Available online at www.sciencedirect.com

SCIENCE DIRECT®

Transportation Research Part B 39 (2005) 479–495

TRANSPORTATION RESEARCH PART B

www.elsevier.com/locate/trb

Solution algorithm for the bi-level discrete network design problem

Ziyou Gao *, Jianjun Wu, Huijun Sun

School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

Received 1 May 2002; received in revised form 1 June 2004; accepted 7 June 2004

Abstract

The discrete network design problem deals with the selection of link additions to an existing road network, with given demand from each origin to each destination. The objective is to make an optimal investment decision in order to minimize the total travel cost in the network, while accounting for the route choice behaviors of network users. Because of the computational difficulties experienced with the solution algorithm of nonlinear bi-level mixed integer programming with a large number of 0–1 variables, the discrete network design problem has been recognized as one of the most difficult yet challenging problems in transport. In this paper, at first a traditional bi-level programming model for the discrete network design problem is introduced, and then a new solution algorithm is proposed by using the support function concept to express the relationship between improvement flows and the new additional links in the existing urban network. Finally, the applications of the new algorithm are illustrated with two numerical examples. Numerical results indicate that the proposed algorithm would be efficient in practice.

Keywords: Bi-level programming; Discrete network design problem; Solution algorithm; Support function

* Corresponding author. Tel.: +86 10 51688193; fax: +86 10 51687127.
E-mail addresses: zygao@center.njtu.edu.cn, hui_junsun@sohu.com (Z. Gao).

0191-2615/$ - see front matter
doi:10.1016/j.trb.2004.06.004

1. Introduction

The network design problem (NDP) involves the optimal decision on the expansion of a street and highway system in response to a growing demand for travel. It has emerged as an important area for progress in handling effective transport planning because the demand for travel on the roads is growing at a rate faster than our urban transport systems can ever hope to accommodate, while resources available for expanding the system capacity remain limited. Historically, this problem has been posed in two different forms: a discrete form dealing with the adding new links or roadway segments to an existing road network which is called as the discrete network design problem (DNDP), and a continuous form dealing with the optimal capacity expansion of existing links which is called as the continuous network design problem (CNDP). In whichever form, the objective of NDP is to optimize a given system performance measure such as to minimize total system travel cost, while accounting for the route choice behaviors of network users.

In this paper, the NDP is concerned with the modification of a transportation system by adding new links, i.e. the DNDP. The objective of the DNDP is to make an optimal investment decision in order to minimize the total travel cost in the network, while accounting for the route choice behaviors of the network users. Because of the computational difficulties experienced with the solution algorithm of nonlinear bi-level mixed integer programming with a large number of 0–1 variables, the bi-level discrete network design problem has been recognized as one of the most difficult yet challenging problems in transport (Magnanti and Wong, 1984; Yang and Bell, 1998). In fact, a large number of scholars have investigated the NDP in one way or another over the past three decades. Some of them proposed the continuous network design problems to avoid the complexity of the NDP (Dantzig et al., 1976; Boyce, 1984; Yang and Bell, 1998; Ben-Ayed et al., 1992).

Bruynooghe (1972) discussed an integer programming model for improving or constructing links in an urban network. He included the construction cost in the objective function, as well as travel time, which resulted in a convex, discontinuous objective function (the construction cost for each link includes a "fixed charge"). Lower bounds were computed by relaxing integrality requirements and by underestimating the objective function with a continuous function (i.e. convex envelope). However, no computational results are reported.

Steenbrink (1974) discussed the DNDP, and reviewed the branch-and-bound techniques for solving this problem. He also gave an excellent introduction to modeling the urban road DNDP, including a discussion of techniques for the traffic assignment and Braess' paradox. He then suggested a new approach to the network design problem in which user optimal flows were approximated by system optimal flows. This technique was an iterative decomposition algorithm. However, it does not always converge to the optimum network.

Poorzahedy and Turnquist (1982) developed a bi-level programming formulation to describe the DNDP. In their method, the upper-level was to minimize the total system cost subject to the investments on new links, and the lower-level was the user equilibrium (UE) problem under the condition of fixed demand. In his method, the bi-level programming model for the DNDP was approximated by the lower-level model with the constraints of the upper-level problem, and the heuristic algorithm based on the branch-and-bound algorithm was given. However, In their algorithm, the relationship between the link flows and the proposed projects cannot be expressed explicitly because of transforming the bi-level model into a single one. Therefore, the solution of the model may not be accurate.

Leblanc (1975) presented a branch-and-bound algorithm for solving the upper-level problem, but the bounding step was dependent upon assumption that additional link improvements would always reduce total user cost. He showed that the upper-level objective function value was monotonically decreasing as further link improvements are made, in addition to being a lower bound on the objective function using flows from the lower-level problem. Thus, his algorithm used flows derived from system-optimal (SO) problem to compute true bounds on the objective function in the upper-level problem, avoiding the potential problems caused by Braess' Paradox. Unfortunately, this algorithm was relatively inefficient, at least in part because the bounds were relatively "loose", and becomes computationally prohibitive for large networks.

Other authors have used other methods to simplify the upper-level problem for solution. For example, Boyce et al. (1973) assumed no congestion on network arcs. This eliminates the differences between (UE) and (SO) problems, and made either much easier to solve. This assumption was quite unrealistic, however, because the basic rationale for considering network improvements was often to reduce congestion. Steenbrink (1974), Abdulaal and LeBlanc (1979) and Dantzig et al. (1979) assumed project investment levels were continuous. This clearly simplified the problem because it removed the combinatorial aspects, and made the problem amenable to a number of nonlinear programming algorithms. The computational benefits of using a continuous approximation to the discrete problem can be substantial. For example, Abdulaal and LeBlanc (1979) reported computational results, which indicated that a continuous approximate solution could be obtained at about 20–25% of the cost of obtaining an exact discrete solution. The disadvantage to this approach, however, was that it may be difficult to translate the continuous solution into particular projects which can be implemented. This is especially difficult if the continuous solution indicates relatively small improvements on a large number of links. Because projects often incurred substantial start-up costs, such a solution may not really be practical. Friesz (1985), Magnanti and Wong (1984) and Boyce (1984) conducted comprehensive surveys on the modeling and algorithmic development for these mathematical programming based NDP. Magnanti and Wong (1984) presented a unified view of modeling the DNDP, and proposed a unifying framework for describing a number of algorithms, of which the Lagrange relaxation and dual ascent procedures have been very successful in providing bounds for the special cases of the DNDP. Chen and Alfa (1991) studied the DNDP using a logit-based stochastic incremental traffic assignment approach and presented some computational experience in solving design problems with some road networks.

