Jrl Syst Sci & Complexity (2009) 22: 446–459

BILEVEL PROGRAMMING MODEL AND SOLUTION METHOD FOR MIXED TRANSPORTATION NETWORK DESIGN PROBLEM*

Haozhi ZHANG · Ziyou GAO

Received: 8 October 2006 / Revised: 28 March 2008
©2009 Springer Science + Business Media, LLC

Abstract By handling the travel cost function artfully, the authors formulate the transportation mixed network design problem (MNDP) as a mixed-integer, nonlinear bilevel programming problem, in which the lower-level problem, comparing with that of conventional bilevel DNDP models, is not a side constrained user equilibrium assignment problem, but a standard user equilibrium assignment problem. Then, the bilevel programming model for MNDP is reformulated as a continuous version of bilevel programming problem by the continuation method. By virtue of the optimal-value function, the lower-level assignment problem can be expressed as a nonlinear equality constraint. Therefore, the bilevel programming model for MNDP can be transformed into an equivalent single-level optimization problem. By exploring the inherent nature of the MNDP, the optimal-value function for the lower-level equilibrium assignment problem is proved to be continuously differentiable and its functional value and gradient can be obtained efficiently. Thus, a continuously differentiable but still nonconvex optimization formulation of the MNDP is created, and then a locally convergent algorithm is proposed by applying penalty function method. The inner loop of solving the subproblem is mainly to implement an all-or-nothing assignment. Finally, a small-scale transportation network and a large-scale network are presented to verify the proposed model and algorithm.

Key words Bilevel programming, network design, optimal-value function, penalty function method.

1 Introduction

To accommodate the growing traffic demand in transportation networks, the way that people usually adopt is to expand the capacities of the existing congested links or build new links. In this case, it is an interesting problem that how to select the location of these new links and how much additional capacity is to be expanded to each of these existing links in order to minimize the total system costs under limited expenditure, while accounting for the route choice behavior of network users. This problem is regarded as the network design problem (NDP) and can be roughly classified into three categories: the discrete network design problem (DNDP) that deals with the selection of the optimal locations (expressed by 0-1 integer decision variables)

Haozhi ZHANG · Ziyou GAO
School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, *China*; *China Urban Sustainable Transport Research Centre, China Academy of Transportation Sciences, Beijing* 100029, *China.*
Email: haozhizhang7@hotmail.com; zygao@center.njtu.edu.cn.
*This research is supported by the National Basic Research Program of China under Grant No. 2006CB705500, the National Natural Science Foundation of China under Grant No. 0631001, the Program for Changjiang Scholars and Innovative Research Team in University, and Volvo Research and Educational Foundations.

of new links to be added; the continuous network design problem (CNDP) that determines the optimal capacity enhancement (expressed by continuous decision variables) for a subset of existing links; and the mixed network design problem (MNDP) in which the enhancement and additions of road segments to an existing transport network is treated jointly rather than separately. MNDP would ensure more efficient utilization of the improved new road system and is regarded as a closer representation to the realistic transportation planning decision, but is more difficult to solve comparing with CNDP and DNDP. In the last three decades, studies have been overwhelmingly focused on the CNDP[1–5], and a large number of researchers have developed solution algorithms of one type or another for both DNDP and CNDP[6–12]. On the contrary, the MNDP has so far been surprisingly little studied. Yang and Meng[13] adopted bilevel programming methods for modeling the contemporary build-operate-transfer (BOT) highway design problem that can be considered one of the MNDPs.

The resultant MNDP model can be formulated as a mixed-integer, nonlinear bilevel programming problem, which has long been recognized to be one of the most difficult and challenging problems in operations research. By far, only several branch-and-bound methods and non-numerical algorithms have been proposed for solving the problem. Edmunds and Bard[14] transformed the mixed-integer nonlinear bilevel programming into a mixed-integer single-level nonlinear programming via replacing the lower-level problem with its Kuhn-Tucker conditions, and developed a branch-and-bound type algorithm (implicit enumeration method) based on the active constraint strategy, however their algorithm did not explore the inherent nature of the transportation NDP and is not practical to solve the realistic transportation problem. In practice, transportation planners developed an explicit enumeration method since there are being considered only limited number of new roads to be built into the basic network. They grouped the new links under consideration for addition into a certain number of feasible candidate projects; for each candidate project, a continuous capacity expansion plan for a subset of existing links is determined by solving a CNDP together with the given new links to be built. Consequently, each project consists of a combination of a subset of new links with joint consideration of capacity enhancement of existing links. After all candidate projects are enumerated and considered through solving the respective sub-CNDPs, the best project can be determined according to some criteria, together with the determination of capacity improvements of existing links associated with that discrete candidate project. It is, therefore, expected that a satisfactory solution to the MNDP could be obtained by solving a number of sub-CNDPs[1]. Naturally, the enumeration methods are expected to suffer from an exponential growth in computational requirements because they require repeatedly and jointly applications of the algorithms for both DNDP and CNDP. In order to avoid the unbearable computational burden, here we propose a locally convergent solution method for the MNDP. Meng et al.[10] proposed a locally convergent algorithm for CNDP via problem transformation. They reformulated the CNDP under the deterministic user equilibrium (DUE) constraints into an equivalent single level continuously differentiable problem by virtue of an optimal-value function tool, and applied a locally convergent augmented Lagrangian method to solve this equivalent problem. Meng et al.[11] presented another single-level continuously differentiable optimization formulation to replace the unified bi-level programming model, so that a unified solution method is obtained. In this paper, we generalize the method of [10–11] to design a locally convergent algorithm for MNDP. First, by handling the travel cost function artfully, the mixed-integer bilevel programming model for MNDP can be reformulated as a continuous version of bilevel programming problem by the continuation method, in which the lower-level problem, comparing with that of conventional bilevel DNDP models, is not a side constrained user equilibrium assignment problem, but a standard user equilibrium assignment problem. The lower-level problem can be expressed as a nonlinear constraint by virtue of the optimal-value function. Therefore, the mixed-integer bilevel programming model for MNDP can be transformed into an equivalent single-level nonlinear optimization problem. For the single-level nonlinear optimization problem, there are many methods to solve it. But it is difficult to find a global minimum because the nonlinear constraint representing the lower-level problem is nonconvex. Shimizu and Lu[15] schemed an auxiliary concave programming with convex constraint set for the equivalent single-level nonlinear optimization problem and solved the concave programming globally by outer approximation algorithm. But in outer approximation algorithm, the vertexes of the polyhedron, which envelops the convex constrained set of the concave programming, increase exponentially as iteration progresses. Compounding the vertexes proliferation problem is another deficiency in outer approximation algorithm that the computational speed fairly depends on the chosen inner point of the constrained set. Therefore, it is hard to apply their method to the realistic transportation problem. Here, we propose a locally convergent algorithm for MNDP based on penalty function concept and convex combination method (such as method of successive averages).

