A Bilevel Model for Toll Optimization on a Multicommodity Transportation Network

Luce Brotcorne • Martine Labbé • Patrice Marcotte • Gilles Savard
CRT, SMG, and LAMIH-ROI, Le Mont Houy, 59313 Valenciennes Cedex 9, France
ISRO and SMG, Université Libre de Bruxelles, CP 210/01, Boulevard du Triomphe, B-1050, Brussels, Belgium
CRT and DIRO, Université de Montréal, C.P. 6128, Succursale Centre-Ville, Montréal, Canada H3C 3J7
GERAD and MAGI, École Polytechnique de Montréal, C.P. 6079, Succursale Centre-Ville, Montréal, Canada H3C 3A7

We consider the problem of determining a set of optimal tolls on the arcs of a multicommodity transportation network. The problem is formulated as a bilevel mathematical program where the upper level consists in a firm that raises revenues from tolls set on arcs of the network, while the lower level is represented by a group of users travelling on shortest paths with respect to a generalized travel cost.

Introduction

There is no denying a renewed interest in toll roads, either managed by governments or private societies. Toll roads may help alleviate congestion while putting the monetary burden on the actual users of the infrastructure. Many projects around cities such as Paris, New York, or Toronto are being contemplated. In all these, electronic toll collection replaces the traditional toll booth of yesteryear and allows for flexible invoicing strategies.

The literature devoted to road pricing by economists and transportation researchers is rich, and mostly focuses on reducing congestion and the associated negative externalities such as pollution (Cropper and Oates 1992) through demand regulation. Morrisson (1986) studied, both from a theoretical and empirical point of view, marginal cost pricing policies which induce an optimal use of the network. In this case, a congestion fee can be viewed as a user charge based on the difference between the social cost and the average cost perceived by the traveller. This analysis can also be considered in a dynamic setting where commuters are allowed to select their individual departure time (Arnott et al. 1990) as well as their route. Some authors (Verhoef et al. 1995, Mcdonald 1995) have investigated real-life situations where marginal cost pricing theory is not applicable, because of technological or political constraints. Pursuing the idea of marginal cost pricing, Larsson and Patriksson (1998) and Hearn and Ramana (1998) proposed a "goal programming" approach where they optimize a secondary criterion over the set of tolls (denoted "valid tolls") that induce a system optimal traffic assignment.

On the other hand, profit maximizing and system optimal tolls have been studied, on simplistic network topologies, by Beckman (1965) and by Verhoef (1996). In this vein of research, Viton (1995) considered the economic viability of a private road competing with a free-access road. In practice, analyses frequently resort to simulation; see, for instance, the article of Mekky (1994) concerning Highway 407 in the vicinity of Toronto.

In contrast with the above-mentioned studies, we consider a sequential game played between a prescient owner (the "leader") and the commuters (the "follower") that fits the framework of bilevel programming. It has been introduced by Labbé

0041-1655/01/3504/0345$05.00
1526-5447 electronic ISSN

et al. (1998) and applied to the optimal tariffing of single-commodity transportation services by Brotcorne (1998) and Brotcorne et al. (2000). More precisely, we consider the situation where the owner of a private toll highway seeks to maximize revenues raised from tolls set on a subset of arcs of a multicommodity, fixed demand transportation network, and where the commuters are assigned to shortest paths with respect to a generalized cost. In our model, we assume that congestion is not affected by the rerouting that could result from the introduction of tolls. An explicit account of congestion would radically transform the mathematical nature of the model and calls for entirely different algorithmic approaches. Note that such a model, involving queueing delays, was introduced by Yan and Lam (1996), but that these authors only addressed a simplistic two-arc situation.

In short, the main contribution of the paper is a robust algorithmic scheme that can solve to near optimality toll-setting problems of significant sizes. The proposed algorithms constitute nontrivial extensions to the multicommodity case of heuristics developed by Brotcorne et al. (2000).

1. A Bilevel Formulation of the Toll-Setting Problem

Let $G = (\mathcal{N}, \mathcal{A})$ be a transportation network where $\mathcal{N}$ denotes the set of nodes and where the arc set $\mathcal{A}$ is partitioned into the subset $\mathcal{A}_1$ of toll arcs and the subset $\mathcal{A}_2$ of toll-free arcs. With each arc a of $\mathcal{A}_1$ is associated a generalized travel cost composed of a fixed part c_a representing the minimal unit travel cost, and an additional unknown toll T_a expressed in time units. Any arc a of $\mathcal{A}_2$ bears a fixed unit travel cost d_a.

Let $\mathcal{K}$ denote the set of commodities. Each commodity k is associated with an origin-destination pair $(o(k), d(k))$. The demand vector b^k associated with each commodity k is specified by:

$$b_i^k = \begin{cases} n^k & \text{if } i = o(k), \\ -n^k & \text{if } i = d(k), \\ 0 & \text{otherwise,} \end{cases}$$

where n^k represents the number of users of commodity k. Finally, x_a^k denotes the number of users of commodity k on arc $a \in \mathcal{A}_1$, and y_a^k denotes the number of users of commodity k on arc $a \in \mathcal{A}_2$.

Neglecting congestion and assuming that demand is fixed, users are assigned to shortest paths linking their departure and arrival nodes, for given values of the tolls T_a set at the upper level of decision making. While the owner and the commuters act in a noncooperative fashion, we assume that, faced with two equally (un)attractive alternatives, a user will select the path that yields the highest revenue for the owner, i.e., in all likelihood, the quickest. This assumption is not unrealistic in that, given two equivalent paths, the one generating the highest revenue could be made the most attractive through a minute reduction of one of its tolls.

Another assumption underlying the above model is that the "value-of-time" parameter, which allows the conversion from time to money units, is uniform through the entire population of network users. However, our methodology could quite easily deal with the more general situation where users are distributed into classes, each endowed with its own perception of the value of one time unit, at the expense of a larger network. In this multiclass model, commuters associated with the same origin-destination pair could yet be assigned to different paths, depending on their perception of travel time.

Based on the above notation, the road pricing problem (RPP) can be formulated as a bilevel program with bilinear objectives and linear constraints, where it is understood that the flows x_a^k must be part of an optimal solution of the lower linear program parameterized by the upper-level toll vector T. If we assume that the toll T_a cannot exceed a prescribed, possibly infinite, upper bound $T_a^{\max}$, this bilevel problem takes the form

$$\begin{aligned}
\text{RPP}: \ & \max_{T,x,y} \sum_{a\in\mathcal{A}_1} T_a \sum_{k\in\mathcal{K}} x_a^k \\
& \text{s.t. } \ T_a \le T_a^{\max} \quad \forall\, a \in \mathcal{A}_1, \\
& \min_{x,y} \sum_{k\in\mathcal{K}} \left(\sum_{a\in\mathcal{A}_1} (c_a + T_a) x_a^k + \sum_{a\in\mathcal{A}_2} d_a y_a^k \right) \\
& \text{s.t. } \ \sum_{a\in i^+} \left(x_a^k + y_a^k\right) - \sum_{a\in i^-} \left(x_a^k + y_a^k\right) = b_i^k \\
& \qquad \forall\, k \in \mathcal{K}, \forall\, i \in \mathcal{N}, \\
& \qquad x_a^k \ge 0 \quad \forall\, k \in \mathcal{K}, \forall\, a \in \mathcal{A}_1, \\
& \qquad y_a^k \ge 0 \quad \forall\, k \in \mathcal{K}, \forall\, a \in \mathcal{A}_2.
\end{aligned}$$

The leader's objective is to maximize the total revenue, which is the sum of the products between toll T_a and the number of users on arc a. The objective of the lower-level problem is to minimize the total cost of the paths selected by the network users. Lower-level constraints are derived from flow conservation (demand) and flow nonnegativity.