In this paper, at first a traditional bi-level programming model for the DNDP is introduced, and then a new solution algorithm for bi-level network design problems is proposed by using the support function concept to express the relationship between improvement flows and the new additional links in the existing urban network.

This paper is organized as follows: the next section introduces the basic idea of bi-level programming approach and the bi-level programming model of the DNDP in the transportation. In Section 3, a new solution algorithm for the bi-level DNDP based on the concept of the support functions is proposed. Computational results on two particular networks are presented in Sections 4 and 5 contains conclusions.

2. A bi-level programming model for the discrete transportation network design problem

2.1. The basic idea of the bi-level programming model for the discrete network design problem

The transportation DNDP can be represented as a leader-follower game where the transportation planning departments are leaders, and the users who can freely chose the path are the followers (Boyce, 1984; Yang and Bell, 1998). It is assumed that the transportation planning managers can influence, but cannot control the users' path-choosing behavior. The users make their decision in an user optimal manner. This interaction game can be represented as the following bi-level programming problem.

$$\text{(U0)} \quad \min_{\mathbf{u}} \quad F(\mathbf{x},\mathbf{u})$$
$$\text{s.t.} \quad \mathbf{G}(\mathbf{x},\mathbf{u}) \leqslant \mathbf{0}$$

where $\mathbf{x} = \mathbf{x}(\mathbf{u})$ is implicitly defined by

$$\text{(L0)} \quad \min_{\mathbf{x}} \quad f(\mathbf{x},\mathbf{u})$$
$$\text{s.t.} \quad \mathbf{g}(\mathbf{x},\mathbf{u}) \leqslant \mathbf{0}$$

Obviously, the bi-level programming model consists of two sub-models, (U0) which is defined as an upper-level problem and (L0) which is a lower-level problem. F and $\mathbf{u}$ are the objective function and decision vectors of upper-level decision-makers or system managers, $\mathbf{G}$ is the constraint set of the upper-level decision vectors. f and $\mathbf{x}$ are the objective function and decision vectors of lower-level decision-makers, $\mathbf{g}$ is the constraint set of the lower-level decision vectors. $\mathbf{x} = \mathbf{x}(\mathbf{u})$ is usually called the reaction or response function.

The upper-level describes leader or policy problem and the lower-level model represents follower or user's behavioral problem. In the DNDP, the upper-level problem is to determine an optimal project for adding new links to make the total cost minimum in the range of investments budget formulated by the government. The lower-level problem represents an user equilibrium assignment problem that describes users' path-choosing behavior, and its objective function is to minimize the users' travel cost. A successful investment programming will greatly depend on how to evaluate the reaction function, or in other words, how to predict flow changes in response to an improvement in urban network capacity.

2.2. The lower-level user equilibrium assignment

It is worth emphasizing that the network design problem must be solved with the network flow pattern constrained to be an user equilibrium. In general, improvement of road network characteristics will definitely induce changes in traffic flow over the network. More importantly, addition of a new road segment, or capacity enhancement to a congested network, without considering the response of network users may actually increase network-wide congestion. This well-known phenomenon has been demonstrated by the ostensible Braess' paradox. Therefore, prediction of traffic patterns via a sound behavior model is essential to the network design process.

Traditionally, the DNDP models hypothesize that the demand is given and fixed, and the users' route choice is characterized by the user equilibrium assignment problem. Let $A = A_1 \cup A_2$ be the set of arcs (links), with $A_1 = \{a{:}a = 1,2,\ldots,n\}$ the set of existing arcs (links) which will not be modified, $A_2 = \{a{:}a = n+1, n+2,\ldots,n+m\}$ the set of new candidate arcs (links). R and S are the sets of vertices which are origins and destinations respectively. The UE problem with fixed demand can be formulated as follows (Sheffi, 1985):

$$\text{(L1)} \quad \min\ F = \sum_{a\in A} \int_0^{x_a} t_a(x)\,\mathrm{d}x \tag{1}$$

$$\text{s.t.} \quad \sum_k f_k^{rs} = q_{rs}, \quad \forall r \in R,\ s \in S \tag{2}$$

$$f_k^{rs} \geqslant 0, \quad \forall r \in R,\ s \in S,\ k \in K_{rs} \tag{3}$$

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}, \quad \forall a \in A \tag{4}$$

$$x_a \leqslant M u_a, \quad \forall a \in A_2 \tag{5}$$

x_a is the total flow on link a; K_{rs} is the set of path between r and s; r is the origin node, $r \in R$; s is the destination node, $s \in S$; $t_a(x)$ is the link travel time (or cost) function which is continuously differentiable and convex for fixed u_a; q_{rs} is the total traffic demand between origin r and destination s; f_k^{rs} is the flows on path k connecting r and s; u_a is the binary variables, if link a is built, then $u_a = 1$ and otherwise $u_a = 0$; M is the arbitrarily positive constant; $\delta_{a,k}^{rs}$ is the path/link incidence variables.

In this model, the users at the lower-level are assumed to follow the user-equilibrium principle of Wardrop under the given network. Constraint (2)–(4) are definitional, nonnegativity and conservation of the flow constraints. Constraint (5) prohibits the flow on any proposed link that is not actually constructed. Here, M is an arbitrarily positive constant. If $u_a = 0$, then x_a cannot be positive; But if $u_a = 1$, then x_a can be as large as desired.

2.3. The upper-level optimization problem

In addition to the aforementioned alternative route choice models, the NDP can be formulated with different forms of decision variables and objective functions. The specific decision variables and objective functions would depend on the characteristics of the particular problem of interest. The DNDP with mixed decision variables (both the discrete and continuous decision variables) deals with the addition of new links to a transportation network in this paper which is particularly sensible for formation of a new road system. The upper-level for the discrete transportation network design problem can be expressed as follows (Boyce, 1984; Magnanti and Wong, 1984; Yang and Bell, 1998):

$$\text{(U1)} \quad \min\ Z = \sum_{a\in A_1} x_a t_a(x_a) + \sum_{a\in A_2} x_a t_a(x_a) \tag{6}$$

$$\text{s.t.} \quad \sum_{a\in A_2} c_a u_a \leqslant B \tag{7}$$

$$u_a = 0 \text{ or } 1 \quad \forall a \in A_2 \tag{8}$$

where **x** is the implicitly function of the **u** which may be obtained by solving the lower-level problem; c_a is the construction cost associated with project link a; B is the total investment budget.