This paper is organized as follows. In Section 2, the bilevel programming formulation for MNDP is introduced. Section 3 transforms the bilevel programming formulation for MNDP into a single-level optimization problem only with continuous variables by virtue of the continuation method and an optimal-value function tool. Section 4 designs a locally convergent algorithm for MNDP based on penalty function concept and method of successive averages (MSA). In Section 5, two test networks are given to verify the proposed algorithm. Conclusions are summarized in Section 6.

2 Model Formulation for MNDP

The road planners can influence, but cannot control, the route choice behavior of network users, which is normally described by a network user equilibrium model. Mathematically, the bilevel programming is a good technique to describe this hierarchical property of the NDP with an equilibrium constraint. Generally, MNDP is formulated as the following mixed-integer, nonlinear bilevel programming model:

$$\min_{\boldsymbol{y},\boldsymbol{u}} \quad Z(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x}) = \sum_{a\in A} x_a t_a(x_a, y_a, u_a) + \theta \sum_{a\in A_1} G_a(y_a) + \theta \sum_{a\in A_2} c_a u_a \tag{1}$$

$$\text{s.t.} \quad \underline{\rho}_a \le y_a \le \overline{\rho}_a, \ \forall a \in A_1, \tag{2}$$

$$u_a \in 0, 1, \quad \forall a \in A_2, \tag{3}$$

where $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{y},\boldsymbol{u})$ is implicitly defined by

$$\min_{\boldsymbol{x}} \quad F(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x}) = \sum_{a\in A} \int_0^{x_a(\boldsymbol{y},\boldsymbol{u})} t_a(w, y_a, u_a)\, dw \tag{4}$$

$$\text{s.t.} \quad \sum_k f_k^{rs} = q_{rs}, \ \forall r \in R, \ s \in S, \tag{5}$$

$$f_k^{rs} \ge 0, \ \forall r \in R, \quad s \in S, \quad k \in K_{rs}, \tag{6}$$

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}, \quad \forall a \in A. \tag{7}$$

Notations:

A : $A = A_1 \cup A_2$, The set of links in the network, where $A_1 = (a|a = 1, 2, \cdots, n)$ is the the set of existing links, $A_2 = (a|a = n+1, n+2, \cdots, n+m)$ is the set of new links to be established, and $a \in A$ is an arbitrary link;

R, S: The set of origins and destinations;
r: Origin node, $r \in R$;
s: Destination node, $s \in S$;
K_{rs}: Set of routes between nodes r and s;
y_a: Upper-level continuous decision variable, representing the continuous capacity enhancement of link a, $a \in A_1$, $\boldsymbol{y} = (y_1, y_2, \cdots, y_a, \cdots, y_n)^{\mathrm{T}}$;
$\underline{\rho}, \overline{\rho}$: The lower and upper bounds for link capacity enhancement;
u_a: Upper-level binary decision variable, if link a is built, then $u_a = 1$ and otherwise $u_a = 0$. $\boldsymbol{u} = (u_{n+1}, u_{n+2}, \cdots, u_{n+m})$ is the vector;
c_a: The construction cost on candidate link a to be established, $a \in A_2$;
$G_a(y_a)$: The cost function of capacity enhancement of existing link a, $a \in A_1$;
θ: The relative weight of total capacity enhancement cost and total travel cost in the system design objective function;
q_{rs}: The fixed travel demand between OD pair (r, s), $\boldsymbol{q}$ is the vector of travel demand;
x_a: The flow on link a, $\boldsymbol{x} = (x_1, \cdots, x_a, \cdots, x_{n+m})$, lower-level decision variable;
f_k^{rs}: The path flow on route k connecting origin r and destination s;
$t_a(x_a, y_a, u_a)$: The travel time (or cost) on link a, $a \in A$, which is continuously differentiable and strictly increasing for fixed (y_a, u_a). In this paper, we use the following form for $t(.)^{\dagger}$.

$$t_a(x_a, y_a) = T_a^0\left(1 + 0.5\Big(\frac{x_a}{K_a + y_a}\Big)^2\right), \ \forall a \in A_1,$$

$$t_a(x_a, u_a) = \begin{cases} T_a^0\left(1 + 0.5\Big(\dfrac{x_a}{K_a}\Big)^2\right), & \text{if } 0 < u_a \le 1, \\ M, & \text{if } \quad u_a = 0, \end{cases} \quad \forall a \in A_2;$$

M: A sufficiently large positive constant.

From Model (1)–(7), observe that it is a mixed-integer, nonlinear bilevel programming problem with both continuous and binary decision variables. In this model, the travel demand is fixed and travel users' route choice behavior follows the user equilibrium principle. In the upper-level problem, transportation planners improve the existing links and construct new links to minimize the total system cost and total expenditure. By taking the travel cost function as the aforementioned form, the side constraints, $x_a \le Mu_a, \forall a \in A_2$, is avoided and the lower-level problem can be formulated as a standard user equilibrium assignment problem. In conventional bilevel DNDP models (such as [7,12]), the constraint, $x_a \le Mu_a, \forall a \in A_2$, must be contained to prohibit flow assignment on non-existent links, if $u_a = 0$, then, $x_a = 0$, if $u_a = 1$, then x_a is unlimited because M is sufficiently large positive constant. The side constraints $x_a \le Mu_a, \forall a \in A_2$ destroy the profitable Cartesian product structure which is inherent in the standard equilibrium assignment problem, and make the lower level problem computationally more demanding. Larsson and Patriksson[16] proposed an augmented Lagrangian dual algorithm for the link capacity side constrained assignment problems.

