

An overview of Stackelberg pricing in networks

Citation for published version (APA):

van Hoesel, C. P. M. (2005). *An overview of Stackelberg pricing in networks*. METEOR, Maastricht University School of Business and Economics. METEOR Research Memorandum No. 036
<https://doi.org/10.26481/umamet.2005036>

Document status and date:

Published: 01/01/2005

DOI:

[10.26481/umamet.2005036](https://doi.org/10.26481/umamet.2005036)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

An overview of Stackelberg pricing in networks

Stan van Hoesel *

September 30, 2005

Abstract

The Stackelberg pricing problem has two levels of decision making: tariff setting by an operator, and then selection of the cheapest alternative by customers. In the network version, an operator determines tariffs on a subset of the arcs that he owns. Customers, who wish to connect two vertices with a path of a certain capacity, select the cheapest path. The revenue for the operator is determined by the tariff and the amount of usage of his arcs. The most natural model for the problem is a (bi-linear) bilevel program, where the upper level problem is the pricing problem of the operator, and the lower level problem is a shortest path problem for each of the customers.

This manuscript contains a compilation of theoretical and algorithmic results on the Stackelberg pricing problem. The description of the theory and algorithms is generally informal and intuitive. We redefine the underlying network of the problem, to obtain a compact representation. