We can assume that the punishment cost is high enough and nobody can endure it. Thus, an additional term is used to formulate the route travel cost in Eq. (2), given by:

$$c_k^{rs} = \sum_a t_a(\mathbf{x}, \mathbf{y}) \delta_{ak}^{rs} + \sum_{\mu} y_{\mu} M \phi_{\mu k}^{rs}, \quad (3)$$

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 cost of all paths in the choice set, so that very few travelers take restricted turnings.

The above SUE formulation can be applied to a variety of route choice models which can meet certain conditions imposed on the satisfaction function. For the general logit model, the satisfaction can be formulated by:

$$S^{rs} = -\frac{1}{\theta} \ln \sum_k \exp(-\theta c_k^{rs}). \quad (4)$$

The set of used paths is a function of the vector of turning restriction decision variables \mathbf{y} , and can be expressed as $K^{rs}(\mathbf{y})$. However, it is not convenient to apply this expression to Eq. (4). Since $K^{rs}(\mathbf{y})$ is an implicit function of \mathbf{y} , and it brings difficulty in theoretically analyzing the relationship between the link flows \mathbf{x} and the turning restriction decision variables \mathbf{y} in the lower level formulation. The application of the punishment cost in Eq. (3) guarantees few travelers to choose paths with turning restrictions and provide an analytical formulation for Eq. (4).

Turning delays due to opposing flows in intersections contribute significantly to travel cost and thus route choice in urban road networks. Different from the traditional traffic assignment models, this paper considers the link travel cost function with an asymmetric Jacobian, and proposes a generalized BRP function to capture the effects of opposing and turning flows on link travel cost as follows:

$$t_a(\mathbf{x}, \mathbf{y}) = t_a^0 \cdot \left(1 + \chi_1 \left(\frac{x_{a,TH} + \phi_{RT} x_{a,RT} + \phi_{LT} x_{a,LT} + \phi_{opp} x_{a,opp} \sum_{\mu=a} (1 - y_\mu)}{s_a} \right)^{\chi_2} \right), \quad (5)$$

where t_a^0 is the free-flow travel cost 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; s_a is the capacity of link a ; ϕ_{LT} , ϕ_{RT} , and ϕ_{opp} are parameters related to the influences of left turn, right turn, and opposing flows on the through flow, respectively; χ_1 and χ_2 are the BPR parameters. The proposed BPR function is the generalized version of the link cost function in Horowitz (1997) and can capture the effect of turning flow on the link travel cost in addition to opposing flow. Note that the link travel cost consists of two components: *free flow travel cost* and *cost of delay* (including congestion delay and turning delay).

2.3 The upper level optimization problem

To solve the TRDP, the primary task for traffic managers is to make decisions about at which intersections should implement turning restrictions. However, not all intersections should be taken into account, but only some crucial intersections should be considered instead. In general, turning

restrictions are inappropriate for uncongested intersections, since turning restrictions have minor contributions towards reducing the system congestion level and may bring inconveniency to travelers who do not have the network information accurately. Moreover, right turning restrictions are not encouraged for the case of drive-to-the-right as in China, because right turns have the lowest influence on other vehicular movements.

Whether a turning movement should be restricted or not can be depicted by a 0-1 variable. Therefore, the TRDP is a discrete problem, which aims to improve the network performance and reduce network-wide congestion. The ULP of the TRDP can be formulated as the following optimization model:

$$\min \quad G(\mathbf{x}, \mathbf{y}) = \sum_{ab} x_{ab}(\mathbf{y}) t_{ab}(\mathbf{x}(\mathbf{y}), \mathbf{y}) \quad (6)$$

$$\text{subject to } y_{\mu} = \{0,1\} \quad \forall \mu \in \Gamma. \quad (7)$$

where $t_{ab}(\mathbf{x}(\mathbf{y}), \mathbf{y}) = t_a(\mathbf{x}(\mathbf{y}), \mathbf{y})$; $x_{ab}(\mathbf{y})$ means that x_{ab} is an implicit function of \mathbf{y} , which can be obtained by solving the LLP. The objective (6) is to minimize the total travel cost. Constraint (7) is the binary decision variable for a turning restriction.

3. The solution method for the lower level SUE problem

The solution algorithms of the SUE problem can be classified into two categories: link-based algorithms (e.g., Sheffi and Powell, 1982; Maher, 1998, and; Chen et al., 2002) and path-based algorithms (e.g., Chen and Alfa, 1991a; Huang, 1995, and; Bekhor and Toledo, 2005). An important advantage of link-based algorithms is that they do not require explicit enumeration of paths, and so, they can be easily applied to large-scale networks. Instead of enumerating paths, link-based algorithms assume implicit path choice sets, such as the use of all efficient paths (e.g., Maher, 1998), or all cyclic and acyclic paths (e.g., Bell, 1995). However, these implicit choice sets may be unrealistic from a behavioral viewpoint (Bekhor and Toledo, 2005). On the contrary, path-based algorithms allow a more flexible definition of the choice set which can accommodate more realistic considerations. Furthermore, a path-based algorithm can obtain path related information such as path flow. In this paper, the path flow information is necessary to determine turning restrictions, which is more important than the information about the number of turning vehicles at individual intersections. Thus, path-based algorithms are more suitable for solving the lower level SUE problem. Although the solution of the LLP is link flow, this solution can be obtained from path flow via the relationship between link flow and path flow.

3.1 Path choice set generation method

Path choice set generation methods received attention in the last decade. Huang (1995) and Chen and Alfa (1991a) used the STOCH method to generate the path set. Bekhor and Toledo (2005) used a combination of the link elimination method and the k-shortest path method. Bekhor et al. (2006) discussed choice set generation and route choice models for large-scale urban networks. Several algorithms (e.g., Dial's (1971) STOCH method and k-shortest path method) were applied to generate the path set for traffic assignment. Han (2007) defined a reasonable path set in the sense of Dial (1971).

In this paper, the paths are generated prior to the assignment, using a combination of the link elimination method (see Bekhor et al., 2001 for details) and Dial's (1971) STOCH method. In a dense network, the k-shortest path method generates routes with a high degree of similarity. The link elimination method consists of successively removing links and finding the shortest path on the remaining links of the network. Only acyclic paths are considered in this method. Since the path set generated by the STOCH method may omit some shorter paths, the STOCH method (see, Dial, 1971; Sheffi, 1985 for details) can be used to generate the basic path set, and the link elimination method can generate some shorter paths omitted by the STOCH method.

