Optim Eng (2007) 8: 241–257
DOI 10.1007/s11081-007-9015-1

A globally convergent algorithm for transportation continuous network design problem

Ziyou Gao · Huijun Sun · Haozhi Zhang

Published online: 10 July 2007

Abstract The continuous network design problem (CNDP) is characterized by a bilevel programming model, in which the upper level problem is generally to minimize the total system cost under limited expenditure, while at the lower level the network users make choices with regard to route conditions following the user equilibrium principle. In this paper, the bilevel programming model for CNDP is transformed into a single level convex programming problem by virtue of an optimal-value function tool and the relationship between System Optimum (SO) and User Equilibrium (UE). By exploring the inherent nature of the CNDP, the optimal-value function for the lower level user equilibrium problem is proved to be continuously differentiable and its derivative in link capacity enhancement can be obtained efficiently by implementing user equilibrium assignment subroutine. However, the reaction (or response) function between the upper and lower level problem is implicit and its gradient is difficult to obtain. Although, here we approximately express the gradient with the difference concept at each iteration, based on the method of successive averages (MSA), we propose a globally convergent algorithm to solve the single level convex programming problem. Comparing with widely used heuristic algorithms, such as sensitivity analysis based (SAB) method, the proposed algorithm needs not strong hypothesis conditions and complex computation for the inverse matrix. Finally, a numerical example is presented to compare the proposed method with some existing algorithms.

Keywords Continuous network design problem · B-level programming · Method of successive averages · Global convergence

Z. Gao (✉) · H. Sun · H. Zhang
State Key Laboratory of Rail Traffic Control and Safety, School of Traffic and Transportation,
Beijing Jiaotong University, Beijing 100044, Peoples Republic of China
e-mail: zygao@center.njtu.edu.cn

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 roughly classified into two different forms: a discrete form dealing with the additions of new links or roadway segments to an existing road network, and a continuous form dealing with the optimal capacity expansion of existing links. 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 behavior of network users (Yang and Bell 1998). The decisions made by road planners influence 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 the upper level problem is to minimize the total system cost and the lower level problem is to characterize the UE traffic flow pattern.

Due to the intrinsic complexity of model formulation, the NDP has been recognized as one of the most difficult yet challenging problems in transport. In fact, a large number of scholars have investigated the NDP in one way or another over the past two decades (Magnanti and Wong 1984; Friesz 1985; Boyce 1984; Wong and Yang 1997; Yang and Bell 1998; Gao et al. 2005).

Up to date, studies have been overwhelmingly focused on the CNDP and substantial achievements in algorithmic development have been made. Abdulaal and LeBlanc (1979) formulated the CNDP under deterministic user equilibrium (DUE) as a bilevel programming model and the Hook-Jeeves heuristic algorithm was also introduced. Generally, bilevel programming problem is difficult to solve, designing efficient algorithms for CNDP is long recognized to be one of the most challenging problems in transportation.

One class of the existing methods tries to derive a set of equivalent differentiable equations for the DUE assignment problem. The CNDP is then reformulated as a constrained differentiable optimization problem, which can be solved by the existing convergent methods. Tan et al. (1979) expressed the DUE problem by a set of nonlinear and nonconvex, but differentiable constraints in terms of path flow variables. Friesz (1981) extended this result to the multiclass DUE problem. As an application, Friesz et al. (1993) used a simulated annealing approach to solve the multiobjective equilibrium network design problem as a single level minimization problem. Since the number of paths in the networks of a realistic size is huge, this approach can only be suitable for small, hypothetical networks. In view of the fact that the DUE problem can be described by a variational inequality, Dafermos (1980), Marcotte (1983) transferred the CNDP into a single level equivalent differentiable optimization problem. The required constraints involve all the extreme points of the closed convex polyhedron for the feasible acyclic multicommodity flow patterns. It is generally difficult to identify all the extreme points for a polyhedron, and in particular, for a moderately large network problem the constraint set might become huge and intractable. Meng et al. (2001) reformulated the CNDP under the DUE constraints into an equivalent single level continuously differentiable problem by virtue of a marginal function tool. The DUE conditions of the lower level problem are represented by a single constraint in terms of the marginal function, but unfortunately, the single constraint is nonconvex. In their paper, the bilevel model is transformed into a single level nonconvex programming model from which the globally optimal solution is hard to obtain. Meng et al. (2001) solve this equivalent problem by the augmented Lagrangian method and only obtain the locally convergent solution. In this paper, we transform the bilevel programming model into an equivalent convex programming and an approximately global optimal solution to the CNDP can be obtained efficiently.

In view of the solution difficulty, various heuristic algorithms for the CNDP are developed to try to produce acceptable solutions for large problems, without necessarily guaranteeing optimality. The relation between DUE link flow and link capacity enhancement is nonlinear and implicit. The derivative of DUE link flow with respect to link capacity enhancement can be obtained by using the sensitivity analysis method under some strong assumptions (Friesz et al. 1990; Cho 1988; Yang 1995, 1997), which is widely used for solving the CNDP. Various sensitivity analysis-based heuristic algorithms are proposed for the CNDP and relevant problems using the derivative information (Tobin and Friesz 1988; Kim and Suh 1988; Friesz et al. 1990; Yang and Yagar 1994; Gao and Song 2002, *etc.*). Unfortunately, because of the large computation of the inverse of matrix, SAB method cannot be used to solve the large transportation network. Moreover, it needs strong assumptions. In addition, the flow-capacity enhancement relation may not always be differentiable. Another class of heuristic algorithms is the so-called iterative optimization assignment (IOA) algorithms that iteratively solve the upper and lower level optimization problems of the CNDP. Marcotte (1986), Friesz and Harker (1985) and Marcotte and Marquis (1992) conducted detailed performance analyses of this type of algorithm. Furthermore, Suwansirikul et al. (1987) developed an alternative heuristic method called the equilibrium decomposed optimization (EDO) algorithm by approximating the derivative of the objective function in the upper level problem. This approximation requires that the approximated derivatives should have the same sign as the original true derivatives, which is difficult to verify, in particular, for realistically large network problems.