3 Single-Level Equivalent Programming Model for MNDP

Let $\boldsymbol{e}$ be a vector of ones with m dimensions. Obviously, $u_i \in 0, 1\, (i = 1, 2, \cdots, n_2)$ is equivalent to $\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{e} - \boldsymbol{u}) = \boldsymbol{0}$ when $\boldsymbol{0} \le \boldsymbol{u} \le 1$. Therefore, Model (1)–(7) for MNDP can be

†Thus, the lower level problem can be formulated as a standard UE assignment problem, different from that of the conventional bilevel DNDP models, which must contain ticklish side constraints rewritten as the following bilevel programming problem only with continuous variables:

$$\min_{\boldsymbol{y},\boldsymbol{u}} \quad Z(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})=\sum_{a\in A}x_a t_a(x_a,y_a,u_a)+\theta\sum_{a\in A_1}G_a(y_a)+\theta\sum_{a\in A_2}c_a u_a \tag{8}$$

$$\text{s.t.} \quad \underline{\rho}_a\le y_a\le\overline{\rho}_a,\ \forall a\in A_1, \tag{9}$$

$$0\le u_a\le 1,\ \forall a\in A_2, \tag{10}$$

$$\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{e}-\boldsymbol{u})=0, \tag{11}$$

where $\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{y},\boldsymbol{u})$ is implicitly defined by

$$\min_{\boldsymbol{x}} \quad F(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})=\sum_{a\in A}\int_0^{x_a(\boldsymbol{y},\boldsymbol{u})}t_a(w,y_a,u_a)\,dw \tag{12}$$

$$\text{s.t.} \quad \sum_k f_k^{rs}=q_{rs},\quad \forall r\in R,\ s\in S, \tag{13}$$

$$f_k^{rs}\ge 0,\ \forall r\in R,\ s\in S,\ k\in K_{rs}, \tag{14}$$

$$x_a=\sum_r\sum_s\sum_k f_k^{rs}\delta_{a,k}^{rs},\ \forall a\in A. \tag{15}$$

Next, we introduce the concept of optimal-value function. By exploring the optimal-value function, Problem (12)–(15) can be expressed by a nonlinear equality constraint. Let Ω_1 denote the set of the feasible link flows for Problem (12)–(15), i.e.,

$$\Omega_1=\{x_a,\ a\in A|x_a \quad \text{satisfies} \quad (13)-(15)\}$$

The optimal-value function of Problem (12)–(15) is defined as follows:

$$\omega(\boldsymbol{y},\boldsymbol{u})=\min_{x_a\in\Omega_1}F(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})=\min_{x_a\in\Omega_1}\sum_{a\in A}\int_0^{x_a(\boldsymbol{y},\boldsymbol{u})}t_a(w,y_a,u_a)dw. \tag{16}$$

$\omega(\boldsymbol{y},\boldsymbol{u})$ is convex and differentiable with respect to $(\boldsymbol{y},\boldsymbol{u})$ (see [17]). For any given $(\boldsymbol{y},\boldsymbol{u})$, the gradient, $\nabla\omega(\boldsymbol{y},\boldsymbol{u})$ of $\omega(\boldsymbol{y},\boldsymbol{u})$ can be obtained easily according to the formula in [10]:

$$\nabla\omega(\boldsymbol{y},\boldsymbol{u})=\left(\cdots,\frac{\partial\omega(\boldsymbol{y},\boldsymbol{u})}{\partial y_a},\cdots,\frac{\partial\omega(\boldsymbol{y},\boldsymbol{u})}{\partial u_a}\right)$$

$$\frac{\partial\omega(\boldsymbol{y},\boldsymbol{u})}{\partial y_a}=\int_0^{x_a^*(\boldsymbol{y},\boldsymbol{u})}\frac{\partial F(w,\boldsymbol{y},\boldsymbol{u})}{\partial y_a}dw =-\frac{1}{3}T_a^0\left(\frac{x_a^*(\boldsymbol{y},\boldsymbol{u})}{K_a+y_a}\right)^3,\ \forall a\in A_1 \tag{17}$$

$$\frac{\partial\omega(\boldsymbol{y},\boldsymbol{u})}{\partial u_a}=\int_0^{x_a^*(\boldsymbol{y},\boldsymbol{u})}\frac{\partial F(w,\boldsymbol{y},\boldsymbol{u})}{\partial u_a}dw =\begin{cases}-\dfrac{T_a^0x_a^*(\boldsymbol{y},\boldsymbol{u})}{(u_a)^2}, & \text{if} \quad 0<u_a\le 1,\\ -M, & \text{if} \quad u_a=0,\end{cases}\quad \forall a\in A_2, \tag{18}$$

where $x_a^*(\boldsymbol{y},\boldsymbol{u}),\ a\in A$ is the equilibrium link flow for fixed $\boldsymbol{y}$ and $\boldsymbol{u}$.