As noticed by Labbé et al. (1998) for the general taxation problem, the leader's objective is neither a continuous nor a convex function of T. However, since it is upper semi-continuous, there exists at least one optimal solution to the above road pricing problem.

Throughout the remainder of this paper, we assume that there exists at least one path composed of untolled arcs for each commodity. This assumption prevents the optimal profit from growing unbounded and allows the derivation of a nontrivial upper bound on the leader's profit. This is expressed as the difference of two follower's optimal objectives, the first corresponding to infinite tolls (access to toll arcs is denied) and the second corresponding to null tolls. Note that this bound need not be reached at an optimal solution of the bilevel program. Note also that an optimal solution may involve negative tolls, as shown on the example of Figure 1, where demand is set to one on origin-destination pairs 1–2 and 3–4, and arcs (5, 6) and (6, 4) are subject to tolls. In this case, compensating interactions between tolls play an active role and the optimal solution, corresponding to a profit of 8 monetary units, is reached for $T_{56} = 5$ and $T_{64} = -2$.

Finally, we assume that there cannot exist a toll-setting scheme that generates profit and creates a negative cost cycle in the network. This assumption implies that the lower level optimal solution corresponds to a set of shortest paths.

Labbé et al. (1998) used a mixed integer programming formulation to solve to optimality small

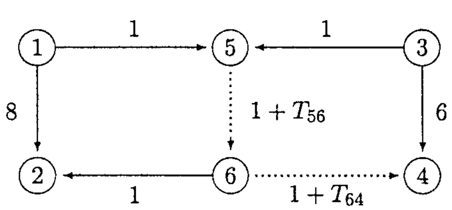

Figure 1 An Optimal Solution with a Negative Toll

instances of the above road pricing problem. Their formulation is based on the structure of the lower-level solution, which corresponds to paths carrying either no flow or all demand associated with a given origin-destination pair. Upon introduction of

$$e_i^k = \begin{cases} 1 & \text{if } i = o(k), \\ -1 & \text{if } i = d(k), \\ 0 & \text{otherwise,} \end{cases}$$

and redefining x_a^k and y_a^k as flow proportions, one can reformulate RPP as

$$\begin{aligned}
\max_{T,x,y} \quad & \sum_{a\in\mathcal{A}_1} T_a x_a \\
\text{s.t.} \quad & T_a \le T_a^{\max} \quad \forall a \in \mathcal{A}_1, \\
\min_{x,y} \quad & \sum_{a\in\mathcal{A}_1} (c_a + T_a) x_a + \sum_{a\in\mathcal{A}_2} d_a y_a \\
\text{s.t.} \quad & \sum_{a\in i^+} (x_a^k + y_a^k) - \sum_{a\in i^-} (x_a^k + y_a^k) = e_i^k \\
& \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{N}, \\
& x_a = \sum_{k\in\mathcal{K}} n^k x_a^k \quad \forall a \in \mathcal{A}_1, \\
& y_a = \sum_{k\in\mathcal{K}} n^k y_a^k \quad \forall a \in \mathcal{A}_2, \\
& x_a^k \ge 0 \quad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}_1, \\
& y_a^k \ge 0 \quad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}_2.
\end{aligned}$$

Next one replaces the lower-level linear program by its optimality conditions. This transformation generates a nonconvex constraint corresponding to the complementarity slackness condition, which can be linearized by introducing the commodity toll variables,

$$T_a^k = T_a x_a^k \quad \forall a \in \mathcal{A}_1, \forall k \in \mathcal{K},$$

as well as the constraints,

$$\begin{aligned}
& -M x_a^k \le T_a^k \le M x_a^k \quad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}_1, \\
& -M(1 - x_a^k) \le T_a^k - T_a \le M(1 - x_a^k) \quad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}_1, \\
& x_a^k \in \{0, 1\} \quad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}_1,
\end{aligned}$$

where M is some constant arbitrarily large with respect to data values. These modifications yield the mixed integer programming formulation:

$$\begin{aligned}
\max_{T,x,y}\quad & \sum_{k\in\mathcal{K}}\sum_{a\in\mathcal{A}_1} n^k T_a^k \\
\text{s.t.}\quad & \sum_{a\in i^+}\left(x_a^k+y_a^k\right)-\sum_{a\in i^-}\left(x_a^k+y_a^k\right)=e_i^k \\
& \qquad\qquad \forall k\in\mathcal{K}, \forall i\in\mathcal{N}, \\
& \lambda_i^k-\lambda_j^k\le c_a+T_a \quad \forall a=(i,j)\in\mathcal{A}_1, \forall k\in\mathcal{K}, \\
& \lambda_i^k-\lambda_j^k\le d_a \quad \forall a=(i,j)\in\mathcal{A}_2, \forall k\in\mathcal{K}, \\
& \sum_{a\in\mathcal{A}_1}\left(c_a x_a^k+T_a^k\right)+\sum_{a\in\mathcal{A}_2} d_a y_a^k=\lambda_{o(k)}^k-\lambda_{d(k)}^k \\
& \qquad\qquad \forall k\in\mathcal{K}, \\
& -Mx_a^k\le T_a^k\le Mx_a^k \quad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \\
& -M\left(1-x_a^k\right)\le T_a^k-T_a\le M\left(1-x_a^k\right) \\
& \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \\
& x_a^k\in\{0,1\} \quad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \\
& y_a^k\ge 0 \quad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_2, \\
& T_a\le T_a^{\max} \quad \forall a\in\mathcal{A}_1.
\end{aligned}$$

2. An Arc-Sequential Heuristic

If $|\mathcal{A}_1|=1$, then a simple procedure proposed by Labbé et al. (1998) yields the optimal toll. Indeed, let $\gamma_a^k(T_a)$ be the cost of a shortest path from $o(k)$ to $d(k)$, for a given value of T_a. The maximum toll that will make arc a attractive to a commuter travelling from $o(k)$ to $d(k)$ is given by

$$\pi_a^k=\gamma_a^k(\infty)-\gamma_a^k(0).$$

Let the commodities be sorted in nonincreasing order with respect to the quantities π_a^k, i.e.,

$$\pi_a^1\ge\pi_a^2\ge\cdots\ge\pi_a^{|\mathcal{K}|}.$$

For a toll level $T_a=\pi_a^l$, arc a will attract the commuters associated with commodity indices less than or equal to l and the revenue generated by arc a is given by

$$P\left(\pi_a^l\right)=\pi_a^l\sum_{k\le l} n^k.$$

If arc a were the sole toll arc, an optimal index l^* would be obtained by setting

$$l^*\in\arg\max_l\left\{\pi_a^l\sum_{k\le l} n^k\right\}, \tag{1}$$

and the corresponding optimal toll would be equal to $T_a=\pi_a^{l^*}$.

If $|\mathcal{A}_1|$ is larger than 1, one may apply this formula iteratively to generate an initial solution. At each iteration of this procedure, one optimizes with respect to toll T_a, fixing the other tolls to their preceding value. One must be careful, however, to take into account the impact of T_a on the revenues generated by the remaining toll arcs. Let P_-^k (respectively P_+^k) denote the sum of the tolls on the shortest path from $o(k)$ to i (respectively j to $d(k)$) and P^k the profit raised from a user associated with commodity k on a shortest path that does not pass through arc a. Then, for fixed tolls T_b, $b\ne a$, one may apply the following modification of Equation (1), which takes into account the fact that customers might not be diverted from their current path to a path passing through a:

$$l^*\in\arg\max_l\sum_{k\le l}\left(P_-^k+\pi_a^l+P_+^k-P^k\right)n^k. \tag{2}$$

A generic iteration of this procedure, together with the initialization phase, is outlined below.

An Arc-Sequential Heuristic

- Initialization
  - — Z (total profit) $\leftarrow 0$.
  - — $T_a\leftarrow\infty$, $\forall a\in\mathcal{A}_1$.
  - — P^k (profit generated from commodity k) $\leftarrow 0$, $\forall k\in\mathcal{K}$.
- Toll optimization
  - — Select a toll arc $a=(i,j)$.
  - — Set tolls (except on arc a) to their current value.
  - — Let $\gamma^k(\infty)$ be the cost of a shortest path from $o(k)$ to $d(k)$ with T_a set at ∞.
  - — For every $k\in\mathcal{K}$, determine the shortest path from $o(k)$ to i with respect to current tolls. Let C_-^k be its cost and compute the sum of the tolls P_-^k on that path.

- For every $k \in \mathcal{K}$, determine the shortest path from j to $d(k)$ with respect to current tolls. Let C_+^k be its cost and compute the sum of the tolls P_+^k on that path.
- $\gamma^k(0) \leftarrow C_-^k + C_+^k + c_a$.
- Let $\tau_{\min}$ be the smallest toll value that induces no negative circuit in the network.
- For every $k \in K$, set $\pi_a^0 = \infty$ and $\pi_a^k = \gamma^k(\infty) - \gamma^k(0)$.
- Order π_a^k in nonincreasing order: $\pi_a^0 \geq \pi_a^1 \geq \pi_a^2 \geq \cdots \geq \pi_a^{|\mathcal{K}|}$.
- $l^* \in \arg\max_{\tau_{\min} \leq \pi_a^l \leq T_a^{\max}} \sum_{k \leq l} (P_-^k + \pi_a^l + P_+^k - P^k) n^k$.
- Update P_k and $T_a \leftarrow \pi_a^{l^*}$.

The procedure is halted whenever no improvement is observed after a full cycle over the toll arcs has been completed. Note that the ordering of the toll arcs might clearly influence the quality of the solution.

The computational complexity of this algorithm is dominated by the computation of shortest paths to and from the arc $a = (i, j)$ being optimized, and is therefore $O(|\mathcal{A}_1||\mathcal{K}||\mathcal{N}|^3)$.

3. A Primal-Dual Heuristic

The main difficulty in solving bilevel programs is due to the complementarity constraints implicit in the first-order optimality conditions of the lower-level linear program. There are several ways to deal with these constraints. In this section, we propose an iterative algorithm which relies on the reformulation of the toll problem as a single-level bilinear program, through the use of an exact penalty function. The algorithm is a multicommodity generalization of a primal-dual method proposed by Brotcorne et al. (2000) for solving a freight tariff setting problem where the lower level consists of a single-commodity transportation problem. In the multicommodity case, we introduce a quadratic penalty term that forces the tolls associated with each commodity to be equal and the resulting nonlinear problem is solved using Frank and Wolfe's (1956) linearization scheme.

Let A_1 (respectively A_2) be the node-arc incidence matrix associated with the arcs in $\mathcal{A}_1$ (respectively $\mathcal{A}_2$). If one redefines commodity flows as the set of flows associated with a given origin, RPP can be written as:

$$
\begin{aligned}
\text{RPP2}: \quad & \max_{T \leq T^{\max}, x, y} \sum_{k \in \mathcal{K}} T x^k, \\
& \min_{x, y} \sum_{k \in \mathcal{K}} ((c+T)x^k + d y^k), \\
& A_1 x^k + A_2 y^k = b^k \quad \forall k \in \mathcal{K}, \\
& x^k, y^k \geq 0 \quad \forall k \in \mathcal{K}.
\end{aligned} \tag{3}
$$

For fixed T, the follower's problem (FP) of (3) is linear and its constraint set is separable by commodities. We denote by FP_k the subproblem associated with commodity k and by λ^k the corresponding dual variables. Replacing FP by its primal-dual optimality constraints yields the following single-level program, which is equivalent to RPP:

$$
\begin{aligned}
\max_{T \leq T^{\max}, x, y} \quad & \sum_{k \in \mathcal{K}} T x^k \\
\text{s.t.} \quad & A_1 x^k + A_2 y^k = b^k \quad \forall k \in \mathcal{K}, \\
& x^k, y^k \geq 0 \quad \forall k \in \mathcal{K}, \\
& \lambda^k A_1 \leq c + T \quad \forall k \in \mathcal{K}, \\
& \lambda^k A_2 \leq d \quad \forall k \in \mathcal{K}, \\
& \sum_{k \in \mathcal{K}} ((c+T)x^k + d y^k - \lambda^k b^k) = 0.
\end{aligned} \tag{4}
$$

The sole nonlinear constraint of the above problem is the constraint stating the equality of the primal and dual objectives. We penalize this constraint into the objective to obtain the single-level bilinear program,

$$
\begin{aligned}
\max_{T \leq T^{\max}, x, y} \quad & \sum_{k \in \mathcal{K}} T x^k - M_1 \sum_{k \in \mathcal{K}} ((c+T)x^k + d y^k - \lambda^k b^k) \\
\text{s.t.} \quad & A_1 x^k + A_2 y^k = b^k \quad \forall k \in \mathcal{K}, \\
& x^k, y^k \geq 0 \quad \forall k \in \mathcal{K}, \\
& \lambda^k A_1 \leq c + T \quad \forall k \in \mathcal{K}, \\
& \lambda^k A_2 \leq d \quad \forall k \in \mathcal{K},
\end{aligned} \tag{5}
$$

where $M_1 > 0$. For each value of the index k, we replace the toll vector T by a commodity vector T^k and penalize the compatibility constraint,

$$T^k = T^1 \quad \forall k \in \mathcal{K},$$

to derive the twin-penalty mathematical program,

$$\begin{aligned}
\text{PEN}: \max_{T\le T^{\max}, x, y, \lambda} \quad & F(T,x,y,\lambda) \\
&= \sum_{k\in\mathcal{K}} \big[T^k x^k - M_1\big((c+T^k)x^k \\
&\qquad + dy^k - \lambda^k b^k\big) - M_2 \|T^k - T^1\|^2\big] \\
\text{s.t.} \quad & A_1 x^k + A_2 y^k = b^k \quad \forall k\in\mathcal{K}, \\
& x^k, y^k \ge 0 \quad \forall k\in\mathcal{K}, \\
& \lambda^k A_1 \le c + T^k \quad \forall k\in\mathcal{K}, \\
& \lambda^k A_2 \le d \quad \forall k\in\mathcal{K},
\end{aligned} \tag{6}$$

where M_2 is a positive penalty factor.

The aim of our general primal-dual scheme is to induce, through updates of the penalty parameters M_1 and M_2, basis changes for the follower's problem. In this process, extremal flow assignments corresponding to distinct values of the toll vector T are generated, and we expect one of these combinations to be of high quality for RPP. These solutions can be improved by noting that, for a given lower-level flow vector (x, y), one can derive the profit-maximizing toll vector that is compatible with (x, y) by solving a simple linear problem (see also Labbé et al. 1998).

The algorithm is composed of three main steps. At Step 1, given a feasible lower level basic solution (x^k, y^k), the algorithm sets the toll vectors T^k to the (partial) optimal solutions of the convex quadratic program PEN. At Step 2, a new flow vector (x^k, y^k) is obtained as the solution of the lower-level problem corresponding to the toll vectors T^k. Finally, Step 3 consists in computing the best common toll vector T that is compatible with the current flow vector (x^k, y^k). The algorithm is outlined below and its main components will be explicit in the forthcoming subsections.

A Primal-Dual Algorithm

Step 0. Initialization

1. $\ell = 0$ and $Z^* = -\infty$.
2. $T_0^k = 0$ for all $k \in \mathcal{K}$.
3. Initialize M_1 and M_2.
4. Go to Step 2.

Step 1. Computation of the commodity tolls T^k. For fixed $x_{\ell-1}^k$ and $y_{\ell-1}^k$, let T_ℓ^k and λ_ℓ^k be solutions for the penalized problem,

$$\begin{aligned}
\text{QP}(x_{\ell-1}, y_{\ell-1}): \max_{T,\lambda} \quad & \sum_{k\in\mathcal{K}} \big[T^k x_{\ell-1}^k - M_1\big((c+T^k)x_{\ell-1}^k \\
&\qquad + dy_{\ell-1}^k - \lambda^k b^k\big) - M_2\|T^k - T^1\|^2\big] \\
\text{s.t.} \quad & T^k \le T^{\max} \quad \forall k\in\mathcal{K}, \\
& \lambda^k A_1 \le c + T^k \quad \forall k\in\mathcal{K}, \\
& \lambda^k A_2 \le d \quad \forall k\in\mathcal{K}.
\end{aligned}$$

(The resolution of $\text{QP}(x_{\ell-1}^k y_{\ell-1}^k)$, including the update of the penalty parameter M_1, is discussed in §3.1.)

Step 2. Computation of commodity flow vectors x^k and y^k. For fixed T_ℓ^k, solve for each commodity $k \in \mathcal{K}$,

$$\begin{aligned}
FP_k: \min_{x,y} \quad & (c + T_\ell^k)x^k + dy^k \\
\text{s.t.} \quad & A_1 x^k + A_2 y^k = b^k, \\
& x^k, y^k \ge 0,
\end{aligned}$$

for x_l^k, y_l^k and the dual vectors λ_l^k.

Step 3. Computation of optimal tolls for given flows. If flows are identical to those of the previous iteration, go to Step 4. Otherwise:

1. Let $\widetilde{T}_\ell(x_\ell, y_\ell)$ be the optimal partial solution of the linear program

$$\begin{aligned}
\max_{T,\lambda} \quad & \sum_{k\in\mathcal{K}} T x_\ell^k \\
\text{s.t.} \quad & T \le T^{\max}, \\
& \lambda^k A_1 \le c + T \quad \forall k\in\mathcal{K}, \\
& \lambda^k A_2 \le d \quad \forall k\in\mathcal{K}, \\
& \sum_{k\in\mathcal{K}} \big((c+T)x_\ell^k + dy_\ell^k - \lambda^k b^k\big) = 0.
\end{aligned}$$

2. $\widetilde{Z} \leftarrow \sum_{k\in\mathcal{K}} \widetilde{T}_\ell x_\ell^k$.
3. If $\widetilde{Z} > Z^*$, then $Z^* \leftarrow \widetilde{Z}$ and $(T^*, x^*, y^*) \leftarrow (\widetilde{T}_\ell(x_\ell, y_\ell), x_\ell, y_\ell)$.

Step 4. Stopping criterion.

— If the stopping criterion is reached, then adopt (T^*, x^*, y^*) as the solution.
— Increase the quadratic penalty term M_2.
— Set $\ell \leftarrow \ell + 1$ and go to Step 1.

At Step 1 of the above procedure, we determine commodity tolls that achieve a high profit while being "nearly equal," and maintaining a low duality gap. However, basis changes could be inhibited if the penalty factor M_1 is too strong (see next subsection), i.e., the duality gap is too small and the algorithm gets trapped into a local minimum. To circumvent this problem, the penalty parameters M_1 and M_2 need to be carefully updated. The adjustment strategies are discussed in §4.

At Step 2, flow vectors that are compatible with the current commodity tolls are computed. In the event where lower-level shortest paths are nonunique, the costs of the toll arcs are perturbed, in order to entice the follower into selecting profit-maximizing paths. More precisely,

$$c_a \leftarrow c_a + (1-\epsilon)T_a \quad \text{if } a \in \mathcal{A}_1.$$

It can be shown that, if ϵ is suitably small, the lower-level solution possesses the required property of maximizing the leader's profit while being optimal with respect to the original arc costs.

At Step 3, the algorithm computes a common toll vector that maximizes total profit while maintaining the lower level optimality of the current commodity flows. The structure of this program is that of an uncapacitated multicommodity network flow problem and is thus easy to solve.

The stopping criterion of the primal-dual heuristic is based on the reduction rate of the objective function of the penalized problem evaluated at the end of Step 2:

$$\begin{aligned} &F(T_\ell, x_\ell, y_\ell, \lambda_\ell) \\ &\quad = \sum_{k\in\mathcal{K}} \big[T_l^k x_l^k - M_1\big((c+T_\ell^k)x_l^k + dy_\ell^k - \lambda_\ell^k b^k\big) \\ &\qquad\qquad - M_2 \|T_\ell^k - T_\ell^1\|^2\big]. \end{aligned}$$

Since $x_\ell^k, y_\ell^k, \lambda_\ell^k$ are the primal and dual optimal solutions for the follower's problem for fixed toll level T_ℓ^k, the duality gap is zero, and the above expression reduces to

$$F(T_\ell, \lambda_\ell, x_\ell, y_\ell) = \sum_{k\in\mathcal{K}} T_\ell^k x_\ell^k - M_2 \sum_{k\in\mathcal{K}} \|T_\ell^k - T_\ell^1\|^2.$$

Note that this monotone increasing function is discontinuous whenever a change in tolls T (Step 1) induces a basis change at the lower level (Step 2).

3.1. Resolution of the Quadratic Subproblem

At Step 1 of the algorithm, the convex quadratic program $\mathrm{QP}(x_{\ell-1}, y_{\ell-1})$ is solved using Frank and Wolfe's (1956) linearization method, whose main advantage is to preserve the uncapacitated network structure of the lower level multicommodity flow problem. Lower bound constraints,

$$T^k \ge T^{\min}, \tag{7}$$

involving a large negative number $T^{\min}$ are appended to $\mathrm{QP}(x_{\ell-1}, y_{\ell-1})$ in order to prevent the linearized problem to yield an unbounded solution.

Let us drop, for ease of presentation, the iteration index ℓ. In the initialization phase, the penalty factor M_1 is reset to its initial value. At iteration m of the Frank-Wolfe procedure, (T_m, λ_m) solve the linearized problem,

$$\begin{aligned} \mathrm{PL}(T_{m-1}, \lambda_{m-1}) : \max_{S,\mu}\ & \nabla_T F(T_{m-1}, \lambda_{m-1})S \\ & + \nabla_\lambda F(T_{m-1}, \lambda_{m-1})\mu \\ \text{s.t.}\ & T^{\min} \le S^k \le T^{\max} \quad \forall k \in \mathcal{K}, \\ & \mu^k A_1 \le c + S^k \quad \forall k \in \mathcal{K}, \\ & \mu^k A_2 \le d \quad \forall k \in \mathcal{K}, \end{aligned} \tag{8}$$

where $S = (S^k)_{k\in\mathcal{K}}$ and $\mu = (\mu^k)_{k\in\mathcal{K}}$ are the auxiliary variables of the linearized problem. Let (S_m, μ_m) be an optimal solution of PL. We have:

$$\begin{aligned} \alpha_m &\in \arg\max_{\alpha\in[0,1]} F(\alpha) \\ &= F(T_{m-1} + \alpha(S_m - T_{m-1}), \lambda_{m-1} + \alpha(\mu_m - \lambda_{m-1})), \\ (T_m, \lambda_m) &= (T_{m-1} + \alpha_m(S_m - T_{m-1}), \lambda_{m-1} \\ &\qquad + \alpha_m(\mu_m - \lambda_{m-1})). \end{aligned}$$

If the duality gap is not too small, the parameter M_1 is increased.

