Netw Spat Econ (2009) 9:243–267
DOI 10.1007/s11067-009-9100-4

A Link-Node Discrete-Time Dynamic Second Best Toll Pricing Model with a Relaxation Solution Algorithm

Xuegang (Jeff) Ban · Henry X. Liu

Published online: 11 February 2009

Abstract Dynamic congestion pricing has become an important research topic because of its practical implications. In this paper, we formulate dynamic second-best toll pricing (DSBTP) on general networks as a bilevel problem: the upper level is to minimize the total weighted system travel time and the lower level is to capture motorists' route choice behavior. Different from most of existing DSBTP models, our formulation is in discrete-time, which has very distinct properties comparing with its continuous-time counterpart. Solution existence condition of the proposed model is established independent of the actual formulation of the underlying dynamic user equilibrium (DUE). To solve the bilevel DSBTP model, we adopt a relaxation scheme. For this purpose, we convert the bilevel formulation into a single level nonlinear programming problem by applying a link-node based nonlinear complementarity formulation for DUE. The single level problem is solved iteratively by first relaxing the strick complementarity by a relaxation parameter, which is then progressively reduced. Numerical results are also provided in this paper to illustrate the proposed model and algorithm. In particular, we show that by varying travel time weights on different links, DSBTP can help traffic management agencies better achieve certain system objectives. Examples are

X. (Jeff) Ban (✉)
Department of Civil and Environmental Engineering,
Rensselaer Polytechnic Institute, Troy, NY, USA
e-mail: banx@rpi.edu

H. X. Liu
Department of Civil Engineering, University of Minnesota,
Twin Cities, Minneapolis, MN, USA
e-mail: henryliu@umn.edu

given on how changes of the weights impact the optimal tolls and associated objective function values.

Keywords Congestion pricing · Dynamic second best toll pricing · Dynamic user equilibrium · Relaxation algorithm

1 Introduction

The advent of congestion pricing and emerging technologies in implementing tolling are among the most promising options to address traffic congestion that has become not only an increasingly critical problem to our quality of life but also has serious consequences in terms of economic development (Federal Highway Administration 2007). Current research on congestion pricing has been mainly concentrating on static pricing, aiming to find an optimal (and fixed) toll on all links (i.e., the first-best toll pricing) or a subset of links (i.e., the second-best toll pricing) in a traffic network. Research in this regard is rich and still growing. One may refer to Lawphongpanich and Hearn (2004), Hearn and Ramana (1998), Yang and Huang (2005), and Sumalee (2007) for more details. Static congestion pricing, however, ignores traffic dynamics and the generated tolls are not suitable if time-varying traffic flows are considered. Rather, time varying tolls generated via dynamic congestion pricing can be more effective to solve ever-increasing congestion problems, especially in heavily-congested urban areas (Friesz et al. 2002).

Day-to-day and within-day toll pricing have been developed in the literature to account for traffic dynamics. The former captures inter-day (long-term) traffic dynamics, while within-day traffic is usually considered as static, which aims to adjust tolls in a day-to-day basis so that certain system objective or equilibrium can be achieved (Friesz et al. 2002). Within-day toll pricing focuses on short-term (i.e. within-day) traffic dynamics, aiming to adjust tolls in a real-time basis. Dual schemes for combining both day-to-day and within-day toll pricing have also been proposed by several researchers (e.g. Friesz et al. 2007). In this paper, we concentrate on the within-day toll pricing problem, which we refer to as "dynamic congestion pricing" hereafter in this paper.

Similar to static congestion pricing, the dynamic congestion pricing may be categorized as first-best toll pricing and second-best toll pricing. The first-best toll pricing assumes that every link in the network can be tolled and marginal cost is the main mechanism for studying first-best toll pricing problems. For example, Carey and Srinivasan (1993) studied marginal costs and tolls in networks with time-varying flows. Wie and Tobin (1998) developed both day-to-day and with-day models for dynamic first-best toll pricing. In particular, the with-day model in Wie and Tobin (1998) assumed fixed demand and was based on instantaneous dynamic user equilibria (DUE).

In reality, however, not every link in a network can be tolled due to technical or policy constraints. Therefore, the dynamic second-best toll pricing

(DSBTP), which imposes tolls only on a subset of links in a network, receives more attention from both practitioners and researchers. Most previous research efforts (Chu 1995; Yang and Huang 1997; Arnott et al. 1998; Liu and McDonald 1999; Kuwahara 2007) have been focusing on studying dynamic congestion pricing for either a single bottleneck or simple networks (e.g. a network with one OD pair and two parallel routes). For these simplified cases, the underlying route choice model can be much simplified so that analytical results can be obtained with relative ease. Therefore, although some insightful results were discovered from these studies, their models may not be applied directly to DSBTP on general networks.

In a general network, it may be more appropriate to model the route choice behavior as DUE as shown in Wie and Tobin (1998). Recently, Friesz et al. (2007) developed a DSBTP model that combines both day-to-day and within day traffic dynamics. The model was developed in the continuous-time domain, which makes them infinite-dimensional mathematical programming problems. Currently, theories and algorithms for solving large-scale infinite-dimensional mathematical programming problems are not yet widely available in the literature.

In this paper, we take a different stand and formulate DSBTP in the discrete-time domain, resulting in a finite-dimensional mathematical programming problem for which a rich set of theories and algorithms have been developed. We particularly model the underlying route choice as a DUE problem to account for user route choices in general networks. The resulting model is a bi-level programming problem or MPEC (mathematical programming with equilibrium constraints, see Luo et al. 1996), similar to that in Friesz et al. (2007). As pointed out in Wie et al. (2002), finite-dimensional and infinite-dimensional mathematical programming problems are essentially different problems, and should be treated separately. Therefore, although our model is similar to certain extent in its form to those in Friesz et al. (2007), the solution existence conditions and solution algorithms are quite different. The work in this paper is also based on the link-node based discrete-time DUE formulation we proposed recently (Ban et al. 2008).

We first show that the bilevel DSBTP model has at least one solution under certain assumptions, independent of the actual formulation of the lower level DUE. This finding may provide some insights on DSBTP models that employ rather complicated traffic flow models in the underlying DUE formulation. We then observe that a recently developed DUE model (Ban et al. 2008) satisfies the solution existence conditions established in this paper for DSBTP. By applying this DUE model, the bilevel formulation can be readily converted to a single level NLP (nonlinear programming problem). To solve the single level model, a relaxation algorithm is adopted as has been used by the authors for solving the static continuous network design problem (Ban et al. 2006). Numerical examples are provided in this paper to illustrate the proposed model and algorithm in this paper. In particular, we show that by varying travel time weights on different links, DSBTP can help traffic management agencies better achieve certain system objectives. Examples are also given on how changes of the weights impact the optimal tolls and associated objective function values.

2 A general bilevel formulation for dynamic second best toll pricing

2.1 Bilevel DSBTP model

Assume a given transportation network can be represented as a connected and directed graph, denoted as $G(N, A)$ where N is the set of nodes and A is the set of links. We divide the study period (during which all traffic in the network is cleared) into K' time intervals with $\triangle$ the length of each time interval. Further denote S the set of destination nodes. Throughout this paper, we use index $i \in N$ or $j \in N$ to denote a node, $a \in A$ to denote a link, $s \in S$ to denote a destination, and k to denote a time interval.

We use U_a^k to denote the total inflow rate to link a at the *beginning* of time interval k (which is also assumed to be constant during the interval) and τ_a^k to denote the travel time of link a at the *end* of interval k. Notice that τ_a^k is defined at the *end* of a time interval, which is to formulate the DUE model in a predictive fashion. For details, one can refer to Heydecker and Verlander (1999), Han (2003), and Ban et al. (2008). Under this notation, τ_a^{k-1} is the travel time of link a at the *beginning* of time interval k. Further, $U = (U_a^k)_{\forall a \in A, 1 \le k \le K'}$ and $\tau = (\tau_a^k)_{\forall a \in A, 0 \le k \le K'-1}$ represent, respectively, the vectors of link inflow rates and travel times.[1] It is well-known that τ_a^k is a critical component in studying DUE, which may not be easily computed in an analytical form. Rather, certain form of dynamic network loading is usually needed to generate τ_a^k (Nie and Zhang 2005; Carey et al. 2003). However, it is reasonable to assume that for a given traffic network, a fixed link inflow vector U should always generate one unique travel time vector τ. In this sense, τ can be expressed as a one-to-one mapping of U, i.e., $\tau = \tau(U)$, whose form may depend on the specific traffic network and possibly not in a close-form. Further denote y_a^k the toll imposed on link a during the time interval k and $y = (y_a^k)_{\forall a \in A, 1 \le k \le K'}$. Since we are interested in the second-best toll pricing in this paper, we denote $P \subseteq A$ the set of links that will be tolled. Thus we have $y_a^k = 0, \forall a \in A \setminus P, 1 \le k \le K'$.