Marcotte and Zhu (1996) and Luo et al. (1996) have obtained some interesting results in optimality conditions and provided some algorithms for a class of general bilevel programming problem in which the lower level problem is described by variational inequalities. Certain exact penalty functions and the corresponding algorithms are investigated and established using the theory of exact penalization for mathematical programs with subanalytic constraints under certain regularity conditions. In addition, some non-numerical algorithms are proposed (Friesz et al. 1993; Cree and Maher 1998).

In this paper, firstly the limitation of the SAB method which is used extensively in solving CNDP (Yang and Bell 1998) is introduced. Then the bilevel programming model for CNDP is transformed into a single level convex programming problem by virtue of a optimal-value function tool and the relationship between SO (the upper level objective function) and UE (the lower level objective function). By exploring the inherent nature of the CNDP, the optimal-value function for the lower level user equilibrium problem is proved to be continuously differentiable and its derivative in link capacity enhancement can be obtained efficiently by implementing a user equilibrium assignment subroutine. However, the reaction (or response) function between the upper and lower level problem is implicit and its gradient is difficult to obtain, so here we approximately express the gradient with the difference concept at each iteration. Based on MSA method, we propose a globally convergent algorithm to solve the single level convex programming problem. Comparing with widely used heuristic algorithms, such as SAB method, the proposed algorithm needs not strong hypothesis conditions and complex computation for the inverse matrix, and its key computational issue is solving the user equilibrium assignment problem with fixed link capacity enhancement and a simple linear programming.

This paper is organized as follows: the next section introduces the basic idea to solve the bilevel programming problems and the bilevel programming model of the CNDP in the transportation. In Sect. 3, firstly, introduce the concept and properties of an optimal-value function. By investigating the characteristics of the lower level problem's optimal-value function, its gradient can be obtained efficiently, then, the bilevel model for CNDP is transformed into a single level convex programming problem and a globally convergent algorithm based on the thought of MSA is proposed. Computational results on a particular network are presented in Sect. 4, and Sect. 5 contains conclusion.

2 A bilevel programming model for the CNDP

2.1 The basic idea of the bilevel programming model

The transportation CNDP can be represented as a leader-follower game where the transportation planning departments are leaders, and the users or travelers who can freely choose 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 a user optimal manner under the given service level of transportation networks. This interaction game can be represented as the following bilevel programming problem.

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

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

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

Obviously, the bilevel programming model consists of two submodels, (U0) which is defined as an upper level problem and (L0) which is a lower level problem. F and

x are the objective function and decision vectors of upper level decision-makers or system managers, **G** and **g** are the constraint sets of the upper level and lower level decision vectors. f and **y** are the objective function and decision vectors of lower level decision-makers. $\mathbf{y} = \mathbf{y}(\mathbf{x})$ 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 CNDP, the upper level problem is to determine an optimal project for improvement link capacities to make the total cost minimum in the range of investments budget formulated by the government. The lower level problem represents a 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 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 CNDP models hypothesize that the demand is given and fixed, and the users' route choice is characterized by the user equilibrium assignment problem. Let A be the set of arcs (links), R and S are the sets of vertices which represent origins and destinations respectively. The UE problem with fixed demand can be formulated as follows (Sheffi 1985):

$$\text{(L)} \quad \min \quad T(\mathbf{x}, \mathbf{y}) = \sum_{a \in A} \int_0^{y_a(\mathbf{x})} t_a(v, x_a) dv \tag{2.1}$$

$$\text{s.t.} \quad \sum_k h_k^{rs} = q_{rs}, \quad \forall r \in R,\ s \in S, \tag{2.2}$$

$$h_k^{rs} \geq 0, \quad \forall r \in R,\ s \in S,\ k \in K_{rs}, \tag{2.3}$$

$$y_a = \sum_r \sum_s \sum_k h_k^{rs} \delta_{a,k}^{rs}, \quad \forall a \in A. \tag{2.4}$$

Notations

y_a: the total flow on link a, $a \in A$;
K_{rs}: the set of path between r and s;
r: the origin node, $r \in R$;
s: the destination node, $s \in S$;
$t_a(\cdot)$: the link travel time (or cost) function which is continuously differentiable and

convex for fixed x_a. Generally, we use the following form for $t_a(\cdot)$

$$t_a(y_a, x_a) = A_a + B_a(y_a/(k_a + x_a))^4$$

where A_a, B_a are parameters, and k_a is the capacity of link a (Sheffi 1985).
q_{rs}: the total traffic demand between origin r and destination s;
h_k^{rs}: flows on path k connecting $r$and s;
x_a: the continuous capacity increase of link a;
$\delta_{a,k}^{rs}$: 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. Constraints (2.2), (2.3) and (2.4) are definitional, non-negativity and conservation of the flow constraints.

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 CNDP deals with the increase of the link capacities to a transportation network. The upper level for the continuous transportation network design problem can be expressed as follows (Yang and Bell 1998):

$$\text{(U)} \quad \min \quad F(\mathbf{x}, \mathbf{y}) = \sum_{a \in A} t_a(y_a(\mathbf{x}), x_a) y_a(\mathbf{x}) \tag{2.5}$$

$$\text{s.t.} \quad \sum_{a \in A} G_a(x_a) \le B \tag{2.6}$$

$$x_a \ge 0, \quad \forall a \in A \tag{2.7}$$

where $\mathbf{y}$ is the implicitly function of the $\mathbf{x}$ which may be obtained by solving the lower level problem; $G_a(x_a)$is the investment function of link $a \in A$; B is the total investment budget. The investment function $G_a(x_a)$ is formulated generally to make the constraint (2.6) convex in practice, such as $G_a(x_a) = 1.5 \cdot d_a \cdot (x_a)^2$, where d_a is the parameter of the investment function.