3.2 Solution algorithm for the lower level SUE problem

The partial derivative of the satisfaction function with respect to route travel cost is the route choice probability, given by:

$$\frac{\partial S^{rs}}{\partial c_k^{rs}} = P_k^{rs}. \quad (8)$$

For the logit SUE model, substituting Eq. (4) into Eq. (8), we have

$$P_k^{rs} = \frac{\partial S^{rs}}{\partial c_k^{rs}} = \frac{\exp(-\theta c_k^{rs})}{\sum_l \exp(-\theta c_l^{rs})}. \quad (9)$$

Then, the path flow and link flow are, respectively, given by:

$$f_k^{rs} = q^{rs} P_k^{rs}, \text{ and} \quad (10)$$

$$x_{ab} = \sum_{rs} \sum_k f_k^{rs} \delta_{ab,k}^{rs} = \sum_{rs} \sum_k q^{rs} P_k^{rs} \delta_{ab,k}^{rs}. \quad (11)$$

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:

$$d_{ab} = \sum_{rs} \sum_k q^{rs} P_k^{rs} \delta_{ab,k}^{rs} - x_{ab}. \quad (12)$$

To solve the lower level SUE problem (1) for a given \mathbf{y} , we use the method of successive weighted averages (MSWA) proposed by Liu et al. (2009), which is much faster than the method of successive averages (MSA). The MSWA is outlined as follows:

Step 1: *Initialization*. Calculate the route choice probability by free flow travel cost. Obtain the initial link flow \mathbf{x}^1 by Eq. (10) and Eq. (11). Set $\kappa = 1$, $\eta > 1$, $0 < \gamma < 1$, $\beta^0 = 1$, and the convergence tolerance $\varepsilon > 0$.

Step 2: *Calculating the descent direction*. Compute the descent direction \mathbf{d}^κ by Eq. (12).

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

$$\beta^\kappa = \begin{cases} \beta^{\kappa-1} + \eta, & \text{if } \|\mathbf{d}^\kappa\| \geq \|\mathbf{d}^{\kappa-1}\|, \\ \beta^{\kappa-1} + \gamma, & \text{otherwise.} \end{cases} \quad (13)$$

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

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

4. The solution method for the TRDP

The difficulty in solving the bi-level programming problem presented in this paper lies in how to evaluate the equilibrium flow pattern $\mathbf{x}(\mathbf{y})$ for a given vector of turning restriction decision variables \mathbf{y} , which is an implicit vector function defined by the lower level stochastic user equilibrium problem. In this paper, \mathbf{y} is a vector of binary decision variables in the ULP. Therefore, the branch and bound technique is a preferential approach to solve the nonlinear mixed integer bi-level programming problem. Sensitivity analysis is also an efficient method for solving network design problems with the lower level SUE problem (see, for example, Patriksson, 2004; Ying and Yang, 2005; Ying et al., 2007, and; Liu et al., 2010). Before designing the branch and bound method, we relax the integrality requirements of the ULP, and employ a sensitivity analysis-based algorithm to solve the relaxed problem. Information provided by sensitivity analysis can help in determining both the upper and lower bounds of the optimal objective value of the ULP, and designing the branch strategy.

4.1 Sensitivity analysis of stochastic user equilibrium flows

The turning restriction decision variables \mathbf{y} are considered as parametric variables in the LLP

of the TRDP. Let $\mathbf{M}(\mathbf{y}) = \begin{bmatrix} \frac{\partial F_{ab}}{\partial x_{cd}} \end{bmatrix}$, and $\mathbf{N}(\mathbf{y}) = \begin{bmatrix} \frac{\partial F_{ab}}{\partial y_\mu} \end{bmatrix}$. The two matrices can then be derived from

Eq. (1), given by:

$$\frac{\partial F_{ab}}{\partial x_{cd}} = \delta_{ab,cd} - \sum_{rs} \sum_k q^{rs} \delta_{ab,k}^{rs} \sum_l \frac{\partial P_k^{rs}}{\partial c_l^{rs}} \cdot \frac{\partial c_l^{rs}}{\partial x_{cd}}, \text{ and} \quad (14)$$

$$\frac{\partial F_{ab}}{\partial y_\mu} = - \sum_{rs} \sum_k q^{rs} \delta_{ab,k}^{rs} \sum_l \frac{\partial P_k^{rs}}{\partial c_l^{rs}} \cdot \frac{\partial c_l^{rs}}{\partial y_\mu}, \quad (15)$$

where, $\delta_{ab,cd} = 1$ if $a=c$ and $b=d$, and $\delta_{ab,cd} = 0$ otherwise.

According to the definition of route travel cost in Eq. (3), we have

$$\frac{\partial c_k^{rs}}{\partial x_{cd}} = \sum_e \delta_{e,k}^{rs} \frac{\partial t_e(\mathbf{x}, \mathbf{y})}{\partial x_{cd}}, \text{ and} \quad (16)$$

$$\frac{\partial c_k^{rs}}{\partial y_\mu} = \sum_a \delta_{a,k}^{rs} \frac{\partial t_a(\mathbf{x}, \mathbf{y})}{\partial y_\mu} + M \phi_{\mu k}^{rs}. \quad (17)$$

Deriving from Eq. (9), we have

$$\frac{\partial P_k^{rs}}{\partial c_l^{rs}} = \begin{cases} \frac{\theta \exp\{-\theta(c_k^{rs} + c_l^{rs})\}}{\{\sum_k \exp(-\theta c_k^{rs})\}^2} & \text{if } l \neq k, \\ \frac{-\theta \exp(-\theta c_k^{rs})}{\sum_k \exp(-\theta c_k^{rs})} + \theta \left\{ \frac{\exp(-\theta c_k^{rs})}{\sum_k \exp(-\theta c_k^{rs})} \right\}^2 & \text{if } l = k. \end{cases} \quad (18)$$

Eq. (18) can be simplified as follows:

$$\frac{\partial P_k^{rs}}{\partial c_l^{rs}} = \begin{cases} \theta P_k^{rs} P_l^{rs} & \text{if } l \neq k, \\ -\theta P_k^{rs} (1 - P_k^{rs}) & \text{if } l = k. \end{cases} \quad (19)$$

Substituting Eq. (19) into Eqs. (14) and (15), we have

$$\frac{\partial F_{ab}}{\partial x_{cd}} = \delta_{ab,cd} + \theta \sum_{rs} \sum_k q^{rs} P_k^{rs} \delta_{ab,k}^{rs} \left\{ \frac{\partial c_k^{rs}}{\partial x_{cd}} - \sum_l P_l^{rs} \frac{\partial c_l^{rs}}{\partial x_{cd}} \right\}, \text{ and} \quad (20)$$

$$\frac{\partial F_{ab}}{\partial y_\mu} = \theta \sum_{rs} \sum_k q^{rs} \delta_{ab,k}^{rs} P_k^{rs} \left\{ \frac{\partial c_k^{rs}}{\partial y_\mu} - \sum_l P_l^{rs} \frac{\partial c_l^{rs}}{\partial y_\mu} \right\}. \quad (21)$$

Eqs. (20) and (21) form the matrices $\mathbf{M}(\mathbf{y})$ and $\mathbf{N}(\mathbf{y})$, respectively. The partial derivative of $\mathbf{x}(\mathbf{y})$ with respect to \mathbf{y} can be calculated by (see Yang and Chen, 2009 and Yin et al., 2009):

$$\nabla_{\mathbf{y}} \mathbf{x}(\mathbf{y}) = -\mathbf{M}(\mathbf{y})^{-1} \mathbf{N}(\mathbf{y}). \quad (22)$$

4.2 Sensitivity analysis-based algorithm 1 (SAAI)

The sensitivity analysis of SUE flows is one of the fundamental works to solve the ULP of the TRDP. The Jacobian $\nabla_{\mathbf{y}} \mathbf{x}(\mathbf{y})$ can be applied to determine the linear approximation of the objective

function of the ULP.

