Risk-Neutral Second Best Toll Pricing

Xuegang (Jeff) Ban∗ Michael Ferris† Lisa Tang‡ Shu Lu§

Abstract

We propose a risk-neutral second best toll pricing (SBTP) scheme to account for the possible nonuniqueness of user equilibrium solutions. The scheme is designed to optimize for the expected objective value as the UE solution varies within the solution set. We show that such a risk-neutral scheme can be formulated as a stochastic program, which complements the traditional risk-prone SBTP approach and the risk-averse SBTP approach we developed recently. The proposed model can be solved by a simulation-based optimization algorithm that contains three major steps: characterization of the UE solution set, random sampling over the solution set, and a two-phase simulation optimization step. Numerical results illustrate that the proposed risk-neutral design scheme is less aggressive than the risk-prone scheme and less conservative than the risk-averse scheme, and may thus be more preferable from a toll designer’s point of view.

1 Introduction

Congestion pricing charges motorists a toll for using a particular stretch of highway or bridge or for entering a particular area (e.g., “cordon tolls” for access to urban areas). It is to apply economic principles (i.e., pricing) to solve traffic congestion or related problems (Knight, 1924; Pigou, 1932; Vickrey, 1969; Yang and Huang, 2005). Congestion pricing is considered as one of the most promising approaches to address traffic congestion (or other related problems) that has become not only an increasingly critical problem to our quality of life but also has serious consequences in terms of economic development (FHWA, 2007). In this article, we focus on link-based toll pricing, i.e., to determine optimal tolls of the entire set or a subset of network links to achieve certain system management objective 1. Such problems can be broadly categorized as static pricing and dynamic pricing. This article focuses on the former, while the reader can refer to Friesz et al. (2007); Lu et al. (2008); Ban and Liu (2009) for recent advances on dynamic toll pricing. Static toll pricing can be further classified as first best toll pricing (FBTP) and second best toll pricing (SBTP). FBTP assumes that all network links can be tolled. Early works on FBTP aimed to determining optimal tolls by applying economic principles such as marginal cost (Beckmann et al., 1956; Arnott, 1979; Smith, 1979). Hearn and Ramana (1998) formulated FBTP

∗Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute, banx@rpi.edu
†Computer Sciences Department, University of Wisconsin at Madison, ferris@cs.wisc.edu
‡Industrial and Systems Engineering Department, University of Wisconsin at Madison, lmtang@wisc.edu
§Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, shulu@email.unc.edu

1Note that this does not necessarily mean that the resulting network model has to be link-based. In certain cases, path-based formulations have to be used; see the second example in Section 6.
as network equilibrium models and found that other optimal toll schemes exist, including the marginal cost toll as a special case.

This article focuses on SBTP, which assumes that tolls can only be imposed on a subset of network links due to technical, policy, or other constraints. It was noted in Johansson-Stenman and Sterner (1998) that SBTP is a trade-off between efficiency gains and system investment/operation costs. The SBTP with two-one level problem, one tolled and one un-tolled, was studied in Marchand (1968); Verhoef et al. (1996); Liu and McDonald (1999). Yang and Lam (1996) proposed a bilevel model to determine optimal tolls for a subset of network links by considering queuing effects. Since then, SBTP has been extensively studied (Verhoef, 2002a,b; Zhang and Ge, 2004; Larsson and Patriksson, 1998; Brotoce et al., 2001). In particular, the elastic demand SBTP problems were investigated in Verhoef (2002a,b). Different design objectives such as those of the system manager and road users were explored in Zhang and Ge (2004) for variable (i.e., elastic) demand SBTP. SBTP with stochastic user equilibrium in the lower level was also studied in Sumalee et al. (2006). Furthermore, while most of existing SBTP models require both traffic demands and link costs as given (either fixed or as a given function), Yang et al. (2009) recently proposed a pricing scheme that can determine the optimal tolls for a set of links without knowing the exact demand or link cost information. For more comprehensive reviews on SBTP, readers may refer to Yang and Huang (2005); Lawphongpanich et al. (2006).

Most existing SBTP models on general traffic networks are formulated as bilevel problems (Bard, 1998) or mathematical programs with equilibrium constraints (MPEC, see Luo et al. (1996)): the upper level is to optimize a system objective, while the lower level is to solve an user equilib-rium (UE) problem to account for drivers’ choice behaviors (such as route choice). Although there are significant variations among the existing SBTP models, in essence, SBTP can be formulated as the following general form:

\[
\begin{align*}
\min_{y,x} & \quad Q(y,x), \\
\text{subject to} & \quad y \in K_y \\
& \quad x \text{ solves } VI(V(y,x), K_x)
\end{align*}
\]

(1)

Here \(VI(V(y,x), K_x)\) denotes a variational inequality (VI) defined as “finding \(x \in K_x\) such that \(V^T(y,x)(x' - x) \geq 0, \forall x' \in K_x,\)” The two defining sets \(K_x\) and \(K_y\), usually convex and compact, are for the lower level variable \(x\) and the upper level variable \(y\) respectively. In the SBTP setting, \(x\) can be considered as the vector of link flows (or path flows), while \(y\) is the link toll vector. Notice that both the upper level system objective (i.e., \(Q(y,x)\)) and the defining function of the lower level VI (i.e., \(V(y,x)\)) are functions of \((y,x)\). Also \(Q(y,x)\) can take various forms besides the total system travel time depending on the system manager’s objective (Zhang and Ge, 2004). For example, a weighted total system travel time is used in Ban et al. (2009) and Ban and Liu (2009) by assigning different weights on different link travel times. In the second example in Section 6 of this article, the objective is to minimize total system emissions.

Model (1) is an MPEC, for which theories and solution algorithms have been extensively studied in the mathematical programming literature. See Luo et al. (1996) for a review, and Ralph and Wright (2004) and Ferris (2004) for more recent advances on theories and solution techniques respectively. Loridan and Morgan (1988) observed that in case the lower level problem (formulated as a nonlinear program (NLP) in Loridan and Morgan (1988) instead of a VI) has multiple solutions for a given \(y\), different lower level solutions may result in different upper level objective values. Model (1) in this case simply focuses on the VI solution that produces the minimum of all these objective values. Model (1) was then called the optimistic design scheme (Loridan and Morgan, 1988) as it aims to optimize for the best case scenario, i.e., the minimum
of the objectives as VI solution varies within its solution set. Accordingly, a pessimistic design scheme was proposed (Loridan and Morgan, 1988) to optimize for the worst case scenario, i.e., the maximum of the objectives as VI solution varies. Ban et al. (2009) explicitly studied this issue for SBTP and observed that if the UE solution is nonunique, different UE solutions may result in different upper level objective values depending on the specific functional form of $Q(y, x)$. As a result, the UE solution set represents an uncertainty set from the toll designer’s perspective when designing optimal tolls (Ban et al., 2009). In this sense, most existing SBTP models is risk-prone or optimistic if UE solution is nonunique (unless the upper level objective admits a unique value over the entire UE solution set). To address this issue, a risk-averse SBTP scheme was developed in Ban et al. (2009) to optimize for the maximum of the objective values as UE solution varies, similar to the pessimistic scheme proposed in Loridan and Morgan (1988). The risk-averse scheme in Ban et al. (2009) however is based on a VI formulation in the lower level, which extends the scheme in Loridan and Morgan (1988) that is based on an NLP in the lower level.

The risk-averse SBTP scheme in Ban et al. (2009) is formulated as a robust optimization problem; similar schemes have also been applied recently to bounded rationality UE (BRUE) (Lou et al., 2010) and dynamic network design problems (Chung et al., 2011). It is well known that the results obtained by solving a robust optimization model is generally too conservative (or too pessimistic) (Yin and Lawphongpanich, 2007), while the risk-prone scheme is too aggressive (or too optimistic). This is because both schemes consider only extreme cases, i.e. a single point in the UE solution set. To address this issue, we propose in this article a “risk-neutral” scheme that explicitly considers the entire UE solution set when designing SBTP. The risk-neutral scheme aims to optimize for the expected objective value as the UE solution changes within the solution set. In particular, by associating certain probability distribution function to the realization of the UE solution over its solution set, we formulate the proposed risk-neutral model as a stochastic program, similar to the problem studied in Deng (2007) and Deng and Ferris (2009). In this setting, the UE solution set and the set of its subsets constitute the sample space. In Deng (2007) and Deng and Ferris (2009), the sample space is fixed. Our risk-neutral problem however has a changing sample space because the UE solution set varies with the toll vector. Hence, the risk-neutral model we study in this article extends the model and results in Deng (2007); Deng and Ferris (2009).

The reason that the issue of nonunique UE solutions in SBTP design was not fully recognized in the past is partially due to the fact that most existing SBTP models are formulated as link-based and assume explicitly or implicitly that UE has a unique solution (e.g., the commonly used BPR function to calculate link travel times) or the objective function admits a unique value over the entire UE solution set (Yang et al., 2004). In these cases, the risk-averse, risk-prone, and risk-neutral schemes coincide. However, there are at least two situations that may make the three schemes possibly different:

- **Nonunique Link Flow Solution.** When link interactions need to be considered (e.g., for urban streets), the BPR function may not be appropriate as link travel times may not be separable. Mathematically, strict monotonicity may not hold and multiple link-based solutions may exist. Depending on the specific functional form of the upper level objective, different objective values may exist for different UE solutions. We illustrate this using the first case study in Section 6. When multiple vehicle classes are considered, vehicle class based link flow is generally nonunique. In this case, the risk-neutral SBTP scheme discussed in this article (and the risk-averse scheme in Ban et al. (2009)) can also apply.

- **Nonunique Path Flow Solution.** It is well-known that path flows are nonunique even when the link travel time function is strictly or strongly monotone. In cases that path flows have to be considered directly in SBTP (e.g., for non-additive path costs cases; see Gabriel and Bernstein (1997); Lo and Chen (2000); Agdeppa et al. (2007)), different path flow solutions may result in different objective values; see the second case study in Section 6.
Notice that recently the BRUE (Mahmassani and Chang, 1987) was revisited (Lou et al., 2010; Guo and Liu, 2011). It was found that under this particular UE assumption, both link-based and path-based solutions are nonunique. Therefore the three risk-taking schemes discussed here can be similarly applied to BRUE. Actually the risk-averse (robust) congestion pricing scheme based on BRUE has been studied in Lou et al. (2010), although risk-neutral models in this particular case have not been discussed yet. The nonuniqueness of path flow solutions was also well recognized in the past, and was mainly addressed by imposing additional assumptions or constraints so that path flows can remain unique (Rossi et al., 1989; Oppenheim, 1995; Bell and Iida, 1997; Larsson et al., 2002). See also recent works in Bar-Gera and Boyce (1999) and Bar-Gera (2010) that attempt to provide behavioral explanations to these additional constraints. We should point out here that UE is just a modeling framework to real world network traffic flow. It is still unclear, to the authors’ understanding, whether a real traffic network will physically always present unique user equilibrium or multiple equilibria can be possible. If unique equilibrium is always the case, adding more assumptions or constraints to enforce unique equilibrium will be more preferable. However, if physically traffic system does present multiple equilibria (similar to the BRUE case), the risk-neutral method as proposed in this article would make more sense since it can deal with the UE solution set explicitly. Arguing which way is better is probably a bit philosophical and is beyond the scope of this article. However the proposed risk-neutral method indeed provides an alternative way to address the issue of nonunique UE solutions, different from existing methods that treat user equilibrium as always unique.

We also notice that the uncertainty due to nonunique UE solutions is just one source of possible uncertainties in toll pricing design. Other uncertainties on demand or supply and possibly the nonunique toll solutions (Penchina, 2004, 2009) are other important uncertainty sources. There are previous works that have dealt with some of those issues, e.g., how to consider demand uncertainty for network design or pricing were discussed in Waller and Ziliaskopoulos (2001), Yin and Lawphongpanich (2007), and Gardner et al. (2008). The risk-neutral and risk-averse SBTP schemes may be applied to these scenarios if a single uncertainty source is considered. In fact, the issue of uncertain demand discussed previously in Yin and Lawphongpanich (2007) is in a similar fashion as the risk-averse scheme in Ban et al. (2009). The risk-neutral scheme proposed here may also be applied to demand or supply or toll solution uncertainties. As shown later in the article, by considering the nonunique UE solutions only, the resulting model (i.e., a stochastic program) is already very complex and computationally demanding. Therefore, it is the authors’ understanding that if two or more sources of uncertainties are considered simultaneously, the resulting model will be most likely more complex. How to develop and solve such models is an interesting future research topic but is out of the scope of this article.

By assuming certain probability distribution function as a realization of the UE solution, the risk-neutral model can be solved by a simulation-based optimization technique. First, the realization of the UE solution can be sampled from its solution set based on the assumed distribution. One critical step in this process is the ability to sample over the UE solution set. For this purpose, we extend the hit-and-run random sampling algorithm originally developed in Smith (1984) from a full dimensional subset in $\mathbb{R}^n$ to a general subset (not necessarily full-dimensional) in $\mathbb{R}^n$. The samples are then evaluated for objective values, which are used in a two-phase simulation optimization algorithm (Deng and Ferris, 2009; Deng, 2007). We test the risk-neutral model and solution algorithm using two case studies in this article to illustrate how the algorithm performs. As shown in Section 6, the risk-neutral approach is less aggressive than the risk-prone scheme and less conservative than the risk-averse scheme. Therefore it can provide more insight for toll authorities to design effective pricing schemes when UE solution is nonunique.

This article is organized as follows. Section 2 start with investigating the characterization of the solution set of an UE under different monotonicity assumptions. Section 3 extends the explicit expression of the UE solution set to UEs under tolls. Section 4 presents the stochastic program for
the risk-neutral design approach. A small illustrative example is also provided in this section. An algorithm based on simulation optimization is proposed in Section 5, including a random sampling method over the solution set and a two-phase simulation optimization algorithm. Two risk-neutral SBTP examples are provided in Section 6. We conclude the article in Section 7.

2 Characterization of UE Solution Set

Assume a transportation network denoted as a directed graph $G(N, A)$, where $N$ is the set of nodes and $A$ is the set of links. We use $a \in A$ to denote a link, and $x_a$ and $t_a$ to denote the total flow and travel time of link $a$ respectively. Link travel time is often modeled as a function of link flow, i.e. $t = t(x)$, where $t = (t_a)_{a \in A}$ and $x = (x_a)_{a \in A}$. Denote $R$ the set of origin-destination (OD) pairs. For any $r \in R$, define $d_r$ the (fixed) travel demand for $r$, $P_r$ the set of paths (routes) for $r$, and $P = \bigcup_{r \in R} P_r$ the set of all paths in the network. To make the problem feasible, we also assume that there is at least one path for an OD pair $r$ with positive demand, i.e., $d_r > 0$. The flow for a path $p \in P$ is defined as $f_p$; the travel time of $p$ is denoted as $F_p$. We then have:

$$x = \Omega f,$$

$$F = \Omega^T t,$$

with $\Omega$ the link-path incidence matrix, and $f = (f_p)_{p \in P}$ and $F = (F_p)_{p \in P}$.

The following two sets define the feasible sets of path flow and link flow respectively.

$$K_f \equiv \{ f | \Lambda f = d, f \geq 0 \}.$$

$$K_x \equiv \{ x | x = \Omega f, \Lambda f = d, f \geq 0 \}.$$

Here $\Lambda$ is the OD-path incidence matrix. It is well known that both $K_x$ and $K_f$ are nonempty, convex, and compact (Nagurney, 1998).

Given the above notation, the link-based UE can be formulated as the following VI by finding an optimal $x \in K_x$ such that (Nagurney, 1998; Patriksson, 1994):

$$t^T(x') - x \geq 0, \forall x' \in K_x. \tag{2}$$

Similarly, the path-based UE can be formulated as the following VI by finding an optimal $f \in K_f$ such that:

$$F^T(f') - f \geq 0, \forall f' \in K_f. \tag{3}$$

It is well-known that, for a feasible UE, it has a unique solution in terms of total link flow if link travel time is a strictly monotone function of link flow (Nagurney, 1998). The solution set is nonempty, convex, and compact if the link travel time is pseudo monotone (but not necessarily strictly monotone) with respect to link flow (Facchinei and Pang, 2003; Nagurney, 1998). For path
flows, however, the equilibrium path flows usually form a nonempty, convex, and compact solution set even if link travel time is strictly monotone with respect to link flow.

We focus on monotone UE problems hereafter in this article. We first give the definitions of different monotonicity conditions; details can be found in Facchinei and Pang (2003).

Definition 1 Suppose $K$ is a subset in the $n$-dimensional space $\mathbb{R}^n$. A function $G : K \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is said to be
(a) pseudo monotone on $K$ if for all vectors $x$ and $y$ in $K$,
\[(x - y)^T G(y) \geq 0 \Rightarrow (x - y)^T G(x) \geq 0;\]
(b) pseudo monotone plus on $K$ if it is pseudo monotone on $K$ and for all vectors $x$ and $y$ in $K$,
\[((x - y)^T G(y) \geq 0 \text{ and } (x - y)^T G(x) = 0) \Rightarrow G(x) = G(y);\]
(c) monotone on $K$ if for all vectors $x$ and $y$ in $K$,
\[(G(x) - G(y))^T (x - y) \geq 0;\]
(d) monotone plus on $K$ if it is monotone on $K$ and for all vectors $x$ and $y$ in $K$,
\[(G(x) - G(y))^T (x - y) = 0 \Rightarrow G(x) = G(y).\]

Remarks. According to Corollary 2.3.10 in Facchinei and Pang (2003), the pseudo monotone plus assumption will guarantee that the defining function of a VI admits a unique value (i.e., the so-called F-uniqueness) over the VI solution set, if $K$ is convex. For the UE problem in particular, a pseudo monotone plus link travel time function $t$ (over $x$) will guarantees that the equilibrium link travel time path travel time is unique, even when the link flow or path flow is not.

Under the pseudo monotone plus assumption, the solution set of UE can be explicitly expressed as shown in the following theorem.

