This article was downloaded by: [144.214.88.242] On: 18 September 2017, At: 20:06
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA

INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Tariff Optimization in Networks

Mustapha Bouhtou, Stan van Hoesel, Anton F. van der Kraaij, Jean-Luc Lutton,

To cite this article:
Mustapha Bouhtou, Stan van Hoesel, Anton F. van der Kraaij, Jean-Luc Lutton, (2007) Tariff Optimization in Networks. INFORMS Journal on Computing 19(3):458-469. https://doi.org/10.1287/ijoc.1060.0177

Full terms and conditions of use: http://pubsonline.informs.org/page/terms-and-conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2007, INFORMS

Please scroll down for article—it is on subsequent pages

INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

INFORMS Journal on Computing
Vol. 19, No. 3, Summer 2007, pp. 458–469
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1903 | 0458

DOI 10.1287/ijoc.1060.0177
© 2007 INFORMS

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

Tariff Optimization in Networks

Mustapha Bouhtou
France Télécom R&D, F-92131 Issy-Les-Moulineaux, France,
mustapha.bouhtou@francetelecom.com

Stan van Hoesel
Department of Quantitative Economics, University Maastricht, 6200 MD Maastricht, The Netherlands,
s.vanhoesel@ke.unimaas.nl

Anton F. van der Kraaij
PriceWaterhouseCoopers Advisory N.V., 3584 BA, Utrecht, The Netherlands,
anton.van.der.kraaij@nl.pwc.com

Jean-Luc Lutton
France Télécom R&D, F-92131 Issy-Les-Moulineaux, France,
jeanluc.lutton@rd.francetelecom.com

We consider the problem of determining a set of optimal tariffs for an agent in the network, who owns a subset of all the arcs, and who receives revenue by setting the tariffs on the arcs he owns. Multiple rational clients are active in the network, who route their demands on the least expensive paths from source to destination. The cost of a path is determined by fixed costs and tariffs on the arcs of the path. We introduce a remodeling of the network, using shortest paths. We develop three algorithms based on this shortest-path graph model: a combinatorial branch-and-bound algorithm, a path-oriented mixed integer program, and a known-arc-oriented mixed integer program. Combined with reduction methods, this remodeling enables us to solve the problem to optimality, for quite large instances. We provide computational results for the methods developed and compare them with the results of the arc-oriented mixed integer programming formulation of the problem, applied to the original network.

Key words: networks-graphs, multicommodity; programming, integer; algorithms, branch-and-bound
History: Accepted by William J. Cook, Area Editor for Design and Analysis of Algorithms; received May 2003; revised December 2004, August 2005; accepted January 2006. Published online in *Articles in Advance* July 20, 2007.

1. Introduction

The tariff-setting problem in a network involves two noncooperative groups, tariff-setting agents and tariff-following clients. Each arc in the network is owned by (at most) one agent. Being the owner of an arc, an agent can set the price for renting capacity on the arc freely, in order to maximize his revenues. The clients wish to route a certain demand for flow capacity on a path connecting two vertices (a commodity). A selected route can involve connections belonging to different agents. Clearly, each client will select a route with minimum cost to satisfy the demand for their commodity. This problem is essentially a game-theoretic problem, see Fisk (1986), where the agents decide on their prices based on (partial) knowledge of the prices of competing agents. We restrict the problem to a single agent who knows the tariffs of his competitors and intends to charge revenue-maximizing tariffs on the subset of the network arcs he owns. This agent is generally referred to as the *leader*, while the clients are referred to as *followers*.

The tarification problem has a wide range of applications such as tariff setting in freight transportation and highway toll optimization; see Brotcorne et al. (2000) and Labbé et al. (1998), respectively. Recently, the problem has become of interest in the telecommunications market. Here, many operators are active, who rent their capacity on connections to customers. In general, a single operator does not own complete connections between all pairs of vertices in a network. Thus, a customer has to rent connections from different operators to establish a complete line.

An interesting variant of the tarification problem occurs in highway traffic routing (Jahn et al. 1999, Roughgarden and Tardos 2000), and IP (Internet etc.) traffic routing (Fortz and Thorup 2000). Here a single operator is active, and tariffs are introduced with the objective to divert traffic from areas of congestion.

A linear bilevel model is a very natural and elegant formulation of the problem. The upper level relates to the leader and fixes the tariffs, the lower level belongs to the clients who, given the tariffs set by the leader, can determine their best (shortest) path. This model was proposed first in Labbé et al. (1998). For the single-commodity case, primal-dual heuristics based on a penalization of the lower-level objective

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

function are proposed by Brotcorne et al. (2000). These heuristics were extended to multiple commodities in Brotcorne et al. (2001). Furthermore, in Labbé et al. (1998), the problem was reformulated as a mixed integer program. This program solves up to medium-sized instances within reasonable time.

The solution methods we propose here are all based on a remodeling of the network that reveals the essential structure of the problem and generally reduces its size. Concretely, we consider shortest paths that are represented by arcs in the so called *shortest-path graph model*, which is equivalent to the original problem. This model is further reduced with dedicated processing techniques. We exploit the model in three solution methods: a combinatorial branch-and-bound algorithm, a path-oriented mixed integer program, and an arc-oriented mixed integer program (Labbé et al. 1998). All three methods allow us to solve fairly large instances of the problem quickly.

In Section 2 we define the tarification problem, address its complexity and introduce the necessary notation. Then, in Section 3, the remodeling of the network is described. In Section 4, model-specific graph-reduction methods are given. In Section 5, these concepts are used in a branch-and-bound algorithm, while Section 6 gives a path-and-arc-oriented mixed integer program. We present some computational results in Section 7, where we compare the three methods developed to the arc-oriented mixed integer program of Labbé et al. (1998) on the original network, to illustrate the efficiency of our methods and remodeling technique.

2. The Tariff-Setting Problem

Consider a network represented by a directed graph $G = (N, A)$ with nodes N and arcs A. The arc set A is partitioned into two sets: the set of *tariff* arcs T, and the set of *fixed* cost arcs F. The tariff arcs belong to the leader in the network and incur a revenue-generating toll for routing a unit of a client's demand. The fixed arcs are owned by other agents in the network, whose tariffs are known a priori and hence can be viewed as fixed per unit costs. The tariffs on the arcs of T are determined such that the total revenue of the leader is maximized. Both the tariffs and the fixed costs are assumed to be nonnegative. The clients on the network route their demands from source to destination according to the shortest path with respect to total cost, where the total cost of a path is defined as the sum of all the tariffs and fixed costs on the arcs of the path. Whenever the client has a choice among multiple shortest paths with the same total cost but with different revenues for the leader, we suppose the client takes the shortest path that is most profitable to the leader. This tie-breaking rule is justified by noting that we can always decrease by ϵ the tariff on one of the arcs of the path with highest revenue for the leader.

Labbé et al. (1998) studied the tarification problem on a transportation network and have shown that the related decision problem is $\mathcal{NP}$ -complete for a single commodity when lower bounds on the tariffs are given. Some variants of the tarification problem have been shown to be solvable by polynomial-time algorithms. Among these is the class of problems with only a single tariff arc and multiple clients, and the class of problems where the leader is dealing with multiple tariff arcs and a single client for which the path taken at optimality is known a priori; see Labbé et al. (1998).