Let \mathbf{y}' be the initial solution of the TRDP and $\mathbf{x}(\mathbf{y}')$ be the optimal solution of the LLP. The linear approximation of the flow \mathbf{x} is given by (Gao et al., 2007):

$$\mathbf{x} \approx \mathbf{x}(\mathbf{y}') + \nabla_{\mathbf{y}} \mathbf{x}(\mathbf{y}')(\mathbf{y} - \mathbf{y}'). \quad (23)$$

The integrality requirements are relaxed, and constraint (7) can be rewritten as:

$$0 \leq y_{\mu} \leq 1, \quad \forall \mu. \quad (24)$$

In order to develop the branch and bound method below, we define Γ_0 ($\Gamma_0 \subseteq \Gamma$) and Γ_1 ($\Gamma_1 \subseteq \Gamma$) as the predetermined sets of turning restrictions rejected and adopted, respectively. The feasible domain of the relaxed ULP can then be represented as follows:

$$\Omega = \{\mathbf{y} \mid y_{\mu} = 0, \forall \mu \in \Gamma_0; y_{\mu} = 1, \forall \mu \in \Gamma_1; 0 \leq y_{\mu} \leq 1, \forall \mu \in \Gamma - \Gamma_0 - \Gamma_1\}. \quad (25)$$

For the relaxed ULP, \mathbf{x} estimated by Eq. (23) will be far away from $\mathbf{x}(\mathbf{y})$ if the value of $\|\mathbf{y} - \mathbf{y}'\|$ is too large. Therefore, we employ a predetermined parameter Δ ($\Delta \in (0,1]$) to ensure that the next solution is not too far away from the initial solution \mathbf{y}' . An additional constraint is applied to the ULP, given by:

$$y'_{\mu} - \Delta \leq y_{\mu} \leq y'_{\mu} + \Delta, \quad \forall \mu \in \Gamma - \Gamma_0 - \Gamma_1. \quad (26)$$

We can then reformulate the feasible domain of the relaxed ULP as follows:

$$\Phi = \Omega \cap \{\mathbf{y} \mid y'_{\mu} - \Delta \leq y_{\mu} \leq y'_{\mu} + \Delta, \forall \mu \in \Gamma - \Gamma_0 - \Gamma_1\}. \quad (27)$$

Substituting Eq. (22) into the Eq. (6), we can obtain the following optimization problem:

$$\min_{\mathbf{y} \in \Phi} G' = \sum_{ab} (x_{ab}(\mathbf{y}') + \nabla_{\mathbf{y}} x_{ab}(\mathbf{y}')(\mathbf{y} - \mathbf{y}')) t_{ab}(\mathbf{x}(\mathbf{y}') + \nabla_{\mathbf{y}} \mathbf{x}(\mathbf{y}')(\mathbf{y} - \mathbf{y}'), \mathbf{y}'). \quad (28)$$

Equation (28) presents a nonlinear optimization problem, which can be solved by many well-known methods, such as outer approximation algorithms, the penalty function algorithms and so on. Then, according to the optimal solution of the relaxed ULP (28), we can solve the LLP again and obtain new equilibrium link flows. We can then obtain the Jacobian $\nabla_{\mathbf{y}} \mathbf{x}(\mathbf{y})$ by the sensitivity method and a new set of turning restrictions by solving the relaxed ULP (28). The procedure is repeated until an optimal solution is obtained. The solution algorithm can be stated as follows:

Step 1: *Initialization*. Determine an initial turning restriction strategy \mathbf{y}^0 and the feasible domain

Ω ; set $\kappa = 0$, and the convergence tolerance σ .

Step 2: *Solving the LLP*. Generate the feasible domain Φ by Eq. (27), and solve the LLP with the

gradient method for a given \mathbf{y}^{κ} to obtain the optimal link flow $\mathbf{x}^{*\kappa}$.

Step 3: *Derivative calculation*. Calculate $\nabla_{\mathbf{y}} \mathbf{x}(\mathbf{y}^{\kappa})$ by using the sensitivity analysis method.

Step 4: *Solving the ULP*. Solve the problem (28) to obtain a new turning restriction strategy $\mathbf{y}^{\kappa+1}$.

Step 5: *Convergence check*. If $\max_{\mu} |y_{\mu}^{\kappa+1} - y_{\mu}^{\kappa}| \leq \sigma$, then stop; otherwise let $\kappa = \kappa + 1$, and go to Step 2.

Remark 1: Because SAA1 is a heuristic algorithm, it may not guarantee convergence (Gao and Song, 2002). Moreover, the optimal solution found may be influenced by the initial solution. A general method is to run the algorithm many times with different initial solutions. If the optimal solutions obtained by the algorithm with different initial solutions are the same, we believe that the proposed algorithm converges to a global optimal solution.

4.3 Sensitivity analysis-based algorithm 2 (SAA2)

SAA1 is a heuristic algorithm and cannot guarantee solution convergence. Indeed, the sensitivity analysis of SUE flows can be used to obtain the gradient of the objective function of the relaxed ULP, and the relaxed ULP can be viewed as a nonlinear programming problem with linear constraints and solved by the Frank-Wolfe algorithm. Based on this idea, we develop an alternative sensitivity analysis-based algorithm (SAA2). Although it is difficult to check the convexity of the relaxed TRDP, the proposed algorithm can converge to a local optimal solution at least.

The gradient of the objective function of the relaxed ULP can be obtained from the sensitivity analysis, given by:

$$\left. \frac{\partial G}{\partial y_{\mu}} \right|_{\mathbf{y}=\mathbf{y}'} = \sum_{ab} \left(\frac{\partial x_{ab}(\mathbf{y}')}{\partial y_{\mu}} \cdot t_{ab}(\mathbf{x}(\mathbf{y}'), \mathbf{y}') + x_{ab}(\mathbf{y}') \cdot \frac{\partial t_{ab}(\mathbf{x}(\mathbf{y}'), \mathbf{y}')}{\partial y_{\mu}} \right). \quad (29)$$

The descent direction can be obtained by solving the following optimization problem:

$$\min_{\mathbf{y} \in \Omega} G'' = \nabla_{\mathbf{y}} G^T \cdot (\mathbf{y} - \mathbf{y}'). \quad (30)$$

After the determination of the descent direction, the optimal step size is determined by solving a one dimensional search problem. The calculation of an optimal step size is time consuming, because the SUE problem is solved again and again during step size searching.

The SAA2 for the relaxed TRDP can be stated as follows:

Step 1: *Initialization*. Determine an initial vector of turning restriction decision variables \mathbf{y}^0 and the feasible domain Ω ; set $\kappa = 0$, and the convergence tolerance σ .

Step 2: *Solving the LLP*. Solve the LLP with the MSWA for a given \mathbf{y}^{κ} to obtain the optimal link flow $\mathbf{x}^{*\kappa}$.

Step 3: *Derivative calculation*. Calculate $\nabla_{\mathbf{y}} \mathbf{x}(\mathbf{y}^{\kappa})$ by using the sensitivity analysis method.

Step 4: *Calculating the descent direction*. Solve the problem (30) to obtain the descent direction

ω^κ .

Step 5: *Searching the optimal step size.* Solve the one dimension search problem $\min_{\lambda} G((1-\lambda)\mathbf{y}^\kappa + \lambda\omega^\kappa)$, and obtain the optimal step λ^κ .

Step 6: *Updating turning restriction decision variables.* Generate a new vector of turning restriction decision variables $\mathbf{y}^{\kappa+1} = (1-\lambda^\kappa)\mathbf{y}^\kappa + \lambda^\kappa\omega^\kappa$.