Theorem 1 Assume the link travel time function is pseudo monotone plus, and $\bar{t}$ is the unique link travel time vector and $\bar{x}$ is any given equilibrium link flow vector. The link-based UE solution set ($S_x$) and path-based UE solution set ($S_y$) can be expressed as follows:
\[S_x = \{ x | \bar{t}^T (x - \bar{x}) = 0, t(x) = \bar{t}, x \in K_x \}, \]
\[S_f = \{ f | \Omega f = v, v \in S_x, f \in K_f \}. \tag{5} \]

Proof. We first look at the link-based solution set $S_x$. Due to the pseudo monotone plus assumption, according to Proposition 2.3.12 in Facchinei and Pang (2003), $S_x$ can be expressed as
\[S_x = \{ x | t(x) = \bar{t} \} \cap \text{argmin}\{ \bar{t}^T x : x \in K_x \}. \tag{6} \]

In other words, $S_x$ is the intersection of two sets: the first one contains all link flow vectors that produce the same and unique link travel time vector $\bar{t}$; the second is the solution set of a linear program since $\bar{t}$ is known. The second set is also the set of link flow vectors that result from all possible shortest path flow patterns of the network by assuming the link cost vector is $\bar{t}$. The first set is convex as long as $t$ is pseudo monotone (Solodov and Svaiter, 1998).

Since $S_x$ is not empty based on the assumption of the theorem (which implies that there exists $x^*$ such that $t(x^*) = \bar{t}$ and $x^* \in \text{argmin}\{ \bar{t}^T x : x \in K_x \}$), $S_x$ in (6) can be rewritten:
\[S_x = \text{argmin}\{ \bar{t}^T x : t(x) = \bar{t}, x \in K_x \}. \]
This a convex nonlinear program because the objective is linear and the two constraints are convex. Then according to Mangasarian (1988), $S_x$ can be explicitly expressed as (4). Known the link-based solution set $S_x$, the path-based solution set $S_y$ can be expressed as (5). □

Theorem 1 shows that both the link-based and path-based solution sets are nonempty and convex, and polyhedral if $t$ is linear. If $t$ is strictly monotone, $\bar{x}$ will be unique. In this case, $S_x$ will reduce to a singleton that contains only $\bar{x}$. The set $S_f$ can also be simplified in this case as:

$$S_f = \{ f | \Omega f = \bar{x}, f \in K_f \}. \quad (7)$$

Note that the same expression of the path-based UE solution set (5) is also given in Tobin and Friesz (1988), which however does not give the explicit expression of the link-based solution set $S_x$. Also (7) is well-known in the network modeling literature, see e.g. Bar-Gera (1999).

We next give two examples to illustrate the characteristics of the UE solution set. The first example is for a pseudo monotone-plus UE on a network with one OD pair and three parallel links (paths). The total demand from the origin to the destination is $d = 1$. The link flows are denoted as $x_1, x_2, x_3$ respectively which also coincide with path flows. The link (path) travel times $t_1, t_2, t_3$ are given as $t_1 = 2x_1 + x_2 + x_3, t_2 = 2x_2 + 2x_3, t_3 = 2x_2 + 2x_3$.

It is easy to check (by the definition) that $t(x)$ is pseudo monotone-plus (actually monotone plus) with respect to $x$. Multiple equilibria thus exist and one of them is $\bar{x}_1 = \bar{x}_2 = \bar{x}_3 = 1/3$ with travel times $t_1 = t_2 = t_3 = 4/3$. According to equation (4), the link solution set is as follows, which also coincides with the path solution set.

$$S_x = \{ x = (x_1, x_2, x_3)^T | x_1 = 1/3, x_2 + x_3 = 2/3 \}.$$

The second example is for the path solution set of a strictly monotone UE. The network is shown in Figure 1. The figure depicts a three-node, three-link network with two OD pairs from node 1 to node 2 and node 1 to node 3 respectively. The demand is 1 for each OD pair. There are two paths from node 1 to 2 coinciding with link 1 (path 1) and link 2 (path 2) respectively. There are also two paths from node 1 to 3 that consist of links 1,3 (path 3) and 2,3 (path 4) respectively. Assume we use a BPR type of link travel time function and in particular $t_a(x_a) = 1 + x_a^\alpha$ for each link $a \in 1, 2, 3$. Under this assumption, the link flow is unique and we have $\bar{x}_1 = \bar{x}_2 = 1, \bar{x}_3 = 2$.

![Figure 1: A network for strictly monotone UE](image)

According to equation (7), the path solution set is:

$$S_f = \{ f = (f_1, f_2, f_3, f_4)^T | f_1 + f_2 = 1, f_3 + f_4 = 1, f_1 + f_3 = \bar{x}_1, f_2 + f_4 = \bar{x}_2, f_1 + f_2 + f_3 + f_4 = \bar{x}_3 \}.$$
This can be simplified as follows:

\[ S_f = \{ f = (f_1, f_2, f_3, f_4)^T | f_1 + f_2 = 1, f_3 = f_2, f_4 = f_1 \}. \]

**Remark:** The above discussions show that, if path flow is considered, the UE solution set is convex as long as the link travel time is pseudo-monotone plus with respect to link flow; the path solution set is convex and polyhedral if the link travel time is strictly monotone, e.g., if the BPR (Bureau of Public Roads) function is used. In other words, the nonuniqueness of path flows is inherent to the UE problem. If link flow is considered, however, the results are more complicated as they depend on the specific monotonicity characteristics of the link travel time function. A set of link-based UE solutions (not a singleton) may exist only if the link travel time function is pseudo-monotone plus but not strictly monotone. In practice, the possibility of this happening can be rare. It is also not straightforward to construct such link travel time functions. This implies that the proposed risk-neutral scheme is more significant for path-based UE formulations than for link-based formulations.

### 3 UE Solution Set under Toll Vector \( y \)

Denote \( y_a \) the toll on link \( a \) and \( y = (y_a)_{a \in A} \). The link travel cost, \( c(y, x) \), can be defined as a weighted summation of link travel time and the toll imposed on the link as \( c(y, x) = t(x) + y/\theta \). Here \( \theta \) is the “value of time.” Accordingly, the path cost can be defined as \( C(y, f) = \sum_{T} c(y, x) \).

The UE under toll \( y \) can then be expressed as a VI by finding an optimal \( x \in K_x \) such that

\[ (t(x) + y/\theta)^T (x' - x) \geq 0, \forall x' \in K_x. \] (8)

For a given toll vector \( y \), the solution existence and uniqueness conditions of the tolled-UE problem (8) are exactly the same as those for a UE without toll. Under the pseudo monotone plus assumption of link travel time function, at least one equilibrium solution exists. Assume \( \bar{x}(y) \) and \( \bar{f}(y) \) are any given link flow and path flow vectors respectively. Also \( \bar{t}(y) \) is the unique link travel time vector due to the pseudo-monotone plus assumption. Then the following theorem provides a way to explicitly express the UE solution sets. The proof of the theorem is similar to that for Theorem 1 and is thus omitted here.

**Theorem 2** Assume the link travel time function is pseudo monotone plus. For a given toll vector \( y \), \( \bar{t}(y) \) is the unique optimal link travel time vector and \( \bar{x}(y) \) is any given equilibrium link flow vector. The link-based and path-based UE solution sets under toll \( y \) can be expressed as follows:

\[ S_x(y) = \{ x | (y/\theta + \bar{t}(y))^T (x - \bar{x}(y)) = 0, t(x) = \bar{t}(y), x \in K_x \}, \] \[ S_f(y) = \{ f | \Omega f = v, v \in S_x(y), f \in K_f \}. \] (9) (10)

The \( y \) in the parenthesis indicates that the solution sets depend on the toll vector \( y \). When the UE is strictly monotone, \( S_x(y) \) will reduce to \( \bar{x}(y) \) only. The following corollary gives an explicit expression for path flow solutions in this case.

**Corollary 1** Assume the link travel time function is strictly monotone. For a given toll vector \( y \), \( \bar{t}(y) \) is the unique optimal link travel time vector and \( \bar{x}(y) \) is the unique equilibrium link flow.
vector. The path-based UE solution sets under toll $y$ can be expressed as follows:

$$S_f(y) = \{ f | \Omega f = \bar{s}(y), f \in K_f \}.$$  \hfill (11)

### 4 Risk-Neutral SBTP Model

#### 4.1 Risk-Taking in SBTP Design

Model (1) represents the commonly-used MPEC formulation for SBTP. The objective $Q(y, x)$ may be the total system travel time or similar objectives the toll designer prefers. Model (1) can be rewritten using link flow as follows, denoted as \textit{MPECSBTP} short for MPEC-based SBTP model:

\begin{align*}
\text{MPECSBTP} \quad & \min_{y, x} Q(y, x) \\
\text{s.t.} \quad & y \in K_y \quad (13) \\
& x \text{ solves UE}(y). \quad (14)
\end{align*}

Here $\text{UE}(y)$ denotes the UE problem under toll vector $y$. We use link flow vector $x$ hereafter to discuss the risk-neutral models and solution algorithms with the understanding that the model properties and solution techniques also apply to path-based models by replacing $x$ with the path flow vector $f$. Since we use $S_x(y)$ to denote the link-based solution set of $\text{UE}(y)$, we may rewrite the constraint that $x$ solves $\text{UE}(y)$ as $x \in S_x(y)$. If we let

$$G = \{ (y, x) \mid x \in S_x(y), \ y \in K_y \}$$

be the graph of the set-valued map $S_x$, we can rewrite \textit{MPECSBTP} into the following single level problem (Ban et al., 2009):

\begin{align*}
\text{RPSBTP} \quad & \min_{y, x} Q(y, x) \\
\text{s.t.} \quad & (y, x) \in G. \quad (16)
\end{align*}

Here the label \textit{RPSBTP} stands for “risk-prone second-best toll pricing.”