The network planners of the upper level are assumed to make the decisions about the improvement of links capacities and investments in order to minimize the total cost. Constraint (2.6) ensures that the total investment cost will not exceed the total budget. Constraint (2.7) is the non-negativity of the decision variables.

3 Solution algorithm for the bilevel problem

In spite of the various intriguing attempts to solve the CNDP, these algorithms are unfortunately either incapable of finding the convergent solution or very computationally intensive and impractical for problems of a realistic size.

The difficulty in solving the bilevel programming problem presented in this paper lies in how to evaluate the equilibrium flow $\mathbf{y}(\mathbf{x})$ for the project $\mathbf{x}$, which is the implicitly function defined by the lower level user path-choosing equilibrium problem. Many solution algorithms for the bilevel model with 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; *etc.*). However, the SAB method needs large computation of the inverse of matrix, and is not suitable to solve the large network and also cannot guarantee the convergence. In addition, SAB method needs many strong assumptions, for example, it requires that the lower level functions is second order continuous differentiable and the lower level problem has a unique solution for any fixed upper level variables. Therefore, a new efficient algorithm is proposed in the following to solve the CNDP.

3.1 Optimal-value function of general nonlinear programming problem

At first, we consider the following general nonlinear programming problem with $\mathbf{x} \in X$ being a parameter as follows:

$$w(\mathbf{x}) = \min_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) \tag{3.1a}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x}, \mathbf{y}) \le \mathbf{0} \tag{3.1b}$$

where X is a nonempty, convex set, $\mathbf{y} \in R^q$ is the decision vector and the functions

$$f : R^n \times R^q \to R^1,$$
$$\mathbf{g} : R^n \times R^q \to R^p.$$

The constraint set defined by (3.1b) is denoted by

$$S(\mathbf{x}) = \{\mathbf{y} \in R^q \mid \mathbf{g}(\mathbf{x}, \mathbf{y}) \le \mathbf{0}\}. \tag{3.2}$$

In general, We call the function $w(\mathbf{x})$ the optimal-value function (Shimizu et al. 1997).

In problem (3.1), the constraint sets of $\mathbf{y}$ depend on the parameters. We refer to such a constraint as parametric constraint.

The optimal solution set of problem (3.1) under the given parameter $\mathbf{x}$ is defined by

$$P(\mathbf{x}) = \{\mathbf{y} \in S(\mathbf{x}) \mid f(\mathbf{x}, \mathbf{y}) = w(\mathbf{x})\}. \tag{3.3}$$

The set of points $\mathbf{y}$ of $S(\mathbf{x})$ that satisfy $\mathbf{g}(\mathbf{x}, \mathbf{y}) < \mathbf{0}$ is denoted by $S^-(\mathbf{x})$, that is

$$S^-(\mathbf{x}) = \{\mathbf{y} \in R^q \mid \mathbf{g}(\mathbf{x}, \mathbf{y}) < \mathbf{0}\}. \tag{3.4}$$

The index set of active inequality constraints is defined as

$$I(\mathbf{x}, \mathbf{y}) = \{i = 1, \ldots, p \mid g_i(\mathbf{x}, \mathbf{y}) = 0\}. \tag{3.5}$$

Let the following assumptions hold in this paper:

(a) functions *f* and $\mathbf{g}$ are convex and continuously differentiable in $(\mathbf{x}, \mathbf{y})$;

(b) $P(\mathbf{x})$ and $S^{-}(\mathbf{x})$ is a nonempty set, $\forall \mathbf{x} \in X$;
(c) $\forall \mathbf{x} \in X$, the vectors $\{\nabla_{\mathbf{y}} g_i(\mathbf{x}, \mathbf{y}), i \in I(\mathbf{x}, \mathbf{y})\}$ are linearly independent.

Theorem 1 $\forall \mathbf{x} \in X$, $P(\mathbf{x})$ *is nonempty, then* $w(\mathbf{x})$ *is convex on* X.

Proof See Mangasarian and Rosen (1964). □

The subgradient denoted by $\partial f(\bar{\mathbf{x}})$ of f at $\bar{\mathbf{x}} \in X$ is defined by

$$\partial f(\bar{\mathbf{x}}) = \{\xi \in R^n \mid f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \xi^T(\mathbf{x} - \bar{\mathbf{x}})\} \tag{3.6}$$

where ξ is a vector. It is noted that $\partial f(\bar{\mathbf{x}})$ is a set of n-dimensional vector.

Next we give a well-known property of the subgradient of a convex function f. For details, see (Rockafellar 1970).

Theorem 2 *Let* $f : \mathbf{x} \to R^1$ *a convex function on* $X.\partial f(\bar{\mathbf{x}})$ *has unique point, then* f *is differentiable at* $\bar{\mathbf{x}} \in X$ *and* $\partial f(\bar{\mathbf{x}}) = \{\nabla f(\bar{\mathbf{x}})\}$.

Theorem 3 *Under the assumptions* (a)–(b), *the subgradient of* $w(\mathbf{x})$ *at* $\mathbf{x}^* \in X$ *is nonempty and given by*

$$\begin{aligned}\partial w(\mathbf{x}^*) = \Bigg\{ & \nabla_{\mathbf{x}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \lambda_i^* \nabla_{\mathbf{x}} g_i(\mathbf{x}^*, \mathbf{y}^*) \;\Bigg| \\ & \nabla_{\mathbf{y}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \lambda_i^* \nabla_{\mathbf{y}} g_i(\mathbf{x}^*, \mathbf{y}^*) = 0, \\ & \lambda_i^* \geq 0,\ \lambda_i^* = 0 \text{ if } i \notin I(\mathbf{x}^*, \mathbf{y}^*) \Bigg\}\end{aligned} \tag{3.7}$$

where λ^* *is the Kuhn–Tucker vectors corresponding to an arbitrary element in* $\mathbf{y}^* \in P(\mathbf{x}^*)$.