Given the above notation, DSBTP can be modeled as a problem to determine an optimal toll vector y such that certain objective is achieved while motorists' route choice behaviors are also respected. It is then straightforward to formulate DSBTP as a bilevel problem. There are various ways to define the objective (Friesz et al. 2007) and in this paper it is to minimize the total weighted system travel time, defined as follows:

$$f(y, U) = \Delta U^T W \tau. \tag{1}$$

[1]Note that the definition of τ iterates over $0 \le k \le K' - 1$, which corresponds to travel times at the beginning of time interval 1 to K'.

Here W is a diagonal matrix representing the weights of network links. If $W = I$ where I is the identify matrix, Eq. (1) is the traditionally-used total system travel time (Friesz et al. 2007). Otherwise, imposing different weights on different links of the network will provide more flexibility to address concerns of traffic management authority in toll pricing design. In Section 5, we will provide more discussions on this.

We further model the motorist's route choice as a DUE. Since the solution of DUE will depend on the toll pricing vector y, we denote the solution set of DUE in terms of U as $SOL(y)$. Therefore, the bilevel DSBTP model, denoted as *BiDSBTP*, can be expressed as follows:

$$BiDSBTP \qquad \min_{y,U} \quad f(y, U) = \Delta U^T W \tau(U) \tag{2}$$

$$\text{s.t.} \quad y \in K_y \tag{3}$$

$$U \in SOL(y). \tag{4}$$

Here Eq. (2) represents the objective function of *BiDSBTP*. In addition, $K_y = \{y | y_l \leq y \leq y_u\}$ is the bound constraint of y, and y_l and y_u are the lower and upper bounds of y. The DUE constraint is represented in Eq. (4), which requires that U must be a solution to the lower level DUE problem. Note that the *BiDSBTP* model above is a general formulation of DSBTP since it does not depend on the actual formulation of DUE. We will also use italic acronyms in this paper to denote mathematical models or algorithms.

2.2 Solution existence condition of *BiDSBTP*

We next establish the solution existence condition for *BiDSBTP*. First, we notice that *BiDSBTP* is a nonlinear programming problem (NLP) and has at least one solution if the DUE constraint Eq. (4) represents a compact set of (y, U). Second, if for any given y, DUE has a unique solution in terms of total link inflow U (i.e., $SOL(y)$ is a singleton), U is a one-to-one mapping of y. In this case, the continuity of U in terms of y will guarantee that the constraint set of *BiDSBTP* is closed, which implies the solution existence of *BiDSBTP* since U and y are clearly bounded.

However, in general, DUE may have multiple solutions in terms of U and thus $SOL(y)$ is a set which may not be a singleton. In this case, $SOL(y)$ is a point-to-set map of y (Facchinei and Pang 2003). The solution existence of *BiDSBTP* will depend on the compactness of the graph of $SOL(y)$, denoted as

$$\Phi(y, U) = \{(y, U) | U \in SOL(y), \forall y_l \leq y \leq y_u\}. \tag{5}$$

Appendix 1 of this paper provides the definitions and properties of a point-to-set map which can be found in Facchinei and Pang (2003). In Luo et al. (1996), various solution existence conditions are established for MPECs based on properties of the graph $\Phi(y, U)$. These conditions (e.g. Theorem 1.4.1 on

Page 58 in Luo et al. 1996), although mathematically sound, lack the direct connection to the properties of DSBTP. In this paper, we extend those conditions by focusing on the properties of DSBTP especially the solution properties of the lower level DUE and the link travel time function. In particular, we can establish the solution existence condition for the bilevel model *BiDSBTP* as the following theorem.

Theorem 1 *BiDSBTP has at least one solution if the following conditions are satisfied:*

(a) $\forall y \in K_y$, *the set* $SOL(y)$ *is nonempty and compact*;
(b) *Assume* $y, \bar{y} \in K_y$ *are two toll vectors. If* $\forall \varepsilon > 0$, *there exists* $\delta > 0$ *such that if* $\|y - \bar{y}\| < \delta$, *then* $\max_{\forall U \in SOL(y)} \min_{\forall \bar{U} \in SOL(\bar{y})} \|U - \bar{U}\| < \varepsilon$; *and*
(c) τ *is a continuous function of* U.

Proof See Appendix 2. □

The first two conditions in Theorem 1 merits further discussions. Condition (a) states that for any given y in K_y, the lower level DUE must have at least one solution in terms of U; and if the solution is not unique, the solution set $SOL(y)$ must be compact. This result was previously established in Dafermos (1984) for static UE problems. Condition (b) can be roughly stated as "If two toll vectors are very close to each other, then their solution sets are also very close" (or if $y \to \bar{y}$, then $SOL(y) \to SOL(\bar{y})$), which is an intuitive assumption. We note here that although mathematically Theorem 1 can be more compactly expressed by "If $\tau(t)$ is continuous and $\Phi(y, U)$ is nonempty and bounded, then $BiDSBTP$ has a solution" (Luo et al. 1996), it is the authors' understanding that Theorem 1 connects more closely to $BiDSBTP$, especially the solution properties of the lower level DUE problem (notice that $\Phi(y, U)$ involves both the lower level and upper level problems). Therefore, Theorem 1 has more practical implications and is thus more preferable.

We also notice that Theorem 1 only depends on the solution properties of DUE and the continuity of the link trave time function, not on the specific DUE formulations. As a result, Theorem 1 actually establishes the solution existence condition for DSBTP that can incorporate a broad range of DUE models as long as the three conditions in the theorem are satisfied. In the next section, we show that the nonlinear complementarity DUE model employed in this paper for *BiDSBTP* satisfies the three conditions in Theorem 1. Therefore, at least one solution exists for such a DSBTP model.

3 A link-node NCP model for dynamic user equilibria

In this paper, we adopt the link-node based NCP formulation for the DUE constraint in bilevel model *BiDSBTP*, which was first discussed in Ban et al.

(2008). We only provide main results of the model in this section and the detailed derivations can be found in Ban et al. (2008).

To discuss the DUE model, we introduce additional notation. First, u_{as}^k represents the destination-specific inflow rate to link a with respect to destination s at the *beginning* of time k. We have $U_a^k = \sum_{s \in S} u_{as}^k$. Also, d_{is}^k is the travel demand from node i to destination s at the *beginning* of time k, while π_{is}^k denotes the minimum travel time from node i to destination s at the *end* of time k. Note that to be consistent with the definition of τ, π is defined at the *end* of time intervals for the ease of applying the predictive DUE concept. Detailed discussions on this can be found in Heydecker and Verlander (1999); Ban et al. (2008). We set $\pi_{ss}^k = 0, \forall s \in S, 0 \le k \le K' - 1$, and $d_{ss}^k = 0, \forall s \in S, 1 \le k \le K'$. Denote vectors $u_s = (u_{as}^k)_{\forall a \in A, 1 \le k \le K'}$, $\pi_s = (\pi_{is}^k)_{\forall i \in N, i \ne s, 1 \le k \le K'}$, and $d_s = (d_{is}^k)_{\forall i \in N, i \ne s, 1 \le k \le K'}$ for destination-specific variables. Therefore, $U = \sum_{s \in S} u_s$. We also define $u = (u_s)_{\forall s \in S}$, $\pi = (\pi_s)_{\forall s \in S}$, $d = (d_s)_{\forall s \in S}$. Next define the exit time function $e_a^k(u) = (k-1)\Delta + \tau_a^{k-1}(u)$ as the exit time of vehicles entering link a at the beginning of time k, which is a function of the link inflow vector u. Further, denote l_a, h_a the tail (starting) and head (ending) nodes of link a, respectively.

We also define three types of indicator functions on u: $\lambda_a^{1,k'}(u)$, $\lambda_a^{2,k,k'}(u)$, and $\lambda_a^{3,k,l}(u)$. First, flow propagation constraints can be represented by $\lambda_a^{1,k'}(u)$, which is defined as:

$$\lambda_a^{1,k'}(u) = \frac{\Delta}{\tau_a^{k'}(u) - \tau_a^{k'-1}(u) + \Delta}, \forall a, k'. \tag{6}$$

The relation between link inflow and exit flow rates can be represented by both $\lambda_a^{1,k'}(u)$ and $\lambda_a^{2,k,k'}$, with the latter defined as:

$$\lambda_a^{2,k,k'}(u) = \frac{\tau_a^{k'}(u) + (k' + 1 - k)\Delta}{\tau_a^{k'}(u) - \tau_a^{k'-1}(u) + \Delta}, \forall a, k, k', e_a^{k'} \le (k-1)\Delta < e_a^{k'+1}. \tag{7}$$

Similarly, $\lambda_a^{3,k,l}$ is used to discretize (and interpolate) the minimum travel time $\pi_{h_a s}(t + \tau_a(t))$. It can be defined as:

$$\lambda_a^{3,k,l}(u) = l - k - \tau_a^k(u)/\Delta, \forall l - 1 \le e_a^{k+1}/\Delta < l. \tag{8}$$