Notice that \textit{RPSBTP} is equivalent to

$$\min_{y \in K_y} \min_{x \in S(y)} Q(y, x).$$  \hfill (17)

Model (17) shows that \textit{RPSBTP} aims to find a toll $y \in K_y$ that optimizes the “best-case” scenario (Ban et al., 2009). The risk-averse approach that adopts the robust optimization concept can be formulated as a \textit{min-max} problem (denoted by \textit{RASBTP}, which stands for “risk-averse second best toll pricing”) as follows (Ban et al., 2009):

\begin{align*}
\text{RASBTP} \quad & \min_{y \in K_y} \max_{x \in S(y)} Q(y, x) \\
\text{s.t.} \quad & (y, x) \in G. \quad (18)
\end{align*}

If we define $\Psi(y) = \max_{x \in S(y)} Q(y, x)$, it is easy to see that the risk-averse model \textit{RASBTP} aims to design the toll so that it is optimal for the worst case scenarios. Here the “worst case” for a given toll represents the largest objective value as UE solution varies under the toll, i.e. $\Psi(y)$. More discussions on \textit{RPSBTP} and \textit{RASBTP} can be found in Ban et al. (2009).
4.2 A Stochastic Program for Risk-Neutral SBTP

From the above discussion, we can see that in case the UE solution is nonunique, the risk-prone SBTP approach is too optimistic, while the risk-averse approach is too conservative. We thus propose a “risk-neutral” SBTP approach, which aims to minimize the expected objective value as the UE solution varies. For this purpose, we assume the realization of the UE solution follows certain distribution over the solution set \( S_x(y) \). That is, the UE solution is assumed to be a random variable defined on \( S_x(y) \). More specifically, we define \( \bar{x}(y) \) as the (random) UE solution that follows the assumed distribution over \( S_x(y) \) under a toll vector \( y \). Based on this setting, the risk-neutral approach can be modeled as the following stochastic formulation:

\[
\text{RNSBTP} \quad \min_{y \in K_y} \bar{Q}(y) = E_{\tilde{x}(y) \sim S_x(y)}[Q(y, \tilde{x}(y))].
\]  

(19)

Here \( E \) denotes “expected value” and \( \tilde{x}(y) \sim S_x(y) \) means that \( \tilde{x}(y) \) follows certain distribution over the definition set \( S_x(y) \). In addition, RNSBTP stands for “risk-neutral second best toll pricing.” In Deng and Ferris (2009) and Deng (2007), a similar problem was studied:

\[
\min_{y \in K_y} Q(y) = E_{\bar{x} \sim X}[Q(y, \bar{x})].
\]  

(20)

Here \( X \) is the definition set of the probability distribution, which is a fixed set. We can see that the only difference between our proposed risk-neutral SBTP model (19) and the stochastic program (20) is that the definition set \( X \) is fixed in (20) while in RNSBTP, the set \( S_x(y) \) is changing with the toll vector. In this sense, the proposed risk-neutral model RNSBTP extends the results in Deng and Ferris (2009) and Deng (2007). We first analyze the solution existence conditions of the RNSBTP model. In particular, the following theorem states that RNSBTP has a solution under certain conditions.

**Theorem 3** If the following conditions hold, the RNSBTP model (19) has at least one solution.

1. The function \( Q(y, x) \) is continuous with respect to \( x \) for each fixed \( y \), \( K_y \) is bounded, and there exists \( M > 0 \) such that \( |Q(y, x)| \leq M \) for any \( y \in K_y \) and \( x \in S_x(y) \).
2. The expectation function \( \bar{Q}(y) \) is lower semicontinuous for any given \( y \in K_y \).

**Proof.** The first condition guarantees that the expectation function \( \bar{Q}(y) \) is properly defined (see page 57 in Ruszczynski and Shapiro (2003)) for each \( y \in K_y \) and bounded on \( K_y \). Then following Theorem 13 in the same reference (page 58), the RNSBTP model (19) has at least one solution based on the lower semicontinuous property of \( \bar{Q}(y) \) as assumed by the second condition. \( \square \)

Theorem 3 merits further discussions. The first condition generally holds for traffic congestion pricing problems. For example, if the total system travel time is used, we will have \( Q(y, x(y)) = x(y)^T t(x, y) \) which is continuous with respect to \( y \) and \( x \). Since both \( K_x \) and \( K_y \) are bounded, link flow \( x \) and travel time function \( t \) (e.g., the BPR function) are bounded for any \( y \in K_y \). This implies that \( Q(y, x(y)) \) is bounded. The second condition, however, may not hold in general. This is due to the dependency of \( S_x(y) \) on \( y \). In fact, \( S_x(y) \) is a set-valued map of \( y \) which does not have global continuity (Ban et al., 2009). Especially \( S_x(y) \) is only outer Lipschitz continuous (Robinson, 1981; Ban et al., 2009), and there is no limit on the contraction rate of \( S_x(y) \) as \( y \) changes. The following example shows that \( \bar{Q}(y) \) is not lower semicontinuous even for a simple problem.
Consider a small network in which two links connect a common origin-destination pair, with each link also being a route. Let the demand from the origin to the destination be \(d\). Suppose that the link travel time does not depend on the link flow \(x\), so \(t(x)\) is a constant function, which is monotone but not strictly monotone. We consider tolls \(y_1\) and \(y_2\) on link 1 and link 2. The link generalized travel times with toll imposed are:

\[
\begin{align*}
  c_1 &= 3 + y_1 \\
  c_2 &= 3 + y_2.
\end{align*}
\]

Let \(y = (y_1, y_2)\) take values in \(K_y = \{(y_1, y_2) \mid 0 \leq y_1 \leq 1, 0 \leq y_2 \leq 1\}\), and define the function \(Q\) to be

\[
Q(y, x) = c_1 x_1 + 2c_2 x_2.
\]

There are the following three cases to consider.

1. When \(y_2 < y_1\), we have \(c_2 < c_1\), so \(S_x(y)\) contains a single point \((0, d)\), and \(\bar{Q}(y) = E_{x \in S(y)} Q(y, x) = 2d(3 + y_2)\).
2. When \(y_2 > y_1\), we have \(c_1 < c_2\), so \(S_x(y)\) contains a single point \((d, 0)\), and \(\bar{Q}(y) = E_{x \in S(y)} Q(y, x) = d(3 + y_1)\).
3. When \(y_2 = y_1\), we have \(c_1 = c_2\), so \(S_x(y) = \{(x_1, x_2) \mid 0 \leq x_1 \leq d, 0 \leq x_2 \leq d, x_1 + x_2 = d\}\).

In this case, it is not hard to verify

\[
\bar{Q}(y) = E_{x \in S(y)} Q(y, x) = 1.5d(3 + y_1).
\]

When \((y_1, y_2)\) lies on the line \(y_1 = y_2\), the set \(S_x(y)\) is a line segment with length \(d\); when \((y_1, y_2)\) leaves the line \(y_1 = y_2\), the set \(S_x(y)\) immediately shrinks to a singleton. Further, the expectation function \(\bar{Q}(y)\) is not lower semicontinuous at each point on the line \(y_1 = y_2\): the lower limit of \(\bar{Q}(y)\) at \((y_1, y_1)\) is \(d(3 + y_1)\), but the function value there is \(1.5d(3 + y_1)\). In particular, \(\bar{Q}(y)\) is not lower semicontinuous at \((0, 0)\), which is the only limit point for any sequence in \(K_y\) whose function value converges to the optimal value \(3d\). Consequently, the \(RNSBTP\) does not attain its optimal objective value in \(K_y\).

To illustrate the risk-neutral SBTP scheme and the difference among the three toll design approaches, we present an example in this section, which was also discussed in Ban et al. (2009) for only the risk-prone and risk-averse schemes. Figure 2 depicts the hypothetical network with one origin-destination (OD) pair (from node \(r\) to node \(s\)) and three routes. A toll booth is located at the very beginning of routes 2 and 3. The distance between node \(r\) and \(i\) is very small so that the travel time can be ignored (assume toll is automatically collected and therefore the delay at the toll booth can be ignored as well). Further assume the total demand \(d = 10\) and the route (also link) flows are \(x_1\), \(x_2\), and \(x_3\). Travel times of the links have the following form:

\[
\begin{align*}
  t_1 &= 2x_1 + x_2 + x_3 \\
  t_2 &= 2x_2 + 2x_3 \\
  t_3 &= 2x_2 + 2x_3.
\end{align*}
\]