Note that, at the end of Step 2 of the primal-dual algorithm, the duality gap is necessarily equal to zero, while it might assume positive values in the subsequent iterations of the Frank-Wolfe procedure. Since a low value of the gap is positively correlated with a small number of basis changes at Step 2, the number of Frank-Wolfe iterates was deliberately set to a small value in order to induce basis changes. Indeed, the larger the number of lower level solutions (basis) visited, the higher the probability of obtaining, through the "inverse optimization" procedure of Step 3, a near-optimal tax vector compatible with this basis.

3.2. Resolution of the Frank-Wolfe Subproblem

The efficiency of the overall algorithm depends on the existence of an efficient method for solving the Frank-Wolfe linear approximation subproblems. For simplicity, we drop the iteration index m. The Frank-Wolfe linearized subproblem takes the form

$$\begin{aligned}\max_{S,\mu}\quad & S^1\frac{\partial F}{\partial T^1}+\sum_{k=2}^{|\mathcal{K}|}S^k\frac{\partial F}{\partial T^k}+\sum_{k=1}^{|\mathcal{K}|}\mu^k\frac{\partial F}{\partial \lambda^k}\\ \text{s.t.}\quad & T^{\min}\le S^k\le T^{\max}\quad \forall k\in\mathcal{K},\\ & \mu^k A_1\le c+S^k\quad \forall k\in\mathcal{K},\\ & \mu^k A_2\le d\quad \forall k\in\mathcal{K},\end{aligned}\tag{9}$$

where, for each commodity k:

$$\frac{\partial F}{\partial \lambda^k}=M_1 b^k.$$

For commodity 1, we have

$$\frac{\partial F}{\partial T^1}=(1-M_1)x^1-2M_2\left(|\mathcal{K}|T^1-\sum_{k=2}^{|\mathcal{K}|}T^k\right)^t,$$

and for the remaining commodities,

$$\frac{\partial F}{\partial T^k}=(1-M_1)x^k-2M_2(T^k-T^1)^t,$$

where the symbol ^{t} denotes the transposition operator.

The problem $\mathrm{PL}(T,\lambda)$ is separable with respect to the commodities. The linear program associated with an arbitrary commodity, whose index is omitted, takes the form

$$\begin{aligned}\max_{S,\mu}\quad & \sum_{(i,j)\in\mathcal{A}_1}S_{i,j}\frac{\partial F}{\partial T_{i,j}}+\sum_{i\in\mathcal{N}}\mu_i\frac{\partial F}{\partial \lambda_i}\\ \text{s.t.}\quad & T^{\min}_{i,j}\le S_{i,j}\le T^{\max}_{i,j}\quad \forall (i,j)\in\mathcal{A}_1,\\ & \mu_i-\mu_j-c_{i,j}\le S_{i,j}\quad \forall (i,j)\in\mathcal{A}_1,\\ & \mu_i-\mu_j\le d_{i,j}\quad \forall (i,j)\in\mathcal{A}_2.\end{aligned}\tag{10}$$

Let us partition the set of toll arcs $\mathcal{A}_1$ into the two subsets:

$$\mathcal{A}_{11}=\left\{(i,j)\in\mathcal{A}_1:\ \frac{\partial F}{\partial T_{i,j}}>0\right\}$$

and

$$\mathcal{A}_{12}=\left\{(i,j)\in\mathcal{A}_1:\ \frac{\partial F}{\partial T_{i,j}}\le 0\right\}.$$

Since the problem is a maximization problem, the $S_{i,j}$-component of the vector S associated with an arc belonging to $\mathcal{A}_{11}$ will be set to its largest possible value, i.e., $S_{i,j}=T^{\max}_{i,j}$. The problem (10) can thus be rewritten as:

$$\begin{aligned}\max_{S,\mu}\quad & \sum_{(i,j)\in\mathcal{A}_{12}}S_{i,j}\frac{\partial F}{\partial T_{i,j}}+\sum_{i\in\mathcal{N}}\mu_i\frac{\partial F}{\partial \lambda_i}\\ \text{s.t.}\quad & \mu_i-\mu_j-c_{i,j}\le T^{\max}_{i,j}\quad \forall (i,j)\in\mathcal{A}_{11},\\ & T^{\min}_{i,j}\le S_{i,j}\le T^{\max}_{i,j}\quad \forall (i,j)\in\mathcal{A}_{12},\\ & \mu_i-\mu_j-c_{i,j}\le S_{i,j}\quad \forall (i,j)\in\mathcal{A}_{12},\\ & \mu_i-\mu_j\le d_{i,j}\quad \forall (i,j)\in\mathcal{A}_2.\end{aligned}\tag{11}$$

Given that the partial derivatives $\partial F/\partial T_{i,j}$ are nonpositive for all arc $(i,j)\in\mathcal{A}_{12}$, the component $S_{i,j}$ will be set to its smallest possible value. Consequently, we may restrict our attention to solutions of the form $S_{i,j}=\mu_i-\mu_j-c_{i,j}$ for all $(i,j)\in\mathcal{A}_{12}$ and $T^{\min}_{i,j}\le\mu_i-\mu_j-c_{i,j}\le T^{\max}_{i,j}$ for all $(i,j)\in\mathcal{A}_{12}$. Moreover, since the constraint $\mu_i-\mu_j-c_{i,j}\ge T^{\min}_{i,j}$ is never tight if the lower bound $T^{\min}$ is sufficiently large (negative), the linearized problem can be expressed as

$$\begin{aligned}\max_{\mu}\quad & \sum_{(i,j)\in\mathcal{A}_{12}}(\mu_i-\mu_j-c_{i,j})\frac{\partial F}{\partial T_{i,j}}+\sum_{i\in\mathcal{N}}\mu_i\frac{\partial F}{\partial \lambda_i}\\ \text{s.t.}\quad & \mu_i-\mu_j-c_{i,j}\le T^{\max}_{i,j}\quad \forall (i,j)\in\mathcal{A}_{11},\\ & T^{\min}_{i,j}\le\mu_i-\mu_j-c_{i,j}\le T^{\max}_{i,j}\quad \forall (i,j)\in\mathcal{A}_{12},\\ & \mu_i-\mu_j\le d_{i,j}\quad \forall (i,j)\in\mathcal{A}_2.\end{aligned}\tag{12}$$

Now, let us decompose the node-toll arcs adjacency matrix A_1 in A_{11} and A_{12} where A_{11} (respectively A_{12}) is the submatrix of A_1 corresponding to the arcs of $\mathcal{A}_{11}$ (respectively $\mathcal{A}_{12}$). This notation is extended to the toll and cost vectors, in the obvious way and we redefine $\partial F/\partial T$ as the vector of components associated with arcs $(i,j)\in\mathcal{A}_{12}$. The problem becomes

$$\begin{aligned}\max_{\mu}\quad & (\mu A_{12}-c_{12})\frac{\partial F}{\partial T}+M_1\mu b\\ \text{s.t.}\quad & \mu A_1\le c+T^{\max},\\ & \mu A_{12}\ge c_{12}+T^{\min}_{12},\\ & \mu A_2\le d,\end{aligned}$$