The three indicator functions satisfy $\lambda_a^{1,k'}(u) > 0, \forall a, k'$, $0 < \lambda_a^{2,k,k'}(u) \le 1, \forall a, k, k', e_a^{k'} \le (k-1)\Delta < e_a^{k'+1}$, and $0 < \lambda_a^{3,k,l}(u) \le 1, \forall l - 1 \le e_a^{k+1}/\Delta < l$. More detailed discussions on these indicator functions can be found in Ban et al. (2008). Based on the above notation, we can define two vector functions:

$$F_1(u, \pi) = \left(\tau_a^k(u) + \sum_{l-1 \le e_a^{k+1}(u)/\Delta < l} \left\{ \lambda_a^{3,k,l}(u) \pi_{h_a s}^l + \left[1 - \lambda_a^{3,k,l}(u)\right] \pi_{h_a s}^{l+1} - \pi_{l_a s}^k \right\} \right)_{\forall a,s,k} \tag{9}$$

$$F_2(u,\pi) = \left(\sum_{a \in A(i)} u_{as}^k - d_{is}^k - \sum_{a \in B(i)} \left\{ \sum_{k': e_a^{k'}(u) \le (k-1)\Delta < e_a^{k'+1}(u)} \left[\lambda_a^{1,k'}(u) \lambda_a^{2,k,k'}(u) u_{as}^{k'} + \lambda_a^{1,k'+1}(u) \left(1 - \lambda_a^{2,k,k'}(u)\right) u_{as}^{k'+1} \right] \right\} \right)_{\forall i,s,i \neq s,k} \tag{10}$$

Although F_1 and F_2 above may seem complicated in the formulations, they actually have straightforward meanings. First, F_1 is a vector function for any combination of link a, destination s, and time instant k. Each of its component represents, for the given a, s, k, the difference of the minimum travel times via two sets of paths from the starting node (or tail node) of link a to destination s at k. The first set of paths must traverse link a, which is a subset of the second set that includes all paths from the starting node of a to destination s. Clearly, this difference should be zero if the inflow to link a at time k is nonzero. On the other hand, F_2 simply represents the flow conservation constraint at any node i, destination $s \neq i$ and time k. Detailed discussion on how F_1 and F_2 are derived can be found in Ban et al. (2008).

The NCP formed DUE model can then be defined as follows (Ban et al. 2008):

$$0 \le u \perp F_1(u,\pi) \ge 0 \tag{11}$$
$$0 \le \pi \perp F_2(u,\pi) \ge 0 \tag{12}$$

Here "$\perp$" reads as "perpendicular" such that $x \perp y = 0 \Longleftrightarrow x^T y = 0$. Denote the above NCP model as *NCPDUE(0)*, and "0" represents the fact that no toll is imposed.

The following theorem is established in Ban et al. (2008). We list it here without proof.

Theorem 2 *NCPDUE*(*0*) *has a nonempty and compact solution set if the following four conditions are satisfied.*

(a) *The link travel time function is positive and finite for any finite u*;
(b) *The* (*fixed*) *demand is nonnegative and bounded from above*;
(c) *λ^1 is bounded from above*; *and*
(d) *F_1 and F_2 are continuous with respect to u and π.*

If toll is considered, the dynamic route choice will depend on both travel times and tolls. For this purpose, we define "generalized travel times," which can be defined as follows:

$$c_a^k(u, y) = \tau_a^k(u) + y_a^{k+1}/\theta, 0 \le k \le K' - 1. \tag{13}$$

Here $c_a^k(u, y)$ is the generalized travel time for link a at the *end* of time k which is a function of u and y, and θ is a (assumed to be fixed) parameter representing the "value of time." Further define ρ_{is}^k as the minimum generalized travel time from node i to destination s at the *end* of time k. Similar to Eqs. (9) and (10), we can define two vector functions:

$$F_u(u, \rho) = \Bigg(\tau_a^k(u) + y/\theta + \sum_{l-1 \le e_a^{k+1}(u)/\Delta < l} \left\{\lambda_a^{3,k,l}(u)\rho_{h_a s}^l + \left[1 - \lambda_a^{3,k,l}(u)\right]\rho_{h_a s}^{l+1} - \rho_{l_a s}^k\right\}\Bigg)_{\forall a,s,k} \tag{14}$$

$$F_\rho(u, \rho) = \Bigg(\sum_{a \in A(i)} u_{as}^k - d_{is}^k - \sum_{a \in B(i)} \Bigg\{ \sum_{k': e_a^{k'}(u) \le (k-1)\Delta < e_a^{k'+1}(u)} \Big[\lambda_a^{1,k'}(u)\lambda_a^{2,k,k'}(u)u_{as}^{k'} + \lambda_a^{1,k'+1}(u)\left(1 - \lambda_a^{2,k,k'}(u)\right)u_{as}^{k'+1}\Big]\Bigg\}\Bigg)_{\forall i,s,i \ne s,k} \tag{15}$$

Define vectors $\rho_s = (\rho_{is}^k)_{\forall i \in N, i \ne s, 0 \le k \le K'-1}$ and $\rho = (\rho_s)_{\forall s \in S}$. The DUE model with toll vector y can be expressed as:

$$0 \le u \perp F_u(u, \rho) \ge 0 \tag{16}$$

$$0 \le \rho \perp F_\rho(u, \rho) \ge 0 \tag{17}$$

Denote the above model as *NCPDUE(y)* to capture the fact that a toll vector y is imposed. We can easily observe that if y is fixed, *NCPDUE(y)* has the exact structure of *NCPDUE(0)*. Therefore, we have the following corollary:

Corollary 1 *For any given toll pricing vector $y \in K_y$,* NCPDUE(y) *has a nonempty and compact solution set if the four conditions in Theorem 2 are satisfied. In particular, condition (d) is rephrased as: $F_u(u, \rho)$ and $F_\rho(u, \rho)$ are continuous with respect to both u and ρ.*

Proof Since $y \in K_y$ is fixed and K_y is bounded, *NCPDUE(y)* has the same structure of *NCPDUE(0)*. Therefore, the proof in Ban et al. (2008) for Theorem 2 also applies here. □

One can easily observe that since $F_u(u, \rho)$ and $F_\rho(u, \rho)$ are continuous with respect to both u and ρ, we have condition (b) in Theorem 1 satisfied. In addition, because condition (a) in Theorem 1 is also met as shown in Corollary 1 and $\tau(u)$ is assumed to be a continuous function of u, we can conclude that *NCPDUE(y)* satisfies the three conditions in Theorem 1. Therefore, the bilevel *BiDSBTP* model will have at least one solution if *NCPDUE(y)* is used as the DUE constraint Eq. (4).

4 A single level model for dynamic second best toll pricing

4.1 Single level NLP model for DSBTP

Substitute the DUE constraint in Eq. (4) using *NCPDUE(y)* and notice the relation between U and u, the bilevel DSBTP can be formulated as:

$$\min_{y,U} \quad f(y, U) = \Delta U^T W \tau(U) \tag{18}$$

$$\text{s.t.} \quad y \in K_y \tag{19}$$

$$U = \sum_{s \in S} u_s \tag{20}$$

$$0 \le u \perp F_u(u, \rho) \ge 0 \tag{21}$$

$$0 \le \rho \perp F_\rho(u, \rho) \ge 0 \tag{22}$$

The above bilevel formulation can be easily converted into a single level NLP as follows:

$$SiDSBTP \qquad \min_{y,U} \quad f(y, U) = \Delta U^T W \tau(U) \tag{23}$$

$$\text{s.t.} \quad y_l \le y \le y_u \tag{24}$$

$$U = \sum_{s \in S} u_s \tag{25}$$

$$u \ge 0 \tag{26}$$

$$F_u(u, \rho) \ge 0 \tag{27}$$

$$\rho \ge 0 \tag{28}$$

$$F_\rho(u, \rho) \ge 0 \tag{29}$$

$$u^T F_u(u, \rho) \leq 0 \tag{30}$$

$$\rho^T F_\rho(u, \rho) \leq 0 \tag{31}$$

Clearly, Eqs. (26)–(31) are equivalent to Eqs. (21)–(22), and *SiDSBTP* is a special case of *BiDSBTP* by applying the nonlinear complementarity DUE model. Therefore, besides Theorem 1, the solution existence of *SiDSBTP* can also be achieved via the observation that *SiDSBTP* is an NLP. In particular, *SiDSBTP* has at least one solution because 1) the objective function is continuous, 2) the constraint set is nonempty as proved in Corollary 1, and 3) the constraint set is also closed and bounded since all functions are continuous.