The network planners at the upper-level are assumed to make the decisions about addition links and investments in order to minimize the total cost. Constraint (7) ensures that the total construction cost will not exceed the total budget. Constraint (8) is the binary restriction of the decision variables.

3. The solution algorithm for the bi-level problem

In general, it is difficult to solve the bi-level programming problem with optimization algorithms. One of reasons is that the bi-level programming problem is a NP-hard problem. Ben-Ayed et al. (1988) had studied this problem deeply, they pointed that even a very simple bi-level problem such as the bi-level linear programming problem is still a NP-hard problem. The nonconvexity is another reason that results in the complex of the solution algorithm, even the upper and lower-level problem are both convex, the whole bi-level problem is possible to be a nonconvex problem. The nonconvexity nature indicates that even we can find the solution of the bi-level problem, it is usually local optimum not global optimum, There are a part of analytic work on the mixed integer bi-level programming problem with exact algorithms (Bard and Moore, 1990; Bard and Moore, 1992; Edmunds and Bard, 1992; Bard, 1998). However, most of these results deal with the linear objective functions. Although Edmunds and Bard provided (1992) an effective algorithm for the mixed-integer nonlinear bi-level programming model, the model required a convex quadratic objective function in the lower-level problem. The objective functions of the upper and lower-level problem in this paper are both nonlinear, obviously, all of the current results do not fit for them. Moreover, because of the computational difficulties experienced with the solution algorithm for the bi-level DNDP with a large number of 0–1 variables, the bi-level DNDP has been recognized as one of the most difficult yet challenging problems in transport (Magnanti and Wong, 1984; Yang and Bell, 1998).

The difficulty in solving the bi-level programming problem presented in this paper lies in how to evaluate the equilibrium flow **x**(**u**) for the project **u**, which is the implicitly function defined by the lower-level user path-choosing equilibrium problem. Although many solution algorithms for the bi-level model with only continuous variables have been developed, such as the sensitivity analysis based algorithm (SAB) (Kim, 1990; Yang and Yagar, 1994; Wong and Yang, 1997; Chiou, 1999; Gao and Song, 2002; Gao et al., 2004). However, the SAB method cannot be applied to solve the proposed bi-level programming model for the DNDP since there exist the mixed discrete and continuous variables in this model. Therefore, a new efficient algorithm is proposed in the following to solve the proposed DNDP.

The common train of thought for solving the bi-level problem is to obtain the optimal solutions of the lower-level for fixed **u** at first and then find the relationship between **x** and **u**. Thus we can substitute the lower variables **x** represented by the function of **u** into the upper-level which transforms the upper-level objection into the form only with variables **u**. However, generally speaking, it is difficult to obtain the relationship between variables **x** and **u** for the DNDP, since it could not use the SAB method directly as CNDP (Yang and Bell, 1998, etc.). Consider model (U) and (L), we know that the forms of their objective functions have some positive connections. In addition, there are not variables **x** in constraint (7). Therefore, we can represent the upper-level objective function with the objective function and the constraints including variables **u** of the lower-level problem. Thus, based on this idea, we will adopt the concept of the support functions in generalized benders decomposition (GBD) method to describe the relationship between flows and addition link projects (Floudas, 1995).

3.1. The basic idea of GBD for mixed-integer nonlinear programming

At first, we consider the following general mixed nonlinear integer programming problem (P1):

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & f(\mathbf{x},\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x},\mathbf{y}) \leqslant \mathbf{0} \\ & \mathbf{x} \in \mathbf{X} \in \Re^n \\ & \mathbf{y} \in \mathbf{Y} = \{0,1\}^q \end{aligned} \tag{9}$$

where $\mathbf{X} \in \Re^{\mathbf{n}}$ is a nonempty, convex set and the functions

$$f : \Re^n \times \Re^q \to \Re$$

$$\mathbf{g} : \Re^n \times \Re^q \to \Re^p$$

are convex for each fixed $\mathbf{y} \in \mathbf{Y} = \{0,1\}^q$.

3.1.1. The primal problem

The primal problem results from fixing the **y** variables to a practical 0–1 combination, which we denote as $\mathbf{y}^n$ where n stands for the iteration counter. Thus, the formulation of problem (P1) at iteration n be P(n):

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x},\mathbf{y}^n) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x},\mathbf{y}^n) \leqslant \mathbf{0} \\ & \mathbf{x} \in \mathbf{X} \end{aligned} \tag{10}$$

Let the solution of P(n) provides information of $\mathbf{x}^n$ and $\boldsymbol{\omega}^n$ which are the vectors of the optimal solution and associated Lagrange multipliers for the inequality constraints of problem P(n).

3.1.2. The master problem

Problem (P1) can be written as (P2):

$$\begin{aligned} \min_{\mathbf{y}} \min_{\mathbf{x}} \quad & f(\mathbf{x},\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x},\mathbf{y}) \leqslant \mathbf{0} \\ & \mathbf{x} \in \mathbf{X} \in \Re^n \\ & \mathbf{y} \in \mathbf{Y} = \{0,1\}^q \end{aligned} \tag{11}$$

Let us define $v(\mathbf{y})$ as

$$
\begin{aligned}
v(\mathbf{y}) = \min_{\mathbf{x}} \quad & f(\mathbf{x},\mathbf{y}) \\
\text{s.t.} \quad & \mathbf{g}(\mathbf{x},\mathbf{y}) \leqslant \mathbf{0} \\
& \mathbf{x} \in \mathbf{X}
\end{aligned} \tag{12}
$$

Note that $v(\mathbf{y})$ is parametric in the variables $\mathbf{y}$ from its definition corresponds to the optimal value of problem (P1) for fixed $\mathbf{y}$. Let us define the set $\mathbf{V}$ as

$$\mathbf{V} = \{\mathbf{y} : \mathbf{g}(\mathbf{x},\mathbf{y}) \leqslant \mathbf{0}, \quad \text{for some } \mathbf{x} \in \mathbf{X}\}$$

Then problem (P2) can be rewritten as (P3):

$$
\begin{aligned}
\min_{\mathbf{y}} \quad & v(\mathbf{y}) \\
\text{s.t.} \quad & \mathbf{y} \in \mathbf{Y} \cap \mathbf{V}
\end{aligned} \tag{13}
$$

The dual representation of $v(\mathbf{y})$ will be in term of the pointwise infimum of a collection of functions that support it, and it is described in the following theorem due to Geoffrion (1972).