We denote by c_a the cost of routing a unit demand on a fixed cost arc $a \in F$, and by t_a, to be determined by the leader, the cost of routing a unit demand on a tariff arc $a \in T$. The commodities are denoted by the set K. The demand of a commodity $k \in K$ is d_k. The source and destination of commodity k are the pair (s_k, t_k). The set of paths that connect s_k and t_k is P_k. For each path $p \in P_k$ we introduce T_p for its set of tariff arcs, and F_p for its set of fixed cost arcs. Furthermore, the cost of routing a unit demand on p is denoted by its length $l_p(t)$, which is a function of the tariffs t. The length of p is determined by the sum of the costs on the fixed arcs of the path, denoted by c_p, and the costs on the tariff arcs of the path, represented by $\pi_p(t)$. Thus, $l_p(t) = c_p + \pi_p(t)$, where $c_p = \sum_{a\in F_p} c_a$, and $\pi_p(t) = \sum_{a\in T_p} t_a$. Note that our model implicitly incorporates arcs with both fixed and tariff costs since we can divide such an arc a with cost c_a and tariff t_a into two arcs: a fixed arc with cost c_a and a tariff arc with tariff t_a. To ensure that the problem is bounded, we assume that for each commodity there exists a path from source to destination that uses only fixed cost arcs. Otherwise, the leader can set arbitrarily high the tariffs on the arcs in T. The following formulation of the tarification problem is a direct implementation of the above description:

$$\begin{aligned} \max_{t\ge 0} \; & \sum_{k\in K} d_k \pi_{p_k^*}(t) \\ \text{s.t.} \; & p_k^* = \arg\min_{p\in P_k} l_p(t) \quad \forall\, k \in K. \end{aligned} \tag{1}$$

Formulation (1) is a bilevel problem where at the upper level the leader strives to maximize his revenue, while at the lower level the clients (followers) seek to minimize the cost of routing their demands. Both objective functions are linear and hence this is a linear-linear bilevel program. The general linear-linear bilevel program has been shown to be $\mathcal{NP}$-hard by Jeroslow (1985). For a reference on bilevel programming, see Vicente and Calamai (1994) who have compiled an annotated bibliography on this subject

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

containing more than 100 references. Note that the bilevel program (1) is not polynomial in its input data, since the set of paths for each client $k \in K$ may be exponential in the size of the problem instance. Labbé et al. (1998) considered the following arc-oriented bilevel model.

$$\max_{t\geq 0} \sum_{k\in K}\sum_{a\in T} t_a y_a^k$$

$$\min_{y^k\geq 0} \sum_{k\in K}\left\{\sum_{a\in T} t_a y_a^k + \sum_{a\in F} c_a y_a^k\right\} \tag{2}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} y_{ij}^k - \sum_{j:(i,j)\in A} y_{ji}^k = \begin{cases} d_k & \text{if } i=s_k \\ -d_k & \text{if } i=t_k \quad \forall k\in K \\ 0 & \text{otherwise.} \end{cases}$$

In this bilevel model the y_a^k $(a \in A)$ represent the flow on the arcs for each commodity k. This formulation has been used in Brotcorne et al. (2000, 2001) for development of primal-dual heuristics. Labbé et al. (1998) have reformulated the bilevel model (2) as a mixed integer program. To this end the lower problem is separated into $|K|$ problems, one for each commodity. This is done by introducing for each commodity its own tariff t^k. Next, the LP for each commodity is restricted to its optimal solution(s) by adding the variables and constraints of the dual LP, and a constraint that sets the primal and dual objective at the same value. Later, constraints are added to force equality of tariffs for all commodities. We henceforth refer to the single-level mixed integer program of Labbé et al. (1998) as AMIP.

3. The Shortest-Path Graph Model

If for given tariffs a client will select the shortest path, say p, between the two end nodes of his commodity, then clearly, the subpaths of p are also shortest paths. Consider two tariff arcs $a_1 = (i_1, j_1)$ and $a_2 = (i_2, j_2)$ that appear consecutively on p. Then the subpath between j_1 and i_2 is a shortest path that contains only fixed arcs. Thus any path p with tariff arcs a_i, $i \in \{1, \ldots, |T_p|\}$, taken by a client can be described as the path

$$p = \{sp_1, a_1, sp_2, a_2, \ldots, sp_{|T_p|}, a_{|T_p|}, sp_{|T_p|+1}\}, \tag{3}$$

where sp_i, $i \in \{1, \ldots, |T_p| + 1\}$ is a shortest subpath using only fixed cost arcs to or from a tariff arc on the path. Since such paths can be computed using the original data, we can construct a new graph model in which this is actually done: the shortest-path graph model (SPGM). We will define this graph model for a single customer first. Consider the original graph $G = (N, A)$ with the tariff arcs in $T \subseteq A$. For a client with demand d from s to t, we define the graph $G^* = (N^*, A^*)$ and the tariff arcs $T^* \subseteq A^*$. In this graph, the tariff arcs are copied from G as a matching. So, arcs with a common vertex are separated. Next, we construct the following fixed cost arcs. For two tariff arcs $a_1 = (i_1, j_1)$ and $a_2 = (i_2, j_2)$ we connect j_1 with i_2, if there is a path in G that uses fixed arcs only. Similarly, we connect j_2 with i_1. From the source s we construct arcs to all the tail nodes of the tariff arcs, and from all the head nodes we construct an arc to the destination t, again only if paths exist using only fixed arcs in G. Any fixed arc in A^* has a cost equal to the length of the shortest path between its end vertices in G, using only fixed cost arcs in G. The new network is the SPGM.

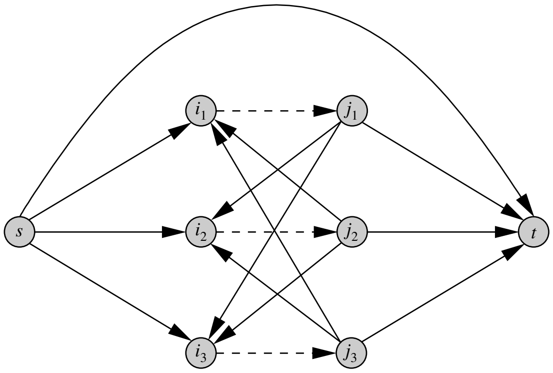


Figure 1 **SPGM for $|T| = 3$**

Example 1. Figure 1 shows the SPGM of any network containing three tariff arcs for a commodity from s to t. The tariff arcs are the (dashed) arcs (i_1, j_1), (i_2, j_2), and (i_3, j_3). All other arcs are representations of the shortest path using only fixed cost arcs between each node. The cost of the arc is the cost of the corresponding shortest path in the original network between the two nodes. If no path exists between two nodes in the original network, the corresponding arc in the SPGM is not present or has infinite cost.

The SPGM can easily be extended to multiple commodities. For each commodity, we create an SPGM. The *inner graph* (consisting of the end vertices of the tariff arcs, and the arcs between them) is equal for all commodities and hence needs to be determined only once. Additional shortest-path calculations are necessary only for the arcs leaving the source or entering the terminal of each commodity. The SPGM is equivalent to the original graph in the sense that both have an optimal solution of the same value: If a path exists in the original graph, then there exists a path in the SPGM that is at least as good. Alternately, if a path exists in the SPGM, then a path with the same cost exists in the original graph.

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

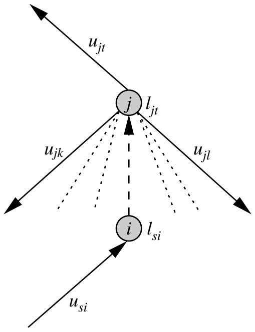


Figure 2 **Detailed View of Tariff Arc**

4. Reduction Methods

We can decrease the size of the SPGM considerably by using reduction methods to limit the amount of potentially shortest paths for a commodity. First, we describe some techniques with which we can remove arcs from the network. Second, we describe a dominance relationship for paths that explicitly removes paths from the set of potentially optimal paths. The aim is to end with a manageable set of potentially optimal paths.