4.2 A relaxation solution algorithm

The single level model *SiDSBTP* is clearly non-convex due to the complementarity constraints (30) and (31). More importantly, certain constraints qualification such as the Magasarian Fromovitz Constraint Qualification (MFCQ) does not hold for the single level model because of Eqs. (30) and (31), as stated in Luo et al. (1996). Therefore, solving the single level model directly will impose difficulties to most existing NLP solution techniques. However, we observe that the single level model has a similar structure to that for the continuous network design problem studied by Ban et al. (2006). Therefore, in this paper, we adopt the relaxation algorithm to iteratively solve *SiDSBTP*. The key of the algorithm is to relax the strict complementarity constraints (30) and (31) using the following two conditions:

$$u^T F_u(u, \rho) \leq \sigma \tag{32}$$

$$\rho^T F_\rho(u, \rho) \leq \sigma \tag{33}$$

for some $\sigma > 0$. For a given σ, a corresponding relaxed sub-problem is defined. As shown in Ralph and Wright (2004), MFCQ does hold for the relaxed sub-problem Eqs. (23)–(29) and (32) and (33), implying that it can be solved by standard NLP solvers. Clearly, if σ is large, the sub-problem is easier to solve. Therefore, the algorithm always starts with some fairly large σ's which are subsequently reduced, controlled by a update factor (see the detailed description of the algorithm below). If σ's are reduced smoothly (i.e., using a relatively large update factor), the solution of one sub-problem can usually provide a "good" starting point to the next iteration. As a result, the relaxed sub-problems can be relatively easily solved even when σ is very small (in which case, the MFCQ is barely satisfied). These are the main motivations for converting the bilevel model to single one and then apply the relaxation scheme. Here we notice that such a relaxation scheme is heuristic and there is no guarantee that the algorithm will converge to the optimal solution or even converges. In practice, however, the algorithm has been often used for solving similar problems, see e.g. (Clegg et al. 2001; Raghunathan and Biegler 2003; Ban et al. 2006). We also notice that Ralph and Wright (2004) established conditions under which the relaxed problem as solved in each iteration of the relaxation scheme in this paper could converge to a stationary point of an MPEC. These conditions may be used to establish the convergence of the relaxation scheme. However, whether they can be applied to the DSBTP problem in this paper is still an open question.

In summary, the relaxation algorithm, denoted as Algorithm ***RELAX***, can be summarized as follows:

Algorithm RELAX:

Step 1 Initialization. Choose an initial relaxation parameter $\sigma^0 > 0$. Set the iteration limit M, update factor $0 < \mu < 1$, and $m = 0$.

Step 2 Major Iteration.

Step 2.1 Solve current relaxed single level NLP Eqs. (23)–(29) and (32) and (33) using the solution from last iteration as the starting point. Use σ^m as the relaxation parameter in Eqs. (32) and (33).

Step 2.2 Update and Move. If $m \leq M$, set $\sigma^{m+1} = \mu\sigma^m$, $m = m + 1$, and go to *Step 2.1*; otherwise, go to Step 3.

Step 3 Final Solve. Set the final relaxation parameter as σ^f which is a pre-defined value representing the desired solution accuracy. If it is successful, we obtain an optimal solution; otherwise, an approximate solution is achieved from the last run of *Step 2.2*.

Algorithm ***RELAX*** merits further discussions. First, the algorithm only provides an iterative framework for solving DSBTP. Depending on the form of the link travel time functions $\tau(U)$, the actual implementation of the algorithm may vary. In particular, if $\tau(U)$ can be analytically expressed by U (as what we will show in the numerical section of this paper), Algorithm ***RELAX*** can be readily implemented using standard NLP solvers to solve the relaxed single level problem in *Step 2.1*. Note that the solution from previous iteration not only provides a starting point in *Step 2.1*, it is also used to temporarily fix the three indicator functions $\lambda^1, \lambda^2, \lambda^3$ so that the relaxed sub-problem can be represented as a regular NLP.

There are four parameters that need to be determined before applying the Algorithm ***RELAX***: the initial relaxation parameter σ^0, final relaxation parameter σ^f, iteration limit M, and update factor μ. Among them, σ^f represents the desired solution accuracy which should be specified by users (usually 10^{-6} should suffice for most applications). Similarly, σ^0 is the initial allowed violation of the complementarity conditions in Eqs. (30) and (31). A larger σ^0 may help solve the initial relaxed problem more easily, but may also require more iterations to achieve a given desired accuracy σ^f. Furthermore, μ represents how fast the relaxation parameter is reduced. A small μ results in fewer iterations. But if the relaxation parameter reduces too abruptly, the relaxed sub-problem in *Step 2.1* may be more difficult to solve since the solution from the previous iteration may not be a good starting point. Lastly, M can be determined accordingly if σ^0, σ^f, μ are given to ensure smooth transition from σ^0 to σ^f. Note that for a given M, $M + 2$ sub-problems will be solved in the ***RELAX*** algorithm, therefore M can be calculated using the equation:

$$\sigma^f \mu^{M-2} = \sigma^0. \tag{34}$$

The selection of the four parameters, therefore, represents certain trade-off between solution efficiency and quality (i.e. whether the problem can be solved or not). The authors of the paper recently conducted extensive numerical experiments on using ***RELAX*** to solve the continuous network design problems (Ban et al. 2009). We found that $\mu = 0.3$ can be used and σ^0 is problem-specific which may be determined via solving an initial traffic user equilibrium problem.

5 Numerical studies

In this section, we present an example for computing the dynamic second best tolls on a small hypothetical network as depicted in Fig. 1. The network has two origins (Node 1 and 2) and one destination (Node 3), which was used in Ban et al. (2008). The length of each time interval is set as $\Delta = 0.25$ min (15 s) and other configuration data of the network is shown in Table 1.

The demand from Node 1 (or Node 2) to Node 3 is assumed to be:

$$d_{is}^k = 40 + 120\left(1 - \left(\frac{k - K/2}{K/2}\right)^2\right), \forall 1 \leq k \leq K. \tag{35}$$

Here $K = 40$ represents the total number of intervals during which there are traffic demands from the origins (Node 1 and 2) to the destination (Node 3).

So far we have not specified the link travel time function $\tau(U)$. Actually, our proposed model is quite general so that any form of link travel time function should be applicable (at least theoretically) as long as it is continuous with respect to link inflow U (also the destination-specific link inflow vector u). In this paper, to simplify our discussion, we choose the following linear form for the link travel time function:

$$\tau_a^k(U) = \alpha_a\left(1 + \beta_a^x x_a^{k+1}\right). \tag{36}$$

Here α_a and β_a^x are constants specific to each link. In particular, α_a denotes the link free flow travel time, and x_a^{k+1} represents link flow, i.e. the total

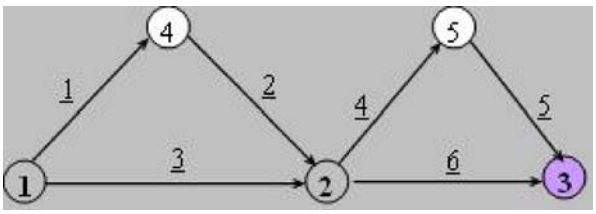

Fig. 1 Layout of the hypothetical network

Table 1 Data of the hypothetical network

Link	α_a (min)	β_a^x (min/vehicle)
1	1.2	0.01
2	1.2	0.01
3	1.8	0.0056
4	1.2	0.01
5	1.2	0.01
6	1.2	0.005

number of vehicles on link a at the *end* of time k (i.e., beginning of time $k+1$). As shown in Ban et al. (2008), the link flow $x = (x_a^k)_{\forall a \in A, 1 \leq k \leq K'}$ can be represented using the link inflow vector U. Therefore, in Eq. (36), the travel time of a link at a given time is a linear function of the link inflow at the same time. Furthermore, the link travel time function defined in Eq. (36) satisfies the First-In-First-Out (FIFO) constraints (Nie and Zhang 2005).

Since we assume that $\tau(U)$ can be analytically expressed by U, Algorithm ***RELAX*** can be straightforwardly implemented by directly applying standard NLP solvers. In this paper, the NLP solver CONOPT (Drud 1992) is used and we further choose $\sigma^0 = 10$, $\sigma^f = 10^{-6}$, $\mu = 0.3$. Using Eq. (34), M can be calculated as $M = 10$. We tested the example in this paper on a PC with an Intel(R) Core(TM)2 DUO 3.0 GHZ CPU and 3.25 GB memory under the Windows XP operating system.

We first show the DUE results without tolling. This is the base scenario. The convergence criteria is defined using a gap function as the difference of link inflows between two consecutive iterations (Ban et al. 2008). In particular, the DUE algorithm stops if the following condition holds:

$$Gap\_U = |U^n - U^{n+1}|_2 \leq \varepsilon \tag{37}$$

Here ε represents the solution accuracy and we choose $\varepsilon = 10^{-5}$ in this paper. Figure 2 depicts the convergence of the DUE algorithm which converges after 20 iterations. The CPU time for solving the DSBTP model is 459.3 s.

Figures 3 and 4 show, respectively, the time-varying inflow rates (U) and link flows (x) of all six links of the network. We first observe that since predictive DUE formulation is adopted in the DSBTP model, the inflows are generally smooth with small fluctuations. The figures also show that the link flows are even smoother than inflows. Furthermore, since Link 5(2) is the downstream of Link 4(1), the inflow rate to Link 5(2) is the exit flow of Link 4(1). We thus observe that the exit flow rate of one link is smoother than the inflow rate of the same link (Ban et al. 2008; Carey et al. 2003). For example, the left dashed circle indicates a relatively large drop of inflow to Link 4 from 100 at $k = 40$ to 78 at $k = 41$. This drop, however, is smoothed in the exit flow (inflow to Link 5) as indicated using the right dashed circle (from 108 at $k = 45$ to 95 at $k = 46$, and then to 81 at $k = 47$).

Figures 5 and 6 further illustrate the DUE route choice conditions from Node 1 to Node 3 and from Node 2 to Node 3, respectively. In these two figures, "Min Travel Time via Link a" represents the minimum travel time from the tail node of Link a to the destination (Node 3) among all paths that

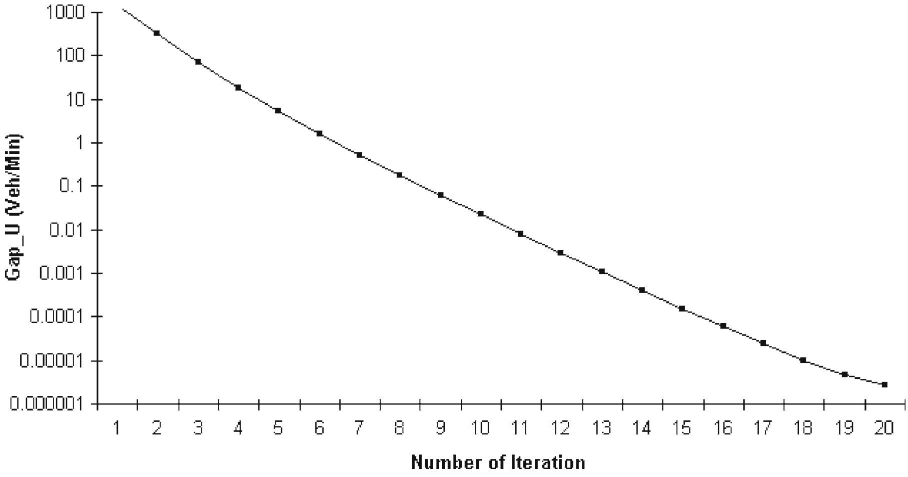

Fig. 2 Convergence of DUE solution

traverse Link a. Figure 5 shows that the travel time via Link 3 represents the minimum travel time from Node 1 to Node 3. As a result, Link 3 carries inflows during the entire study period. At time interval $k = 9$ (or $k = 37$) the minimum travel time via Link 1 becomes the same as (or larger than again) that via Link 3. Therefore, the inflow to Link 1 is positive between $k = 9 - 37$. These two transition points are indicated using dashed lines in Fig. 5. For demands from Node 2 to 3, the minimum travel times are equal via Link 4 and Link 6. Therefore, inflows to both links are positive for the study period. These two figures illustrate that DUE route choice are satisfied for the obtained solution.

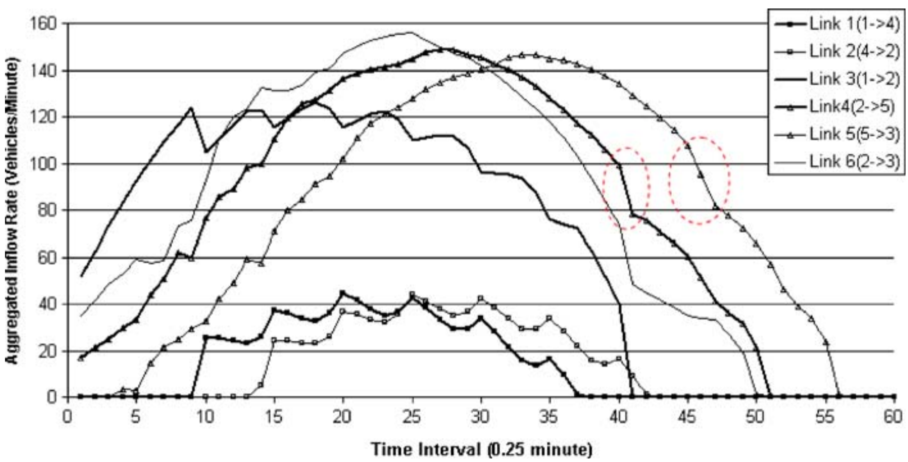

Fig. 3 DUE inflow rates(U)

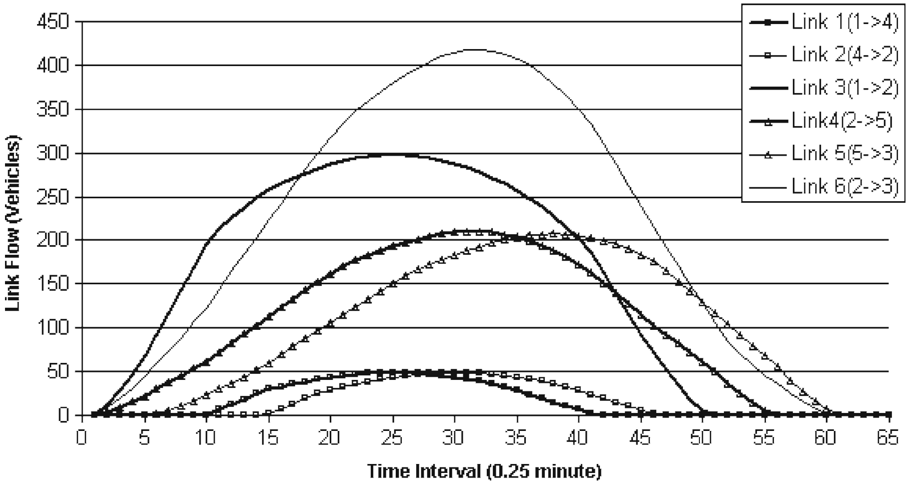

Fig. 4 DUE link flow (x)

We next show that if Link 3 (from Node $1 \rightarrow 2$) is tolled, what the optimal toll pricing will be. In the objective function, without loss of generality, we set the value of time $\theta = 1$ dollar/minute. We also set the weight for Link 3 as 1.6, while weights for other links are set as 1. We will show later how different weights of Link 3 will impact the optimal toll and its associated objective value.

Figure 7 first depicts the time-varying optimal tolls generated by the relaxation algorithm. The tolls change smoothly as time evolves. The resulting objective value for DSBTP is 166.1 vehicle-hours (VH). We can also compute

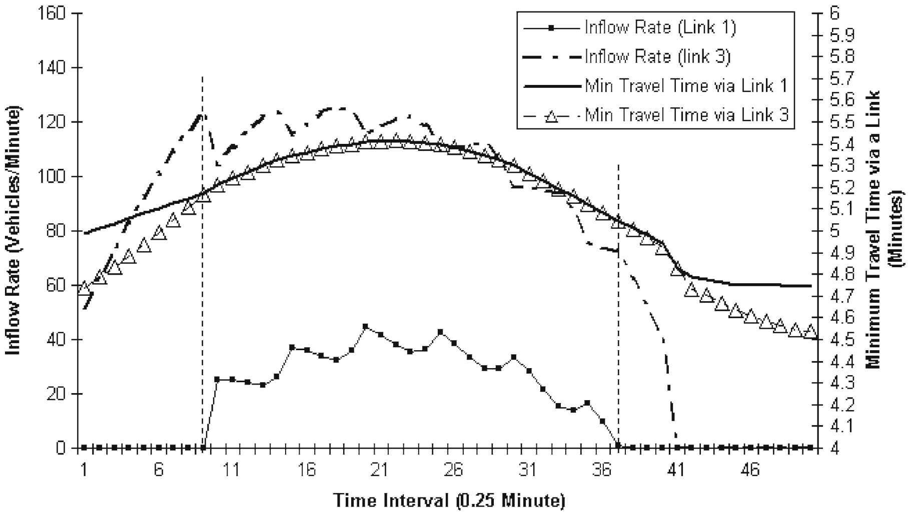

Fig. 5 DUE route choice condition from Node 1 to 3

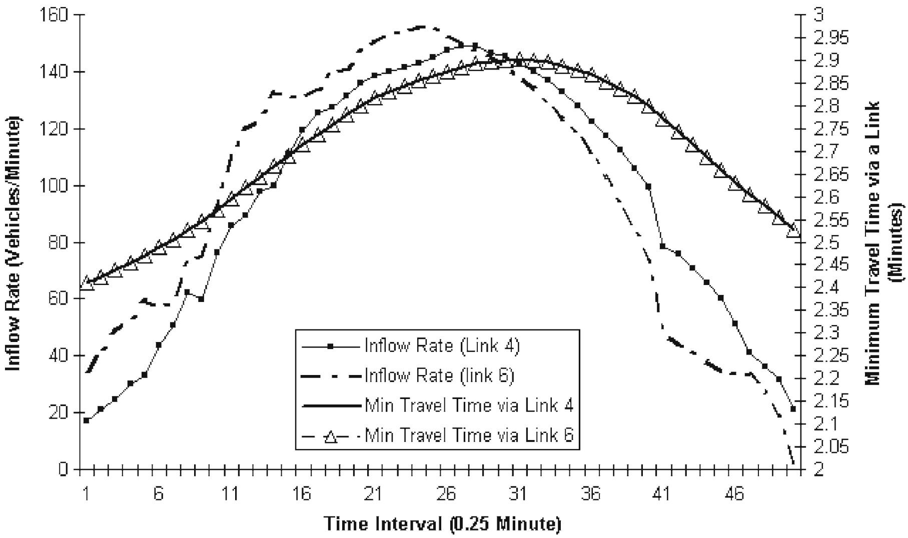

Fig. 6 DUE route choice condition from Node 2 to 3

the objective value for DUE as 184.2 VH. Therefore, if the optimal tolls in Fig. 7 are imposed, the objective value will be improved by 9.8%.

Figures 8 and 9 display, respectively, the inflow rates and link flows for all links if the optimal tolls in Fig. 7 are imposed. We can see that in this case the inflow rates to Link 1 is increased significantly, while the inflow to Link 3 is reduced substantially.

To illustrate that the DSBTP solution indeed satisfies the DUE route choice, we show in Figs. 10 and 11 the DUE conditions for demands from Node 1 to

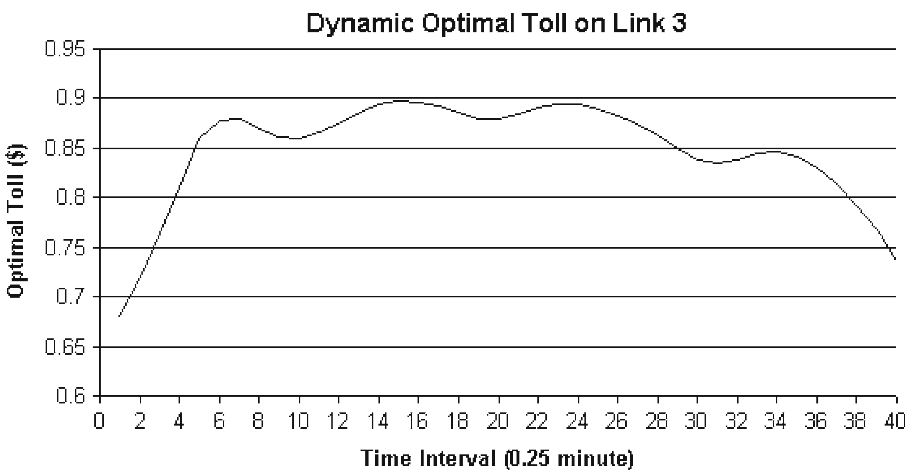

Fig. 7 Time-varying optimal toll

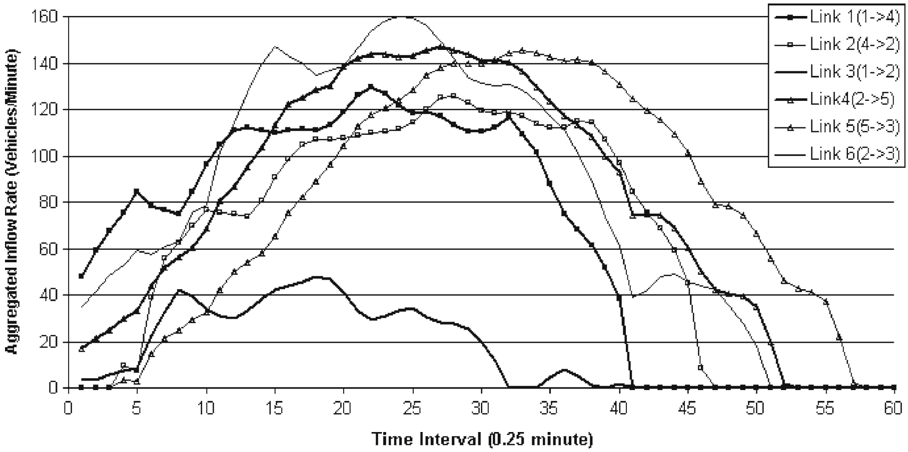

Fig. 8 Inflow rates (U) under optimal toll

3 and from Node 2 to 3, respectively. From these two figures, it is easy to see that the DUE route choice is satisfied for demands from Nodes 1 and 2 to the destination. For example, for demands from Node 1 to 3, since the minimum travel times via Link 1 and 3 are the same until after $k = 40$, the inflows to both links are positive for $k = 1 - 40$. For demands from Node 2 to 3, the

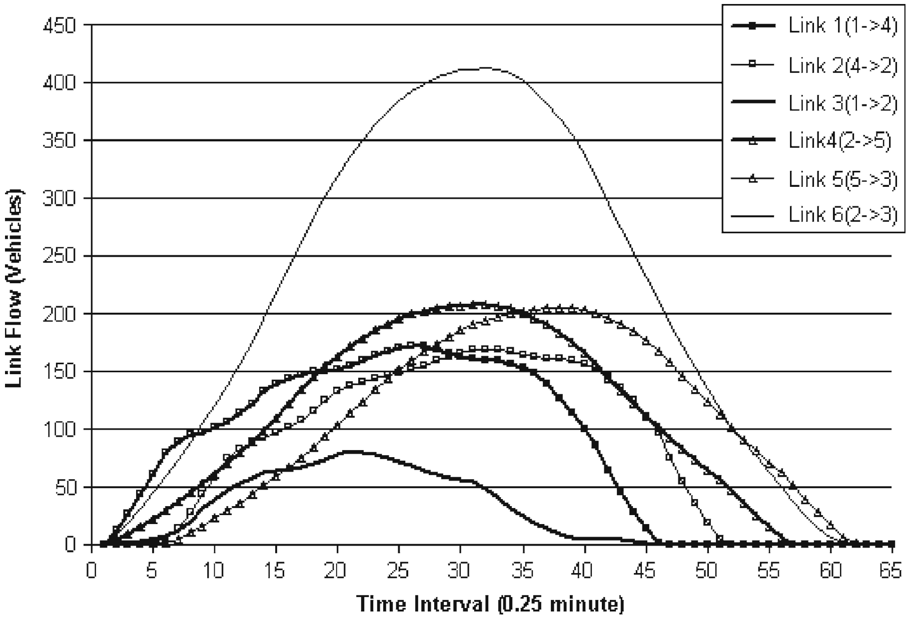

Fig. 9 Link flow (x) under optimal toll

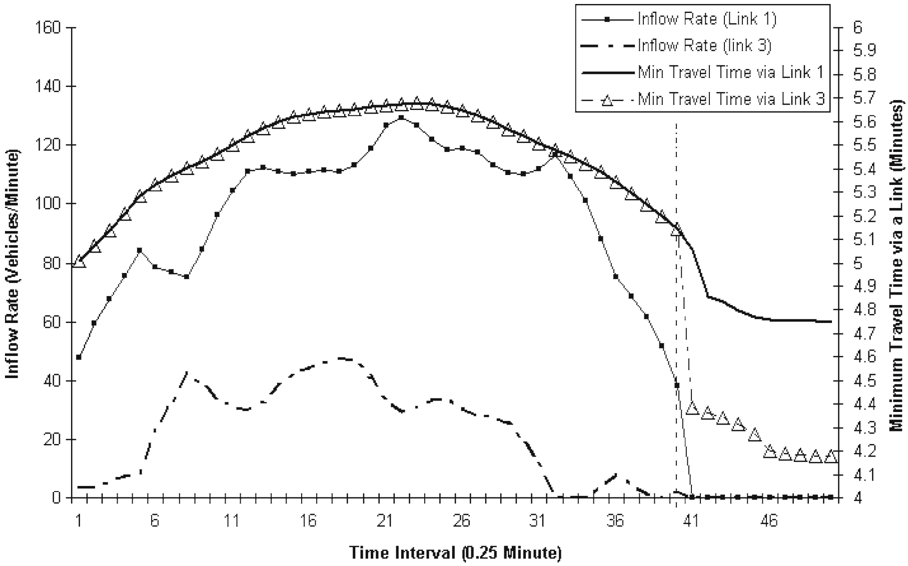

Fig. 10 Route choice condition for DSBTP solutions from Node 1 to 3

same result can be observed. Note that in Figs. 10 and 11, "Min. Travel Time" represents the minimum generalized travel time.