In other words, link interactions do exist among the three links. For simplicity, we assume the "value of time" \(\theta = 1\). Then the link generalized travel times, with toll imposed, are:
Here $y$ is the toll and $y \in K_y = \{y | 0 \leq y \leq 15\}$. Denote $c = (c_1, c_2, c_3)^T$ and $x = (x_1, x_2, x_3)^T$. To determine the "optimal" toll, we assume the objective function for the upper level as follows:

$$G(y, x) = t_1 x_1 + 3t_2 x_2 + t_3 x_3. \tag{21}$$

We notice that the problem can actually be solved analytically due to its special structure. As shown in Ban et al. (2009), the risk-prone and the risk-averse solutions for this toll design problem are $y_p^* = 5$ and $y_a^* = 13.57$ respectively. The corresponding objective values are 125 and 167.86 respectively. This is shown in Figure 3. In this figure, the three axes represent the link flow on the three links of the small network in Figure 2. As shown in Appendix A of the article, the solution set under a given toll vector $y$ is $S_x(y) = \{x = (x_1, x_2, x_3)^T \geq 0 | x_1 = (10+y)/3, \quad x_2 + x_3 = (20-y)/3\}$, which is a line in the three dimensional space. Appendix A also presents how the analytical risk-neutral solution $y_n^* = 11$ can be derived. The objective value for the risk-neutral approach is 155. The example illustrates that if the UE solution is nonunique (e.g., the lines in Figure 3), the three toll design approaches may generate different solutions. In general, the risk-neutral solution lies in between the risk-prone and risk-averse solutions. The objective value of the risk-neutral approach also lies in between those of the risk-prone and risk-averse approaches, indicating that the risk-neutral approach is indeed less aggressive than the risk-prone approach and less conservative than the risk-averse approach. In this sense, it provides another alternative for SBTP design. In addition, while risk-prone and risk-averse optimal tolls (i.e. $y_p^*$ and $y_a^*$) have their associated optimal UE solutions (i.e. $x_p^*$ and $x_a^*$ respectively), there is no a single UE solution that is "optimal" under the optimal risk-neutral toll $y_n^*$. Rather, the risk-neutral optimal toll is optimized over the entire solution set $S_x(y_n^*)$ in terms of the expected objective value. It is this capability of explicitly considering all UE solutions that makes the risk-neutral SBTP approach more appealing than the other two design approaches.

Quantifying the actual distribution of $\hat{x}(y)$ for a given toll vector $y$ is crucial to the risk-neutral model. In this article, we simply assume $\hat{x}(y)$ follows a uniform distribution over $S_x(y)$, which implies that all solutions within $S_x(y)$ have the same probability to be realized. This simplified assumption may not be valid in practice. As shown in Bie and Lo (2010), some of the equilibria may have less probability to be realized or not "physical" at all, and others have higher probability to be realized. This needs to be captured by a more properly constructed distribution function. The model presented in this article however can capture any distribution form that can be properly
identified. Furthermore, the solution algorithm we present in Section 5 can also be extended to solve the RNSBTP model with distributions other than the uniform distribution \(^2\).

# 5 Solution Algorithm

In this section, we present the solution algorithm for the risk-neutral SBTP model. The algorithm is designed based on the simulation-based optimization technique.

## 5.1 Outline of the Algorithm

The solution algorithm, called the Simulation Optimization based Risk-Neutral Algorithm (**SORNA**), is first given below:

**SORNA Algorithm**

Step 1. *Initialization.* Select an initial toll vector \( y^0 \) and set the iteration counter \( k = 0 \).

Step 2. *Construct solution set* \( S_x(y^k) \). Solve the UE problem under toll \( y^k \), \( UE(y^k) \). Denote the solution as \( \bar{x}^k \). Then construct the solution set of \( S_x(y^k) \) based on \( \bar{x}^k \) using the methods presented in Sections 2 and 3.

Step 3. *Random Sampling.* Perform uniform random sampling over \( S_x(y^k) \) (See Section 5.2). Denote the samples as \( x_{i}^{k}, 1 \leq i \leq M_{k} \). Here \( M_{k} \) is the number of samples which is set by the two-phase simulation optimization in Step 4 and may be different for different \( y^k \)'s.

Step 4. *Two-Phase Simulation Optimization.* The first phase is a global exploration step to identify promising subregions. The second step is to solve a quadratic approximation of the original problem in each subregion by derivative-free algorithms. This step will call Step 3 to generate

\(^2\)This requires to construct a particular distribution from the uniform distribution generated by the EHR algorithm in Section 5.2, which is a standard operation.
random samples in the solution set. An approximate solution $\hat{y}^k$ will be generated in this step.

Step 5. Convergence Test and Move. If $\|y^k - \hat{y}^k\| \leq \varepsilon$, stop. Otherwise, set $y^{k+1} = \hat{y}^k$ and $k = k + 1$, go to Step 2.

In Step 3, the sampling method is based on the Hit-and-Run (HR) approach originally developed in Smith (1984) for sampling over a full dimensional subset in $\mathbb{R}^n$. We extend in Section 5.2 the original approach to sample over a general subset of $\mathbb{R}^n$ which is not necessarily full-dimensional. The process of the two-phase simulation optimization in Step 4 is discussed in Deng and Ferris (2009); Deng (2007), and is thus omitted here. It suffices to say that the algorithm first identifies promising subregions and then performs a derivative-free optimization on a quadratic approximation of the original problem. This process repeats itself until certain convergence criteria are met. In this article, we directly use the WISOPT solver developed in Deng (2007) to perform the subregion identification and optimization. In particular, we set $M_k$ a constant for different $y_k$'s. Notice that WISOPT may sample over additional $y$'s to construct the quadratic approximation of the objective. The sample process will be the same as in Step 3. In Step 5, $\varepsilon$ is a user-defined threshold for convergence tolerance. We use $10^{-5}$ in this article.

5.2 Random Sampling from the UE Solution Sets

Sampling over a convex set in $\mathbb{R}^n$ to follow a certain distribution is a classical problem in operations research. Smith (1984) proposed the "hit-and-run" algorithm, a Monte-Carlo process to uniformly sample points from a full dimensional convex set of $\mathbb{R}^n$. Denote such a set as $W \subseteq \mathbb{R}^n$. The hit-and-run (HR) algorithm for generating $M$ uniformly distributed sample points can be summarized as follows (Smith, 1984):

**HR Algorithm**

Step 1. Choose a starting point $w_0 \in W$ and set $m = 0$.

Step 2. Generate a random direction $p$ uniformly distributed over a direction set $P \subseteq \mathbb{R}^n$.

Step 3. Find the line set $L = W \cap \{w|w = w_m + \lambda p\}$ and generate a random point $w_{m+1}$ uniformly distributed over $L$.

Step 4. If $m = M$, stop. Otherwise, set $m = m + 1$ and go to Step 2.

In the above algorithm, since $W$ is full-dimensional, the direction set $P$ in Step 2 can be chosen as the unit sphere $P = \{p| \|p\| = 1\}$. However, the UE solution sets $S_x(y)$ and $S_f(y)$ are not full-dimensional in general. Therefore we need to generate the random direction from the subspace that is parallel to the affine hull of the sampling set. (The affine hull of a convex set is the smallest affine set containing it.) Another observation is that the algorithm above applies to general convex sets, not restricted to polyhedrons. For illustration purposes, we only show in this article the random point generation process for a polyhedral set. This applies to two cases: (1) nonunique link flows when link interactions are considered and the link travel time function is affine; and (2) nonunique path flows when the link travel time function is strictly monotone (e.g., the BPR function) or affine. For other cases with non-polyhedral convex UE solution sets (such sets are convex if the link travel time function is pseudo-monotone plus), the proposed method needs to be properly modified. The reader can refer to Smith (1984); Zabinsky (2003) for how the modification may be done for sampling over a general convex set.
To show how the sampling method works, we first consider the path-based UE solution set $S_f(y)$ for a given $y$. We will then discuss how to handle the link-based UE solution set $S_x(y)$.

When the link travel time function is strictly monotone, $\bar{t}(y)$ is a constant in Corollary 1. We can rewrite $S_f(y)$ in (11) as
$$S_f(y) = \{f|\Omega f = \bar{t}(y), \Lambda f = d, f \geq 0\}.$$  
When the link travel time function is pseudo-monotone plus and affine, $\bar{t}(y)$ is a constant in Theorem 2 and the constraint $t(x) = \bar{t}(y)$ in (9) becomes a linear equation. The set $S_f(y)$ in (10) can be rewritten as
$$S_f(y) = \{f|(\theta + \bar{t}(y))^T(\Omega f - \bar{x}(y)) = 0, t(\Omega f) = \bar{t}(y), \Lambda f = d, f \geq 0\}.$$  
In both (22) and (23), $S_f(y)$ is a polyhedron defined by finitely many linear equalities and inequalities. Hence, we can consider the two cases together, by writing $S_f(y)$ in the general form:
$$S_f(y) = \{f|Bf = b, Cf \geq c\},$$  
where $B, C$ are matrices and $b, c$ are vectors, all with proper dimensions. Note that from (22) and (23), $C = I, c = 0$. However, $C, c$ are used here as the method can work for general polyhedrons. We should also point out here that although it appears (22)-(25) involve all paths, what really matters is the set of paths in the UE solution set. This would be the so-called minimum-cost paths between each OD pair. Since we deal with UE with pseudo-monotone link travel time functions, the set of the minimum-cost paths between each OD pair is unique. Such sets can be relatively easily constructed (see Section 6.2) once the unique UE link travel times are found by a standard UE algorithm. This means that in (24), $f$ is the vector that only contains minimum-cost paths, not all paths for each OD pair. In other words, the sampling method does not need to enumerate all paths. Although the total number of paths between an OD pair may be large, the number of used paths in an UE solution is typically very small. This can be seen more clearly from the path-based example in Section 6.2.

We need to find the subspace that is parallel to the affine hull of $S_f(y)$. Denote this subspace by $S_0$ and write it as
$$S_0 = \{f'|\bar{B}f' = 0\},$$  
where $\bar{B}$ is a matrix of proper dimensions. Clearly, $S_0$ must be a subspace of the null space of the matrix $B$ in (24). In other words, the row space of $\bar{B}$ must contain the row space of $B$. It is possible for $\bar{B}$ to contain some rows of $C$ as well. This will happen if some of the inequality constraints hold as equality. To check if each of the inequality constraints is satisfied as an equality by all points in $S_f(y)$, we solve the following linear program:
$$\max_f C_i f - c_i \text{ subject to } Bf = b, Cf \geq c,$$
where $C_i$ stands for the $i$th row of $C$. If the optimal value of the above LP is exactly zero, then we add the row $C_i$ into $\bar{B}$. When applied to the path-based UE problem, this procedure is to check if each minimum-cost path carries positive flow in some UE solution. Of course, if a path already carries positive flow in the current UE solution, we know a priori that the optimal value of the above linear program is not zero and there is no need to solve it. If a minimum-cost path carries zero flow in the current UE solution, then we need to solve the above linear program to find whether this path carries zero flow in all UE solutions by checking if the optimal value of the linear program is zero. Once we find the set $S_0$ in (25), we can modify the original HR algorithm, by using $S_0$ to replace the direction set $P$ in the algorithm. This results in the following Extended Hit-and-Run (EHR) algorithm. Note that we assume the path-based UE solution and the minimum-cost path set between each OD pair have been obtained before running the EHR algorithm.

**EHR Algorithm**
Step 1. Let $f_0 \in S_f(y)$ be a known UE solution under the toll vector $y$. Construct matrices $B, C$, and vectors $b, c$ from the representation of $S_f(y)$ in (22) or (23), using minimum-cost path sets only. Set $m = 0$.

Step 2. Find the matrix $\bar{B}$, and construct an orthogonal basis of its null space. Denote the basis as $\{h_0^0, h_1^0, \ldots, h_k^0\}$, where $k$ is the dimension of the null space of $\bar{B}$.

Step 3. Generate $k$ random numbers from the standard normal distribution $N(0,1)$: $\gamma_1, \gamma_2, \ldots, \gamma_k$. Construct a random direction in $S_0$ as

$$p_m = \frac{\sum_{i=1}^{k} \gamma_i h_i^0}{\sqrt{\sum_{i=1}^{k} \gamma_i^2}}.$$  \tag{26}

Step 4. Compute the following two scalars (Zabinsky, 2003):

$$\lambda_{\text{min}} = \max \left\{ \frac{c_i - C_i f_m}{C_i p_m}, \forall i \text{ such that } C_i p_m > 0 \right\},$$ \tag{27}

$$\lambda_{\text{max}} = \min \left\{ \frac{c_i - C_i f_m}{C_i p_m}, \forall i \text{ such that } C_i p_m < 0 \right\},$$ \tag{28}

where $C_i$ is the row vector formed by the $i$th row of $C$. Generate a random scalar $\lambda_m$ that follows the uniform distribution defined on the range $[\lambda_{\text{min}}, \lambda_{\text{max}}]$.

The new sample can then be constructed as $f_{m+1} = f_m + \lambda_m p_m$.

Step 5. If $m = M$, stop. Otherwise, set $m = m + 1$ and go to Step 3.

In Step 3 of the EHR algorithm, the random direction is generated as a linear combination of elements in a basis of $S_0$. The weights follow standard normal distribution. As shown in Knuth (1969), the direction generated this way follows uniform distribution in the subspace. Therefore $p_m$ as expressed in (26) is a uniformly distributed direction in $S_0$. Furthermore, as shown in Zabinsky (2003), $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$ represent the intersecting points of the line $f_m + \lambda p_m$ with the boundaries of $S_f(y)$, which are calculated using the inequality constraints of $S_f(y)$, i.e. $Cf \geq c$. This implies $L$ is the line segment between $f_m + \lambda_{\text{min}} p_m$ and $f_m + \lambda_{\text{max}} p_m$, which indicates that $f_{m+1} = f_m + \lambda_m p_m$ is a uniformly distributed point along line set $L$ as long as $\lambda_m$ is uniformly distributed over $[\lambda_{\text{min}}, \lambda_{\text{max}}]$. We should point out here that for large scale problems, the matrix $\bar{B}$ may be quite large. In this case, calculating the null space of $\bar{B}$ can be computationally demanding. This is particularly critical since the null space of matrix $B$ needs to be computed for every different toll value. Recently the algorithm developed by Foster and Davis (2011) can deal with the computation of null spaces of large size matrices. Another way to address this issue is to explore the special structure of $\bar{B}$, such as sparsity, to develop specialized algorithms. Research in this direction may be pursued in future research.

Before we end this section, we explain how to sample on the link-based UE solution set $S_x(y)$, when the link travel time is affine and pseudo monotone plus. As shown in (9), for each fixed $y$ the set $S_x(y)$ is a polyhedron since $\bar{f}(y)$ is a constant. However, the expression (9) depends on $K_x$, which is the image of $K_f$ under the linear map $\Omega$. It is not easy to rewrite (9) as a polyhedron defined by linearly equalities and inequalities. Accordingly, we modify the above method when we deal with $S_x(y)$. First, note that $S_x(y)$ is related with $S_f(y)$ in the following way:

$$S_x(y) = \Omega S_f(y).$$
where $S_f(y)$ is given in (22). In other words, $S_x(y)$ is the image of $S_f(y)$ under the linear map $\Omega$. Hence, we can first find a basis of the subspace parallel to the affine hull of $S_f(y)$, and then compute the images of elements in that basis under $\Omega$. This will give a basis for the subspace parallel to the affine hull of $S_x(y)$. We can then obtain a random direction by following Step 3 above. To obtain a random step-size, we cannot use the formulas in (27) and (28) in Step 4, since we do not have a equality/inequality constrained representation of $S_x(y)$. Instead, we find a box that encloses $S_x(y)$ in it (such a box always exists because $S_x(y)$ is bounded), and pick a random point from the intersection of that enclosing box and the line along the random direction. We adopt this random point as a new sample if it belongs to $S_x(y)$, and abandon it and reselect if otherwise. This is similar to the “acceptance-rejection” method proposed in Zabinsky (2003) to sample over a general convex set (not necessarily full-dimensional).

6 Numerical Examples

To illustrate the risk-neutral SBTP scheme and the SORNA solution algorithm, we apply the algorithm to a link-based and a path-based SBTP application in this section.

6.1 Link-Based $RNSBTP$ Model

The link-based model is tested on the small-size network as shown in Figure 2. It turns out that, as shown in Section 4, the problem can be solved analytically and exactly, which can be used to analyze the effectiveness of the solution approach.

We first apply SORNA to solve the risk-neutral model $RNSBTP$ for the test example. We set the number of samples for each given toll (Step 3 of SORNA) as $M_k = M = 300$. The algorithm converges after 8 iterations with an obtained solution 11.12 and the corresponding objective value is 154.37. The results are very close to the optimal solution (11) and the optimal objective value (155) as shown in Appendix A. The deviations are only 1.1% and 0.4% respectively. Figure 4 depicts the convergence of the SORNA algorithm.

![Figure 4: Convergence of the SORNA Algorithm](image)

We now check whether the random sampling algorithm EHR works properly. We need to answer two questions: (i) are the generated samples within the solution set $S(y)$? and (ii) do the samples follow a uniform distribution within $S(y)$? Here we use $y = 11$, i.e. the optimal toll, as
a test case. First, based on (33) in Appendix A, we know that the solution set when \( y = 11 \), i.e. \( S(11) \), can be expressed as:

\[
S(11) = \{ x = (x_1, x_2, x_3)^T \mid x_1 = 7, \quad x_2 + x_3 = 3 \}.
\] (29)

The \( x_1 \) component of all \( M = 300 \) samples generated by the EHR algorithm is 7. Figure 5 further depicts the \((x_2, x_3)\) component of the samples using plus signs. The solid line represents the “theoretical” line of \( x_2 + x_3 = 3 \), which \((x_2, x_3)\) should follow. It is clear that all samples lie on the theoretical line, indicating that the random samples are within \( S(11) \) as defined in (29).

To answer the second question, we notice that if \( x_2 \) follows the uniform distribution within \([0, 3]\), the generated samples will indeed follow the uniform distribution within \( S(11) \). We plot in Figure 6 the histogram of the \( x_2 \) component of the 300 randomly generated samples. We particularly arrange the samples into 10 bins. The bold solid line in the figure is the “theoretical” (uniform) distribution that \( x_2 \) should follow. The thin solid line with plus signs is the histogram calculated by the generated samples. We can see that although there are significant variations, the histogram does follow and fluctuate over the theoretical line. This means that the generated samples do follow, approximately, a uniform distribution.