Step 7: *Convergence checking.* If $\max_{\mu} |y_{\mu}^{\kappa+1} - y_{\mu}^{\kappa}| \leq \sigma$, then stop; otherwise let $\kappa = \kappa + 1$, and go to Step 2.

4.4 An approximate integer solution method

In general, the optimal solutions obtained from the sensitivity analysis-based algorithms are not integer solutions due to the integrality requirements relaxed for the ULP. However, we can obtain nearly optimal integer solutions by using an approximate integer solution method (AISM). For $\forall \mathbf{y}^* \in \{\mathbf{y} \mid 0 \leq y_{\mu} \leq 1, \forall \mu\}$, the following method can be used to get an nearly optimal integer solution for the ULP:

$$\hat{y}_{\mu}^* = \begin{cases} 0 & \text{if } y_{\mu}^* < y_{\text{crit}} \\ 1 & \text{otherwise} \end{cases} \quad \forall \mu \in \Gamma, \quad (31)$$

where $\hat{\mathbf{y}}^*$ is the nearly optimal integer solution, and y_{crit} is the critical value satisfying $y_{\text{crit}} \in (0,1)$.

We can set $\Gamma_0 = \emptyset$ and $\Gamma_1 = \emptyset$, and obtain the solution of the relaxed TRDP by the proposed sensitivity analysis-based algorithms. The AISM can then be used to get a nearly optimal integer solution for the TRDP. If SAA1 (or SAA2) is used, the approximate integer solution method combined with SAA1 (or SAA2) is named as SAA1+AISM (or SAA2+AISM) in short in this paper.

The AISM can also be applied to determine the upper bound of the optimal objective value of the ULP of the TRDP in the branch and bound method (BBM) in the next subsection, where Γ_0 and Γ_1 may not be empty.

4.5 Branch and bound method

The ULP of the TRDP can be considered as a generalized nonlinear integer problem, and can be solved by the BBM. The following notations are adopted to describe the BBM:

- LB* lower bound of the optimal upper-level objective value;
- UB* upper bound of the optimal upper-level objective value;
- UT* set of unbranched nodes in the search tree;

- BT set of branched nodes in the search tree;
- α node index in the search tree;
- IS_α input sequence of search tree node α ;
- LB_α lower bound of the optimal upper-level objective value for search tree node α ;
- UB_α upper bound of the optimal upper-level objective value for search tree node α ;
- $\bar{\mu}_\alpha$ turning restriction for search tree node α ;
- \bar{y}_α 0-1 variable; if the turning restriction $\bar{\mu}_\alpha$ is applied, then $y_\mu = 1$; otherwise $y_\mu = 0$;
- \mathbf{y}_α^* optimal solution of the relaxed TRDP for search tree node α , and;
- $\hat{\mathbf{y}}_\alpha^*$ nearly optimal integer solution of the TRDP for search tree node α .

A bound strategy and a branch strategy are developed to improve the performance of the BBM. In the bound strategy, the nearly optimal integer solution is applied to determine a better upper bound, which is useful to reduce the number of search tree nodes. In the branch strategy, we choose the turning restriction with the minimum non-integer value for the corresponding turning restriction decision variable as the constrained turning restriction at the branch node. This strategy is useful to obtain the optimal integer solution of the ULP quickly. The following discusses the bound and branch strategies.

(i) *Bound strategy*

In the proposed BBM, the input sequence is used to add constraints to the relaxed ULP. For a given search tree node $\alpha \in UT$, the turning restriction sets Γ_0 and Γ_1 can be obtained from the input sequence IS_α , given by:

$$\Gamma_0 = \{\bar{\mu}_\beta \mid \bar{y}_\beta = 0, \forall \beta \in IS_\alpha\}, \text{ and} \quad (32)$$

$$\Gamma_1 = \{\bar{\mu}_\beta \mid \bar{y}_\beta = 1, \forall \beta \in IS_\alpha\}. \quad (33)$$

Substituting Eq. (32) and Eq. (33) into Eq. (25) and Eq. (27), and using the sensitivity analysis algorithm (i.e., SAA1 or SAA2), we can obtain the optimal solution \mathbf{y}_α^* , and the lower bound of the optimal objective value of the ULP, LB_α . Consequently, the AISM can be used to get the nearly optimal integer solution $\hat{\mathbf{y}}_\alpha^*$. Substituting $\mathbf{y} = \hat{\mathbf{y}}_\alpha^*$ into the LLP, we can obtain the optimal link flow $\hat{\mathbf{x}}_\alpha^*$ and the upper bound of the optimal objective value of the ULP, UB_α . The upper and lower bounds of the optimal upper-level objective value for the TRDP can be derived from the solution of each unbranched search tree node, given by:

$$UB = \min_{\alpha \in UT \cup BT} \{UB_\alpha\}, \text{ and} \quad (34)$$

$$LB = \min_{\alpha \in UT} \{LB_\alpha\}. \quad (35)$$

(ii) *Branch strategy*

In the BBM, the unbranched search tree node with the minimum lower bound is chosen as a

branch node. The index of the branch node can be obtained from:

$$\beta = \arg \min_{\alpha \in UT} \{LB_{\alpha}\}. \quad (36)$$

If the branch node β has a non-integer solution, we can choose the turning restriction with the minimum non-integer value for the corresponding turning restriction decision variable as the constrained turning restriction, given by:

$$\bar{\mu} = \arg \min_{\mu \in \Gamma} \{y_{\beta\mu}^* \mid y_{\beta\mu}^* > 0\}. \quad (37)$$

The branch node β generates two new search tree nodes, denoted as β_1 and β_2 . We respectively set $\bar{y}_{\beta_1} = 0$ and $\bar{y}_{\beta_2} = 1$. The turning restrictions and the input sequences of the two generated nodes are set as:

$$\bar{\mu}_{\beta_1} = \bar{\mu}_{\beta_2} = \bar{\mu}, \quad (38)$$

$$IS_{\beta_1} = IS_{\beta} + \{\beta_1\}, \text{ and} \quad (39)$$

$$IS_{\beta_2} = IS_{\beta} + \{\beta_2\}. \quad (40)$$

The above strategies are incorporated into the proposed BBM for the TRDP, which can be stated as follows:

Step 0: *Initialization.*

Step 0.1: Set the initial upper bound $UB = +\infty$, the initial lower bound $LB = -\infty$ and the branch node set $BT = \emptyset$.

Step 0.2: Generate an initial search tree node α_0 such that $\bar{\mu}_{\alpha_0} \in \emptyset$. Set $IS_{\alpha_0} = \emptyset$ and $UT = \{\alpha_0\}$.

Step 0.3: Solve the TRDP for the search tree node α_0 and obtain the optimal solution $\mathbf{y}_{\alpha_0}^*$ and the nearly optimal integer solution $\hat{\mathbf{y}}_{\alpha_0}^*$.

Step 0.4: Determine the upper bound UB_{α_0} and the lower bound LB_{α_0} .

Step 0.5: If $\mathbf{y}_{\alpha_0}^* = \hat{\mathbf{y}}_{\alpha_0}^*$, stop and output the optimal solution $\mathbf{y}_{\alpha_0}^*$; otherwise, go to Step 1.