4.1. Arc Reduction

Let u_{ij} be the cost of the shortest path using only fixed arcs from node i to node j in G, i.e., u_{ij} is the length of the arc (i, j) in G^*. Let l_{ij} denote the cost of the shortest path from i to j in G, possibly using tariff arcs, when the tariffs are set to zero. We restrict ourselves to a single commodity, where node s represents the source node and node t the destination node. In Figure 2 we depict the values defined here: the u_{ij} are arc values, and the l_{ij} are node values. Note that l_{ij} is a lower bound for the cost of a path from i to j, and u_{ij} is an upper bound for the cost of a path from i to j taken by the client.

Proposition 1. *If $l_{jt} = u_{jt}$, then any optimal path from s to t using node j can use arc (j, t); all other arcs leaving j can be removed.*

Proof. A lower bound on the cost of a path, possibly using tariff arcs, from j to t is l_{jt}. An upper bound is u_{jt}. If $l_{jt} = u_{jt}$, there is no room for taxation on any path from j to t. Thus, (j, t) is an optimal choice. □

Proposition 2. *If $l_{si} = u_{si}$, then any optimal path from s to t using node i can use arc (s, i); all other arcs entering i can be removed.*

Proof. Analogous to the proof of Proposition 1. □

Proposition 3. *Consider two tariff arcs (i_1, j_1) and (i_2, j_2). If $u_{j_1t} \le u_{j_1i_2} + l_{j_2t}$, we can delete arc (j_1, i_2).*

Proof. The lower bound on the cost of a path from j_1 to t taking the arc (j_1, i_2) is equal to $u_{j_1i_2} + l_{j_2t}$. An upper bound on the cost of a shortest path is u_{j_1t}. Hence, there is no room for taxation on a path from j_1 to t using the arc (j_1, i_2). Thus, (j_1, t) is an optimal choice. □

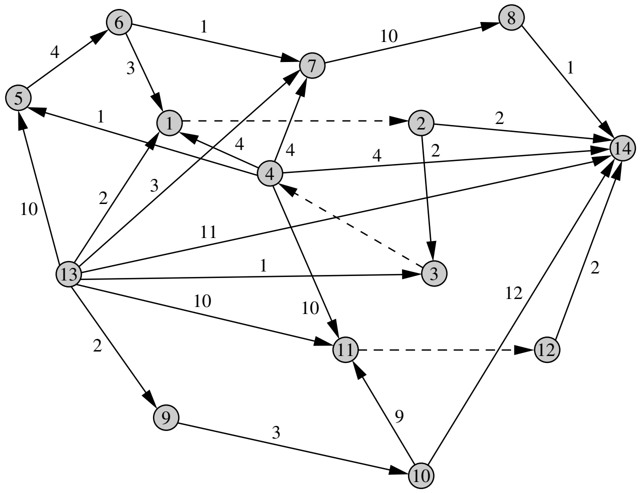


Figure 3 **A Small Network with $|T| = 3$ (Example 2)**

Proposition 4. *Consider two tariff arcs (i_1, j_1) and (i_2, j_2). If $u_{si_1} \le u_{j_2i_1} + l_{si_2}$, we can delete arc (j_2, i_1).*

Proof. Analogous to the proof of Proposition 3. □

Proposition 5. *If $u_{st} \le l_{si_1} + l_{j_1t}$, we can delete the tariff arc (i_1, j_1).*

Proof. The lower bound on the cost of a path going through the tariff arc (i_1, j_1) is $l_{si_1} + l_{j_1t}$. Hence, there is no room for taxation on the tariff arc (i_1, j_1), and thus (s, t) is always at least as good. □

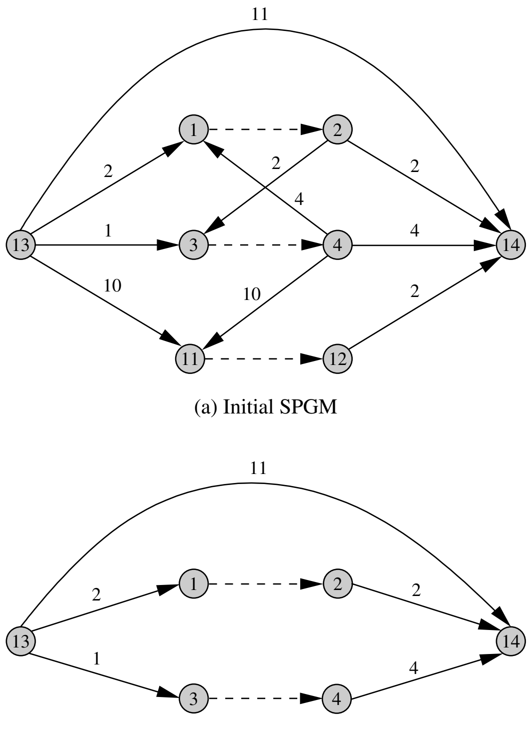
Proposition 6. *Consider two tariff arcs (i_1, j_1) and (i_2, j_2). If $u_{st} \le l_{si_1} + u_{j_1i_2} + l_{j_2t}$, we can delete the arc (j_1, i_2).*

Proof. The upper bound on the cost of a shortest path from s to t is u_{st}. A lower bound on the cost of a path using the tariff arc (i_1, j_1) and going to the tariff arc (i_2, j_2) is $l_{si_1} + u_{j_1i_2} + l_{j_2t}$. Hence, there is no room for taxation on a path from s to t using the arc (j_1, i_2). Thus, this arc need not be used in an optimal solution. □

Example 2. Consider the network in Figure 3, which has 14 nodes, 21 fixed cost arcs and three tariff arcs. The tariff arcs are $(1, 2)$, $(3, 4)$, and $(11, 12)$, represented by dashed arcs in Figure 3. The solid arcs in the network represent the fixed cost arcs. The leader on the network is dealing with one client who has a unit demand from node 13 to node 14.

The SPGM, for each client, obtained after application of the reduction methods, is denoted as the final SPGM. Figure 4(a) shows the initial SPGM for the network of Example 2. The solid arcs in the SPGM are representations of the shortest path using no tariff arcs between each node in the original network given in Figure 3. The cost of the arc is the cost of the

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.



(a) Initial SPGM

(b) Final SPGM

Figure 4 SPGM for Example 2

corresponding shortest path in the original network between the two nodes. If no path exists between two nodes in the original network, the corresponding arc in the SPGM is not present.

To arrive at the final SPGM, we need to apply the reduction methods described in Section 4 to the initial SPGM. Looking at Figure 4(a), we see we can delete arcs $(4, 1)$, $(2, 3)$, and $(4, 11)$ of the initial SPGM by applying, e.g., Proposition 3 or 4. Furthermore, Proposition 5 allows us to delete the tariff arc $(11, 12)$ from the graph.

The final graph is shown in Figure 4(b). The optimal solution is to set the tariff on the arc $(1, 2)$ to 7, while setting all the other tariffs to a suitable large value, e.g. 11, yielding a revenue of 7 to the leader.

4.2. Path Reduction

By applying the propositions described in Section 4.1 and finding all paths in the graph, we obtain a reduced set of paths that remains relevant for the commodity. Hopefully, this set of paths is small. It is, however, possible to reduce the size of this set in some instances even more by eliminating dominated paths. The notion of path dominance is given by the following definition.

Definition 1. If we can replace in all feasible solutions the path q by the path p without violating the feasibility constraints or decreasing the value of the objective function, then path p dominates path q.