Theorem 1. (Dual of $v(\mathbf{y})$)

$$
v(\mathbf{y}) = \begin{bmatrix} \inf\limits_{\mathbf{x}} f(\mathbf{x},\mathbf{y}) \\ \text{s.t.} \quad \mathbf{g}(\mathbf{x},\mathbf{y}) \leqslant \mathbf{0} \\ \mathbf{x} \in \mathbf{X} \end{bmatrix} = \left[\sup_{\boldsymbol{\omega} \geqslant 0} \inf_{\mathbf{x} \in \mathbf{X}} L(\mathbf{x},\mathbf{y},\boldsymbol{\omega}) \right], \quad \forall \mathbf{y} \in \mathbf{Y} \cap \mathbf{V} \tag{14}
$$

where $L(\mathbf{x},\mathbf{y},\boldsymbol{\omega}) = f(\mathbf{x},\mathbf{y}) + \boldsymbol{\omega}^{\mathrm{T}}\,\mathbf{g}(\mathbf{x},\mathbf{y})$

Substituting (14) for $v(\mathbf{y})$ into problem (P3), which is equivalent to (P2):

$$\min_{\mathbf{y} \in \mathbf{Y}} \sup_{\boldsymbol{\omega} \geqslant 0} \inf_{\mathbf{x} \in \mathbf{X}} L(\mathbf{x},\mathbf{y},\boldsymbol{\omega}) \tag{15}$$

Using the definition of supremum as the lowest upper bound and introducing a scalar μ_B we obtain (M):

$$
\begin{aligned}
\min_{\mathbf{y} \in \mathbf{Y}, \mu_B} \quad & \mu_B \\
\text{s.t.} \quad & \mu_B \geqslant \inf_{\mathbf{x} \in \mathbf{X}} L(\mathbf{x},\mathbf{y},\boldsymbol{\omega}), \quad \forall \boldsymbol{\omega} \geqslant 0
\end{aligned} \tag{16}
$$

where $L(\mathbf{x},\mathbf{y},\boldsymbol{\omega}) = f(\mathbf{x},\mathbf{y}) + \boldsymbol{\omega}^{\mathrm{T}}\,\mathbf{g}(\mathbf{x},\mathbf{y})$ which is called the master problem and denoted as (M).

If we assumed that the optimum solution of $v(\mathbf{y})$ in (12) is bounded for all $\mathbf{y} \in \mathbf{Y} \cap \mathbf{V}$, then we can replace the infimum with a minimum. Subsequently, the master problem will be as follows:

$$
\begin{aligned}
\min_{\mathbf{y} \in \mathbf{Y}, \mu_B} \quad & \mu_B \\
\text{s.t.} \quad & \mu_B \geqslant \min_{\mathbf{x} \in \mathbf{X}} L(\mathbf{x},\mathbf{y},\boldsymbol{\omega}), \quad \forall \boldsymbol{\omega} \geqslant 0
\end{aligned}
$$

Generally, for a given $\mathbf{y}$, there are two cases of ① feasible problem and ② infeasible problem for (P1). However, for the lower-level problem of the DNDP proposed in this paper, we are sure to obtain the optimal solution of the flows, so we only consider the first condition: the feasible problem.

Then $\xi^n(\mathbf{y},\boldsymbol{\omega}^n)$ can be interpreted as the support functions of $v(\mathbf{y})$ at iteration n as follows:

$$\xi^n(\mathbf{y},\boldsymbol{\omega}^n) = \min_{\mathbf{x}\in\mathbf{X}} L(\mathbf{x},\mathbf{y},\boldsymbol{\omega}^n) = \min_{\mathbf{x}\in\mathbf{X}}\{f(\mathbf{x},\mathbf{y}) + \boldsymbol{\omega}^{n^T}\mathbf{g}(\mathbf{x},\mathbf{y})\} \tag{17}$$

where $\xi^n(\mathbf{y},\boldsymbol{\omega}^n)$ is the form of the support function at iteration n, namely, it is the function expression of variables $\mathbf{y}$ when $\mathbf{x} = \mathbf{x}^n$ and $\boldsymbol{\omega} = \boldsymbol{\omega}^n$ ($\xi(\mathbf{y})$ is a support function of $v(\mathbf{y})$ at point $\mathbf{y_0}$ if and only if $\xi(\mathbf{y}_0) = v(\mathbf{y_0})$ and $\xi(\mathbf{y}) \leqslant v(\mathbf{y})$, $\forall \mathbf{y} \neq \mathbf{y}_0$). The determination of the support functions cannot be achieved in general, since these are parametric functions of $\mathbf{y}$. Although a number of special cases for which the support functions can be obtained explicitly as functions of the $\mathbf{y}$ variables (Floudas, 1995), these methods can not be applied to the problem in this paper directly. Therefore, we propose a new algorithm to solve the DNDP.

3.2. The algorithm of bi-level programming for DNDP

If we can transform Eq. (14) which corresponds to the optimal value of the lower-level problem for fixed $\mathbf{y}$ into the value of the objective function of the upper-level problem, then the objective function of the upper-level can be substituted by Eq. (14) only with the variables of $\mathbf{y}$ which corresponds to the variables $\mathbf{u}$ in the upper-level model (U1). Therefore, the upper-level model is easy to solve.