Step 1: *Branching.*

Step 1.1: Choose the optimal branch node β by Eq. (36).

Step 1.2: Search the best constrained turning restriction $\bar{\mu}$ by Eq. (37).

Step 1.3: Generate the search tree node β_1 , subject to $\bar{y}_{\beta_1} = 0$, $\bar{\mu}_{\beta_1} = \bar{\mu}$, and $IS_{\beta_1} = IS_{\beta} + \{\beta_1\}$. Generate also another search tree node β_2 , subject to $\bar{y}_{\beta_2} = 1$, $\bar{\mu}_{\beta_2} = \bar{\mu}$, and $IS_{\beta_2} = IS_{\beta} + \{\beta_2\}$.

Step 1.4: Solve the TRDP for each newly generated search tree node. Obtain the optimal

solutions $\mathbf{y}_{\beta_1}^*$ and $\mathbf{y}_{\beta_2}^*$, and the nearly optimal integer solutions $\hat{\mathbf{y}}_{\beta_1}^*$ and $\hat{\mathbf{y}}_{\beta_2}^*$.

Step 1.5: Determine the lower bounds LB_{β_1} and LB_{β_2} , and the upper bounds UB_{β_1} and UB_{β_2} .

Step 2: *Updating search tree node set.* Set $BT = BT \cup \{\beta\}$ and $UT = UT \cup \{\beta_1\} \cup \{\beta_2\} - \{\beta\}$.

Step 3: *Determination of the upper and lower bounds.* Calculate the upper bound UB and the lower bound LB by Eq. (34) and Eq. (35), respectively.

Step 4: *Branch removal.* For all $\alpha \in UT$, if $LB_\alpha \geq UB$, then set $UT = UT - \{\alpha\}$.

Step 5: *Convergence checking.* If $UB = LB$, set $\beta = \arg \min_{\alpha \in UT} \{UB_\alpha\}$, output the optimal solution \mathbf{y}_β^* and stop; otherwise, go to Step 1.

Remark 2. Both SAA1 and SAA2 proposed in Section 4.2 and Section 4.3, respectively, can be applied to solve the relaxed TRDP for the search tree node in step 0.3 and step 1.4.

Remark 3. SAA1+BBM (SAA2+BBM) is adopted to simplify the presentation of the algorithm that SAA1 (SAA2) and the AISM are embedded into the BBM. The performance of the two SAAs embedded into the above BBM will be tested in the next section.

5. Numerical study

To illustrate the effects of the model parameters and the performance of the proposed algorithms, we developed a numerical example using the Sioux Falls network as shown in Fig. 1, which was also used in Leblanc et al. (1975). The Sioux Falls network consists of 24 nodes, 76 links, and 528 OD pairs. The feasible turning restrictions are listed in Table 1. The average number of paths per OD pair is 3.6, and the maximum number of routes between an OD pair is 15.

The two BPR parameters are the same as those in Leblanc et al. (1975). That is, $\chi_1 = 0.15$ and $\chi_2 = 4$. We follow Horowitz (1997) using a ϕ_{opp} -value of 0.4. ϕ_{LT} and ϕ_{RT} are set to 1 for simplicity. We set the punishment cost $M = 50.0$ as the maximum route travel cost is less than 2.0 if there are no turning restrictions in the network. This value of M is believed to be large enough in this study. Nevertheless, all the above parameters can be calibrated based on the actual scenario without conceptual difficulty. The values of the parameters for solution algorithms are as follows: $\Delta = 0.25$, $\eta = 1.5$, $\gamma = 0.1$, $\varepsilon = 1.0 \times 10^{-5}$, $\sigma = 1.0 \times 10^{-3}$, and $y_{crit} = 0.1$.

To demonstrate the improvement of the network performance by implementing turning restrictions, the system costs with and without implementing turning restrictions were compared. As shown in Fig. 2, the system cost can be decreased by implementing turning restrictions with the dispersion parameter θ varied from 0.1 to 2.0. The results presented in Fig. 2 indicate that

SAA2+BBM outperforms both SAA1+BBM and SAA2+AISM. The results by SAA1+AISM are not presented because of the bad performance in solving the TRDP. Meanwhile, SAA2+BBM obviously outperforms the other two algorithms in term of accuracy. As described in Section 4.4, the solution obtained from SAA2+AISM is an approximate integer solution for the TRDP, and it is difficult to evaluate the accuracy of the solution obtained by this approach. Therefore, SAA2+BBM is recommended from the perspective of the solution accuracy for the TRDP.

The performance of the algorithms for the TRDP was further tested, and the results are listed in Table 2, where we set a value of 1.0 for the dispersion parameter θ . In this example, SAA1 is notably inferior to SAA2 in solving the relaxed TRDP. The objective value of the relaxed ULP obtained by SAA2 is less than that by SAA1. As mentioned before, SAA1 is a heuristic algorithm, and it is difficult to guarantee solution convergence. On the contrary, SAA2 is a Frank-Wolfe based algorithm which solves the relaxed TRDP as a convex programming problem. Although, it is hard to analyze the convexity of the proposed bi-level problem, we can obtain a local optimal solution for the relaxed TRDP at least.

From the SAA1 and SAA2 results presented in Table 2, we use SAA1+AISM and SAA2+AISM to obtain the nearly optimal integer solutions. It is shown that less than 10% of decision variables of the ULP of the TRDP take incorrect values corresponding to the optimal solution solved by SAA1+BBM and SAA2+BBM (Table 2). Therefore, the proposed bound strategy can aid SAA1+BBM and SAA2+BBM in searching a lower upper bound and obtain the optimal solution efficiently. Note that the value of y_{crit} has no impact on the convergence of SAA1+BBM and SAA2+BBM, because we just use the value to find the nearly optimal integer solution and reduce the upper bound. An appropriate value of y_{crit} allows SAA2+BBM to get a good starting upper bound and obtain the nearly optimal integer solution close to the global optimal integer solution quickly.

Many factors affect the performance of the proposed algorithms and the solution of the TRDP, including the value of the dispersion parameter θ and the level of network congestion. The essential fact is that the above two factors influence the implementation of turning restrictions by impacting the lower level SUE problem on both the solution and the computational time. Therefore, we conducted a test for the sensitivity of the solution of the TRDP with respect to the value of θ . Table 3 presents the optimal solutions for various values of the dispersion parameter θ . The result shows that the solution of the TRDP is robust when the value of the dispersion parameter θ varies. This observation implies that the calibration of the dispersion parameter θ needs not be too precise in

applications. Fig. 3 illustrates the effect of the value of the dispersion parameter θ on system cost saving due to implementing the optimal turning restriction strategy, where system cost saving is defined as the difference between system cost without and with the implementation of turning restrictions. It shows that implementing turning restrictions can increase system cost saving when the value of the dispersion parameter θ decreases as a whole. As mentioned before, system cost saving consists of two components: free flow travel cost saving and delay saving of all travelers. The result illustrates that free flow travel cost saving takes a small percentage of total cost saving. This indicates that the optimal turning restriction strategy mainly reduces travelers' delay.