Proof See Theorem 6.6.2 in (Shimizu et al. 1997). □

Even though $w(\mathbf{x})$ is not necessary differentiable in general (Shimizu et al. 1997), furthermore, we can obtain a characteristic property for the optimal-value function $w(\mathbf{x})$ under assumptions (a)–(c).

Theorem 4 *Under the assumptions* (a)–(c), *if* $\forall \mathbf{x}^* \in X$ *and* $\mathbf{y}^* \in P(\mathbf{x}^*)$, *then* $w(\mathbf{x})$ *is differentiable at* $\mathbf{x}^*$, *and the gradient of* $w(\mathbf{x})$ *at* $\mathbf{x}^*$ *is given by*

$$\nabla_{\mathbf{x}} w(\mathbf{x}^*) = \nabla_{\mathbf{x}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \lambda_i^* \nabla_{\mathbf{x}} g_i(\mathbf{x}^*, \mathbf{y}^*), \tag{3.8a}$$

$$\nabla_{\mathbf{y}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \lambda_i^* \nabla_{\mathbf{y}} g_i(\mathbf{x}^*, \mathbf{y}^*) = 0, \tag{3.8b}$$

$$\lambda_i^* \geq 0, \qquad \lambda_i^* = 0 \quad \text{if } i \notin I(\mathbf{x}^*, \mathbf{y}^*). \tag{3.8c}$$

Proof In view of Theorem 2, we only prove that there exists unique point in the set of $\partial w(\mathbf{x})$.

From Theorem 3, if $\xi_1 \in \partial w(\mathbf{x}^*)$, then

$$\xi_1 = \nabla_{\mathbf{x}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \lambda_i^* \nabla_{\mathbf{x}} g_i(\mathbf{x}^*, \mathbf{y}^*), \tag{3.9a}$$

$$0 = \nabla_{\mathbf{y}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \lambda_i^* \nabla_{\mathbf{y}} g_i(\mathbf{x}^*, \mathbf{y}^*), \tag{3.9b}$$

$$\lambda_i^* \geq 0, \qquad \lambda_i^* = 0 \quad \text{if } i \notin I(\mathbf{x}^*, \mathbf{y}^*). \tag{3.9c}$$

Assume that there exists another point $\xi_2 \in \partial w(x^*)$ and $\xi_1 \neq \xi_2$. According to Theorem 3, we have

$$\xi_2 = \nabla_{\mathbf{x}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \bar{\lambda}_i^* \nabla_{\mathbf{x}} g_i(\mathbf{x}^*, \mathbf{y}^*), \tag{3.10a}$$

$$0 = \nabla_{\mathbf{y}} f(\mathbf{x}^*, \mathbf{y}^*) + \sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} \bar{\lambda}_i^* \nabla_{\mathbf{y}} g_i(\mathbf{x}^*, \mathbf{y}^*), \tag{3.10b}$$

$$\bar{\lambda}_i^* \geq 0, \qquad \bar{\lambda}_i^* = 0 \quad \text{if } i \notin I(\mathbf{x}^*, \mathbf{y}^*). \tag{3.10c}$$

Obviously, from (3.9) and (3.10), we obtain

$$\sum_{i \in I(\mathbf{x}^*, \mathbf{y}^*)} (\lambda_i^* - \bar{\lambda}_i^*) \nabla_{\mathbf{y}} g_i(\mathbf{x}^*, \mathbf{y}^*) = 0. \tag{3.11}$$

This means $\lambda_i^* = \bar{\lambda}_i^*$, $i \in I(\mathbf{x}^*, \mathbf{y}^*)$ under assumption (c).

Therefore, it yields $\xi_1 = \xi_2$.

The conclusion is correct. □

3.2 The algorithm of bilevel programming for CNDP

Define the optimal-value function of the lower level problem as follows:

$$w(\mathbf{x}) = \min T(\mathbf{x}, \mathbf{y}) = \sum_{a \in A} \int_0^{y_a(\mathbf{x})} t_a(v, x_a) dv \tag{3.12a}$$

$$\text{s.t.} \quad \sum_k h_k^{rs} = q_{rs}, \quad \forall r \in R,\ s \in S, \tag{3.12b}$$

$$h_k^{rs} \geq 0, \quad \forall r \in R,\ s \in S,\ k \in K_{rs}, \tag{3.12c}$$

$$y_a = \sum_r \sum_s \sum_k h_k^{rs} \delta_{a,k}^{rs}, \quad \forall a \in A. \tag{3.12d}$$

Obviously the lower level user equilibrium problem holds the assumptions (a) and (b). It can be proved that assumption (c) also holds in problem (3.12) (see Yang et al. 2004). Observe that the constraints in problem (3.12) do not include the upper level variables $\mathbf{x}$, therefore when calculating the gradient $\nabla w(\mathbf{x})$ in accordance with (3.8a) in Theorem 4, the second term, $\sum_{i\in I(\mathbf{x}^*,\mathbf{y}^*)}\lambda_i^*\nabla_{\mathbf{x}}g_i(\mathbf{x}^*,\mathbf{y}^*)$, equals to zero. So we can obtain$\nabla w(\mathbf{x})$ easily by the following formulation:

$$\frac{\partial w(\mathbf{x})}{\partial x_a} = \sum_{a\in A}\int_0^{y_a^*(\mathbf{x})} \frac{\partial t_a(v,x_a)}{\partial x_a} dv, \quad a \in A \tag{3.13}$$

where $y_a^*(\mathbf{x})$is the user equilibrium link flow for fixed link capacity enhancement pattern $\mathbf{x}$.