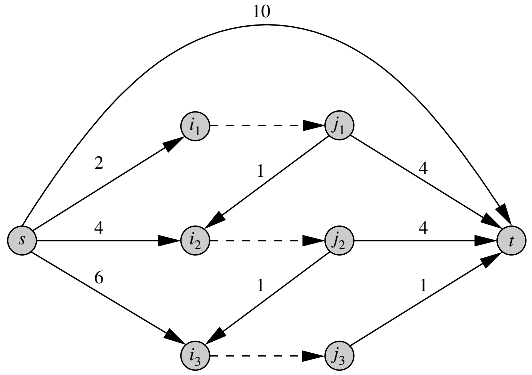


Figure 5 Dominated-Paths Example

The following proposition allows us to eliminate dominated paths. Recall that T_p is the set of tariff arcs from path p, and that c_p is the total cost of the fixed arcs from p.

Proposition 7. *Consider paths p and q. If $T_p \subseteq T_q$ and $c_q \ge c_p$, then path p dominates path q for all tariff values.*

Proof. Suppose that path q is the shortest path taken by the client. Then

$$\begin{gathered} c_q + \sum_{a\in T_q} t_a \le c_p + \sum_{a\in T_p} t_a \quad \text{or equivalently} \\ c_q + \sum_{a\in T_q\setminus T_p} t_a \le c_p. \end{gathered} \tag{4}$$

Since $c_q \ge c_p$ and $t_a \ge 0$ for all tariff arcs $a \in T$, (4) holds only when $c_q = c_p$ and $t_a = 0$ for all $a \in T_q \setminus T_p$. Hence, we can replace path q by path p without violating the feasibility constraints or changing the value of the objective function since path p has the same revenue for the leader and the same cost for the client as path q. Furthermore, (4) shows that whenever $c_q > c_p$, path q can never be the path taken by a client in a feasible solution, since it will always be at least as expensive as path p. □

Example 3. An instance where this dominance of paths occurs is in Figure 5. The tariff arcs are (i_1, j_1), (i_2, j_2), and (i_3, j_3). The leader is dealing with one client who wants to route his demand from node s to node t. For this graph, the path $\{s, i_1, j_1, i_2, j_2, t\}$ is dominated by the path $\{s, i_1, j_1, t\}$.

In the SPGM, the maximum number of paths for a client $k \in K$ is bounded by $e|T|!$, the number of ordered subsets of the tariff arcs. The number of *undominated* paths in a network is bounded by the number of possible subsets of T, i.e., by $2^{|T|}$: If two paths p and q have an identical set of tariff arcs, then the undominated path is the path with smallest fixed cost. Example 4 shows that this number of undominated paths can be reached.

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

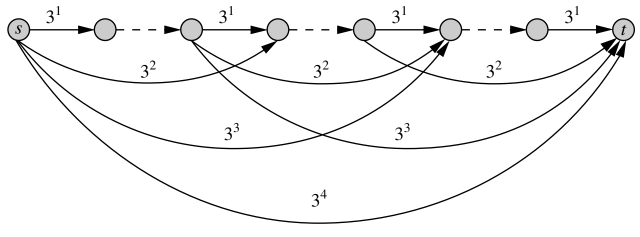


Figure 6 Network for Example 4, for $|T| = 3$

Example 4. Consider a network with $|T|$ tariff arcs, denoted by $a_1, \ldots, a_{|T|}$. A unit demand from s to t is to be routed. The fixed cost arcs are defined as follows. The cost of the fixed cost arcs from the source s to the tarif arc a_i, $i \in \{1, \ldots, |T|\}$ is 3^i. The cost of the fixed cost arcs from the tariff arc a_i, $i \in \{1, \ldots, |T|\}$, to the target t is $3^{|T|+1-i}$. Furthermore, a fixed cost arc with cost 3^{j-i} connects a tariff arc a_i to a_j, where $i, j \in \{1, \ldots, |T|\}$ and $i < j$. Finally, the cost of the fixed cost arc (s, t) is set to $3^{|T|+1}$. See Figure 6 for a network with 3 tariff arcs.

As stated earlier, any path p with tariff arcs a_i, $i \in \{1, \ldots, |T_p|\}$, taken by a client can be described as the path

$$p = \{sp_1, a_1, sp_2, a_2, \ldots, sp_{|T_p|}, a_{|T_p|}, sp_{|T_p|+1}\}. \tag{5}$$

For the network in Example 4, any subpath of p of the form $\{sp_m, a_m, sp_{m+1}\}$, $m \in |T_p|$, has fixed cost of the form $3^i + 3^j$, where $i, j \in \{1, \ldots, |T|\}$.

Now let $1 \leq i < j < k \leq |T|$. The direct arc from a_i to a_k has a fixed cost of 3^{k-i}. The path using the direct arc from a_i to a_j, then tariff arc a_j and then the direct arc from a_j to a_k, has fixed cost $3^{j-i} + 3^{k-j}$, which is smaller than 3^{k-i}. Thus, replacing a fixed cost arc with two fixed cost arcs and a tariff arc reduces the fixed costs (also for arcs starting at s or ending at t). Thus, for any paths p and q with $T_p \subseteq T_q$ the fixed cost of p is higher than the fixed cost of q. So, neither p nor q dominates the other. Concluding, for each subset of the tariff arcs, there is a path using them that is not dominated.

5. Branch-and-Bound Algorithm

In this section we describe a two-phase branch-and-bound algorithm for our tariff-setting problem that uses the SPGM from Section 3 and the reduction methods from Section 4. First, we create for each client an SPGM and apply to it the reduction methods of Section 4. For each client we thus find the relevant shortest paths. Then, we solve the problem to optimality by a classical branch-and-bound method.

As stated in Section 2, denote the clients by the set K and the set of paths a client $k \in K$ can take by P_k. The reduction methods applied to the SPGM allow us to determine the set of relevant paths for each commodity. We suppose that P_k is reduced to contain the relevant paths only. Recall furthermore from Section 2 the linear function $l_p(t) = c_p + \pi_p(t)$ denoting the cost of a path p as a function of all tariff values. Let p_k^l be the path for client $k \in K$ with the smallest fixed cost, i.e., $p_k^l = \arg\min_{p \in P_k} c_p$, and let p_k^u be the path with the largest fixed cost, i.e., $p_k^u = \arg\max_{p \in P_k} c_p$. Note that p_k^u has no revenues for the leader, since it denotes the path with fixed cost arcs only. Clearly, $c_{p_k^u} - c_{p_k^l}$ is an upper bound on the revenues that can be generated from client k. This is an important measure in the branch-and-bound algorithm.

5.1. Branching Rules

In each node of the branch-and-bound tree, we select a client and create a branch for each of the relevant paths of the client. The selection method of the clients is based on the upper bound $c_{p_k^u} - c_{p_k^l}$ on the revenue generated by each client for the leader: the client for which this upper bound is highest, is selected first. Next, we walk through the search tree in a depth-first manner.

5.2. Node Processing

Due to our branching rules, in each node of the branch-and-bound tree for some clients the path taken in the solution is fixed, whereas for other clients this choice is still to be made. In each node, we denote by the set $K_f \subseteq K$ the set of clients for which we have fixed the path taken in the solution. Suppose that for any client $k \in K_f$, we have fixed the path p_k^*. We can find the optimal, revenue-maximizing tariffs for the problem restricted to the clients in K_f by solving the following linear problem:

$$\begin{aligned} \max \quad & \sum_{k \in K_f} d_k \pi_{p_k^*}(t) \\ \text{s.t.} \quad & l_p(t) \geq l_{p_k^*}(t) \quad \forall\, k \in K_f,\ \forall\, p \in P_k. \\ & t_a \geq 0 \quad \forall\, a \in T. \end{aligned} \tag{6}$$

The linear program (6) forces the path p_k^* to be the shortest path in P_k, while maximizing the leader's revenue.