We also studied the influence of the value of the dispersion parameter θ on the computational time of solving the TRDP by various algorithms. The results are presented in Fig. 4, which show that both SAA1+BBM and SAA2+AIISM obtained solutions faster than SAA2+BBM with a slight advantage of SAA2+AIISM over SAA1+BBM. This is because (1) the link flow is linearly approximated in SAA1, which can reduce the time required for solving the lower level SUE problem in SAA1+BBM, and (2) SAA2 is only executed once in SAA2+AIISM. However, since SAA2 is repeatedly used when solving the relaxed TRDP by SAA2+BBM, the computational time of SAA2+BBM can be very long. Moreover, since SAA2 is a Frank-Wolfe based algorithm, the computational time of SAA2+BBM is greatly consumed on the stepsize determination. However, SAA1+BBM may not be able to get an optimal solution for the TRDP. Thus, SAA1+BBM is not recommended to use.

We examined the impact of travel demand on system cost saving due to implementing turning restrictions, where the demand level reflects the degree of network congestion. The Sioux Falls demand matrix was uniformly varied by multiplying by a constant demand 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 0.05. Fig. 5 presents the system cost saving due to implementing the optimal turning restriction strategy for different demand factors when $\theta = 1.0$. The results show that implementing turning restrictions is more necessary at a higher demand level and is more effective in improving more congested traffic networks. Furthermore, the robustness of the turning restriction strategy for the base scenario was studied under various demand factors. We performed the SUE assignment to obtain the total system cost at each demand factor under this strategy. The result shows that this strategy can bring a similar system cost saving to the optimal turning restriction strategy when the demand factor varies from 0.65 to 1.15. Thus, the turning

restriction strategy is not needed to adjust when travel demand slightly increases. In addition, we can observe from Fig. 5 that system travel cost saving is increasing when the demand factor grows up. Therefore, implementing turning restrictions may work better if the network is more congested.

The proposed bi-level programming model can also be solved by solution algorithms based on artificial intelligence techniques, such as genetic algorithm (GA), simulated annealing, and neural network. To further illustrate the merits of the proposed SAA2+BBM, we compared SAA2+BBM with GA, which has been successfully applied to a wide range of transportation engineering problems (see, for example, Drezner and Wesolowsky, 2003; Ceylan and Bell, 2005). Herein, we applied the general GA (Gen and Cheng, 2000) to solve the TRDP.

The parameters for GA are set as: population size = 20, maximum number of generations = 100, crossover probability = 0.4, and mutation probability = 0.08. Two convergence criteria are used to stop the GA procedure: 1) *the number of generations equals the predetermined maximum number of generations* (convergence criterion 1) and 2) *the best upper-level objective value obtained from GA equals that obtained by SAA2+BBM* (convergence criterion 2). We adopted GA to solve the TRDP with the value of the dispersion parameter θ varied from 0.1 to 2.0. The test results show that the solutions obtained by GA cannot be better than those obtained by SAA2+BBM. Most of the time the solutions obtained by GA can converge to the solutions obtained from SAA2+BBM. These results indicate that the proposed SAA2+BBM may obtain global optimal solutions in this studied case.

The computation time of GA performance is presented in Fig. 6. We can see that a higher value of the dispersion parameter θ can bring a longer computation time for the TRDP solved by GA. SAA2+BBM can solve the TRDP faster than GA in general. The number of generations required by GA to meet the convergence criterion is not evidently different when the value of θ varies. However, the lower level SUE problem requires more iterations to achieve an optimal solution with a fixed precision when the value of θ grows up. Therefore, GA may be slightly slower to solve the TRDP with a higher value of θ . Compared with SAA2+BBM, GA has an intrinsic weakness: we do not know when GA can get optimal solutions. A high number of generations (i.e., long computation time) is required to ensure the solution quality. On the contrary, the TRDP solution obtained from SAA2+BBM is local optimal. In conclusion, SAA2+BBM performs better than the GA-based heuristic algorithm, especially when the dispersion parameter takes a high value.

6. Conclusions

This paper presents a bi-level programming model and proposes a branch and bound algorithm for solving the TRDP. For the lower level SUE problem, a penalty term is used to formulate the route travel costs, and can be considered as the punishment cost for a traveler taking a restricted turning. The SAA is applied to solve the relaxed TRDP and obtain the nearly optimal integer solution of the bi-level model. The solution of the relaxed TRDP is employed to determine the lower and upper bounds and choose the branch strategy in the BBM.

Four algorithms based on the combination of the SAA and BBM are applied to solve the TRDP in a sample network. The test results show that SAA2+BBM performs the best in terms of higher solution accuracy among all the tested algorithms. Compared with the GA based heuristic algorithm, SAA2+BBM performs the best in terms of convergence and computation time. Furthermore, numerical results show that implementing turning restrictions can obviously reduce system cost by reducing congestion cost. The sensitivity analyses of the dispersion parameter and the demand level show that turning restriction strategies are robust during the variation of traffic environment (e.g., the precision of information shared by travelers and the demand level).

It will be interesting to examine TRDP that also considers signal control or road channelization at intersections. This will be our future research direction. The future work also includes considering the TRDP in dynamic traffic networks and evaluating its effectiveness.

Acknowledgements

The authors would like to thank Dr. Shlomo Bekhor for his help on solving the lower level SUE problem, and the anonymous referees for their helpful comments and suggestions on revising this paper. This work is jointly supported by the National Basic Research Program of China (2006CB705500), the National Natural Science Foundation of China (70631001, 70801004) and the Innovation Foundation of Science and Technology for Excellent Doctoral Candidates of Beijing Jiaotong University (48040).

Reference

- Al-Madani, H.M.N., 2003. Dynamic vehicular delay comparison between a police-controlled roundabout and a traffic signal, *Transportation Research Part A* 37 (8), 681-688.
- Bekhor, S., Toledo, B., 2005. Investigating path-based solution algorithms to the stochastic user equilibrium problem. *Transportation Research Part B* 39 (3), 279-295.
- Bekhor, S., Ben-Akiva, M.E., Ramming, M. S., 2006 Evaluation of choice set generation