Table 1 Symmetric Networks with 10 O-D Pairs

		SR 2			SR 5			PD			CPLEX	
$T^{\max}$	% T	# T	%OPT	CPU	# T	%OPT	CPU	# T	%OPT	CPU	# Nodes	CPU
		10	97.39	11	10	99.58	25	10	100.00	15	2	1
		10	90.02	13	10	91.20	30	10	100.00	18	38	1
		9	99.01	8	9	99.01	23	10	99.50	18	167	3
10	5	9.67	95.47	10.34	9.67	96.60	25.92	10.00	99.83	17.00	69.00	2.01
		20	99.80	27	20	99.80	62	21	100.00	16	200	12
		20	93.61	22	20	93.61	55	21	100.00	17	271	20
		21	92.38	28	21	95.69	66	21	100.00	14	400	21
10	10	20.33	95.26	25.67	20.33	96.37	60.80	21.00	100.00	15.67	290.33	17.69
		26	91.82	31	23	95.01	100	31	97.44	18	600	49
		28	93.05	46	28	97.38	98	31	100.00	14	168	12
		26	96.80	35	26	99.95	86	30	100.00	19	937	67
10	15	26.67	93.89	37.20	25.67	97.45	94.64	30.67	99.15	17.00	568.33	42.39
		40	85.72	53	40	92.01	150	42	100.00	19	2400	316
		38	96.18	59	38	96.18	148	39	100.00	18	1700	192
		38	95.20	54	38	95.20	136	42	100.00	15	101	13
10	20	38.67	92.37	56.76	38.67	94.46	141.96	41.00	100.00	17.33	1400.33	173.60
		10	99.51	11	10	99.51	27	10	97.48	17	110	3
		10	94.75	9	10	94.75	26	10	100.00	19	209	5
		9	99.07	9	9	99.07	22	10	99.07	16	5400	80
20	5	9.67	97.78	9.89	9.67	97.78	25.14	10.00	98.85	17.33	1906.33	29.17
		21	96.26	25	21	97.59	57	21	99.26	19	52600	1743
		21	85.88	24	21	96.51	61	21	98.44	19	3553	214
		21	87.20	23	21	91.42	57	21	98.07	18	319	10
20	10	21.00	89.78	24.16	21.00	95.17	58.34	21.00	98.59	18.67	18824.00	655.80
		24	93.19	40	23	93.59	88	31	99.13	22	42354	4990
		28	97.90	39	28	97.90	99	31	99.82	21	22730	9520
		26	94.99	38	26	97.24	87	30	96.43	22	39800	2071
20	15	26.00	95.36	38.79	25.67	96.24	91.31	30.67	98.46	21.33	38318.00	2682.59
	*	36	88.96	69	36	88.96	157	40	99.55	25	62451	4516
		31	89.74	63	35	95.47	135	37	97.89	26	800	47
		38	88.37	53	38	91.23	167	42	99.24	22	1000	73
20	20	35.00	89.35	62.03	36.33	92.22	152.82	39.67	98.89	24.33	21417.00	1545.46

or

$$\begin{aligned} \max_{\mu} \quad & \mu\left(A_{12}\frac{\partial F}{\partial T}+M_1 b\right), \\ \text{s.t} \quad & \mu A_1 \le c+T^{\max}, \\ & -\mu A_{12} \le -c_{12}-T_{12}^{\min}, \\ & \mu A_2 \le d. \end{aligned} \tag{13}$$

Let $z \in \mathbb{R}^{m_1}$, $v \in \mathbb{R}^{m_{12}}$, $u \in \mathbb{R}^{m_2}$ be the dual variables associated with the constraints of (13), where m_2 is the number of toll-free arcs, m_1 the number of toll arcs and m_{12} the number of toll arcs belonging to $\mathcal{A}_{12}$. The dual of this linear program is

$$\begin{aligned} \text{TPP}: \quad \min_{u,z,v} \quad & du+(c+T^{\max})z+(-T_{12}^{\min}-c_{12})v \\ \text{s.t.} \quad & A_2 u+A_1 z-A_{12}v = A_{12}\frac{\partial F}{\partial T}+M_1 b, \\ & z, u, v \ge 0. \end{aligned}$$

The problem TPP is simply a transshipment problem on a modified network $\widetilde{G}=(\mathcal{N},\widetilde{\mathcal{A}})$ whose arc set is composed of

- the untolled arcs $(i,j)\in\mathcal{A}_2$, with costs $d_{i,j}$, whose flow variables are denoted by $u_{i,j}$;

Table 2 Symmetric Networks with 20 O-D Pairs

		SR 2			SR 5			PD			CPLEX	
$T^{\max}$	% T	# T	%OPT	CPU	# T	%OPT	CPU	# T	%OPT	CPU	# Nodes	CPU
		10	100.00	18	10	100.00	41	10	100.00	27	213	30
		10	93.14	17	10	93.14	41	10	98.88	26	66	12
		9	97.90	15	10	100.00	38	10	98.50	33	700	72
10	5	9.67	97.01	16.88	10.00	97.71	40.00	10.00	99.13	28.67	326.33	38.12
		21	95.35	40	21	96.59	99	21	98.94	35	8700	1078
		21	92.04	36	21	95.79	89	21	98.53	40	60900	5903
	*	19	81.98	35	21	95.61	106	21	97.01	41	100500	18000
10	10	20.33	89.79	37.20	21.00	96.00	97.92	21.00	98.16	38.67	56700.00	8326.97
	*	28	89.96	67	29	90.44	154	31	94.28	39	59200	18000
		31	95.57	71	31	95.57	156	31	98.64	37	12900	3923
	*	31	92.47	71	31	92.47	166	31	97.95	36	50500	18000
10	15	30.00	92.67	69.44	30.33	92.83	158.96	31.00	96.95	37.33	40866.67	13307.76
	*	38	92.17	78	38	92.17	254	42	94.29	39	34300	18000
	*	39	90.65	79	40	92.63	220	42	98.71	39	42700	18000
		39	91.09	100	39	92.69	252	42	99.70	35	6500	936
10	20	38.67	91.30	85.44	39.00	92.50	241.92	42.00	97.57	37.67	27833.33	12312.14

- the toll arcs $(i, j) \in \mathcal{A}_1$, with costs $c_{i,j} + T^{\max}_{i,j}$, whose flow variables are denoted by $z_{i,j}$;
- the inverse arcs (j, i) such that $(i, j) \in \mathcal{A}_{12}$, with costs $-c_{i,j} - T^{\min}_{i,j}$, whose flow variables are denoted by $v_{i,j}$.

4. Numerical Results

The heuristics developed in this paper have been tested on two varieties of randomly generated grid networks with 60 nodes (5×12), 208 two-way arcs, 10 to 20 origin-destination pairs, and where the proportion of toll arcs varies from 5 to 20%. In Type I problems, toll arcs are scattered throughout the network while, in problems of Type II, chains of toll arcs corresponding to toll highways are generated. In the latter case, a symmetric cost structure has been adopted for the sake of realism. The process of randomly generating toll arcs is described in Brotcorne et al. (2000). The upper bound $T^{\max}$ has been set either to a low value (10) or a high value (20), that is, the maximum value of an arc's initial cost c_a.