We then show how the weight of Link 3 impacts the computed optimal tolls and objective value. For this purpose, we vary the weight of Link 3 from 1.0

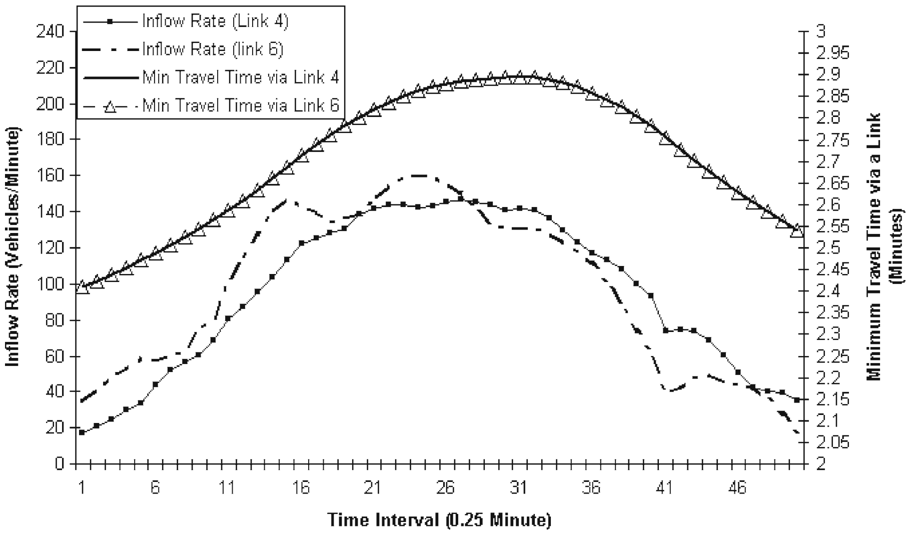

Fig. 11 Route choice condition for DSBTP solutions from Node 2 to 3

to 2.0 using 0.1 as the increment. Figure 12 depicts the changes of the time-varying optimal tolls generated by the DSBTP model when the weight of Link 3 is 1.0, 1.4, 1.6, and 2, respectively. We can observe that if the weight is small, e.g. 1.0, since Link 3 is shorter and the flows on Link 1 and 3 contribute equally to the objective function, tolls on Link 3 are not effective in this case to reduce the objective value. As a result, the computed tolls from the DSBTP are small and positive for only $k = 13 - 31$. However, as the weight increases, the flow on Link 3 contributes more to the objective function than other links. Consequently, tolls on Link 3 tend to help divert traffic from Link 3 to Link 1 more substantially. Therefore, the optimal toll increases if so does the weight of Link 3. This is shown in Fig. 12. Figure 13 further illustrates the inflow rates to Link 3 when the weight of Link 3 is 1.0, 1.4, 1.6, and 2, respectively. As one may expect, as the toll on Link 3 increases, inflow to Link 3 decreases in general.

Figure 14 shows the change of objective values vs. the weights on Link 3, for both the DUE and DSBTP solutions. The figure also depicts the improvement of the DSBTP objective over the DUE objective vs. the weights of Link 3. It is clear that as the weight increases from 1.0 to 2.0, the objective improvement of DSBTP increases monotonically from 4% to 16%.

Therefore, by imposing different weights on some link(s) of a network, dynamic tolls can help to divert traffic from one area to another in the network. This may help to resolve not only traffic congestion, but also other concerns from traffic management point of view. For example, if Link 3 is in a environmental sensitive area that desires less traffic, imposing more weights on Link 3 can generate dynamic toll pricing that helps to effectively reduce traffic in the area. Of course, a practical toll pricing scheme will need to address

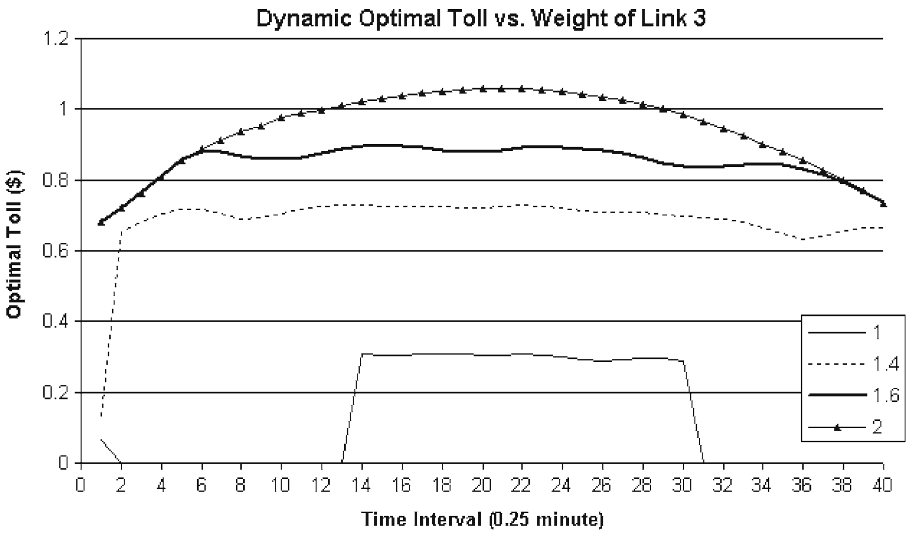

Fig. 12 Optimal toll vs. weight of link 3

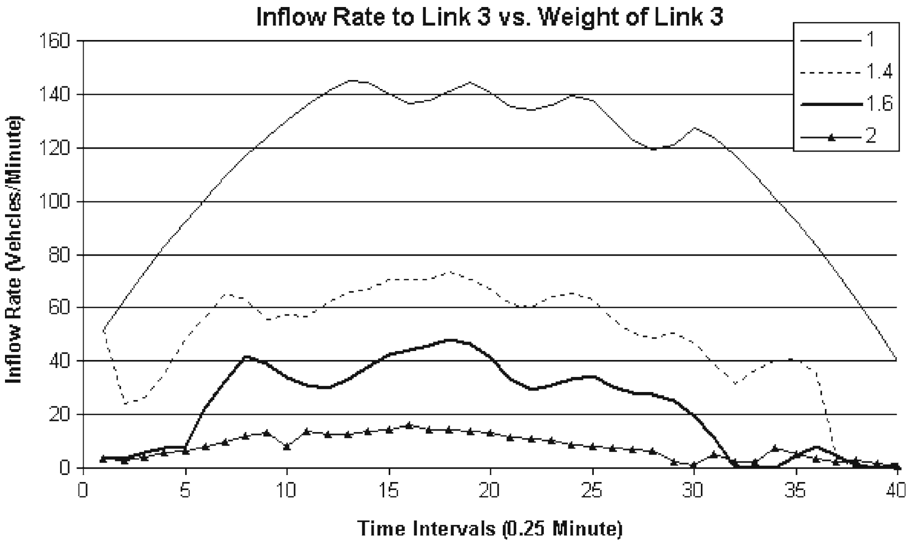

Fig. 13 Inflow rates to link 3 vs. weight of link 3

concerns from different aspects, such as congestion, environment, equity, etc. We hope, however, the DSBTP model and weighing scheme proposed here can provide some insights in this regard.

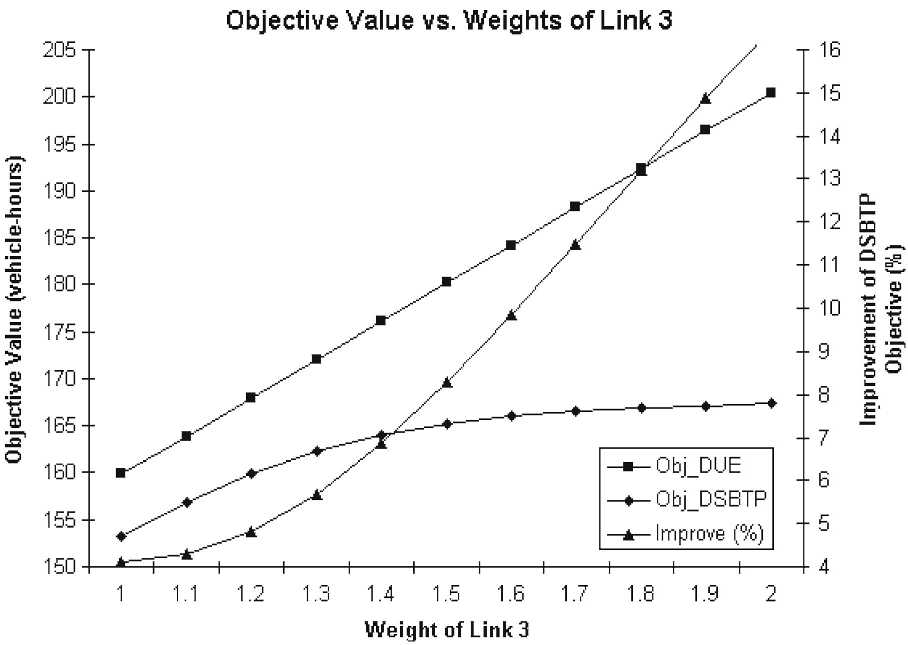

Fig. 14 Objective value vs. weight of link 3

6 Conclusion

In this paper, we studied the dynamic second best toll pricing (DSBTP) problem that aims to achieve certain traffic management objective, while still considering the choice behavior of individual motorists. To specifically account for motorists' roue choice decisions on general networks, we formulated the route choice as a dynamic user equilibrium (DUE) problem. The resulting model is then a bilevel formulation.

We first established conditions under which the proposed bilevel model has at least one solution. These conditions are mild, focusing on the solution properties of the lower level DUE and the continuity of the link travel time function, which makes it potentially applicable for a wide range of traffic flow models. In particular, we observed that the link-node DUE model developed recently satisfies the required conditions and was thus adopted in this paper for the bilevel DSBTP model. We then showed that it is straightforward to convert the bilevel model to a single level problem, which can be solved by a relaxation algorithm, recently used by the same authors to solve static problems.