- algorithms for route choice models. *Annals of Operations Research* 144 (1), 235-247.
- 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.*
- Bell, M.G.H., 1995. Alternatives to Dial's logit assignment algorithm. *Transportation Research Part B* 29 (4), 287-295.
- Ceylan, H., Bell, M.G.H., 2005. Genetic algorithm solution for the stochastic equilibrium transportation networks under congestion. *Transportation Research Part B* 39 (2), 169–185.
- 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, K.M., Luo, Z.Z., 2006. Analysis of feature of left turn traffic at level crossing and countermeasures. *Technology of Highway and Transport* 22 (2), 114-118.
- 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.
- Cipriani, E., Fusco G., 2004. Combined signal setting design and traffic assignment problem. *European Journal of Operational Research* 155 (3), 569-583.
- Dial, R.B., 1971. A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research* 5 (2), 83-111.
- Drezner, Z., Wesolowsky, G.O., 2003. Network design: selection and design of links and facility location. *Transportation Research Part A* 37 (3), 241-256.
- Edmunds, T.A., Bard, J.F., 1992. An algorithm for the mixed-integer nonlinear bilevel programming problem. *Annals of Operations Research* 34 (1-4), 149-162.
- Gallo, M., D'Acierno, L., Montella, B., 2010. A meta-heuristic approach for solving the urban network design problem. *European Journal of Operational Research* 201 (1), 144–157.
- Gao, Z.Y., Song, Y.F., 2002. A reserve capacity model of optimal signal control with user-equilibrium route choice. *Transportation Research Part B* 36 (4), 313-323.
- Gao, Z., Sun, H., Zhang, H., 2007. A globally convergent algorithm for transportation continuous network design problem. *Optimization and Engineering* 8 (3), 241-257.
- Gen, M., Cheng, R., 2000. *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons, Inc., New York.
- Han, S., 2007. A route-based solution algorithm for dynamic user equilibrium assignments. *Transportation Research Part B* 41 (10), 1094-1113.
- Harwood, D.W., Mason, J.M., Brydia R.E., 1999. Design policies for sight distance at stop-controlled intersections based on gap acceptance, *Transportation Research Part A* 33 (3), 199-216.
- Horowitz, A. J., 1997. Intersection delay in regionwide 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.
- Leblanc, L.J., 1975. An algorithm for the discrete network design problem. *Transportation Science* 9 (3), 183-199.
- 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.
- Liu, J., Ma, S., Huang, C., Ma, S., 2010. A dimension-reduced method of sensitivity analysis for stochastic user equilibrium assignment model. *Applied Mathematical Modelling* 34 (2), 325-333.
- Maher, M., 1998. Algorithms for logit-based stochastic user equilibrium assignment. *Transportation Research Part B* 32 (8), 539-549.
- Patriksson, M., 2004. Sensitivity analysis of traffic equilibria. *Transportation Science* 38 (3), 258-281.
- Poorzahedy, H., Turnquist, M.A., 1982. Approximate algorithms for the discrete network design problem. *Transportation Research Part B* 16 (1), 45-55.
- Poorzahedy, H., Rouhani, O.M., 2007. Hybrid meta-heuristic algorithms for solving network design problem. *European Journal of Operational Research* 182 (2), 578-596.
- Sheffi, Y., Powell, W.B., 1982. An algorithm for the equilibrium assignment problem with random link times. *Networks* 12 (2), 191-207.
- Sheffi, Y., 1985. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.
- Troutbeck, R.J., Kako, S., 1999. Limited priority merge at unsignalized intersections, *Transportation Research Part A* 33 (3-4), 219-304.
- Yang, C., Chen, A., 2009. Sensitivity analysis of the combined travel demand model with applications, *European Journal of Operational Research* 198 (3), 909-921.
- Yin, Y., Madanat, S.M., Lu, X.Y., 2009. Robust improvement schemes for road networks under demand uncertainty. *European Journal of Operational Research* 198 (2), 470-479.
- Ying, J.Q., Lu, H.P., Shi, J., 2007. An algorithm for local continuous optimization of traffic signals. *European Journal of Operational Research* 181 (3), 1189-1197.
- Ying, J.Q., Yang, H., 2005. Sensitivity analysis of stochastic user equilibrium flows in a bi-modal network with application to optimal pricing. *Transportation Research Part B* 39 (9), 769-795.
- Zhang, H.Z., Gao, Z.Y., 2007. Two-way road network design problem with variable lanes. *Journal of Systems Science and Systems Engineering* 16 (1), 50-61.

Table 1 Turning restrictions

μ	μ_t	μ_h	μ	μ_t	μ_h
1	36	31	12	47	21
2	10	32	13	22	50
3	27	34	14	52	48
4	40	33	15	28	45
5	32	26	16	67	44
6	25	29	17	72	67
7	48	28	18	46	68
8	43	27	19	63	69
9	24	19	20	63	70
10	16	20	21	68	61
11	17	22	22	56	62

Table 2 Comparison between various sensitivity analysis algorithms ($\theta = 1.0$)

Method	SAA1	SAA1+AISM	SAA1+BBM	SAA2	SAA2+AISM	SAA2+BBM
Objective	155.5330	155.5368	155.5368	153.9007	155.9272	154.4151
y_1	1	1	1	1	1	1
y_2	1	1	1	1	1	1
y_3	1	1	1	1	1	1
y_4	1	1	1	1	1	1
y_6	1	1	1	1	1	1
y_7	1	1	1	1	1	1
y_8	0.0015	0	0	0.0116	0	0
y_9	1	1	1	1	1	1
y_{11}	1	1	1	1	1	1
y_{13}	1	1	1	1	1	1
y_{14}	1	1	1	1	1	1
y_{15}	1	1	1	0.0235	0	1
y_{16}	1	1	1	1	1	1
y_{18}	1	1	1	I	I	0
y_{19}	1	1	1	1	1	1
y_{20}	1	1	1	1	1	1
y_{21}	1	1	1	1	1	1
y_{22}	1	1	1	0.0116	0	0

* The turning restrictions not listed in the table take a value of 0.

Table 3 The optimal solutions for various values of the dispersion parameter θ .

The dispersion parameter θ	The optimal turning restriction strategy \mathbf{y}^*
0.1~0.4	1111010010101111001110
0.5~1.2	1111011010101111001110
1.3~2.0	1111011010101111011110

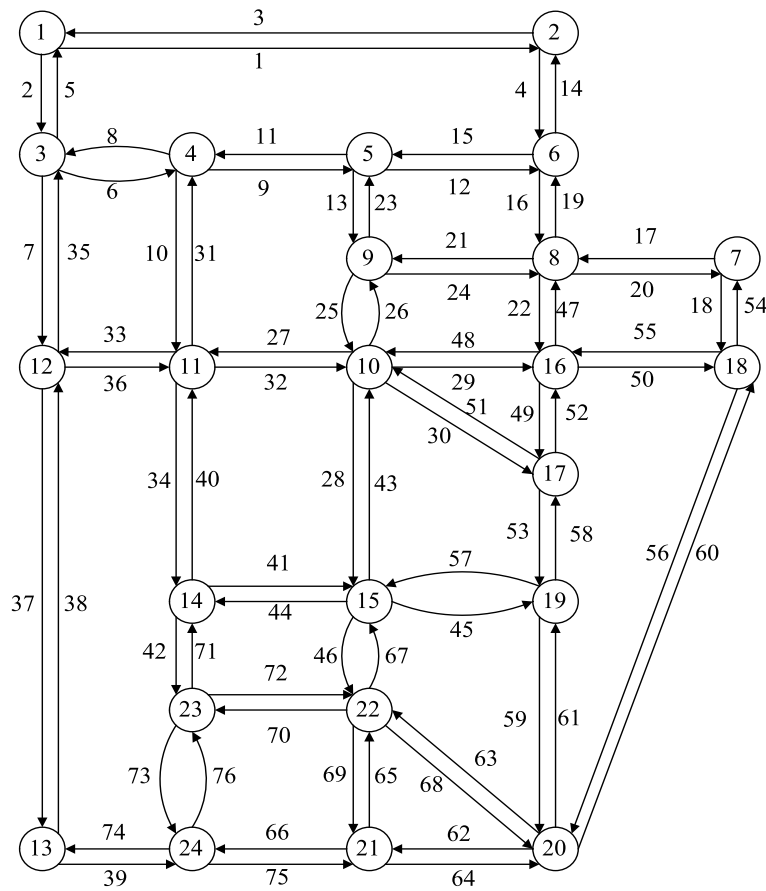


Fig. 1. The Sioux Falls network.