For the primal-dual heuristic, the penalty factor M_1 has been initialized to 1.1 and incremented by 0.1 at the end of each Frank-Wolfe iteration, while the number of Frank-Wolfe iterations m_{FW} has been set to a low value (2) in order to induce basis changes at the lower level. The setting of these parameters achieves a trade-off between two conflicting objectives: maximizing the number of bases visited and reducing the time-consuming process of optimization with respect to each basis. Our choice is consistent with our computational experiments; indeed we observed that the best lower level bases generally occur at the beginning of the process and that a rapid increase of the penalty parameter M_1 could prevent the algorithm of discovering promising solutions (bases). As for the quadratic penalty factor M_2, it has been initialized to 10 and increased by units of 1. Finally the primal-dual heuristic is halted as soon as the improvement in the objective function value of the penalized problem evaluated at the end of each main iteration becomes smaller than 10% for blocks of 30 main iterations.

The heuristics developed in this paper were coded in C and computational results obtained on a SUN ULTRA (360 Mhz). The transshipment subproblems were solved using the minimum cost flow code of Goldberg and Tarjan (1990). The numerical results are summarized in Tables 1–6, where the first two columns of each table provide the value of $T^{\max}$ and the percentage of toll arcs, respectively. The "arc-sequential" heuristic has been performed for various permutations of the set of toll arcs; the results

Table 3 Asymmetric Networks with 10 O-D Pairs

		SR 2			SR 5			PD			CPLEX	
T^{max}	% T	# T	%OPT	CPU	# T	%OPT	CPU	# T	%OPT	CPU	# Nodes	CPU
		10	100.00	9	10	100.00	23	10	100.00	17	72	2
		10	86.21	10	10	88.97	24	10	99.82	19	102	2
		10	97.98	9	10	97.98	20	10	100.00	12	40	3
10	5	10.00	94.73	9.28	10.00	95.62	22.08	10.00	99.94	16.00	71.33	2.37
		20	96.53	26	20	96.53	60	21	100.00	14	2	3
		21	92.07	20	21	99.64	55	21	100.00	17	31	2
		20	91.54	28	20	91.54	63	21	99.02	16	85	7
10	10	20.33	93.38	24.80	20.33	95.91	59.04	21.00	99.67	15.67	39.33	3.85
		28	91.14	34	27	98.24	88	31	100.00	16	27	5
		28	93.91	33	29	99.92	87	31	98.80	16	545	135
		28	98.02	32	28	98.02	83	31	99.67	16	63	8
10	15	28.00	94.35	32.80	28.00	98.73	86.16	31.00	99.49	16	211.67	49.37
		37	84.35	54	37	88.70	139	41	100.00	19	800	134
		36	89.36	81	34	92.21	137	40	99.86	19	12	7
		34	88.34	59	33	93.68	146	41	99.00	22	1282	210
10	20	35.67	87.35	64.56	34.67	91.53	140.48	40.67	99.62	20	698.00	117.25
		10	95.52	8	10	100.00	21	10	100.00	16	719	23
		10	100.00	9	10	96.90	23	10	100.00	16	165	6
		10	97.94	9	10	97.94	24	10	100.00	16	377	13
20	5	10.00	97.82	8.80	10.00	98.28	22.64	10.00	100.00	16.00	420.33	14.08
		20	93.84	24	20	93.84	64	21	98.63	19	400	12
		20	91.25	20	21	94.04	54	21	99.00	19	3176	169
		20	95.82	28	20	95.82	61	20	100.00	20	1429	89
20	10	20.00	93.64	23.76	29.33	94.57	59.76	20.66	99.21	19.33	1668.33	89.96
		28	95.37	36	27	97.33	104	30	100.00	22	10288	563
		29	85.28	36	29	89.70	88	31	98.08	17	2005	96
		27	96.48	40	27	96.48	88	31	99.97	24	4200	257
20	15	28.00	92.38	37.44	25.67	94.50	93.68	30.67	99.35	21.00	5497.67	305.12
	*	37	84.57	58	37	84.57	155	41	96.46	27	125000	18000
		34	95.37	60	34	95.37	141	41	96.74	23	1300	106
	*	31	83.94	66	31	88.34	144	40	92.58	26	163500	18000
20	20	34.00	87.96	61.36	34.00	89.43	146.72	40.66	95.44	25.33	96600.00	12035.33

corresponding to 2 (respectively 5) random permutations are displayed in column SR2 (respectively column SR5). The last line of each subtable contains the average statistics for the corresponding data set, and the column label "%OPT" refers to the ratio of the heuristic objective over the optimal solution achieved by the mixed integer programming code CPLEX 6.5, which was halted whenever a time limit of 5 hours was reached or memory requirements became excessive. In both these cases, the optimum value was replaced by the best upper bound achieved. This is indicated by a star (*) in the tables' second columns. The label "#T" refers to the number of toll arcs with nonzero flow in the final solution. The label "NODES" refers to the number of nodes of the branch-and-bound tree explored while solving the MIP formulation of the toll setting problem; the label "CPU" refers to CPU times expressed in seconds.

As a general rule, the results were independent of the network's topology, and the primal-dual heuristic sharply outperformed the "arc-sequential" one. Typically, the primal-dual heuristic produces solutions within 1.5% of optimality, whereas the arc-sequential heuristic provides solutions within 7% of optimality for 2 toll arc permutations and within 5% of optimality for 5 toll arc permutations. With the exception of

Table 4 Asymmetric Networks with 20 O-D Pairs

		SR 2			SR 5			PD			CPLEX	
T^{max}	% T	# T	%OPT	CPU	# T	%OPT	CPU	# T	%OPT	CPU	# Nodes	CPU
		9	98.23	16	9	98.23	40	10	98.58	39	318	41
		10	98.93	14	10	98.95	36	10	100.00	23	118	11
		10	94.22	16	10	98.21	40	10	95.49	24	232	29
10	5	9.67	97.13	15.28	9.67	98.46	38.48	10.00	98.04	28.67	222.67	27.19
		21	96.73	39	20	97.01	107	21	100.00	29	3000	403
		21	89.26	33	21	89.26	82	21	100.00	28	59	18
		21	91.66	43	21	95.51	102	21	99.93	45	900	128
10	10	21.00	92.55	38.24	20.67	93.93	97.04	21.00	99.98	34.00	1319.67	183.09
		29	93.51	59	30	94.36	161	31	99.82	31	2906	7834
		29	95.22	81	28	96.24	176	30	97.83	42	21176	86400
		30	93.25	66	30	93.25	181	31	98.66	46	8229	14957
10	15	29.33	93.99	68.56	29.33	94.61	172.48	30.67	98.77	36.33	19786.33	5570.38
	*	39	94.17	89	39	95.88	214	42	98.94	36	23900	18000
	*	41	90.12	102	41	90.12	234	42	97.00	34	26000	18000
		38	89.36	99	39	93.35	261	42	100.00	42	11400	9878
10	20	39.33	91.22	97.04	39.67	93.12	236	42.00	98.65	37.33	20433.33	15292.76