Obviously, for any feasible point $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$, it always has

$$F(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) - \omega(\boldsymbol{y}, \boldsymbol{u}) \geq 0. \tag{19}$$

Noting that the lower-level problem has unique link flows, thus, $F(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) - \omega(\boldsymbol{y}, \boldsymbol{u}) = 0$ holds only and if only $\boldsymbol{x}$ are the user equilibrium link flows for fixed $\boldsymbol{y}$ and $\boldsymbol{u}$. Therefore, the bilevel model (1)–(7) for MNDP can be transformed into a single-level nonlinear programming as follows:

$$\min_{\boldsymbol{y}, \boldsymbol{u}} \quad Z(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) = \sum_{a \in A} x_a t_a(x_a, y_a, u_a) + \theta \sum_{a \in A_1} G_a(y_a) + \theta \sum_{a \in A_2} c_a u_a \tag{20}$$

$$\text{s.t.} \quad \underline{\rho}_a \leq y_a \leq \overline{\rho}_a, \ \forall a \in A_1, \tag{21}$$

$$0 \leq u_a \leq 1, \ \forall a \in A_2, \tag{22}$$

$$\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{e} - \boldsymbol{u}) = 0, \tag{23}$$

$$F(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) - \omega(\boldsymbol{y}, \boldsymbol{u}) = 0, \tag{24}$$

$$\sum_k f_k^{rs} = q_{rs}, \ \forall r \in R, \ s \in S, \tag{25}$$

$$f_k^{rs} \geq 0, \ \forall r \in R, \ s \in S, \ k \in K_{rs}, \tag{26}$$

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}, \ \forall a \in A, \tag{27}$$

where Constraints (24)–(27) are equivalent to the lower-level problem (4)–(7). $\omega(\boldsymbol{x}, w)$ has not explicit formulation in general and Constraint (24) is nonconvex, thus, it is difficult to solve Problem (20)–(27) globally.

4 Solution Algorithm

Here, we employ the augmented Lagrangian algorithm[18] to design a locally convergent algorithm for the aforementioned MNDP. One of the advantages of the method is that the nonlinear, implicit Constraints (23) and (24) can be incorporated into the objective function as penalty terms.

For simplicity of notation, we denote $F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) - \omega(\boldsymbol{y}, \boldsymbol{u})$ as

$$h(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) = F(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) - \omega(\boldsymbol{y}, \boldsymbol{u}). \tag{28}$$

We introduce two large positive constants $\gamma_1 > 0$ and $\gamma_2 > 0$ as multipliers, $\rho_1 > 0$ and $\rho_2 > 0$ as penalty parameters. Let

$$\begin{aligned} & L(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\rho}) \\ = {} & Z(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) + \gamma_1 h(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) + \gamma_2 \boldsymbol{u}^{\mathrm{T}}(\boldsymbol{e} - \boldsymbol{u}) + \frac{1}{2}\rho_1 (h(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}))^2 + \frac{1}{2}\rho_2 (\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{e} - \boldsymbol{u}))^2, \end{aligned}$$

where $Z(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{x}) = \sum\limits_{a \in A} x_a t_a(x_a, y_a, u_a) + \theta \sum\limits_{a \in A_1} G_a(y_a) + \theta \sum\limits_{a \in A_2} c_a u_a$ is the above mentioned upper-level objective function.

Then, we obtain the following auxiliary problem with simple linear constraints:

$$\min_{\boldsymbol{y},\,\boldsymbol{u},\,\boldsymbol{x}} \quad L(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x},\boldsymbol{\gamma},\boldsymbol{\rho}) \tag{29}$$

$$\text{s.t.} \quad \underline{\rho}_a \le y_a \le \overline{\rho}_a,\ \forall a \in A_1, \tag{30}$$

$$0 \le u_a \le 1,\ \forall a \in A_2, \tag{31}$$

$$\sum_k f_k^{rs} = q_{rs},\ \forall r \in R,\ s \in S, \tag{32}$$

$$f_k^{rs} \ge 0,\ \forall r \in R,\ s \in S,\ k \in K_{rs}, \tag{33}$$

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}. \tag{34}$$

The objective function of Problem (29)–(34) is nonlinear, but the constraints of Problem (29)–(34) are linear and separable with respect to the variables $\boldsymbol{y},\ \boldsymbol{u}$, and $\boldsymbol{x}$. Since the gradient of the optimal-function $\omega(\boldsymbol{y},\boldsymbol{u})$ can be figured out from formulas (17) and (18), the gradient of $L(\cdot)$ can be obtained easily. Based on MSA (method of successive averages), we scheme the following procedure to find a locally optimal solution to the original MNDP.

Main Algorithm

Step 0 Initialization: Let $\delta > 0$ be a termination scalar. Select a feasible link flow x_a^0, $a \in A$, capacity enhancement y_a^0, $a \in A_1$ and new link addition pattern u_a^0, $a \in A_2$. Choose penalty parameters $\gamma_1^0 > 0$ and $\gamma_2^0 > 0$, and a scalar $\alpha > 1$. Set $k = 0$.

Step 1 Solve the lower-level problem: For fixed y_a^k, $a \in A_1$ and u_a^k, $a \in A_2$, solve the lower-level problem (4)–(8) by implementing a user equilibrium assignment subroutine and obtain the functional value of $\omega(\boldsymbol{y}^k,\boldsymbol{u}^k)$ and the equilibrium link flow $(x_a^k)^*$, $a \in A$. Then, we calculate the gradient $\nabla\omega(\boldsymbol{y}^k,\boldsymbol{u}^k)$ according to formulas (17) and (18).

Step 2 Solve the subproblem: For fixed penalty parameters γ_1^k and γ_2^k, set $(\boldsymbol{y}^k,\boldsymbol{u}^k,\boldsymbol{x}^k)$ as an initial point, solve the following subproblem with MSA:

$$\min_{\boldsymbol{y},\,\boldsymbol{u},\,\boldsymbol{x}} \quad L(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x},\gamma_1^k,\gamma_2^k) \quad \text{subject to Constraints (30) – (34).}$$

Let $(\boldsymbol{y}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{x}^{k+1})$ denote the solution.

Step 3 Verify termination criterion: If $\gamma_1^k h(\boldsymbol{y}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{x}^{k+1}) < \delta$ and $\gamma_2^k \boldsymbol{u}^{k+1}(\boldsymbol{e}-\boldsymbol{u}^{k+1}) < \delta$ both hold, then terminate, take x_a^{k+1}, $a \in A$, y_a^{k+1}, $a \in A_1$ and rounded u_a^{k+1}, $a \in A_2$ as an optimal solution to the MNDP. Otherwise, let $\gamma_1^{k+1} = \alpha\gamma_1^k,\ \gamma_2^{k+1} = \alpha\gamma_2^k$, $k := k+1$, and go to Step 1.