Therefore, the objective function and the constraints of the lower-level problem (L1) can be written as

$$\begin{aligned} \min \quad & F(\mathbf{x},\mathbf{u}(\mathbf{x})) = \sum_{a\in A_1}\int_0^{x_a} t_a(x)\,\mathrm{d}x + \sum_{a\in A_2}\int_0^{x_a} t_a(x)\,\mathrm{d}x \\ \text{s.t.} \quad & \sum_{k\in K_{rs}} f_k^{rs} - q_{rs} = 0, \quad \forall r\in R,\ s\in S \end{aligned} \tag{18}$$

$$x_a - Mu_a \leqslant 0, \quad \forall a \in A_2$$

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

where $\mathbf{u}$ and $\mathbf{x}$ are the upper and lower-level variables respectively.

If we define $\mathbf{X} = \{\sum_k f_k^{rs} - q_{rs} = 0, f_k^{rs} \geqslant 0,\ \forall k\in K_{rs},\ r\in R,\ s\in S\}$, then we can obtain the support functions $\xi^n(\mathbf{u},\boldsymbol{\omega}^n)$ for the lower-level model (18) at iteration n as follows:

$$\begin{aligned} \xi^n(\mathbf{u},\boldsymbol{\omega}^n) &= \min_{\mathbf{x}\in\mathbf{X}}\left\{\sum_{a\in A}\int_0^{x_a(\mathbf{f})} t_a(x)\,\mathrm{d}x + \sum_{a\in A_2}\omega_a^n(x_a - Mu_a)\right\} \\ &= \min_{\mathbf{x}\in\mathbf{X}}\left\{\sum_{a\in A_1}\int_0^{x_a(\mathbf{f})} t_a(x)\,\mathrm{d}x + \sum_{a\in A_2}\int_0^{x_a(\mathbf{f})} t_a(x)\,\mathrm{d}x + \sum_{a\in A_2}\omega_a^n(x_a - Mu_a)\right\} \end{aligned} \tag{19}$$

where $x_a(\mathbf{f})$ expresses Eq. (4), and $\xi^n(\mathbf{u}, \boldsymbol{\omega}^n)$ is the function expression of the upper-level variables $\mathbf{u}$ when $\mathbf{x} = \mathbf{x}^n$ and $\boldsymbol{\omega} = \boldsymbol{\omega}^n$.

Based on Eq. (19), in order to simplify the computations of the algorithm, $\xi^n(\mathbf{u}, \boldsymbol{\omega}^n)$ can be rewritten as follows:

$$\xi^n(\mathbf{u}, \boldsymbol{\omega}^n) = -\sum_{a\in A_2} \omega_a^n M u_a + \min_{\mathbf{x}\in\mathbf{X}} \left\{ \left(\sum_{a\in A_1} \int_0^{x_a(\mathbf{f})} t_a(w) + \sum_{a\in A_2} \int_0^{x_a(\mathbf{f})} (t_a(w) + \omega_a^n) \right) \mathrm{d}w \right\} \tag{20}$$

where ω_a^n are the optimal Lagrange multipliers for the inequality constraints at iteration n for fixed u_a. Based on problem (P2), for fixed $\mathbf{u}$ given by the upper level, $\xi^n(\mathbf{u}, \boldsymbol{\omega}^n)$ can represent the optimal function value of the lower-level problem at iteration n.

Therefore, in view of the relationship between UE and SO, we can represents the support functions $\bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n)$ of the upper-level objective function (system optimal) at iteration n, which is the optimal value of the upper-level objective function at iteration n for fixed $\mathbf{u}$.

$$\bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n) = \xi^n(\mathbf{u}, \boldsymbol{\omega}^n) + \min_{\mathbf{x}\in\mathbf{X}} \sum_{a\in A} \int_0^{x_a(\mathbf{f})} \left(w \frac{\mathrm{d}t_a(w)}{\mathrm{d}w} \right) \mathrm{d}w \tag{21}$$

Then, $\bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n)$ can be expressed as

$$\begin{aligned} \xi'^n(\mathbf{u}, \boldsymbol{\omega}^n) &= -\sum_{a\in A_2} \omega_a^n M u_a + \min_{\mathbf{x}\in\mathbf{X}} \left\{ \sum_{a\in A_1} \int_0^{x_a(\mathbf{f})} \left[t_a(w) + w \frac{\mathrm{d}t_a(w)}{\mathrm{d}w} \right] \mathrm{d}w \right. \\ &\quad \left. + \sum_{a\in A_2} \int_0^{x_a(\mathbf{f})} \left(t_a(w) + w \frac{\mathrm{d}t_a(w)}{\mathrm{d}w} \right) \mathrm{d}w + \sum_{a\in A_2} \int_0^{x_a(\mathbf{f})} \omega_a^n \mathrm{d}w \right\} \\ &= -\sum_{a\in A_2} \omega_a^n M u_a + \min_{\mathbf{x}\in\mathbf{X}} \left\{ \sum_{a\in A_1} t_a(x_a) x_a + \sum_{a\in A_2} (t_a(x_a) + \omega_a^n) x_a \right\} \end{aligned} \tag{22}$$

where only u_a is the variable.

Let

$$\bar{\xi}^n(\mathbf{u}, \omega^n) = -\sum_{a\in A_2} \omega_a^n M u_a + \min_{\mathbf{x}\in\mathbf{X}} \left\{ \sum_{a\in A_1} t_a(x_a) x_a + \sum_{a\in A_2} (t_a(x_a) + \omega_a^n) x_a \right\} \tag{23}$$

Obviously, the second term of the right side in Eq. (23) can be treated as a system optimal problem and can be solved directly with the Frank–Wolfe algorithm for SO (Sheffi, 1985).

Then the master problem for the upper-level model can be written as

$$\begin{aligned} &\min_{\mathbf{u}\in U, \mu_B} \quad \mu_B \\ &\text{s.t.} \quad \mu_B \geqslant \bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n) \quad n = 1, 2, \ldots, N \end{aligned} \tag{24}$$

where $U = \{u : \sum_{a\in A_2} c_a u_a \leqslant B, u_a = 0 \text{ or } 1, \ \forall a \in A_2\}$ and μ_B is the lower bound of the upper level problem. N is the times of the iteration.