Table 5 Symmetric Networks with Highways: 10 O-D Pairs

		SR 2			SR 5			PD			CPLEX	
T^{max}	% T	# T	%OPT	CPU	# T	%OPT	CPU	# T	%OPT	CPU	# Nodes	CPU
		19	93.19	21	19	97.31	51	20	100.00	18	120	6
		20	88.43	20	20	93.99	55	21	100.00	17	21	4
		21	82.38	20	21	87.01	64	21	98.86	17	400	27
10	10	20.00	88.00	20.48	20.00	92.77	56.67	20.67	99.62	17.33	180.33	12.47
		27	89.62	40	30	93.84	87	29	99.88	18	394	27
		28	92.61	54	28	92.61	129	31	100.00	15	401	39
		23	91.76	37	23	97.44	88	27	100.00	17	500	45
10	15	26.00	91.33	43.68	27.00	94.63	101.32	29.00	99.96	16.67	431.67	36.98
		37	95.79	60	37	95.79	125	41	100.00	18	428	35
		35	94.38	59	35	96.02	139	40	99.43	17	152	15
		37	97.46	47	37	97.46	120	41	99.49	15	778	100
10	20	36.33	95.88	55.20	36.33	96.43	127.80	40.67	99.64	16.67	452.67	49.88
		19	96.33	20	19	96.33	50	20	98.53	21	2400	95
		20	90.45	25	20	93.87	64	21	100.00	19	368	17
		18	95.43	24	18	95.43	57	18	97.62	23	32300	922
20	10	19.00	94.07	23.20	19.00	95.21	56.88	19.67	98.72	21.00	11689.33	344.64
		30	89.82	50	29	91.66	128	31	100.00	19	7100	484
		25	94.80	39	25	94.80	93	31	100.00	20	71923	2033
		19	85.38	48	21	87.77	133	27	100.00	22	4500	123
20	15	24.67	90.00	45.60	25.00	91.41	118.00	29.66	100.00	20.33	27841.00	880.06
	*	33	91.60	60	33	91.60	143	41	96.95	23	375200	18000
		31	95.54	58	31	95.54	126	40	99.52	23	2400	117
	*	37	93.88	57	37	94.40	135	41	98.40	18	2100	111
20	20	33.67	93.67	58.48	33.67	93.85	134.48	40.67	98.29	21.33	126566.67	6075.71

Table 6 Symmetric Networks with Highways: 20 O-D Pairs

		SR 2			SR 5			PD			CPLEX	
$T^{\max}$	% T	# T	%OPT	CPU	# T	%OPT	CPU	# T	%OPT	CPU	# Nodes	CPU
		19	95.75	36	19	95.75	89	20	99.17	33	14800	2291
		20	94.53	37	20	96.26	95	19	100.00	31	3600	265
		20	96.59	32	20	96.59	90	21	99.40	45	4500	755
10	10	19.67	95.63	35.28	19.67	96.20	91.44	20.00	99.52	36.33	7633.33	1103.46
		30	97.43	48	30	97.43	126	30	99.46	38	12114	3149
	*	28	84.91	99	28	87.87	251	29	92.00	45	55200	18000
	*	31	88.59	68	31	88.59	163	30	91.54	58	2560	13908
10	15	29.67	90.31	71.92	29.67	91.30	180.16	29.67	97.66	33.79	30971.33	11685.66
	*	36	89.34	75	38	90.66	205	39	95.45	32	32000	18000
		38	95.69	79	38	95.69	201	40	99.11	40	6200	2914
	*	33	91.29	79	35	92.29	229	38	97.24	41	23000	18000
10	20	35.67	91.77	77.68	37.00	92.88	211.84	39.33	97.32	39.37	20400.00	12971.33

the smallest problem (10 O-D pairs and $T^{\max} = 10$), the proposed heuristics are much faster than the exact MIP algorithm. It has been observed that the CPU time required by CPLEX increases with the percentage of toll arcs, the number of O-D pairs and the value of $T^{\max}$. Since the exact resolution was too costly for symmetric and asymmetric networks with 20 O-D pairs and $T^{\max} = 20$, the corresponding results are not reported. While the computing time of the heuristics also increases with the number of toll arcs as well as the number of O-D pairs, this increase is more modest for the primal-dual heuristic than for the arc-sequential method. Indeed, the computing time of the "arc-sequential" heuristic exceeds that of the primal-dual method as soon as the number of arc permutations is larger than two.

While the primal-dual heuristic produces high-quality solutions quite rapidly and consistently, the situation is more contrasted with the "arcs sequential" method for which a deviation from optimality as large as 12% has been observed for both Type I and Type II problems. Even though the variation decreases with the number of arc permutations considered, it remains significant for a value as high as 5 permutations.

5. Conclusion

Tarification problems, which are pervasive in decision making, naturally lend themselves to a bilevel programming formulation with a specific structure. In the current work, we showed that this structure is amenable to numerical algorithms that can solve to near-optimality large instances of this problem within reasonable computing times. In particular, we developed a primal-dual heuristic procedure based on concepts that can be extended to more general models involving, for instance, congestion at the lower level. This research avenue will be explored in the near future.

References

Arnott, R., A. de Palma, R. Lindsey. 1990. Departure time and route choice for the morning commute. *Transportation Res. B* **24** 209–228.

——, ——, ——. 1994. The welfare effects of congestion tolls with handerogeneous commuters. *J. Transport Econom. Policy* **28** 139–161.

Beckmann, M. J. 1965. On optimal tolls for highway tunnels and bridges. L. Edie, R. Herman, R. Rothery, eds. *Vehicular Traffic Science.* Elsevier, New York. 331–341.

Brotcorne, L. 1998. Approches opérationnelles et stratégiques des problèmes de trafic routier. Ph.D. thesis, Université Libre de Bruxelles, Brussels, Belgium.

——, M. Labbé, P. Marcotte, G. Savard. 2000. A bilevel model for toll optimization: A freight tariff-setting problem. *Transportation Sci.* **34** 289–302.

CPLEX Optimization Inc. 1993. Using the CPLEX Callable Library and CPLEX Mixed Integer Library.

Chopper M. L., W. Oates. 1992. Environmental economics: A survey. *J. Econom. Literature* **30** 675–740.

Frank, M., P. Wolfe. 1956. An algorithm for quadratic programming. *Naval Res. Logist. Quart.* 3 **1** 95–110.

Gendreau, M., P. Marcotte, G. Savard. 1996. An hybrid tabu ascent algorithm for the linear bilevel programming problem. *J. Global Optim.* **8** 217–233.

Goldberg, A. V., R. E. Tarjan. 1990. Finding minimum-cost circulation by successive approximation. *Math. Oper. Res.* **15** 430–466.

Hearn, D. W., M. V. Ramana. 1998. Solving congestion toll pricing models. Patrice Marcotte, Sang Nguyen, eds. *Equilibrium and Advanced Transportation Modelling.* Kluwer, Dordrecht, The Netherlands, 109–123.

Labbé, M., P. Marcotte, G. Savard. 1998. A bilevel model of taxation and its application to optimal highway pricing. *Management Sci.* **44** 1608–1622.

——, ——, ——. 2000. On a class of bilevel programs. G. Di Pillo and F. Giannessi, eds. *Nonlinear Optimization and Related Topics.* Kluwer Academic Publishers, 183–206.

Larsson, T., M. Patriksson. 1998. Traffic management through link tolls—an approach utilizing side constrained traffic equilibrium models. Patrice Marcotte, Sang Nguyen, eds. *Equilibrium and Advanced Transportation Modelling.* Kluwer, Dordrecht, The Netherlands, 125–151.

Mcdonald, J. F. 1995. Urban highway congestion, an analysis of second-best tolls. *Transportation* **22** 353–369.

Mekky, A. 1994. Toll revenue and traffic study of highway 407 in Toronto. *Transportation Record* **1498**, 5–15.

Morrison, S. A. 1986. A survey of road pricing. *Transportation Res. A* **20** 87–97.

Verhoef, E. 1996. Economic efficiency and social feasibility in the regulation of road transport externalities. Ph.D. Thesis, University of Amsterdam, Amsterdam, The Netherlands.

——, P. Nijkamp, P. Rietveld. 1995. Second-best regulation of road transport externalities. *J. Transport Econom. Policy* **29** 147–167.

Viton, P. A. 1995. Private roads. *J. Urban Econom.* **37** 260–289.

Yan, H., W. H. K. Lam. 1996. Optimal road tolls under conditions of queueing and congestion. *Transportation Res. A* **30** 319–332.

Received: May 2000; revisions received: October 2000; accepted: January 2001.