Remark 3.1 The formulation (3.13) is just the same as the formulation in Meng et al. (2001), but our proof complexity is less than that of Meng et al.'s.

Noted that there exists the following relationship between the upper and lower level objective functions:

$$F(\mathbf{x},\mathbf{y}^*(\mathbf{x})) = w(\mathbf{x}) + \sum_a \int_0^{y_a^*(\mathbf{x})} v\frac{dt_a(v,x_a)}{dv}dv \tag{3.14}$$

where $w(\mathbf{x})$ is the optimal-value function of the lower level model.

Therefore, the bilevel model for CNDP can be equivalently transferred into the following single level optimization problem:

$$\min_{\mathbf{x}} \quad F(\mathbf{x},\mathbf{y}^*(\mathbf{x})) = w(\mathbf{x}) + \sum_a \int_0^{y_a^*(\mathbf{x})} v\frac{dt_a(v,x_a)}{dv}dv \tag{3.15a}$$

$$\text{s.t.} \quad \sum_{a\in A} G_a(x_a) \le B, \tag{3.15b}$$

$$x_a \ge 0, \quad \forall a \in A. \tag{3.15c}$$

Obviously, in view of the definition of the convex function, we can obtain the following theorem.

Theorem 5 *If* $\mathbf{y} = \mathbf{y}(\mathbf{x})$ *is a convex function, and* $F(\mathbf{x},\mathbf{y})$ *is convex in* $(\mathbf{x},\mathbf{y})$ *and nondecreasing for any* $\mathbf{y}$, *then* $F(\mathbf{x},\mathbf{y}(\mathbf{x}))$ *is also a convex function in* $\mathbf{x}$.

Remark 3.2 In some widely used methods, such as SAB method, $\mathbf{y} = \mathbf{y}(\mathbf{x})$ is often expressed by a linearization method, i.e., a first-order Taylor expansion of $\mathbf{y}(\mathbf{x})$ at the iteration, then, of course $\mathbf{y}(\mathbf{x})$ is a convex function.

Assume that $\mathbf{y} = \mathbf{y}(\mathbf{x})$ is a convex function, and $F(\mathbf{x},\mathbf{y}(\mathbf{x}))$ is convex in $(\mathbf{x},\mathbf{y})$ and nondecreasing for any $\mathbf{y}$. Based on Theorem 5, we can conclude that problem (3.15) is a convex programming problem. However $\mathbf{y} = \mathbf{y}(\mathbf{x})$ is generally an implicit and nonlinear function.

Dafermos and Nagurney (1984) pointed out: if the travel cost function is strict monotone, then $y_a(\mathbf{x})$ is a continuous function of the travel demand and link capacity improvements. Furthermore, we assume that $y_a(\mathbf{x})$ is not only continuous but also differential. Based on (3.13), the gradient of $F(\mathbf{x},\mathbf{y}(\mathbf{x}))$ can be computed straightforwardly as follows:

$$\frac{\partial F(\mathbf{x},\mathbf{y}^*(\mathbf{x}))}{\partial x_a} = \frac{\partial \omega(\mathbf{x})}{\partial x_a} + \sum_{a\in A} y_a^*(\mathbf{x}) \frac{dt_a(y_a,x_a)}{dy_a}\bigg|_{y_a=y_a^*(\mathbf{x})} \frac{\partial y_a(\mathbf{x})}{\partial x_a} + \int_0^{y_a^*(\mathbf{x})} v \frac{dt_a(v,x_a)}{dvdx_a} dv, \quad a \in A. \tag{3.16}$$

As referred above, $\mathbf{y} = \mathbf{y}(\mathbf{x})$ is an implicit and nonlinear function, thus $\frac{\partial y_a(\mathbf{x})}{\partial x_a}$ cannot be obtained efficiently. Although in common methods such as SAB (sensitivity analysis based method), the differentiability of $\mathbf{y} = \mathbf{y}(\mathbf{x})$ can be guaranteed under certain strong assumptions and $\frac{\partial y_a(\mathbf{x})}{\partial x_a}$ can be figured out approximately, its computational expense becomes unendurable and even impossible as the problem's scale increases. For example, it is inescapable to calculate the inverse matrix in SAB (Tobin and Friesz 1988; Yang 1995). Moreover, SAB cannot guarantee the globally convergence (Friesz et al. 1990; Yang and Yagar 1994; Gao and Song 2002). In this paper, we use the difference $\frac{\Delta y_a(\mathbf{x}^k)}{\Delta x_a^k}$ to approximate the differential $\frac{\partial y_a(\mathbf{x})}{\partial x_a}$ at each iteration, thus the aforementioned deficiency can be avoided. Computational errors may result from the approximate expression of $\frac{\partial y_a(\mathbf{x})}{\partial x_a}$, for example, the auxiliary iteration point $\tilde{\mathbf{z}}^k$ which determines can not be computed exactly, but the errors can be coped with by exploring the characteristic of MSA (method of successive average).