The algorithm for the bi-level programming model is summarized as follows:

Step 1: Determine an initial set of projects pattern $\mathbf{u}^1 \in U$, and set $n = 1$. Solve the lower-level problem (L1) for given $\{\mathbf{u}^1\}$ and obtain $\{\mathbf{x}^1\}$, $\{\boldsymbol{\omega}^1\}$ and the support functions $\xi^1(\mathbf{u}, \boldsymbol{\omega}^1)$, $\bar{\xi}^1(\mathbf{u}, \boldsymbol{\omega}^1)$. Set the current upper bound $\text{UBD} = \bar{\xi}^1(\mathbf{u}^1, \boldsymbol{\omega}^1)$. Select the convergence tolerance $\varepsilon \geqslant 0$.

Step 2: Solve the relaxed master problem

$$\begin{aligned} \min_{\mathbf{u}\in U, \mu_B} \quad & \mu_B \\ \text{s.t.} \quad & \mu_B \geqslant \bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n), \quad n = 1, 2, \ldots, N \end{aligned}$$

Let $(\hat{\mathbf{u}}, \hat{\mu}_B)$ be an optimal solution of the aforementioned relaxed master problem. $\hat{\mu}_B$ is a lower bound of problem (U1); that is, the current bound is $\text{LBD} = \hat{\mu}_B$. If UBD–LBD $\leqslant \varepsilon$, then terminate.

Step 3: Solve the lower-level problem for $\mathbf{u} = \hat{\mathbf{u}}$. Then we obtain an optimal solution $\hat{\mathbf{x}}$, the optimal multiplier vectors $\hat{\boldsymbol{\omega}}$ and the support function $\bar{\xi}(\hat{\mathbf{u}}, \hat{\boldsymbol{\omega}})$. Update the upper bound $UBD = \min\{UBD, \bar{\xi}(\hat{\mathbf{u}}, \hat{\boldsymbol{\omega}})\}$. If $UBD - LBD\varepsilon$, then terminate. Otherwise, set $n = n + 1$, $\boldsymbol{\omega}^n = \hat{\boldsymbol{\omega}}$. Return to step 2.

Remark 1. Note that the relaxed master problem (see step 2) in the first iteration will have as a constraint one support function that corresponds to feasible primal.

$$\begin{aligned} \min_{\mathbf{u}\in U, \mu_B} \quad & \mu_B \\ \text{s.t.} \quad & \mu_B \geqslant \bar{\xi}^1(\mathbf{u}, \boldsymbol{\omega}^1) \end{aligned} \tag{25}$$

In the second iteration, the relaxed problem will feature two constraints.

$$\begin{aligned} \min_{\mathbf{u}\in U, \mu_B} \quad & \mu_B \\ \text{s.t.} \quad & \mu_B \geqslant \bar{\xi}^1(\mathbf{u}, \boldsymbol{\omega}^1) \\ & \mu_B \geqslant \bar{\xi}^2(\mathbf{u}, \boldsymbol{\omega}^2) \end{aligned} \tag{26}$$

Note that in this case, the relaxed master problem (25) will have a solution that is greater or equal to the solution of problem (26). Therefore, we can see that the sequence of lower bounds that is created from the solution of the relaxed master problem is nondecreasing.

Remark 2. Note that since the upper bounds are produced by fixing the **u** variables to different 0–1 combinations, there is no reason for the upper bounds to satisfy any monotonicity property. If we consider however the updated upper bounds, then the sequence for the updated upper bounds is monotonically nonincreasing since by their definition we always keep the best upper bound.

Remark 3. The termination criterion for the algorithm in this paper is based on the difference between the updated upper bound and the current lower bound. If this difference is less than or equal to a prespecified tolerance $\varepsilon \geqslant 0$ then we terminate.

Remark 4. In the proposed algorithm, we need to obtain the optimal multipliers ω_a^n at each iteration for the lower problem. For fixed u_a, the multipliers in constraint (5), an inequality constraints, can be solved by the multiplier method (Bazaraa et al., 1993) as follows:

Define the augmented Lagrange function for the lower-level model, we have

$$F(\mathbf{x},\boldsymbol{\omega},\sigma)=\sum_{a\in A}\int_0^{x_a} t_a(w)\,\mathrm{d}w+\frac{1}{2\sigma}\sum_{a\in A_2}\left\{[\max(0,\omega_a-\sigma(Mu_a-x_a))]^2-\omega_a^2\right\} \tag{27}$$

where σ is the penalty factor, and ω_a is the multiplier for inequality which must be revised with the following equation in the course of iteration.

$$\omega_a^{(k+1)}=\max\left(0,\omega_a^{(k)}-\sigma\left(Mu_a^k-x_a^{(k)}\right)\right) \tag{28}$$

It can be proved that $\omega_a^{(k+1)}$ will converge to the optimal multiplier $\bar{\omega}_a$ (Bazaraa et al., 1993). Therefore, the lower-level problem can be solved by the known methods such as the augmented Frank–Wolfe method (Patriksson, 1994) or Origin-based algorithm (Bar-Gera, 1999) directly. Thus, if we use the augmented Frank–Wolfe method to solve the lower-level problem for given $\mathbf{u}^n$, then we can obtain the optimal $\mathbf{x}^n$ and the optimal multiplier $\boldsymbol{\omega}^n$ simultaneously.

By the finite convergence for the GBD algorithm in Geoffrion (1972), we can obtain the following theorem:

Theorem 2. (Finite convergence) *If* $\mathbf{u}$ *is a discrete set, then the proposed algorithm in this paper terminates in a finite number of iterations for any given* $\varepsilon>0$ *and even for* $\varepsilon = 0$.

4. Numerical examples

In this section, to illustrate the applications of the model and algorithm proposed in this paper, we present two numerical examples for optimal network design problem. One is a simple example, and another is the well-known Sioux–Falls test network.

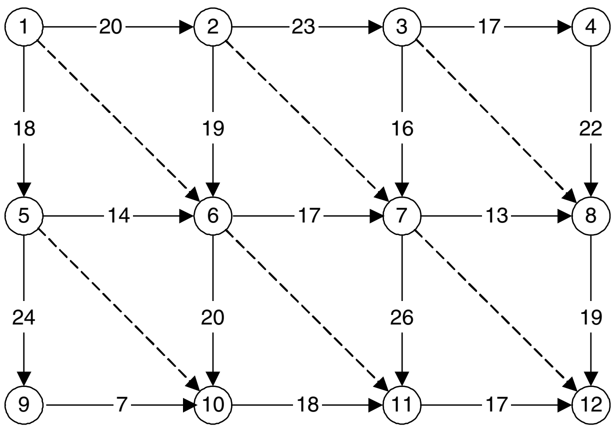

Fig. 1. The test network. Solid lines indicate existing links, and dashed lines potential new links.

Table 1
The parameters of new links

Index	1	2	3	4	5	6
(i,j)	(1, 6)	(5, 10)	(2, 7)	(6, 11)	(3, 8)	(7, 12)
t_{ij}	19	25	30	32	21	28
Cost	7	12	7	15	11	18

Table 2
Movement process of the upper and lower bounds, iteration numbers and results (1 stands for the new link should be built and 0 should not)

Budget	Upper bound $\bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n)$	lower bound μ_B	Results
10	5499.583806	3999.583806	1 0 0 0 0 0
	4076.597932	3999.583806	0 0 1 0 0 0
	4076.597932	4076.597932	1 0 0 0 0 0
	4076.597932		
	Result: 4076.597932; Iteration number: 3		
20	5499.583806	2499.583806	0 1 1 0 0 0
	5223.373259	2499.583806	1 0 0 0 1 0
	3967.851301	3723.373259	1 0 1 0 0 0
	3952.524772	3723.373259	0 0 1 0 1 0
	3952.524772	3723.373259	1 1 0 0 0 0
	3952.524772	3952.524772	1 0 1 0 0 0
	3952.524772		
	Result: 3952.524772; Iteration number: 6		
30	5499.583806	999.583806	1 1 1 0 0 0
	3928.412936	2428.412936	1 0 1 0 1 0
	3823.385857	2428.412936	0 1 1 0 1 0
	3823.385857	2428.412936	1 0 1 1 0 0
	3772.234736	2499.583806	1 0 0 0 0 1
	3590.502069	2668.584294	0 1 0 1 0 0
	2668.584294	2668.584294	1 0 0 0 0 1
	2668.584294		
	Result: 2668.584294; Iteration number: 7		
40	5499.583806	0.000000	1 0 1 1 1 0
	3636.678035	999.583806	0 1 1 0 0 1
	3636.678035	2136.678035	1 0 0 1 0 1
	2524.586692	2136.678035	1 0 0 0 1 1
	2524.586692	2136.678035	0 1 0 1 1 0
	2524.586692	2319.345791	1 1 0 0 0 1
	2524.586692	2136.678035	1 1 1 0 1 0
	2524.586692	2319.345791	0 0 1 1 0 1
	2524.586692	2524.586692	1 0 0 1 0 1
	2524.586692		
	Result: 2524.586692; Iteration number: 9		

(*continued on next page*)

Table 2 (*continued*)

Budget	Upper bound $\bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n)$	lower bound μ_B	Results
50	5499.583806	0.000000	1 1 0 0 1 1
	2559.447791	1059.447791	0 1 1 0 1 0
	2559.447791	590.502069	1 0 1 1 0 1
	2404.815485	2090.502069	1 0 1 1 1 0
	2404.815485	2090.502069	0 1 0 1 0 1
	2404.815485	2090.502069	1 1 1 0 0 1
	2404.815485	2136.678035	1 1 1 1 0 0
	2404.815485	2136.678035	1 0 1 0 1 1
	2404.815485	2241.038685	1 1 0 1 1 0
	2404.815485	2404.815485	1 0 1 1 0 1
	2404.815485		
	Result: 2404.815485; Iteration number: 10		
60	5499.583806	0.000000	1 0 1 0 1 1
	2413.750330	0.000000	0 1 0 1 1 1
	2413.750330	843.189223	1 1 1 1 0 0
	2413.750330	913.750330	1 0 1 1 1 1
	2281.727646	913.750330	1 1 1 0 1 1
	2281.727646	2241.038685	1 1 1 1 1 0
	2281.727646	2241.038685	1 1 1 1 0 1
	2281.727646	2281.727646	1 0 1 1 1 1
	2281.727646		
	Result: 2281.727646; Iteration number: 8		
70	5499.583806	0.000000	0 1 0 1 1 1
	3843.189223	843.189223	1 1 1 0 1 0
	3799.268780	843.189223	1 1 1 1 0 1
	2380.739275	880.739275	1 1 1 1 1 1
	2256.961857	2256.961857	1 1 1 1 1 1
	2256.961857		
	Result: 2256.961857; Iteration number: 5		

Example 1. Note that the data here are assumed for simplification of computation. In practice, the data should be identified from observations of the actual network design projects. The discrete network is shown in Fig. 1 with only one O–D pair $(1, 12)$ and the O–D demand is assumed 20. There are a total of six involve construction of new links, these links are numbered in Fig. 1 by dashed lines.

The form of cost functions is used as follows:

$$t_{ij}(x_{ij}) = t_{ij}^0 + 0.008x_{ij}^4$$

t_{ij}^0 of existing links is labeled in Fig. 1. And the parameters of the new links are given in Table 1. In Table 2, we provide the iteration results and the process of the movement of the upper and lower bounds with different budgets. In the process of obtaining the Lower bound, the approximation method is adopted. Therefore, it need much time to converge. It can be known, in this example, with the increase of the budgets, the more links are built, the less system cost becomes.

In order to test the convergence of the algorithm, we use different initial points with the budgets of 40, the results are same, which can partly illustrate the convergence of the algorithm. Therefore, we can believe that the solution may be close to the global solution of the problem. Table 3 gives the iteration results with different initial points.

Example 2. To compare the efficiency of the procedures outlined above with the previous algorithm, the well-known Sioux–Falls network has been chosen. The original network includes 24 vertices and 76 links. Links in the network are in pairs representing two-way traffic movements. All the data of the test network are the same as Leblanc (1975). There are total of five candidate projects and a budget of \$3,000,000 was assumed for the problem.

The result obtained by the proposed algorithm is $(1, 0, 1, 1, 0)$ as shown in Table 4. with different initial solutions, we obtain the same optimal solutions. So projects 1, 3 and 4 should be undertaken, which is the same as the result of Leblanc (1975). Only two iteration steps are needed for this example, which suggests the proposed algorithm is efficient for this DNDP.