In Step 1, the lower-level problem with fixed $\boldsymbol{y}$ and $\boldsymbol{u}$ is a standard assignment problem because we handle the travel cost function as the aforementioned form in Section 2. In Step 3, we terminate the computation when the penalty terms $\gamma_1^k h(\boldsymbol{y}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{x}^{k+1}) < \delta$ and $\gamma_2^k \boldsymbol{u}^{k+1}(\boldsymbol{e}-\boldsymbol{u}^{k+1}) < \delta$ get sufficiently small. This idea has been widely used in penalty methods, e.g., [18]. In Step 2, the constraint set of subproblem contains only simple bounds of $\boldsymbol{y}$ and $\boldsymbol{u}$, and flow conservation equations, so the linear programming for finding the descent direction in each iteration can be decomposed into three simple linear programming subproblems.

For fixed penalty parameters, the gradient of $L(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})$ at the point $(\boldsymbol{y}^k,\boldsymbol{u}^k,\boldsymbol{x}^k)$ is

$$\begin{aligned}\frac{\partial L}{\partial x_a}\bigg|_{(x_a^k,y_a^k,u_a^k)} &= \Big(\frac{\partial Z(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial x_a} + \gamma_1^k \frac{\partial F(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial x_a}\Big)_{(x_a^k,y_a^k,u_a^k)} \\ &= T_a^0\Big(\frac{x_a^k}{K_a+y_a^k}\Big)^2 + (1+\gamma_1^k)T_a^0\Big(1+0.5\Big(\frac{x_a^k}{K_a+y_a^k}\Big)^2\Big),\quad \forall a \in A_1; \end{aligned} \tag{35}$$

$$\begin{aligned}
\frac{\partial L}{\partial x_a}\Big|_{(x_a^k,y_a^k,u_a^k)} &= \Big(\frac{\partial Z(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial x_a}+\gamma_1^k\frac{\partial F(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial x_a}\Big)_{(x_a^k,y_a^k,u_a^k)} \\
&= \begin{cases}\dfrac{T_a^0}{u_a^k}\Big(\dfrac{x_a^k}{K_a}\Big)^2+(1+\gamma_1^k)\dfrac{T_a^0}{u_a^k}\Big(1+0.5\Big(\dfrac{x_a^k}{K_a}\Big)^2\Big),\ 0<u_a\le 1,\\ M,\ u_a=0,\ \forall a\in A_2;\end{cases} \qquad (36)\\
\frac{\partial L}{\partial y_a}\Big|_{(x_a^k,y_a^k)} &= \Big(\frac{\partial Z(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial y_a}+\gamma_1^k\frac{\partial F(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial y_a}-\gamma_1^k\frac{\partial \omega(\boldsymbol{y},\boldsymbol{u})}{\partial y_a}\Big)_{(x_a^k,y_a^k)} \\
&= -T_a^0\Big(\frac{x_a^k}{K_a+y_a^k}\Big)^3+\theta\frac{dG_a(y_a)}{dy_a}\Big|_{(x_a^k,y_a^k)} \\
&\quad -\frac{\gamma_1^k}{3}T_a^0\Big(\frac{x_a^k}{K_a+y_a^k}\Big)^3-\gamma_1^k\frac{\partial\omega(\boldsymbol{y},\boldsymbol{u})}{\partial y_a}\Big|_{(x_a^k,y_a^k)},\quad \forall a\in A_1; \qquad (37)\\
\frac{\partial L}{\partial u_a}\Big|_{(x_a^k,u_a^k)} &= \Big(\frac{\partial Z(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial u_a}+\gamma_1^k\frac{\partial F(\boldsymbol{y},\boldsymbol{u},\boldsymbol{x})}{\partial u_a}-\gamma_1^k\frac{\partial \omega(\boldsymbol{y},\boldsymbol{u})}{\partial u_a}+\gamma_2^k(\boldsymbol{e}-2\boldsymbol{u})\Big)_{(x_a^k,u_a^k)} \\
&= \begin{cases}\theta c_a-(\gamma_1^k+1)\dfrac{T_a^0x_a^k}{(u_a)^2}-\gamma_1^k\dfrac{\partial\omega(\boldsymbol{y},\boldsymbol{u})}{\partial u_a}+\gamma_2^k(1-2u_a^k),\quad 0<u_a\le 1,\\ -M,\quad u_a=0,\quad \forall a\in A_2.\end{cases} \qquad (38)
\end{aligned}$$

In MSA, solving the following linear programming can generate the descent direction at the current solution $(\boldsymbol{y}^k,\boldsymbol{u}^k,\boldsymbol{x}^k)$:

$$\begin{aligned}
\min_{y_a,u_a,x_a}\ & \sum_{a\in A}\Big(\frac{\partial L}{\partial x_a}\Big|_{(x_a^k,y_a^k,u_a^k)}\Big)x_a+\sum_{a\in A_1}\Big(\frac{\partial L}{\partial y_a}\Big|_{(x_a^k,y_a^k)}\Big)y_a+\sum_{a\in A_2}\Big(\frac{\partial L}{\partial u_a}\Big|_{(x_a^k,u_a^k)}\Big)u_a\\
\text{s.t.}\ & \underline{\rho}_a\le y_a\le\overline{\rho}_a,\ \forall a\in A_1,\\
& 0\le u_a\le 1,\ \forall a\in A_2,\\
& \sum_k f_k^{rs}=q_{rs},\ \forall r\in R,\ s\in S,\\
& f_k^{rs}\ge 0,\ \forall r\in R,\ s\in S,\ k\in K_{rs},\\
& x_a=\sum_r\sum_s\sum_k f_k^{rs}\delta_{a,k}^{rs},\ \forall a\in A.
\end{aligned}$$