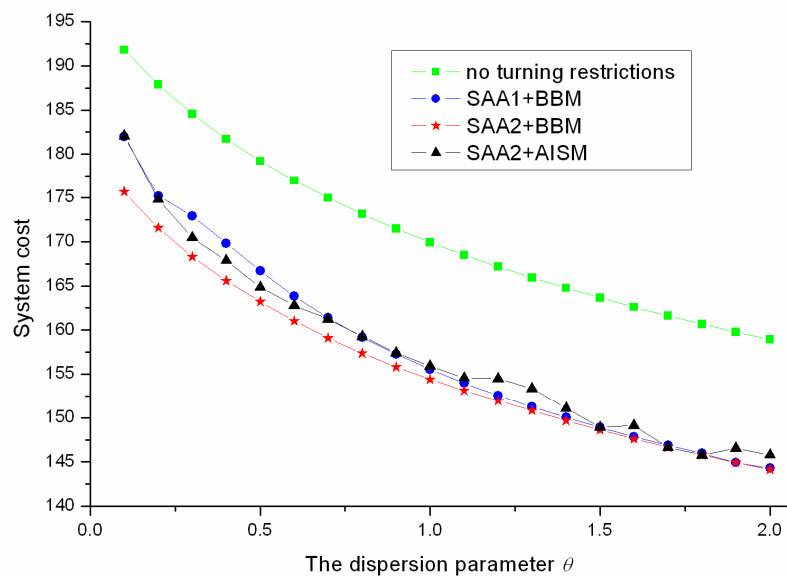


Fig. 2. The influence of the value of the dispersion parameter θ on system cost for various algorithms.

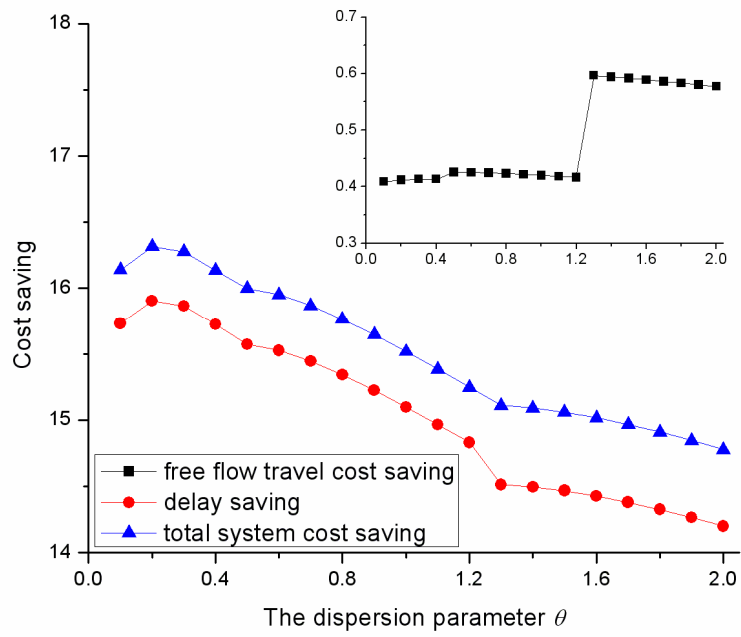


Fig. 3: The effect of the value of the dispersion parameter θ on system cost saving due to implementing turning restrictions

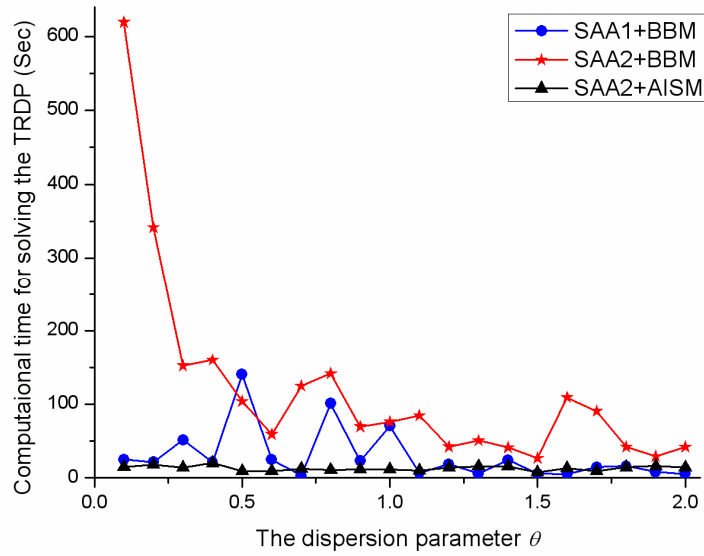


Fig. 4. The influence of the value of the dispersion parameter θ on the computational time of solving the TRDP by various algorithms.

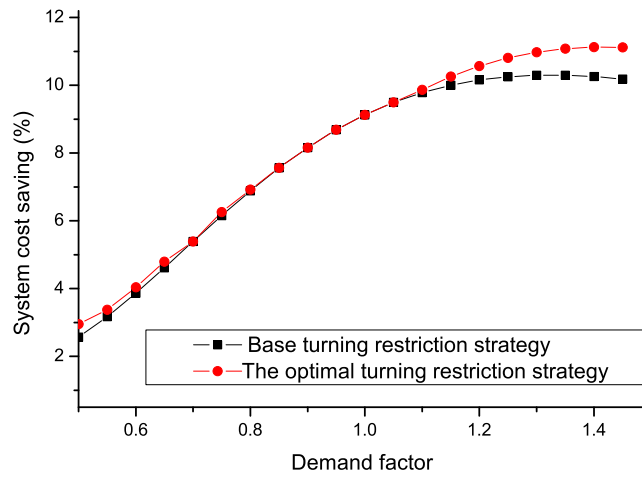


Fig. 5. The effect of travel demand on system cost saving due to implementing turning restrictions ($\theta=1.0$).

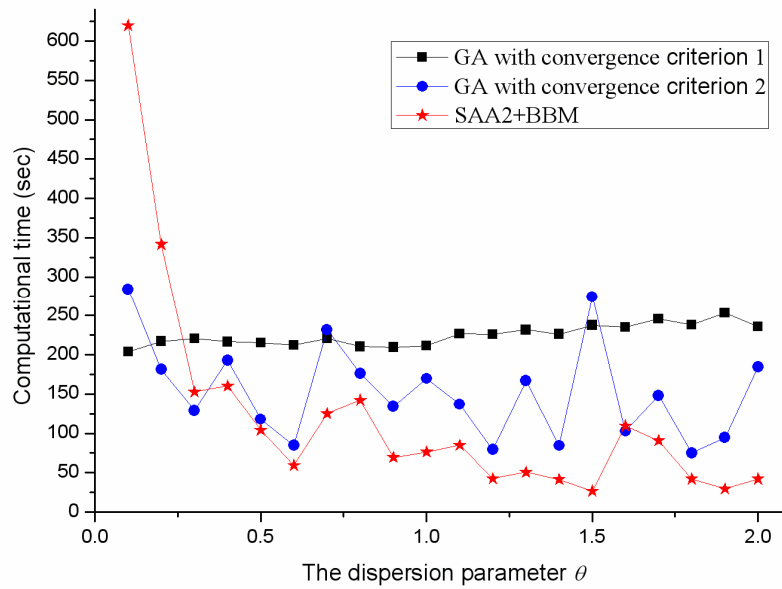


Fig. 6. The influence of the value of the dispersion parameter θ on the computational time of solving the TRDP by GA and SAA2+BBM.