We generate lower bounds in each node of the branch-and-bound tree by computing a feasible solution. Such a feasible solution can be created by solving (6) and then adding the revenues from the tariffs of (6) for the clients in $K \backslash K_f$. However, a better lower bound may be generated by fixing for all clients the path taken in the solution. For each client $k \in K_f$ we already know which path is taken in the solution and we denote it by p_k^*. For the clients $k \in K \backslash K_f$, we fix the path to the one for which the possible revenue is

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

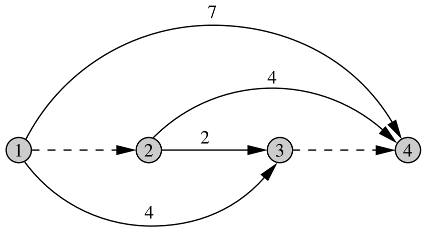


Figure 7 **Upper Bound on Revenue or Cost Is Not Reached**

highest. This is the path with smallest fixed cost, i.e., the path p_k^l:

$$\begin{aligned} \max\ & \sum_{k\in K_f} d_k \pi_{p_k^*}(t) + \sum_{k\in K\setminus K_f} d_k \pi_{p_k^l}(t) \\ \text{s.t.}\ & l_p(t) \ge l_{p_k^*}(t) \quad \forall k \in K_f,\ \forall p \in P_k. \\ & l_p(t) \ge l_{p_k^l}(t) \quad \forall k \in K\setminus K_f,\ \forall p \in P_k \\ & t_a \ge 0 \qquad \forall a \in T. \end{aligned} \tag{7}$$

If this linear program is infeasible no sensible lower bound is generated.

For a client $k \in K$ an upper bound for the revenue generated by that client is $c_{p_k^u} - c_{p_k^l}$. As is shown by Labbé et al. (1998), this upper bound is not necessarily reached. Even the upper bound on the cost of the path, $c_{p_k^u}$, is not tight. This is shown by the example in Figure 7, where the tariff arcs are dashed. For a single client with a unit demand from node 1 to 4, the optimal tarification scheme is to set the tariffs on the tariff arcs to $t_1 = t_2 = 2$. Hence, the cost of the path taken by the client is 6, yielding a revenue of 4 for the leader. The upper bound on the cost of the path is, however, 7, while the upper bound on the revenue is $7 - 2 = 5$.

In each node of the branch-and-bound tree, let Π^* be the optimal value of (6), i.e., the optimal revenue obtained from the commodities with fixed paths. The remaining customers can contribute no more per unit than $\sum_{k\in K\setminus K_f}\{c_{p_k^u} - c_{p_k^l}\}$. An upper bound on the total revenue for the leader in a node is thus

$$\Pi^* + \sum_{k\in K\setminus K_f} d_k\{c_{p_k^u} - c_{p_k^l}\}. \tag{8}$$

This upper bound can be tightened by using the information we can retrieve from the constraints of the linear program (6). These constraints must be satisfied by any feasible solution, for all clients and their relevant paths. Thus, the constraints of (6) also hold for the commodities for which a path is not fixed yet, as can be seen from the constraints in (7). Each of the constraints in (6) involves two paths, say p and q, and states that the total cost of one path (say p) is at most the total cost of the other path (say q): $c_p(t) \le c_q(t)$. Now, let T_p be the set of tariff arcs on path p. The constraint is, in more detail,

$$c_p + \sum_{a\in T_p} t_a \le c_q + \sum_{a\in T_q} t_a \tag{9}$$

or equivalently,

$$\sum_{a\in T_p\setminus T_q} t_a - \sum_{a\in T_q\setminus T_p} t_a \le c_q - c_p. \tag{10}$$

If $\sum_{a\in T_q\setminus T_p} t_a$ can be bounded from above, then we have an upper bound on the sum of a set of tariffs. For instance, if $T_q\setminus T_p$ is empty, then we get $c_q - c_p$ as an upper bound. Thus, this way we find a number of constraints of the type

$$\sum_{a\in T_i\subseteq T} t_a \le b_i \quad (i \in \Phi), \tag{11}$$

where we define $b_i = c_q - c_p$ for the index set Φ.

Next, consider a path $p \in P_k$ from commodity $k \in K\setminus K_f$, i.e., a commodity for which no path has been fixed yet. An upper bound on the total cost per unit demand was previously given by the path using only fixed arcs, i.e., the path $c_{p_k^u}$. With the constraints of type (11) obtained from the commodities in K_f and by defining $b_i = 0$ for $T_i = \varnothing$, the upper bound on the total cost of a path $p \in P_k$, denoted by $\gamma(p)$, is

$$\gamma(p) = \min_{i\in\Phi:\, T_p\subseteq T_i} (c_p + b_i) \quad \forall k \in K\setminus K_f,\ \forall p \in P_k. \tag{12}$$

For each commodity $k \in K$, the total cost of each path taken in a solution can be at most the minimum cost of all other possible paths for that commodity. Hence, an upper-bound on the total cost of *any* path in P_k is

$$\min_{p\in P_k} \gamma(p). \tag{13}$$

For each commodity $k \in K$, the upper-bound on the revenue is still highest for the path with smallest fixed cost, i.e., the path p_k^l. We can thus improve the upper-bound computation in (8) by using the upper-bound

$$\Pi^* + \sum_{k\in K\setminus K_f} d_k \left\{ \min_{p\in P_k} \gamma(p) - c_{p_k^l} \right\}. \tag{14}$$

We will use (14) for our branch-and-bound algorithm.

6. Mixed Integer Programming Formulations

In the path-based branch-and-bound algorithm of the previous section, an LP solver is called in each subproblem. In this section, we introduce a mixed integer programming formulation using path variables, where the paths are generated from the SPGM. This formulation does not incorporate all features of the branch-and-bound algorithm of the previous section, but it can be handed to an ILP-solver directly (we use CPLEX 7.5). We rewrite the bilevel formulation (1) to the following single-level program, using path variables. To this end, we introduce the binary variable h_p,

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

which indicates whether a given path is taken in the optimal solution.

$$
\begin{aligned}
\max\ & \sum_{k\in K}\sum_{p\in P_k} d_k \sum_{a\in T_p} h_p t_a \\
\text{s.t.}\ & \sum_{p\in P_k} h_p = 1 \quad \forall k\in K \\
& l_p(t) \geq \sum_{q\in P_k} h_q l_q(t) \quad \forall k\in K,\ \forall p\in P_k \\
& h_p \in \{0,1\} \quad \forall k\in K,\ \forall p\in P_k \\
& t_a \geq 0 \quad \forall a\in T.
\end{aligned}
\tag{15}
$$

The first constraint in (15) indicates that a client selects only one path. The second constraint ensures that the path taken by the client is indeed the least expensive path available. Note that this formulation is not only nonlinear in its objective function, but also in this constraint. It is however possible to use standard linearization techniques for (15) by introducing the variable r_{pa} for each path $p\in P_k$, and tariff arc $a\in T$. The variable r_{pa} is equal to t_a if and only if client $k\in K$ takes the path $p\in P_k$ going through the tariff arc $a\in T$ and is equal to 0 otherwise. After adding the constraints enforcing this relationship, we can write the following mixed integer programming formulation. We will refer to this formulation as PMIP.