There exist many methods to solve problem (3.15). Among these methods, MSA is one of the efficient and globally convergent methods (Powell and Sheffi 1982). From the definition, we know that $\frac{\partial y_a(\mathbf{x})}{\partial x_a} = \lim_{x_a' - x_a} \frac{y_a(\mathbf{x}') - y_a(\mathbf{x})}{x_a' - x_a}$, where other elements of $\mathbf{x}'$ and $\mathbf{x}$ are equal except $x_a' \neq x_a$. In the iteration procedure of MSA (iteration step $\alpha = \frac{1}{k}$), the difference between iteration points, $\mathbf{x}^{k+1} - \mathbf{x}^k = \frac{1}{k}(\mathbf{z}^k - \mathbf{x}^k)$, tends to zero as the iteration proceeds. Denote the exactly auxiliary iteration point as $\tilde{\mathbf{z}}^k$, and the approximately auxiliary iteration point as $\mathbf{z}^k$, which is obtained by MSA when replacing $\frac{\partial y_a(\mathbf{x})}{\partial x_a}$ with $\frac{\Delta y_a(\mathbf{x}^k)}{\Delta x_a^k}$. Assume that the error $\zeta^k = \tilde{\mathbf{z}}^k - \mathbf{z}^k$ is a stochastic variables. Observing constraints (3.15a) and (3.15b), it can be concluded that the approximately auxiliary iteration point $\mathbf{z}^k$ is bounded. It follows that the variance of the auxiliary iteration point $\mathbf{z}^k$ is bounded too. For expression convenience, we assume that the upper bound of $\mathbf{z}^k$ is $\sigma^2 < \infty$. Obviously,

$$\mathrm{Var}(\mathbf{x}^k) = \frac{1}{(k-1)^2} \sum_{l=1}^{k-1} \mathrm{Var}(\mathbf{z}^l) < \frac{1}{(k-1)^2} \sum_{l=1}^{k-1} \sigma^2$$

holds, therefore $\mathrm{Var}(\mathbf{x}^k)$ approaches zero as k grows, that is, the variance approaches zero as the iteration proceeds. So the error can be ignored when replacing $\frac{\partial y_a(\mathbf{x}^k)}{\partial x_a^k}$ with

$\frac{\Delta y_a(\mathbf{x}^k)}{\Delta x_a^k}$, and then (3.16) can be rewritten as follows:

$$\frac{\partial F(\mathbf{x},\mathbf{y}^*(\mathbf{x}))}{\partial x_a} = \frac{\partial \omega(\mathbf{x})}{\partial x_a} + \sum_{a\in A} y_a^*(\mathbf{x}) \frac{dt_a(y_a,x_a)}{dy_a}\bigg|_{y_a=y_a^*(\mathbf{x})} \frac{\Delta y_a(\mathbf{x})}{\Delta x_a} + \int_0^{y_a^*(\mathbf{x})} v\frac{dt_a(v,x_a)}{dv dx_a} dv, \quad a\in A \tag{3.17}$$

where $\nabla_{\mathbf{x}} w(\mathbf{x}^k)$ can be obtained according to (3.13).

Based on this, the near-global optimum to the CNDP can be obtained by our algorithm. The advantage of this algorithm is that part of coefficients is exact and part of that is approximate and (3.17) is easy to compute, which should be better than the existing algorithms such as SAB.

Therefore, the gradients of $F(\mathbf{x},\mathbf{y}(\mathbf{x}))$ at each trial point can be approximately computed based on the above results. By solving the lower level user equilibrium problem at a trial point $\mathbf{x}^k$, we can obtain an optimal solution $\mathbf{y}^* \in P(\mathbf{x}^k)$ and figure out $\nabla_{\mathbf{x}} F(\mathbf{x}^k,\mathbf{y}^*(\mathbf{x}^k))$ from (3.17).

Therefore, we propose the following algorithm for CNDP based on MSA. (This algorithm can be called MSAB.)

Step 1: *Initiation.* Give initial points $\mathbf{x}^0 \in X$, and let $k=1$.

Step 2: *Calculate* $\nabla_{\mathbf{x}} F(\mathbf{x}^k,\mathbf{y}^*(\mathbf{x}^k))$. Fixed the upper level variables $\mathbf{x}=\mathbf{x}^k$, solve the following problem by implementing the user equilibrium assignment procedure

$$\begin{aligned} & w(\mathbf{x}^k) = \min_{\mathbf{y}} T(\mathbf{x}^{\mathbf{k}},\mathbf{y}) \\ \text{s.t.} \quad & \sum_k h_k^{rs} = q_{rs}, \quad \forall r\in R,\ s\in S, \\ & h_k^{rs} \ge 0, \quad \forall r\in R,\ s\in S,\ k\in K_{rs}, \\ & y_a = \sum_r\sum_s\sum_k h_k^{rs}\delta_{a,k}^{rs}, \quad \forall a\in A \end{aligned}$$

and obtain the optimal solution $\mathbf{y}^*(\mathbf{x}^k)$, then calculate $\nabla_{\mathbf{x}} w(\mathbf{x}^k)$ from (3.13) and then $\nabla_{\mathbf{x}} F(\mathbf{x}^k,\mathbf{y}^*(\mathbf{x}^k))$ can be figured out according to (3.17).

Step 3: *Solve linear programming.* Solve the transferred upper level problem (3.15) with MSA method. Obtain the descent direction by solving the following linear programming:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \nabla_{\mathbf{x}} F(\mathbf{x}^k,\mathbf{y}^*(\mathbf{x}^k))^T(\mathbf{x}-\mathbf{x}^k) \\ \text{s.t.} \quad & \textstyle\sum_{a\in A} G_a(x_a) \le B, \\ & x_a \ge 0, \quad \forall a\in A \end{aligned}$$

obtain the optimal $\mathbf{z}^k$.

Step 4: *Convergence check.* If $\nabla_{\mathbf{x}} F(\mathbf{x}^k,\mathbf{y}^*(\mathbf{x}^k))^T(\mathbf{z}^k-\mathbf{x}^k)=0$ stop; Otherwise, let $\mathbf{x}^{k+1}=\mathbf{x}^k+\frac{1}{k}(\mathbf{z}^k-\mathbf{x}^k)$, $k=k+1$, go to step 2.

Remark 3.3 In step 2, for any fixed link capacity enhancement $\mathbf{x}$, the corresponding user equilibrium link flow $\mathbf{y}^*(\mathbf{x}^k)$ can be obtained easily by implementing an efficient DUE traffic assignment procedure (such as Frank–Wolfe method) (Sheffi 1985; Patriksson 1994). Then the derivative of the optimal-value function $w(\mathbf{x})$can be obtained straightforwardly through (3.13).

Remark 3.4 In step 3, because the constraints are always linear, we only solve the linear programming by existing algorithms, such as the simplex method.

From the mathematical analysis, it can be showed that the algorithm we proposed here has the following advantages compared with the traditional algorithms:

(1) The algorithm is very simple and need not strong hypothesis conditions. However, other algorithms such as SAB algorithm have to calculate the inverse of the matrix and only can be used with many assumptions (It requires that the lower level functions is second order continuous differentiable and the lower level has a unique solution for fixed upper level variables), which limits the application in engineering.

(2) Because of the implicit non-convex of the bilevel network design problem, some algorithms, such as SAB, generally can not ensure the convergence. The main thought of this paper is to transform the bilevel programming into a convex programming problem and then MSA method is used and the near-global optimum of CNDP can be obtained. The possible shortcoming may occur when the derivative $\nabla_{\mathbf{x}} F(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))$ is approximated. However, the error can be ignored by exploring the characteristics of MSA.

(3) The algorithm is based on the MSA, therefore it is suit to the transportation problem.

4 Numerical example

As a test, the network shown in Fig. 1 (Suwansirikul et al. 1987) was used as a basis to compare the results of the algorithm (MSAB) proposed in this paper with those obtained from other existing algorithms, such as MINOS method (a Modular In-Core Nonlinear Optimization System, Tan et al. 1979), EDO algorithm (Equilibrium Decomposed Optimization, Suwansirikul et al. 1987), BDA (Bilevel Descent Algorithm, Kim and Suh 1988) and BLABD (Bilevel Linear Approximation Based on Difference, Gao et al. 2000) etc.

Fig. 1 Test network

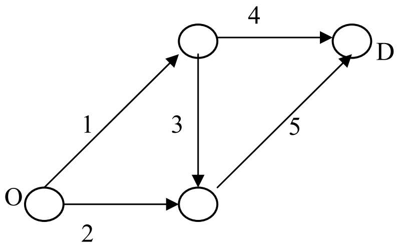

Table 1 Data for test network

Link	A_a	B_a	K_a	d_a
1	4.0	0.60	40.0	2.0
2	6.0	0.90	40.0	2.0
3	2.0	0.30	60.0	1.0
4	5.0	0.75	40.0	2.0
5	3.0	0.45	40.0	2.0

$t_a(y_a, x_a) = A_a + B_a(y_a/(k_a + x_a))^4, a \in A$
$Z(x) = \sum_a t_a(y_a, x_a)y_a, a \in A$
$G_a(x_a) = 1.5 \cdot d_a \cdot (x_a)^2, a \in A$

Table 2 Comparison results of different algorithms (1)

Demand	Link	MINOS	EDO	BDA	BLABG	MSAB
	y_1	1.34	1.31	1.34	1.34	1.34
	y_2	1.21	1.19	1.21	1.21	1.23
100.00	y_3	0.00	0.06	0.00	0.00	0
	y_4	0.97	0.94	0.97	0.97	0.96
	y_5	1.10	1.06	1.08	1.10	1.09
	Z	1200.58	1200.64	1200.58	1200.58	1200.58
150.00	y_1	6.05	5.98	6.04	6.05	6.05
	y_2	5.47	5.52	5.46	5.47	5.47
	y_3	0.00	0.02	0.00	0.01	0.02
	y_4	4.64	4.61	4.64	4.64	4.64
	y_5	5.27	5.27	5.27	5.27	5.27
	Z	3156.38	3156.24	3156.21	3156.21	3156.21

Table 1 presents the functional forms of the travel and investment costs, as well as the parameter values for each arc, used in this numerical test. We experimented two different traffic demand levels from node O to node D 100 or 150 respectively. The corresponding results are shown in Table 2.

The MINOS algorithm is regard as the most precise one and the BDA algorithm the less computation. Table 2 presents the comparisons of the proposed algorithm with the four existing algorithms. From the table, we can observe that the solution obtained by our algorithm is quite close to the solution by MINOS. It is needed to solve the UE problem in CNDP, moreover, the common method to solve UE is the Frank–Wolfe method (F-W). Therefore, the iteration number of F-W can be treated as the measure of the computation for CNDP. Table 3 give comparison results of different algorithm in terms of the iteration number. From Table 3, we can see that MSAB method has the smallest number of iteration.

Table 3 Comparison results of different algorithms (2)

Demand	Algorithm	Iteration number of Frank–Wolfe
100.00	EDO	24
	BDA	17
	BLABG	10
	MSAB	9
150.00	EDO	29
	BDA	19
	BLABG	10
	MSAB	9

5 Conclusion

The transportation continuous network design problem (CNDP) is characterized by a bilevel programming model and recognized to be one of the most difficult and challenging problems in transportation. The main difficulty stems from the fact that the bilevel formulation for the CNDP is nonconvex and nondifferentialbe, and indeed only some heuristic methods have been so far proposed. In this paper, the bilevel programming model for CNDP is transformed into a convex programming problem by virtue of an optimal-value function tool and the relationship between System Optimum (the upper level objective function) and User Equilibrium (the lower level objective function). By exploring the inherent nature of the CNDP, the optimal-value function for the lower level user equilibrium problem is proved to be continuously differentiable and its derivative in capacity enhancement can be obtained efficiently by implementing a user equilibrium assignment subroutine. However, the reaction (or response) function between the upper and lower level problem is implicit and its gradient is difficult to obtain. Although, here we approximately express the gradient with the difference concept at each iteration, based on the method of successive averages (MSA), we propose a globally convergent algorithm to solve the single level convex programming problem. Comparing with widely used heuristic algorithms, such as SAB method, the proposed algorithm needs not strong hypothesis conditions and complex computation for the inverse matrix, and its key computational issue is solving the user equilibrium assignment problem with fixed link capacity enhancement. These favorable characteristics indicate the potential of the algorithm to solve large CNDPs. Numerical results have indicated the efficiency of the technique. However, it is pity that the proposed method has its intrinsic weakness because it must explore the relationship between the upper and lower level objective functions to transform the bilevel model into a single level optimization problem. Thus, the proposed method is hard to deal with other bilevel optimization problems.

Acknowledgements This paper is partly supported by National Basic Research Program of China (2006CB705500), NSFC of China (70631001 and 70501005), Changjiang Scholars and Innovative Research Team in University under Grant (IRT0605).

References

Abdulaal M, LeBlanc LJ (1979) Continuous equilibrium network design models. Transp Res B 13:19–32

Boyce DE (1984) Urban transportation network equilibrium and design models: recent achievements and future prospectives. Environ Plan A 16:1445–1474

Chiou SW (1999) Optimization of area traffic control for equilibrium network flows. Transp Sci 33:279–289

Cho HJ (1988) Sensitivity analysis of equilibrium network flows and its application to the development of solution methods for equilibrium network design problems. PhD dissertation. University of Pennsylvania, Philadelphia

Cree ND, Maher MJ (1998) The continuous equilibrium optimal network design problem: a genetic approach. In: Transportation networks: recent methodological advances. Elsevier, Netherlands, pp 163–174

Dafermos S (1980) Traffic equilibria and variational inequalities. Transp Sci 14:42–54

Dafermos S, Nagurney A (1984) Sensitivity analysis for the asymmetric network equilibrium problem. Math Program 28:174–184

Friesz TL (1981) An equivalent optimization problem with combined multiclass distribution assignment and modal split which obviates symmetry restriction. Transp Res B 15:361–369

Friesz TL (1985) Transportation network equilibrium, design and aggregation: key developments and research opportunities. Transp Res A 19:413–427

Friesz TL, Harker PT (1985) Properties of the iterative optimization equilibrium algorithm. Civ Eng Syst 2:142–154

Friesz TL et al (1990) Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. Math Program 48:265–284

Friesz TL et al (1993) The multiobjective equilibrium network design problem revisited: a simulated annealing approach. Eur J Oper Res 65:44–57

Gao ZY, Song YF (2002) A reserve capacity model of optimal signal control with user-equilibrium route choice. Transp Res B 36:313–323

Gao ZY, Song YF Si BF (2000) Urban transportation continuous equilibrium network design problem: theory and method. China Railway Press, Beijing

Gao ZY, Wu JJ, Sun HJ (2005) Solution algorithm for the bi-level discrete network design problem. Transport Res B 39:479–495

Kim TJ (1990) Advanced transport and spatial systems models: applications to Korea. Springer, New York

Kim TJ, Suh S (1988) Toward developing a national transportation planning model: a bilevel programming approach for Korea. Ann Reg Sci XXSPED:65–80

Luo ZQ, Pang JS, Ralph D (1996) Mathematical programs with equilibrium constraints. Cambridge University Press, Cambridge

Magnanti TL, Wong RT (1984) Network design and transportation planning: models and algorithms. Transp Sci 18:1–55

Mangasarian OL, Rosen JB (1964) Inequalities for stochastic nonlinear programming problems. Oper Res 12:143–154

Marcotte P (1983) Network optimization with continuous control parameters. Transp Sci 17:181–197

Marcotte P (1986) Network design problem with congestion effects: a case of bi-level programming. Math Program 34:142–162

Marcotte P, Marquis G (1992) Efficient implementation of heuristics for the continuous network design problem. Ann Oper Res 34:163–176

Marcotte P, Zhu DL (1996) Exact and inexact penalty methods for the generalized bilevel programming problems. Math Program 74:141–157

Meng Q, Yang H, Bell MGH (2001) An equivalent continuously differentiable model and a locally convergent algorithm for the continuous networks design problem. Transp Res B 35:83–105

Patriksson M (1994) The traffic assignment problem models and methods. VSB BV, Netherlands

Powell WB, Sheffi Y (1982) The convergence of equilibrium algorithms with predetermined step size. Transp Sci 6:5–55

Rockafellar RT (1970) Convex analysis. Princeton University Press, Princeton

Sheffi Y (1985) Urban transportation networks: equilibrium analysis with mathematical programming methods. Prentice–Hall, Englewood Cliffs

Shimizu K, Ishizuka Y, Bard JF (1997) Nondifferentiable and two-level mathematical programming. Kluwer Academic, Massachusetts

Suwansirikul C, Friesz TL, Tobin RL (1987) Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. Transp Sci 21:254–263

Tan HN, Gershwin SB, Athans M (1979) Hybrid optimization in urban traffic networks. Report No. DOT-TSC-RSPA-79-7. Laboratory for Information and Decision System, MIT, Cambridge, MA

Tobin RL, Friesz TL (1988) Sensitivity analysis for equilibrium network flows. Transp Sci 22:242–250

Wong SC, Yang H (1997) Reserve capacity of a signal-controlled road network. Transp Res Part B 31:397–402

Yang H (1995) Sensitivity analysis for queuing equilibrium network flow and its application to traffic control. Math Comput Model 22:247–258

Yang H (1997) Sensitivity analysis for the elastic demand network equilibrium problem with applications. Transp Res B 31:55–70

Yang H, Bell MGH (1998) Models and algorithm for road network design: a review and some new developments. Transp Rev 18(3):257–278

Yang H, Yagar S (1994) Traffic assignment and traffic control in general freeway-arterial corridor systems. Transp Res B 28:463–486

Yang H, Meng Q, Liu GS (2004) The generalized transportation network optimization problem: models and algorithms. Working Paper, The Hong Kong University of Science and Technology