![Random Samples Generated by the EHR Algorithm](image)

**Figure 5: Random Samples Generated by the EHR Algorithm**

To show how the sampled objective values compare with the *true* objective value for a given toll, we show in Figure 7 the sampled objective values vs. toll \( y \). In this figure, the dotted line and the solid line with triangle represent respectively the average objective value of all the samples for \( M = 3 \) and \( M = 50 \). The bold solid line is the true objective value, i.e. calculated via equation (35) in Appendix A. We can see that the average objective value can approximate the true objective curve very well, in spite the fact that the average value has variations at different \( y \)'s. We can also observe that as toll \( y \) becomes larger, the variation becomes smaller. This is because, as can be seen from the solution set \( S(y) \) in (33) in Appendix A, the range of the solution set becomes smaller as \( y \) increases, leading to a smaller variation of the objective value. Notice that this observation is specific to this small example and should not be extended to RNSBTP problems in general.

The value of \( M \) is an important factor to the solution algorithm. Intuitively, a larger \( M \) will generate more samples for a given toll which may produce more accurate approximation to the underlying true distribution. On the other hand, however, larger \( M \)'s will also require more computational time to perform the sampling, which may not be appropriate for large scale problems.
Therefore, a proper $M$ will be most likely problem-specific, which will reflect the user’s tradeoff between the solution quality and the available computational resources. To see how different $M$’s may impact the solution quality, we show in Table 1 how the solution and objective value change as the value of $M$ varies. Notice that for this small example, the computational efforts is less interesting, which can be reasonably represented by the actual values of $M$ (i.e. $M = 100$ roughly requires twice of the computational time used by $M = 50$).

We can see from the table that as $M$ increases, both the obtained solution (toll) and the objective value become closer to the optimal solution and the optimal objective value. In particular, the deviation for the obtained toll decreases from 4.32% to 1.1%, while the deviation for the obtained objective value decreases from 4.18% to 0.4%. This indicates that increasing the number of samples at each iteration does produce more accurate approximation to the underlying distribution, and as a result, a better solution.

Finally, to compare the results of the three SBTP design approaches, we depict in Figure 8 the changes of the objective value as a function of the imposed toll, for the risk-prone, risk-neutral, and risk-averse approaches. The curves are based on the analytical results of the three SBTP
Table 1: Impact of the Number of Samples ($M$) on Solution Quality

<table>
<thead>
<tr>
<th>Value of $M$</th>
<th>30</th>
<th>50</th>
<th>100</th>
<th>300</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtained sol</td>
<td>10.53</td>
<td>10.71</td>
<td>10.59</td>
<td>11.12</td>
</tr>
<tr>
<td>Optimal sol.</td>
<td>11.00</td>
<td>11.00</td>
<td>11.00</td>
<td>11.00</td>
</tr>
<tr>
<td>Diff. with opt. sol. (%)</td>
<td>4.32</td>
<td>2.64</td>
<td>3.76</td>
<td>1.10</td>
</tr>
<tr>
<td>Obtained obj</td>
<td>148.53</td>
<td>151.91</td>
<td>153.30</td>
<td>154.37</td>
</tr>
<tr>
<td>Optimal obj.</td>
<td>155.00</td>
<td>155.00</td>
<td>155.00</td>
<td>155.00</td>
</tr>
<tr>
<td>Diff. with opt. obj. (%)</td>
<td>4.18</td>
<td>1.99</td>
<td>1.09</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Table 1: Impact of the Number of Samples ($M$) on Solution Quality

6.2 Path-Based $RNSBTP$ Model

To illustrate the path-based $RNSBTP$ model, we introduce an application of vehicle emission control via $SBTP$ on the Sioux Falls network (Leblanc et al., 1975). The network data can be seen on the website of Transportation Network Test Problems (http://www.bgu.ac.il/~bargera/tntp/). This example can be considered as a generalized form of congestion pricing since the objective is to reduce emissions rather than to reduce “congestion.” To simplify our discussion, we assume that all vehicles are plug-in hybrid vehicles (PHEV) with the range $Z$ miles. That is, within the first $Z$ miles, all vehicles are operated under electric mode with zero emission; after $Z$ miles, vehicles are under the regular fuel mode with standard emissions. Tolls are imposed on links grouped by zones. Within each zone, the same toll will be imposed to all links. The problem is to determine the optimal toll charge for each zone in order to minimize the total system emissions. Figure 9 depicts the tolling and zoning schemes for the Sioux Falls network. All links within the dotted circle are included in zone 1; all links within the dashed circle but not in the dotted circle are
included in zone 2; the remaining links are in zone 3. The same toll will be imposed to links in the same zone; between zones, tolls can be different. In other words, the toll vector for this network is three-dimensional, i.e., $y = [y_1 \ y_2 \ y_3]^T$.

![Figure 9: Tolling Scheme on Sioux Falls Network to Control Emissions](image)

We should point out here that the above application is very simplistic because (i) PHEVs will probably never get to 100% penetration in the market; (ii) the PHEV range may vary depending on the actual models/types of PHEVs; (iii) if PHEVs are widely used, charging stations may be built so that their batteries can be recharged or replaced during the travel (see He et al. (2013) for a recent study to determine the optimal locations of charging stations); and (iv) other pricing schemes may be more applicable in this case, such as path-based pricing, since as will be shown later, the emissions of PHEVs are path-dependent. By considering all the above factors, the resulting model will be more complicated and rigorous treatment of such model is beyond the scope of this article. However, it is the authors’ understanding that the fundamental issues due to nonunique path solutions should still remain (and likely more complicated). Therefore it suffices to illustrate the key features of the path-based RNSBT model by this simplified scenario. Note also that although we assume all links can be tolled, it is different from FBTP as tolls are grouped into zones. The model can also apply if only a subset of links within each zone is tolled.

To mathematically formulate the above problem, we need to introduce more notation. Denote $l_a$ the length of link $a$, and $e_a$ the amount of emissions a single PHEV will generate by traversing a unit distance (say one mile) on link $a$. Here $e_a$ represents the standard emission level when a PHEV is operated under the fuel mode, similar to the “emission factor” used in Nagurney et al. (1998). Let $e_p^r$ be the amount of emissions a single PHEV will produce by traversing (the entire length) of path $p$ between OD pair $r$. Due to our assumption that a PHEV will produce no emission within its range (i.e., the first $Z$ miles) and will produce emissions similar to regular gasoline vehicles
beyond the range, $e_p^e$ can be expressed as:

$$e_p^e = \sum_{a \in p > z} e_a l_a. \quad (30)$$

Here $p > z$ is the sub-path of path $p$ by excluding the links of the first $Z$ miles. Clearly, a PHEV taking $p$ will only produce emissions when it is on $p > z$; if the path length is smaller than $Z$, the PHEV will produce no emission. Based on (30), the objective, i.e., the total system emissions, can be expressed as a function of the toll and path flows as:

$$Q(y, f(y)) = \sum_r \sum_p f^r_p \sum_{a \in p > z} e_a l_a. \quad (31)$$

Clearly because of the definition of the sub-path $p > z$ is path-dependent, equation (31) has to be described using path-flows. The path-based RNSBT P model for this particular problem can then be written as:

$$\min_{y \in K_y} \tilde{Q}(y) = E_{\tilde{f}(y) \sim S_f(y)}[Q(y, \tilde{f}(y))]. \quad (32)$$

We test the above path-based model on the Sioux Falls network using the tolling scheme shown in Figure 9. The link travel time function for this example is the BPR type. Other parameters used in the test are: the value of time $\theta = 1$, upper bound and lower bound of toll $y_u = 5$ and $y_l = 0$ respectively, and the range of PHEV $Z = 20$ miles. The link emission factor $e_a$ is random generated following a uniform distribution between 0.1 to 10. $M_k$ is set to vary between 500 - 5000: WISOPT can determine an optimal value for each $M_k$ mainly depending on the need to approximate the quadratic objective function for iteration $k$.

One of the challenges for solving the path-based RNSBT P model in (32) is that we need to explicitly enumerate paths. However, here we only need to enumerate all minimum-cost paths, NOT all paths, between an OD pair. This is because the RNSBT P model operates on the UE solution set which consists only the minimum-cost paths. Since we deal with UE with pseudo-monotone link travel time functions, path costs will be unique; so is the set of minimum-cost paths between each OD pair. This has been observed before for UE with BPR type link travel time functions (Bar-Gera, 1999). Generating the set of minimum-cost paths (after the UE is solved) can be done via some minor revisions to the $K$-shortest path algorithm (Jimenez and Marzal, 1999). That is, one can set $K$ to be very large, but can terminate the search once the latest found path cost is larger than the minimum path cost. Previous research (Bar-Gera, 1999; Ban et al., 2006) showed that the minimum-cost paths consist only of a very small portion of all paths between an OD pair. For example, for Sioux Falls, this number is usually smaller than 10, while the total number of paths between an OD pair is usually in the magnitude of a few thousands. Since the $K$-shortest path algorithm can run in polynomial time (Jimenez and Marzal, 1999) for a given OD pair, enumerating all minimum-cost paths for all OD pairs can also be done in polynomial time, at least in theory. Here we adopt directly the algorithm proposed in Jimenez and Marzal (1999); further testing of the performance of the algorithm on large traffic networks will be conducted in future research.