We also provided numerical examples in this paper to demonstrate the proposed model and algorithm. We showed that dynamic tolls are helpful to reduce total weighted system travel times. In particular, we illustrated that by imposing different weights on different links, DSBTP can help traffic management authorities to better achieve their specific goals in terms of toll pricing design.

In this paper, we assume fixed demands for the lower level UE. This assumption can be relaxed by incorporating departure-time choice into the link-node DUE formulation. The authors are investigating this issue and results will be reported in subsequent papers. The numerical experiments in this paper focus on a small test network; it is important to test the model and especially the relaxation algorithm on large-size networks. In addition, we only tested the proposed model by assuming that link travel time can be expressed as a linear function of link flows. Although this much simplifies the implementation of the solution algorithm, the actual formulation of link travel times may be much more complicated and has to be evaluated via simulation (in a micro-, meso- or macro-level). Exploring how these more practical models will impact the performance of the proposed model and algorithm should be further pursued.

Appendices

Appendix 1: Definition and properties of a point-to-set map

This appendix provides definitions and properties of a point-to-set map. Interested readers may refer to Facchinei and Pang (2003) for details and proofs.

Definition 1 A map Φ is a point-to-set map from $\mathbb{R}^n$ to $\mathbb{R}^m$ if for any $x \in \mathbb{R}^n$, $\Phi(x)$ is a subset of $\mathbb{R}^m$ (possibly empty). The domain of Φ, denoted by domΦ, the range of Φ, denoted by ranΦ, and the graph of Φ, denoted by gphΦ, are defined as follows:

$$\text{dom}\Phi \equiv \{x \in \mathbb{R}^n : \Phi(x) \neq \emptyset\} \tag{38}$$

$$\text{ran}\Phi \equiv \bigcup_{x \in dom\Phi} \Phi(x) \tag{39}$$

$$\text{ghp}\Phi \equiv \left\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m : y \in \Phi(x)\right\} \tag{40}$$

Definition 2 A point-to-set map $\Phi : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is said to be

(a) *closed* at at point $\bar{x}$ if

$$\left.\begin{array}{c} x^k \longrightarrow \bar{x} \\ y^k \in \Phi(x^k)\ \forall k \\ y^k \longrightarrow \bar{y} \end{array}\right\} \Longrightarrow \bar{y} \in \Phi(\bar{x})\,;$$

(b) *closed* on a set S if Φ is closed at every point of S.
(c) *upper semicontinuous* at a point $\bar{x}$ if for every open set υ containing $\Phi(\bar{x})$, there exists an open neighborhood $\mathcal{N}$ of $\bar{x}$ such that, for each $x \in \mathcal{N}$, υ contains $\Phi(x)$.
(d) *lower semicontinuous* at a point $\bar{x}$ if for every open set υ meeting $\Phi(\bar{x})$, there exists an open neighborhood $\mathcal{N}$ of $\bar{x}$ such that, for each $x \in \mathcal{N}$, υ meets $\Phi(x)$.

Theorem 3 *The following statements are true for a point-to-set map Φ.*

(a) *Suppose $\Phi(\bar{x})$ is a closed set. If Φ is upper semicontinuous at $\bar{x}$, then Φ is closed at $\bar{x}$;*
(b) *Φ is closed if and only if its graph is a closed set.*

Appendix 2: Proof of Theorem 1

Under condition (b) of Theorem 1, the point-to-set mapping of $SOL(y)$ is upper-semicontinuous (see Definition 2(c) in Appendix 1). To see this, assume $\upsilon = SOL(\bar{y}) + IB(0, \varepsilon)$ is an open set containing $SOL(\bar{y})$, where $IB(0, \varepsilon)$ is an open ball with radius ε. We then define another open set $\mathcal{N} \equiv \{y | \|y - \bar{y}\|_2 < \delta\}$ containing $\bar{y}$. According to condition (b) in the theorem, for any $\varepsilon > 0$, there exists $\delta > 0$ such that $\max_{\forall U \in SOL(y)}\ \min_{\forall \bar{U} \in SOL(\bar{y})} \|U - \bar{U}\| < \varepsilon$, which is equivalent to $\bigcup_{y \in \mathcal{N}} SOL(y) \subseteq \upsilon$.

Then together with condition (a), it implies that the point-to-set mapping $SOL(y)$ is closed on set K_y (see Theorem 3(a) and Definition 2(b) in Appendix 1). Therefore, the graph $\Phi(y, U)$ defined in (5) is closed (see Theorem 3(b) in the Appendix). Also under (a), $SOL(y)$ is bounded for any $y \in K_y$. Since K_y is a bounded set, the graph $\Phi(y, U)$ is bounded as well. Thus, we proved that graph $\Phi(y, U)$ is compact. Together with (c) and the fact that K_y is compact, *BiDSBTP* must has at least one solution since it is an NLP with a continuous objective function defined on a compact constraint set (i.e. from Weierstrass' Theorem). □

References

Arnott R, de Palma A, Lindsey R (1998) Recent developments in the bottleneck model. In: Button KJ, Verhoef ET (eds) Road pricing, traffic congestion and the environment: issues of efficiency and social feasibility. Edward Elgar, Aldershot, pp 79–110

Ban X, Liu HX, Ferris MC, Ran B (2006) A general mpcc model and its solution algorithm for continuous network design problem. Math Comput Model 43:493–505

Ban X, Liu HX, Ferris MC, Ran B (2008) A link-node complementarity model and solution algorithm for dynamic user equilibria with exact flow propagations. Transp Res Part B 42(9): 823–842

Ban X, Ferris MC, Liu HX (2009) An MPCC formulation and numerical studies for continuous network design with asymmetric user equilibria. Submitted for publication

Carey M, Srinivasan A (1993) Externalities, average and marginal costs, and tolls on congested networks with time-varying flows. Oper Res 41(1):217–231

Carey M, Ge YE, McCartney M (2003) A whole-travel time model with desirable properties. Transp Sci 37(1):83–96

Chu X (1995) Endogenous trip scheduling: the henderson approach reformulated and compared with the vickrey approach. J Urban Econ 37:324–343

Clegg J, Smith M, Xiang Y, Yarrow R (2001) Bilevel programming applied to optimising urban transportation. Transp Res Part B 35(1):41–70

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

Drud AS (1992) Conopt—a large-scale grg code. ORSA J Comput 6:207–216

Facchinei F, Pang JS (2003) Finite-dimensional variational inequalities and complementarity problems, vol I, II. Springer, New York

Federal Highway Administration (FHWA) (2007) Congestion pricing. Technical report, Publication number: FHWA-HOP-07-074

Friesz TL, Bernstein D, Kydes N (2002) Dynamic congestion pricing in disequilibrium. Netw Spat Econ 4:181–202

Friesz TL, Kwon C, Mookherjee R (2007) A computable theory of dynamic congestion pricing. In: Proceedings of 17th international symposium on traffic and transportation theory, London, 23–25 July 2007

Han S (2003) Dynamic traffic modeling and dynamic stochastic user equilibrium assignment for general road networks. Transp Res Part B 37(3):225–249

Hearn DW, Ramana MV (1998) Solving congestion toll pricing models. In: Marcotte P, Nguyen S (eds) Equilibrium and advanced transportation modeling. Kluwer Academic, Dordrecht, pp 109–124

Heydecker BG, Verlander N (1999) Calculation of dynamic traffic equilibrium assignments. In: Proceedings of the european transport conferences, seminar F, pp 79–91

Kuwahara M (2007) A theory and implications on dynamic marginal cost. Transp Res Part A 41(7):627–643

Lawphongpanich S, Hearn D (2004) An mpec approach to second-best toll pricing. Math Program B 101:33–55

Liu LN, McDonald, JF (1999) Economic efficiency of second-best congestion pricing schemes in urban highway systems. Transp Res B 33:157–188

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

Nie XJ, Zhang HM (2005) Delay-function-based link models: their properties and computational issues. Transp Res B 39:729–751

Raghunathan AU, Biegler LT (2003) Interior point method for mathematical programs with complementarity constraints. Technical report, Department of Chemical Engineering, Carnegie Mellon University

Ralph D, Wright SJ (2004) Some properties of regularization and penalization schemes for MPECs. Optim Methods Softw 19:527–556

Sumalee A (2007) Multi-concentric optimal charging cordon design. Transportmetrica 3(1):41–71

Wie BW, Tobin RL (1998) Dynamic congestion pricing models for general traffic networks. Transp Res B 32(5):313–327

Wie BW, Tobin RL, Carey M (2002) The existence, uniqueness and computation of an arc-based dynamic network user equilibrium formulation. Transp Res B 36:897–918
Yang H, Huang HJ (1997) Analysis of the time-varying pricing of a bottleneck with elastic demand using optimal control theory. Transp Res B 31:425–440
Yang H, Huang HJ (2005) Mathematical and economic theory of road pricing. Elsevier, Amsterdam