Table 3
The iteration result with different initial solution for the budget 40

Initial solution	Budget 40 Results
100011	100101
011001	100101
100010	100101
100000	100101

Table 4
The result with different initial solutions

Different initial solutions	Upper bound $\bar{\xi}^n(\mathbf{u}, \boldsymbol{\omega}^n)$	Lower bound μ_B	Results
$(0, 0, 0, 0, 0)$	3.0686×10^9	4.3408×10^8	11100
	4.3408×10^8	4.3408×10^8	10110
	4.3408×10^8		
	Result: 4.3408×10^8; Iteration number: 2		
$(0, 0, 0, 0, 1)$	2.9008×10^9	4.3200×10^8	11100
	4.3408×10^8	4.3408×10^8	10110
	4.3408×10^8		
	Result: 4.3408×10^8; Iteration number: 2		

5. Conclusion

In this paper, we introduce a bi-level programming model and propose a new solution algorithm for the DNDP in which the effects of new adding link projects on the equilibrium flows are taken into account. Based on the concept of the support functions, the upper-level programming of the DNDP can be transformed to the usual nonlinear programming problem, so that many standard constrained optimization techniques can be applied. Numerical results have indicated the efficiency of the technique.

In this paper, we assume that the demand is fixed and do not consider the capacity restrictions of the road. Therefore, it would be interesting to examine the elastic demand and the road capacity. The further work also includes considering the mixed problem of the continuous and discrete network design and some combined assignment problems.

Acknowledgments

The work described in this paper was partly supported by National Outstanding Young Investigator Grant (70225005) of National Natural Science Foundation of China, Teaching & Research Award Program for Outstanding Young Teachers (2001) in Higher Education Institutions of Ministry of Education, Natural Science Foundation of Beijing (9042006) and school foundation of Beijing Jiaotong University, China (2003SM032).

The authors wish to thank three anonymous referees for their helpful comments and suggestions for improvement.

References

Abdulaal, M., LeBlanc, L.J., 1979. Continuous equilibrium network design models. Transportation Research 13B, 19–32.

Bard, J.F., 1998. Practical Bilevel Optimization, Algorithm and Applications. Kluwer Academic Publishers.

Bard, J.F., Moore, J.T., 1990. A branch and bound algorithm for the bilevel programming problem. SIAM, Journal on Scientific Computing 11 (2), 281–292.

Bard, J.F., Moore, J.T., 1992. An algorithm for the discrete bilevel programming problem. Naval Research Logistics 39, 419–435.

Bar-Gera, H., 1999. Origin-Based Algorithm for Transportation Network Modeling. Technical Report 103, National Institute of Statistical Sciences, Chicago, October 1999.

Bazaraa, M.S., Sherali, H.D., Shetty, C.M., 1993. Nonlinear Programming Theory and Algorithms. John Wiley, Sons, New York.

Ben-Ayed, O., Blair, C.E., Boyce, D.E., LeBlanc, L.J., 1992. Construction of a real-world bi-level linear programming model of the highway network design problem. Annals of Operations Research 34, 219–254.

Ben-Ayed, O., Boyce, D.E., Blair, C.E., 1988. A general bilevel programming formulation of the network design problem. Transportation Research 22B, 311–318.

Boyce, D.E., 1984. Urban transportation network equilibrium and design models: Recent achievements and future prospectives. Environment and Planning 16A, 1445–1474.

Boyce, D.E., Farhi, A., Weischedel, R., 1973. Optimal network problem: a branch-and-bound algorithm. Environment and Planning 5, 519–533.

Bruynooghe, M., 1972. An Optimal Method of Choice of Investments in a Transport Network. In: PTRC Proceedings.

Chen, M., Alfa, A.S., 1991. A network design algorithm using a stochastic incremental traffic assignment approach. Transportation Science 25, 215–224.

Chiou, S.W., 1999. Optimization of area traffic control for equilibrium network flows. Transportation Science 33, 279–289.

Dantzig, G.B., Maier, S.F., Lansdowne, Z.F., 1976. Application of Decomposition to Transportation Network Analysis. Control Analysis Corporation, Palo Alto, California, Technical Report No. 1, March 1976.

Dantzig, G.B., Harvey, R.P., Lansdowne, Z.F., Robinson, D.W., Maier, S.F., 1979. Formulating and solving the network design problem by decomposition. Transportation Research 13B, 5–17.

Edmunds, T.A., Bard, J.F., 1992. An Algorithm for the mixed-integer nonlinear bilevel programming problem. Annals of Operations Research 34, 149–162.

Floudas, C., 1995. Nonlinear and mixed-integer optimization fundamentals and applications. Oxford University Press, New York, USA.

Friesz, T.L., 1985. Transportation network equilibrium, design and aggregation: key developments and research opportunities. Transportation Research 19A, 413–427.

Gao, Z.Y., Song, Y.F., 2002. A reserve capacity model of optimal signal control with user-equilibrium route choice. Transportation Research 36B, 313–323.

Gao, Z., Sun, H., Shan, L., 2004. A continuous equilibrium network design model and algorithm for transit systems. Transportation Research 38B, 235–250.

Geoffrion, A.M., 1972. Generalized benders decomposition. Journal of Optimization Theory and Application 10 (4), 237–250.

Poorzahedy, H., Turnquist, M.A., 1982. Approximate algorithms for the discrete network design problem. Transportation Research 16B, 45–55.

Kim, T.J., 1990. Advanced Transport and Spatial Systems Models: Applications to Korea. Springer-Verlag, New York, USA.

Leblanc, L.J., 1975. An Algorithm for the Discrete Network Design Problem. Transportation Science 9, 183–199.

Magnanti, T.L., Wong, R.T., 1984. Network design and transportation planning: models and algorithms. Transportation Science 18, 1–55.

Patriksson, M., 1994. The Traffic Assignment Problem: Models and Methods. VSP, Utrecht, The Netherlands.

Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall, Englewood Cliffs, NJ, USA.

Steenbrink, A., 1974. Transport network optimization in the Dutch integral transportation study. Transportation Research 8B, 11–27.

Wong, S.C., Yang, H., 1997. Reserve capacity of a signal-controlled road network. Transportation Research 31B, 397–402.

Yang, H., Bell, M.G.H., 1998. Models and algorithms for road network design: a review and some new developments. Transport Reviews 18, 257–278.

Yang, H., Yagar, S., 1994. Traffic assignment and traffic control in general freeway-arterial corridor systems. Transportation Research 28B, 463–486.