$$
\begin{aligned}
\max\ & \sum_{k\in K}\sum_{p\in P_k} d_k \left(\sum_{a\in T_p} r_{pa}\right) \\
\text{s.t.}\ & \sum_{p\in P_k} h_p = 1 \quad \forall k\in K \\
& l_p \geq \sum_{q\in P_k}\sum_{a\in F_q} h_q c_a + \sum_{q\in P_k}\sum_{a\in T_q} r_{qa} \quad \forall k\in K,\ \forall p\in P_k \\
& l_p = \sum_{a\in F_p} c_a + \sum_{a\in T_p} t_a \quad \forall k\in K,\ \forall p\in P_k \\
& r_{pa} - t_a \leq (1-h_p)M \quad \forall k\in K,\ \forall p\in P_k,\ \forall a\in T_p \\
& r_{pa} - t_a \geq -(1-h_p)M \quad \forall k\in K,\ \forall p\in P_k,\ \forall a\in T_p \\
& r_{pa} \leq h_p M \quad \forall k\in K,\ \forall p\in P_k,\ \forall a\in T \\
& h_p \in \{0,1\} \quad \forall k\in K,\ \forall p\in P_k \\
& t_a \geq 0 \quad \forall a\in T \\
& r_{pa} \geq 0 \quad \forall k\in K,\ \forall p\in P_k,\ \forall a\in T.
\end{aligned}
\tag{16}
$$

Note that for the PMIP, in theory, there is an exponential number of constraints. However, the reformulation of the network into an SPGM, combined with the reduction methods, will allow us to generate the relevant paths for each commodity and thus reduce the necessary variables and constraints to a manageable amount.

The arc-oriented bilevel program (2) is turned into an ILP by applying the ideas of Labbé et al. (1998). The shortest-path problems of the clients are replaced by the shortest-path optimality conditions. This introduces nonlinearity: product of tariff variables (t_a $(a\in T)$) and design variables (x_a^k $(a\in T, k\in K)$). The product is then replaced with separate tariff variables for all tariff arcs and for all commodities. We will refer to the thus obtained formulation as AMIP. If the AMIP formulation is applied to the SPGM, then we use the notation AMIP+. The results of AMIP+ and PMIP, together with the numerical results of the branch-and-bound algorithm developed in this paper, will show the efficiency of the SPGM when compared to AMIP on the original network.

7. Numerical Results

The reduction methods and the branch-and-bound algorithm (PBB) were implemented in C++. CPLEX 7.5 was used to solve the linear programs (6) in each node of the tree. The formulations AMIP, AMIP+, and PMIP were handed directly to CPLEX 7.5, after application of reduction methods for AMIP+, and PMIP. So no further processing was done on subproblems created during the branch-and-bound phase. All computational results were established on an AMD Athlon 2400XP+ with 1 Gb RAM, running Debian GNU/Linux 3.0 with kernel 2.4.18.

The solution methods developed here were tested on three different types of networks: telecom networks, random networks, and grids. For each of the three types of networks we provide statistics on the problem instances and on solution times for all methods.

7.1. Telecom Networks

The first set of instances was provided by France Télécom Research and Development. They represent real-life instances from the international interconnections market in which France Télécom participates. In these instances, the graph represents a telecommunications (backbone) network and the tariff arcs are the interconnections between the different operators. A description of these instances is in Table 1, where, for each instance, we give the number of nodes and

Table 1 **Description and SPGM Statistics of France Télécom Instances**

Data	#Nodes	#Arcs	#Tariff arcs	#Clients	MIN	MAX	AVRG
D1	29	92	6	13	2	3	2.5
D2	29	98	11	13	2	5	3.1
D3	43	176	10	20	2	8	5.5
D4	60	212	9	22	2	15	4.2
D5	60	212	10	8	2	20	5.0
D6	60	212	17	8	2	30	12.0
D7	60	212	21	18	2	9	4.8
D8	49	116	9	23	2	10	4.0
D9	33	116	15	30	2	8	3.1

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

arcs in the network, the number of tariff arcs, and the number of clients. These data determine the size of AMIP. Table 1 also shows some statistics on the number of paths generated for each instance using the SPGM and its reduction methods for both data sets. In this table, the column MIN, resp. MAX, indicates the minimum, resp. maximum, number of paths generated over all commodities. The column AVRG gives the average number of paths generated for each commodity for the whole data set. The product of AVRG and the number of commodities (the total number of paths) gives a good idea of the size of the networks after constructing the SPGM and application of the reduction methods. The actual size of each instance in number of variables and constraints is in Table 2 for all three formulations. The statistics of the solution methods are in Table 3, which shows an overview of the results for all algorithms for the real-life instances. In this table, the column OPT indicates the value of the optimal solution for the given data set. The columns *Time* and *Nodes* indicate the execution time in seconds, and the number of nodes in the B&B tree for each algorithm. For the PBB algorithm, the PMIP, and the AMIP+, the CPU time includes the time needed for generation of the SPGM and all relevant paths.

From Table 2 it can be seen that the path model is generally smaller than AMIP+, and AMIP+ is smaller than AMIP. This applies to integer variables, integer plus continuous variables, and constraints. This behavior is not surprising, given the success of the reduction techniques on these instances. As can be seen in Table 3, the PBB algorithm efficiently uses the few relevant paths generated to find the optimal solution for each data set. When we compare it to AMIP, we see that the execution time of the PBB algorithm is less than or equal to the running time of CPLEX for the AMIP for each data set considered. The difference is especially large for the sets D2, D3, D6, D7, and D8. The most striking difference is for D3, for which the execution time of the AMIP is 113,423 seconds, whereas the PBB algorithm takes only 30 seconds. The PMIP and AMIP+ columns show the efficiency and power of the SPGM. The execution time of the PMIP is 3 seconds or less for all data sets, except the data set D6. For this data set the execution time is, however, still much less than the execution time of the AMIP and less than the time needed for the PBB algorithm to execute. The AMIP+ column shows furthermore that the AMIP formulation can benefit very much from the SPGM: For all data sets, the time needed for the AMIP+ formulation is a little more than the time needed for the PMIP formulation and much less than the time needed for the same formulation on the original network (AMIP). Concerning the performance difference between AMIP and AMIP+: Although when considering the LP relaxation of AMIP on the original graph and on the complete SPGM, the subpaths between the tariff arcs are shortest paths for both methods, but the AMIP+ formulation uses more information than does AMIP. Proposition 5 allows us to delete tariff-arcs in the network for a specific client, essentially setting a lot of tariff arc variables to zero beforehand in the AMIP+ formulation. For AMIP+, this diminishes the number of variables on which to branch. Concluding, PMIP does best, closely followed by AMIP+, which has comparable computation times, but a higher number of nodes in the B&B. Then PBB follows, with slightly higher times, but certainly more nodes. AMIP does considerably worse than the other algorithms.