The algorithm finds the optimal toll as $y = (4.208, 2.582, 4.990)$. Figure 10 depicts the convergence of the algorithm. It clearly shows that the convergence is achieved after about 15 iterations. To compare the three risk-taking tolling schemes, we fix the tolls for Zone 2 and Zone 3 as the optimal values (2.582 and 4.990 respectively) and plot in Figure 11 how the minimum, average,
and maximum objective values change with the toll for Zone 1. To better illustrate the results, we tested the toll for Zone 1 from 3.6 to 5 with 0.1 as the increment. For each given toll, we randomly sampled 5000 times from the UE solution set and the plot shows the objective variation based on the samples. Figure 12 and Figure 13 show similar plots for Zone 2 and Zone 3 respectively by fixing the tolls for the other two zones as the optimal values. We particularly plotted the toll for Zone 2 from 1.6 to 3 in Figure 12 and the toll for Zone 3 from 3.6 to 5.6 in Figure 13. From the three figures, we can observe that (i) The curves are neither monotonic nor convex implying that finding the global optimal solutions is generally difficult; (ii) The solution by the proposed model and algorithm in the article, i.e., \( y = (4.208, 2.582, 4.990) \), appears to be close to the global optimal solution; (iii) Under a given toll, there can be significant differences among the three objectives, indicating that the UE solution set indeed contains multiple solutions that can result in different toll design objectives; (iv) Although the general trends of the three objectives are similar, there could be some distinct difference also. For example, in Figure 11, the minimum and average objectives reach the lowest values when the toll for Zone 1 is around 4.2 - 4.3. However, the maximum objective reaches its lowest value when the toll is around 4.9. In Figure 12, all three objectives reach their lowest values when the toll for Zone 2 is around 2.6. In Figure 13, the minimum and average objectives reach their lowest values when the toll for Zone 3 is around 5, while the maximum objective obtains its lowest value when the toll is around 4.7. These observations confirm that the three risk-taking toll design schemes are significant and can produce different optimal tolls: the risk-neutral toll (objective) is usually in between the risk-averse and risk-prone tolls (objectives).

![Convergence](image)

**Figure 10:** Convergence of the RNSBTP algorithm (Sioux Falls)

The Sioux Falls network has 528 OD pairs. The number of paths between any OD pair ranges from over 1,500 to nearly 5,000. There are about 12% of the OD pairs, each of which has more than 4,000 paths, and nearly 50% of them has more than 3,000 paths each. The number of minimum-cost paths is however very small compared to the total number of paths: 78% have only 1 path, and 90% have 2 or less, 97% have 3 or less, 99% have 4 or less. Only 2 out of the 528 OD pairs have 5 minimum-cost paths and another 2 have 6 minimum-cost paths. Figure 14 further depicts the distribution (histogram) of the ratio between the number of minimum-cost paths and the total number of paths, for all OD pairs. The ratio is very small: over 58% of OD pairs have the ratio of 0.04% (4%%) or less. This ratio is expected to be even smaller when the network is larger. This indicates that although it is not feasible to enumerate all paths for all OD pairs for a large network, it is probably possible to enumerate all minimum-cost paths even for a reasonably large network. This has been actually shown previously in Bar-Gera (1999).
Figure 11: Objective Variation vs Toll for Zone 1 (Sioux Falls)

Figure 12: Objective Variation vs Toll for Zone 2 (Sioux Falls)

7 Conclusion

We proposed a risk-neutral scheme for the SBTP design to account for the possible nonuniqueness of the UE solution. The scheme aims to minimize the expected objective value as the UE solution varies over the solution set. The scheme provides an alternative way, from the toll designer’s perspective, for addressing the uncertainty due to nonunique UE solutions. Therefore, the proposed risk-neutral scheme complements the traditionally used risk-prone SBTP scheme and the risk-averse
We showed that the risk-neutral scheme can be formulated as a stochastic program, which extends the recent simulation-based optimization methods studied in Deng (2007); Deng and Ferris (2009). The stochastic program can be solved iteratively via three major steps: characterization of the UE solution set, uniform sampling over the UE solution set, and a two-phase simulation optimization algorithm using the random samples. It turns out that a proper expression of the UE solution set requires that the link travel time function is pseudo-monotone plus with respect to link flow. To sample uniformly over the UE solution set, we extended the Hit-and-Run (HR) sampling algorithm in Smith (1984) from a full dimensional subset in $\mathbb{R}^n$ to a subset of a subspace in $\mathbb{R}^n$. We tested the model and solution algorithm on a small example (for link-based formulations) and the Sioux-Falls network (for path-based formulations). The results showed that (i) the extended sampling algorithm works well as it only generates samples within the UE solution set and the
generated samples follow an approximate uniform distribution; (ii) the solution algorithm can produce an approximate model that matches well the general shape and trend of the true objective function; see Figure 7; (iii) when the UE solution is nonunique, the three design schemes could indeed produce quite different optimal toll solutions and objective values: in general, the risk-neutral toll (objective) is in between the risk-prone and risk-averse tolls objective values); and (iv) with respect to the path-based formulations, the minimum-cost path set for an OD pair plays an important role and needs to be explicitly generated. Since the minimum-cost path set usually consists of a very small portion of the entire path set for the OD pair, it can be relatively easily generated (i.e., enumerated).

By introducing the concept of toll designer’s risk-taking behavior, one has alternative ways (i.e., the risk-averse, risk-prone, and risk-neutral schemes) to address the uncertainty due to nonunique UE solutions. There are several questions in this direction however that still remain unresolved. Some of them are summarized as follows:

(a) The efficiency of the random sampling algorithm needs to be further improved. For example, in Figure 6, there are still significant variations in the histogram even with 300 random samples. Investigations on sampling methods that can converge faster should be conducted in future research. Other issues related to apply the proposed model and algorithm to larger scale networks, such as fast computation of the null space of a large matrix, need also to be further investigated.

(b) In this article, we assume that the realization of the UE solutions follows a uniform distribution over the solution set. In practice, some UE solutions may be unstable (Watling, 1999), implying that the probability of these solutions being realized is small. As a result, this simplified assumption may not be valid in practice. The authors are currently investigating this issue by linking the realization of the UE solutions with user choice behaviors especially the day-to-day route choice adjustment (Ma and Ban, 2011). This can hopefully result in more realistic distribution of UE solutions.

(c) The solution algorithm for the risk-neutral model requires an explicit expression of the solution set of the lower level UE. As shown in Section 2, such an expression can be readily constructed for pseudo-monotone plus UEs, resulting in convex UE solution sets for either link-based or path-based formulations. However, in this article we only tested the cases that the UE solution set is polyhedral. Although it is our understanding that the proposed method (especially the sampling algorithm) can be readily extended to a general convex set, such extension can still be an interesting future research topic.

(d) We focus on link-based tolling strategies in this article, which may result in link-based or path-based formulations as illustrated using the first and second examples respectively in Section 6. With the advent of mobile sensing (Herrera et al., 2010; Ban and Gruteser, 2012), tracking of individual vehicles is possible (if associated issues such as privacy can be satisfactorily resolved). This will enable the implementation of path-based pricing or charging. One of such emerging applications is the mileage fee (also called distance-based or VMT-based charges) that charges individual vehicles based on where and how long they have traveled. Due to the tracking nature of such applications, path-based formulations will probably need to be applied, for which the nonuniqueness of path flows discussed here will be critical in terms of designing the optimal pricing/charging schemes of these applications. The authors will study such applications in their future research.

(e) We study static congestion pricing problems in this article. The proposed risk-neutral concept can also be applied to other types of bilevel network design problems. One of such problems is dynamic congestion pricing. As the solution nonuniqueness issues can be expected to be
more common for dynamic problems (such as the dynamic user equilibrium (DUE) problem), extending the results/findings of this article and investigating the issues presented above to dynamic congestion pricing problems will be crucial. This will not only be scientifically interesting but also have significant practical implications as it can help decision makers to design more robust pricing schemes. However, due to the complex nature of DUE, such extensions can be a challenging task and call for further investigations.

References


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


Appendices

A Analytical Solution of the Test Example

As shown in Ban et al. (2009), the solution set $S(y)$ of the test example in Figure 2 can be expressed as:

$$S(y) = \{x = (x_1, x_2, x_3)^T \geq 0 | x_1 = (10 + y)/3, \ x_2 + x_3 = (20 - y)/3\}. \quad (33)$$

Clearly, for any given $y \in K_y$, $S(y)$ is a straight line (i.e., a nonempty polyhedral set) in the three dimension space $x_1 - x_2 - x_3$ (see Figure 3).

Here we want to minimize the objective function (21) over all $x \in S(y)$ that follows uniform distribution. This is equivalent to say, by (33), that $x_2$ follows uniform distribution in the range $[0, (20 - y)/3]$. Subsequently, the expected value of the objective for a given $y$ will be:

$$F(y) = E_x \text{ is uniform distributed over } S(y)[f(y, x)] = \int_0^{(20-y)/3} f(y, x) dx_2. \quad (34)$$

Substituting $x_1 = (10 + y)/3$ and $x_3 = (20 - y)/3 - x_2$ into (34), we can obtain

$$F(y) = \int_0^{(20-y)/3} ((y^2 - 10y + 400)/3 + (40 - 2y)x_2/3) * 3/(20 - y)d(x_2).$$

Note that in (34), $3/(20 - y)$ is the probability density function of $x_2$ over its possible range $[0, (20 - y)/3]$. The above equation can be simplified as:

$$F(y) = 5(y - 11)^2/9 + 155. \quad (35)$$

Therefore, the risk neutral solution is obtained at $y^{*\text{n}} = 11$, with objective value $z^{*\text{n}} = 155$. 

32