By inspection of its structure, the linear programming model can be decomposed into three independent linear programming subproblems by portioning the coefficients in the objective function and the constraints below:

$$\begin{aligned}
\text{LP1}:\quad & \min_{x_a}\ \sum_{a\in A}\Big(\frac{\partial L}{\partial x_a}\Big|_{(x_a^k,y_a^k,u_a^k)}\Big)x_a\\
& \text{s.t.}\ \sum_k f_k^{rs}=q_{rs},\ \forall r\in R,\ s\in S,\\
& \qquad f_k^{rs}\ge 0,\ \forall r\in R,\ s\in S,\ k\in K_{rs},\\
& \qquad x_a=\sum_r\sum_s\sum_k f_k^{rs}\delta_{a,k}^{rs},\ \forall a\in A;\\
\text{LP2}:\quad & \min_{y_a}\ \sum_{a\in A_1}\Big(\frac{\partial L}{\partial y_a}\Big|_{(x_a^k,y_a^k)}\Big)y_a\\
& \text{s.t.}\ \underline{\rho}_a\le y_a\le\overline{\rho}_a,\ \forall a\in A_1;
\end{aligned}$$

$$\text{LP3}: \min_{u_a} \sum_{a\in A_2} \Big(\frac{\partial L}{\partial u_a}\Big|_{(x_a^k, u_a^k)}\Big) u_a$$
$$\text{s.t.} \quad 0 \le u_a \le 1, \ \forall a \in A_2.$$

In terms of path flow variables, Model (LP1) can be rewritten as

$$\begin{aligned} \min_{f_k^{rs}} \quad & \sum_r \sum_s \sum_k (\widetilde{c}_k^{rs})^k f_k^{rs} \\ \text{s.t.} \quad & \sum_k f_k^{rs} = q_{rs}, \ \forall r \in R, \ s \in S \\ & f_k^{rs} \ge 0, \ \forall r \in R, \ s \in S, \ k \in K_{rs}, \end{aligned}$$

where $\widetilde{c}_k^{rs} = \sum_{a\in A} \widetilde{t}_a \delta_{a,k}^{rs}$, $\forall r \in R$, $s \in S$, $k \in K_{rs}$ is the path cost associated with the generalized link cost between OD pair (r, s). Therefore, the optimal solution $\overline{x}_a^k$, $a \in A$ of Model (LP1) can be simply obtained by implementing all-or-nothing assignment procedure.

Let $\overline{y}_a^k$ $(a \in A_1)$ denote the optimal solution of Model (LP2), which can be expressed in the form of

$$\overline{y}_a^k = \begin{cases} \underline{\rho}_a, & \text{if } \dfrac{\partial L}{\partial y_a}\Big|_{(x_a^k, y_a^k)} > 0, \\ \overline{\rho}_a, & \text{else}, \end{cases} \quad a \in A_1. \tag{39}$$

In the same way, the optimal solution $\overline{u}_a^k$, $a \in A_2$ of Model (LP3) is

$$\overline{u}_a^k = \begin{cases} 0, & \text{if } \dfrac{\partial L}{\partial u_a}\Big|_{u_a = u_a^k} > 0, \\ 1, & \text{else}, \end{cases} \quad a \in A_2. \tag{40}$$

Let

$$\begin{aligned} x_a^{k+1} &= x_a^k + \frac{1}{k+1}(\overline{x}_a^k - x_a^k), \quad a \in A, \\ y_a^{k+1} &= y_a^k + \frac{1}{k+1}(\overline{y}_a^k - y_a^k), \quad a \in A_1, \\ u_a^{k+1} &= u_a^k + \frac{1}{k+1}(\overline{u}_a^k - u_a^k), \quad a \in A_2. \end{aligned}$$

then, the optimal solution $(\boldsymbol{y}^{k+1}), \boldsymbol{u}^{k+1}, \boldsymbol{x}^{k+1}$ to the subproblem in Step 2 is obtained, which is also the next iteration point.

In brief, the subproblem in Step 2 can be solved efficiently because of exploring the structure of its constraint set. The key computational issue of our algorithm is solving the standard traffic assignment problem. These favorable characteristics indicate that the potential of the algorithm to solve large-scale MNDPs.

5 Test Example

In order to investigate the efficiency of the proposed model and algorithm, we consider two test examples. The first example involves a small network shown in Figure 1, which derives from Gao and Song[3]. The second example involves a large size network, the network of Sioux Falls city, shown in Figure 2.

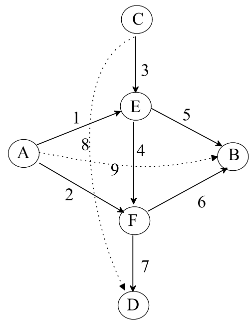

Figure 1 The first test network

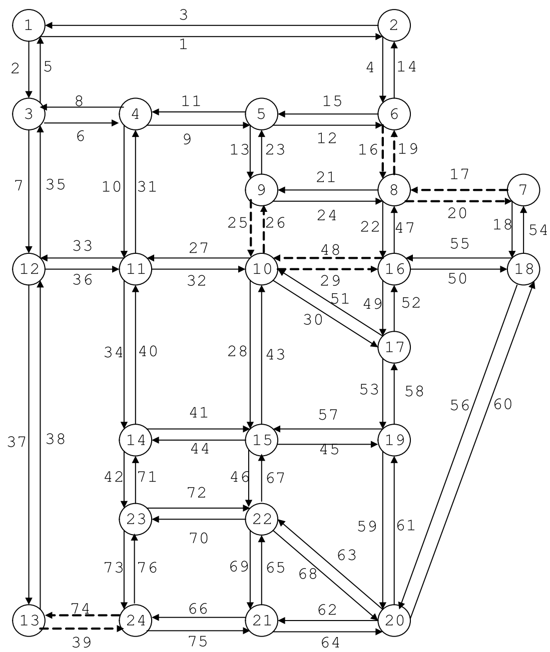

Figure 2 The second test network

The first network has two OD pairs, six nodes and seven links. The current OD demand from nodes A to B is 18 veh/min, from nodes C to D is 6 veh/min. There are three paths AEB, AFB, and AEFB between OD pair (A, B) while there is only one path CEFD between OD pair (C, D). Different from the original example network, there are two candidate links 8 and 9 under consideration to be built (shown in dashed).

The cost function, free-flow travel time, saturation flow for all links are shown in Table 1. The data about links $1 - 7$ are the same as that of Gao and Song (2002), but the data about links 8 and 9 are new given.

For convenience, we abbreviate our algorithm proposed in this paper as PF-AP (penalty function method where the lower level problem is a standard assignment problem), and ab-

Table 1 Input data to the first test network

link	1	2	3	4	5	6	7	8	9
T_a^0	2.0	1.0	2.0	3.0	1.0	2.0	1.0	4.0	3.0
K_a	24.0	30.0	30.0	35.0	24.0	30.0	30.0	10.0	12.0
Cost Function	$t_a(x_a) = T_a^0(1.0 + 0.5(\frac{x_a}{K_a+y_a})^2)$								
d_a	3.0	3.0	3.0	3.0	3.0	3.0	3.0	10	10
Investment Function	$G_a(y_a) = d_a(y_a)^2$								

breviate the penalty function method for conventional bilevel MNDP models as PF-SCAP, in which the lower level problem is a side constrained assignment problem. We solve the example network with the two different algorithms respectively for comparison using MATLAB v6.5 compiler on the computer with Pentium(R) 4, CPU 1.6 GHz. The numerical results are shown in Table 2. It is obvious that PF-AP cost less computational time than PF-SCAP.

Table 2 The comparison results of the example network when $\theta = 1$

Variable	PF-SCAP	PF-AP
y_1	0.0115	0.003
y_2	0.0115	0.009
y_3	0.0000	0.003
y_4	0.0000	0.003
y_5	0.0115	0.009
y_6	0.0115	0.018
y_7	0.0000	0.003
u_1	1, i.e., link 8 added	1, link 8 added
u_2	0, i.e., link 9 not added	0, link 9 not added
Total system cost	85.3171	87.4802
CPU time	963 second	290 second

Note: the upper bound for all y_a is 30, and CPU is 1.6.

The second test network, the network of Sioux Falls city, consists of 24 nodes, 76 links, and 528 O-D trip pairs. Among all the links, the dashed lines in Fig. 2 are candidate links to be built, and the other 66 links are the links to be expanded. The detailed data for the test example including OD demand and link travel cost function (BPR type) for capacity enhancement and construction cost are presented in Table 3. Table 4 lists the optimal results for the second test network obtained by PF-AP method. The number of solved user equilibrium problems in our method is 2978, which almost equals to the number 2700 required by the CNDP method of Ref. [10] when solving CNDP of Sioux Falls city network. From the second test network, we observed that the computational cost of solving MNDP by our method is almost the same as that of solving CNDP though MNDP is far more complicate than CNDP. Fig. 3 presents the changes of the system cost with iteration for the second test network. It can be observed that the system cost decreases as iteration processes.

Table 3 Input data of link cost function and construction cost function for the 2nd test network

Cost Function: $t_a(x_a) = t_a^0(1 + 0.15(\frac{x_a}{K_a})^4); G_a(y_a) = d_a(y_a)^2$

link	t_a^0	K_a	d_a	c_a	link	t_a^0	K_a	d_a	c_a
1 and 3	0.06	25.9002	22		33 and 36	0.06	4.9088	50	
2 and 5	0.04	23.4035	31		34 and 40	0.04	4.8765	44	
4 and 14	0.05	4.9582	29		37 and 38	0.03	25.9002	28	
6 and 8	0.04	17.1105	30		39 and 74	0.04	5.0913		149.6
7 and 35	0.04	23.4035	20		41 and 44	0.05	5.1275	26	
9 and 11	0.02	17.7828	34		42 and 71	0.04	4.9248	36	
10 and 31	0.06	4.9088	48		45 and 57	0.04	15.6508	38	
12 and 15	0.04	4.9480	48		46 and 67	0.04	10.3150	25	
13 and 23	0.05	10.0000	40		49 and 52	0.02	5.2299	54	
16 and 19	0.02	4.8986		124.8	50 and 55	0.03	19.6799	45	
17 and 20	0.03	7.8418		48	53 and 58	0.02	4.8240	50	
18 and 54	0.02	23.4035	50		56 and 60	0.04	23.4035	32	
21 and 24	0.10	5.0502	24		59 and 61	0.04	5.0026	48	
22 and 47	0.05	5.0458	28		62 and 64	0.06	5.0599	39	
25 and 26	0.03	13.9158		50 and 65	63 and 68	0.05	5.0757	34	
27 and 32	0.05	10.0000	36		65 and 69	0.02	5.2299	26	
28 and 43	0.06	13.5120	30		66 and 75	0.03	4.8854	25	
29 and 48	0.05	5.1335		230.4	70 and 72	0.04	5.0000	54	
30 and 51	0.08	4.9935	52		73 and 76	0.02	5.0785	45	

Table 4 Results for the 2nd test network

link	Capacity increment	Candidate link	link	Capacity increment	Candidate link
1 and 3	0.125 and 4.375		33 and 36	19.125 and 17.875	
2 and 5	20 and 4.125		34 and 40	14.75 and 15.875	
4 and 14	0.25 and 15.625		37 and 38	15.375 and 14.25	
6 and 8	14.25 and 14		39 and 74		Added
7 and 35	14 and 18.75		41 and 44	10.00 and 13.50	
9 and 11	24 and 3.75		42 and 71	11.50 and 13.375	
10 and 31	5.5 and 8.875		45 and 57	11.875 and 16.00	
12 and 15	9.5 and 14.125		46 and 67	11.125 and 13.625	
13 and 23	23.625 and 4.125		49 and 52	30.75 and 20.125	
16 and 19		Added	50 and 55	19.875 and 6.625	
17 and 20		Not added	53 and 58	13.125 and 11.375	
18 and 54	6.75 and 3.375		56 and 60	11.125 and 5.375	
21 and 24	7.875 and 24.625		59 and 61	5.50 and 8.75	
22 and 47	21.75 and 9.875		62 and 64	9.25 and 8.25	
25 and 26		Not added	63 and 68	13.375 and 17.375	
27 and 32	14.375 and 8.875		65 and 69	10.125 and 7.00	
28 and 43	6.0 and 3.50		66 and 75	5.00 and 9.625	
29 and 48		Added	70 and 72	13.375 and 14.50	
30 and 51	16.5 and 16.125		73 and 76	6.375 and 10.875	

Total system cost: $SC = \sum_{a \in A_0} t_a(x_a, y_a)x_a + \sum_{a \in A_1} t_a(x_a, u_a)x_a = 54.3396$

Number of solved DUE: 2978.

Note: set the upper bound for all $y_a, a \in A_0$ is 25 and $\phi = 0.001$.

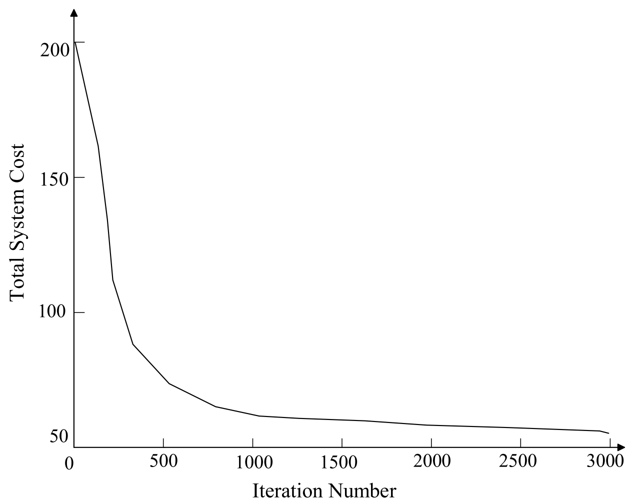

Figure 3 Changes of the system cost with iteration for the second test network

6 Conclusion

MNDP is generally formulated as a mixed-integer nonlinear bilevel programming model, which has long been recognized as one of most difficult yet challenging problems because of its intrinsic complexity. In this paper, we propose a locally convergent algorithm for MNDP by exploring the characteristic of the NDP. First, we reformulate the mixed-integer bilevel programming model for urban transportation mixed network design problem (MNDP) as a continuous version of bilevel programming problem by the continuation method, in which the lower-level problem, comparing with the conventional bilevel DNDP models, is not a link capacitated assignment problem, but a standard user equilibrium assignment problem because of handling the travel cost function artfully. The standard equilibrium assignment problem can be expressed as a nonlinear constraint by virtue of the optimal-value function tool. Therefore, the mixed-integer bilevel programming model for MNDP can be transformed equivalently into a single-level nonlinear optimization problem. Thus, a locally convergent penalty function method is applied to solve this equivalent problem. The descent direction in each step of the inner loop can be found by doing an all-or-nothing assignment. These favorable characteristics indicate the potential of the algorithm to solve large MNDPs. Finally, two transportation network design problem (one is small and the other is large network) are presented to verify the proposed algorithm. The result shows that the proposed algorithm is efficient.

References

[1] H. Yang and M. G. H. Bell, Models and algorithms for road network design: A review and some new developments, *Transport Review*, 1998, **18**(3): 257–278.

[2] Q. Meng, *Bilevel Transportation Modeling and Optimization*, Ph.D. Thesis, the Hong Kong University of Science and Technology, Hong Kong, 2000.

[3] Z. Y. Gao and Y. F. Song, A reserve capacity model of optimal signal control with user equilibrium route choice, *Transportation Research-B*, 2002, **36**(4): 313–323.

[4] A. V. Lim, *Transportation Network Design Problems: An MPEC Approach*, Ph.D. Thesis, the Johns Hopkins University, Maryland, 2002.
[5] S. W. Chiou, Bilevel programming for the continuous transport network design problem, *Transportation Research-B*, 2005, **39**(4): 361–383.
[6] L. J. Leblanc, An algorithm for the discrete network design problem, *Transportation Science*, 1975, **9**(3): 183–199.
[7] H. Poorzahedy and M. A. Turnquist, Approximate algorithms for the discrete network design problem, *Transportation Research-B*, 1982, **16**(1): 45–55.
[8] T. L. Magnanti and R. T. Wong, Network design and transportation planning: Models and algorithms, *Transportation Science*, 1984, **18**(1): 1–55.
[9] H. Yang and M. G. H. Bell, Transportation bilevel programming problems: Recent methodological advances, *Transportation Research-B*, 2001, **35**(1): 1–4.
[10] Q. Meng, H. Yang, and M. G. H. Bell, An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem, *Transportation Research-B*, 2001, **35**(1): 83–105.
[11] Q. Meng, D. H. Lee, H Yang, and H. J. Huang, Transportation network optimization problems with stochastic user equilibrium constraints, *Journal of Transportation Research Record*, 2004, **1882**: 113–119.
[12] Z. Y. Gao, J. J. Wu, and H. J. Sun, Solution algorithm for the bi-level discrete network design problem, *Transportation Research-B*, 2005, **39**(6): 479–495.
[13] H. Yang and Q. Meng, Highway pricing and capacity choice in a road network under a Build-Operate-Transfer Scheme, *Transportation Research-A*, 2000, **34**(3): 207–222.
[14] T. A. Edmunds and J. F. Bard, An algorithm for the mixed-integer nonlinear bilevel programming problem, *Annals of Operations Research*, 1992, **34**(2): 149–162.
[15] K. Shimizu and M. Lu, A global optimization method for the Stackelberg problem with convex functions via problem transformation and concave programming, *IEEE Transaction on Systems, Man, and Cybernetics*, 1995, **25**(12): 1635–1640.
[16] T. Larsson and M. Patriksson, An augmented Lagrangian dual algorithm for link capacity side constrained traffic assignment problems, *Transportation Research-B*, 1995, **29**(6): 433–455.
[17] O. L. Mangasarian and J. B. Rosen, Inequalities for stochastic nonlinear programming problems, *Operations Research*, 1964, **12**(1): 143–154.
[18] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley, New York, 1993.