Table 2 Size Statistics for France Télécom Instances

		PMIP/PBB			AMIP+			AMIP		
Data	OPT	Int vars	Total vars	Cons	Int vars	Total vars	Cons	Int vars	Total vars	Cons
D1	6.6225	32	230	102	29	248	313	78	1,657	1,898
D2	7.7928	40	491	152	55	443	555	143	1,805	2,236
D3	728.435	109	1,209	396	147	1,110	1,414	200	4,590	5,200
D4	1,321	92	929	657	127	1,060	1,327	198	6,191	6,798
D5	995	40	450	276	41	348	428	80	2,266	2,504
D6	1,426	96	1,745	977	104	1,104	1,303	136	2,329	2,728
D7	1,565	86	1,913	737	171	1,445	1,784	378	5,295	6,426
D8	664,406	93	939	338	142	1,102	1,400	207	4,011	4,646
D9	189,180	93	1,503	408	122	1,026	1,285	450	4,935	6,300

Table 3 Solution Statistics for France Télécom Instances

	PMIP		PBB		AMIP+		AMIP	
Data	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*
D1	0	1	0	52	1	8	2	187
D2	0	10	1	201	0	49	8	1,060
D3	1	55	30	14,856	2	786	113,423	13,619,345
D4	1	78	37	21,523	6	2,445	208	10,305
D5	1	6	0	185	1	16	1	35
D6	35	134	39	1,195	39	1,246	399	71,013
D7	3	257	25	9,689	8	1,254	1,426	76,003
D8	0	110	23	16,709	1	32	317	29,509
D9	1	112	11	6,129	1	132	56	3,141

7.2. Random Networks

The second set of instances consists of randomly generated graphs. A graph $G = (N, A)$ was generated by taking a random subset of the edges of the complete graph K_n. For each edge, we created two arcs, one for each direction between the two nodes. The tariff arcs were assigned randomly. The cost on each fixed cost arc was uniformly generated from $[20, 100]$. The source and destination of each demand was assigned randomly, rejecting commodities for which no path with only fixed cost arcs was available. The demand of each commodity was uniformly generated from $[1, 30]$. These instances are much larger than the real-life instances, and are used to illustrate what happens with each algorithm when the number of (tariff) arcs

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

Table 4 Solution Statistics for Random Networks

	SPGM	PMIP		PBB		AMIP+		AMIP	
Data	*Time*	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*
				Networks with 100 nodes and 2,000 arcs					
20T15D	3.4	0.2	2	0.2	165	0.4	5	38.6	121
40T15D	11.6	0.2	2.2	0.4	216	0.4	3	142	934
60T15D	26.2	0.2	5	1.6	720	0.8	7.6	204	1,096
80T15D	51.4	0.4	9.6	2.4	786	0.6	29.4	773	5,764
100T15D	90	0.8	9	4.2	1,038	1.2	29.2	754	5,457
90T20D	90.4	0.6	2.8	2.6	555	1.6	9.4	856.6	3,208
90T40D	178	1.6	11	1,244	206,068	2.8	50.8	(4/5)	27.5%
90T60D	265	3.6	54	(4/5)	3.8%	5.2	163	(5/5)	31.5%
90T80D	353	5.2	378	(5/5)	10.9%	10	705	(5/5)	34.6%
90T100D	441	13	3,169	(5/5)	16.4%	51.8	7,460	(5/5)	35.4%
				Networks with 75 nodes and 4,000 arcs					
100T30D	317	3.2	8.6	46.8	9,717	4.8	17.4	332	179
100T40D	420	4.6	27.4	706	114,278	6.4	44	(2/5)	35.6%
100T60D	628	7	1,648	(5/5)	5.9%	8.6	193	(5/5)	26.3%
100T80D	836	12.4	2,142	(5/5)	15.2%	16.8	2,370	(5/5)	17.4%
20T20D	9.4	0.4	8.2	0.8	477	0.4	6.2	466	960
40T20D	28.4	1	10.4	2	1,129	1.2	14.8	1,268	2,909
60T20D	64.2	1.4	21.6	3.6	1,385	1.4	20.4	(1/5)	4.4%
80T20D	124	1.2	23	5.6	1,760	1.8	32.6	(2/5)	21.3%
100T20D	214	2.6	45.2	10.4	2,499	3.6	25.8	(3/5)	39.4%

or the number of clients in the network increases. Each of these data sets has a name of the form $xTyD$, where x represents the number of tariff arcs and y the number of clients in the network. Each data set actually consists of five instances. Statistics and execution times for each data set are averages over all five instances. Hence, each formulation developed in this paper has been tested on a total of 165 random instances.

The SPGM statistics of random graphs are not interesting: Average and maximum number of paths per commodity are at most 3.5 and 8.0. Solution statistics are in Table 4, which presents the behavior of all formulations developed for larger networks. The results are averages over five instances per size. Table 4 shows what happens when we increase the number of tariff arcs or clients in the network. For these instances, the CPU time of the PMIP, AMIP, and AMIP+ does *Nodesot* include the time needed for the SPGM. This value is given in the column SPGM. The maximum execution time for each solution method was 3,600 seconds. If, for a certain formulation or algorithm, an instance could not be solved within this time, the column CPU indicates the number of instances where the time limit was exceeded, while the column Nodes gives the average relative gap (in percentage) between the best solution as found by any of the algorithms within the time limit.

For the PMIP, AMIP+, and PBB algorithms, the time needed for generation of the SPGM is the largest part of the execution time of the algorithm. When the size of the network increases, the SPGM takes more time to generate, but its execution time is more dependent on the number of shortest paths that need to be calculated (depending on the number of tariff arcs or clients in the network) than on the size of the network on which each shortest path calculation is executed. The execution time of the SPGM increases when the number of clients is increased, but is especially sensitive to the number of tariff arcs in the network. We can furthermore conclude that, although the time needed for the SPGM generation increases when the number of tariff arcs or clients increases, the AMIP+ and PMIP formulation still use the information generated by the SPGM very efficiently when compared to the PBB algorithm or AMIP formulation, where an increase in the number of clients in the network is soon a problem. For example, for the networks with 100 nodes and 2,000 arcs, the data set $90T60D$, consisting thus of 90 tariff arcs and 60 clients, cannot be solved in four out of five instances for the PBB algorithm, and five out of five instances for the AMIP formulation.

Finally, the relative behavior of the algorithms is the same as for the telecom instances.

The numerical results show that the performance of the algorithms developed is consistent with the numerical results for the real-life instances: all instances were solved to optimality by PMIP and AMIP+. Moreover, PMIP and AMIP+ are fastest, PMIP doing slightly better, followed by respectively PBB and AMIP.

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

Table 5 SPGM Statistics for Grid Graphs

	36 nodes, 120 arcs			64 nodes, 224 arcs			100 nodes, 360 arcs		
% $\|T\|$	MIN	MAX	AVRG	MIN	MAX	AVRG	MIN	MAX	AVRG
2.5	1	3	2.00	1	4	2.25	2	6	3.25
5	1	4	2.25	4	10	6.25	4	19	13.25
7.5	3	8	7.25	3	28	15.75	5	52	27.00
10	3	25	11.25	4	81	41.25	43	114	68.75

Table 7 SPGM Statistics for Unit Grid Graphs

	36 nodes, 120 arcs			64 nodes, 224 arcs			100 nodes, 360 arcs		
% $\|T\|$	MIN	MAX	AVRG	MIN	MAX	AVRG	MIN	MAX	AVRG
2.5	2	4	3	2	6	4.3	8	18	11
5.0	3	7	4.8	12	48	22.3	20	61	40
7.5	5	19	12	17	76	42.3	27	211	132.5
10	5	40	19.8	24	115	70	126	418	268.3

7.3. Grids

The third data set has been constructed to see what happens to the models when dominance criteria are less effective. We selected grid networks, which were created as follows. The number of clients in the network (all with unit demand) was set to 4, i.e., from each corner of the grid to the opposite corner. The grids created are of size 6 × 6, 8 × 8, and 10 × 10. The tariff arcs were assigned randomly in the grid. The percentage of tariff arcs ranges from 2.5% to 10%. There are two types of instances differing only in the costs of the nontariff arcs. First, the costs are uniformly generated from $[1, 50]$. Second, the costs are fixed to 1. We can expect the number of undominated paths to be high for these instances, especially for the set with unit fixed costs: The optimal solution for these unit grid graphs is easy, namely set all tariffs to 1, but many paths will have the same (optimal) length in the grid.

The data on these two sets of instances are found in Tables 5 and 6. Tables 5 and 7 show that the number of paths generated by the SPGM for the grid graphs is higher on average than for the other two data sets. The average number of paths is highest for the unit grid graphs, as shown in Table 7. Tables 6 and 8 show the performance of the algorithms developed for the (unit) grid graphs. For these instances, the CPU time of the PMIP, AMIP, and AMIP+ does *Nodesot* include the time needed for the SPGM. This value is once again given in the column SPGM. All instances were solved to optimality for all algorithms implemented.

The first set of instances gives already an idea what happens if dominance criteria are less successful, and the number of undominated paths in a network increases. The path-based algorithms take more time to find the optimal solution than does the arc-based algorithm: AMIP+ now outperforms PMIP, and though AMIP is still much worse than PMIP, it does even better than PBB now.

The second set of instances confirm this behavior. For the 10 × 10 grid, with 10% tariff arcs, the number of undominated paths is on average 268.3 (see Table 7) and the PMIP and PBB take respectively 29.68 and 1.79 seconds to solve. The AMIP+ and AMIP algorithms find the optimal solution in respectively 0.39 seconds and 0.35 seconds. Note that the PBB algorithm finds the optimal solution in the first node of the branch-and-bound tree for these instances, since the optimal paths are the ones with the lowest fixed costs. Hence, instances where dominance criteria are not very effective are better solved with

Table 6 Solution Statistics for Grid Graphs

	SPGM	PMIP		PBB		AMIP+		AMIP	
% $\|T\|$	*Time*	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*
				36 nodes, 120 arcs					
2.5	0.02	0.01	0	0.05	1	0.01	0	0.03	0
5	0.03	0.01	0	0.02	9	0.01	0	0.08	1
7.5	0.04	0.08	0	0.09	33	0.03	3	0.13	7
10	0.06	0.19	23	0.48	220	0.05	16	0.19	28
				64 nodes, 224 arcs					
2.5	0.03	0.02	0	0.03	1	0.02	0	0.06	0
5	0.08	0.11	1	0.12	28	0.04	1	0.21	2
7.5	0.16	0.17	15	0.42	99	0.14	40	0.89	129
10	0.27	0.92	7	1.22	154	0.2	21	0.72	52
				100 nodes, 360 arcs					
2.5	0.06	0.01	0	0.06	1	0.01	0	0.15	0
5	0.2	0.04	0	0.33	58	0.03	0	1.27	134
7.5	0.34	0.22	0	0.52	46	0.1	0	0.6	6
10	11.3	1.64	2	20.38	646	0.78	61	10.4	1,394

Table 8 Solution Statistics for Unit Grid Graphs

	SPGM	PMIP		PBB		AMIP+		AMIP	
% $\|T\|$	*Time*	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*	*Time*	*Nodes*
				36 nodes, 120 arcs					
2.5	0.01	0.03	0	0.01	1	0.02	0	0.03	0
5	0.02	0.01	0	0.02	1	0.04	0	0.04	0
7.5	0.04	0.06	0	0.14	1	0.06	0	0.05	0
10	0.08	0.08	0	0.06	1	0.03	0	0.03	0
				64 nodes, 224 arcs					
2.5	0.03	0.10	0	0.03	1	0.04	0	0.07	0
5	0.09	0.23	0	0.09	1	0.15	0	0.15	0
7.5	0.17	0.35	0	0.17	1	0.03	0	0.06	0
10	0.26	1.06	0	0.29	1	0.08	0	0.13	0
				100 nodes, 360 arcs					
2.5	0.06	0.08	0	0.09	1	0.11	0	0.17	0
5	0.22	0.48	0	0.24	1	0.36	3	0.22	0
7.5	0.76	3.20	0	0.77	1	0.15	1	0.36	0
10	1.76	29.68	0	1.79	1	0.39	7	0.35	0

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:06 . For personal use only, all rights reserved.

arc-oriented formulations, where AMIP+ does better than AMIP if the SPGM is smaller than the original network, i.e., the number of tariff arcs is small compared to the size of the network.

8. Conclusions and Final Remarks

AMIP+ shows the most stable behavior under all circumstances. If reduction methods work well, then PMIP is the (slightly) better choice.

Note that the PBB algorithm works well if the number of paths is limited, and the algorithm does not need a powerful mixed integer programming solver. Since very basic techniques have been used both with respect to concept and implementation, there is still room for improvement on this algorithm.

The SPGM gives deeper insight in the structure of the problem, which can be used for more complex situations, e.g., more general combinatorial bilevel programs. To solve the game-theoretic version of the pricing problem, where multiple operators can set prices on the arcs in the network, an efficient method to solve the pricing problem of each operator is furthermore needed. France Télécom uses the SPGM for strategic studies, such as finding new profitable connections in their network.

Our problem assumes that the capacity on the arcs in the network is unlimited. For certain networks in telecommunications, the capacity available on the arcs in the network is very large in comparison with the amount of data needed to be routed on the network. This is often the case for telecommunications backbone networks where, for some applications, an operator may therefore consider his network to be uncapacitated. For the case of leased lines, customers request capacity from point to point in the network. Their demand usually varies from 2 Mbps to 30 Mbps, while the capacity of the fiber-optic network currently ranges from 2.5 Gbps to 40 Gbps times the number of wavelengths that can be sent over a fiber simultaneously (currently 100). However, the capacitated version of the problem has important applications as well. The SPGM is less suitable in its current form to solve the capacitated tariff-setting problem for two reasons. First, bifurcation may occur: a commodity may see its demand routed over multiple paths. Second, even if one disallows bifurcation, then the SPGM has the problem that its arcs represent paths. Thus, it may be the case that two arcs in the SPGM contain the same arc in the original network. This causes the need for capacity restrictions on multiple arcs simultaneously. In other words, arc capacities complicate the problem considerably.

Acknowledgments

This research was supported by France Télécom Research and Development. The authors thank the Discrete Optimization Network for additional financial support. The authors also thank M. Labbé, P. Marcotte, and G. Savard for their suggestions and comments.

References

Brotcorne, L., M. Labbé, P. Marcotte, G. Savard. 2000. A bilevel model and solution algorithm for a freight tariff-setting problem. *Transportation Sci.* **34** 289–302.

Brotcorne, L., M. Labbé, P. Marcotte, G. Savard. 2001. A bilevel model for toll optimization on a multicommodity transportation network. *Transportation Sci.* **35** 345–358.

Fisk, C. 1986. A conceptual framework for optimal transportation systems planning with integrated supply and demand models. *Transportation Sci.* **20** 37–47.

Fortz, B., M. Thorup. 2000. Internet traffic engineering by optimizing OSPF weights. *INFOCOM'00 Proc. 19th IEEE Annual Joint Conf. Comput. Comm., Tel Aviv, Israel*, 519–528.

Jahn, O., R. Möhring, A. Schulz. 1999. Optimal routing of traffic flows with length restrictions in networks with congestion. *Proc. Sympos. Oper. Res. (SOR'99), Magdeburg, Germany*, 437–442.

Jeroslow, R. G. 1985. The polynomial hierarchy and a simple model for competitive analysis. *Math. Programming* **32** 146–164.

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

Roughgarden, T., E. Tardos. 2000. How bad is selfish routing? *Proc. 41st Annual IEEE Sympos. Foundations Comput. Sci., Redondo Beach, CA*, 93–102.

Vicente, L. N., P. H. Calamai. 1994. Bilevel and multilevel programming: Bibliography review. *J. Global Optim.* **5** 291–306.

RIGHTSLINK
Copyright Clearance Center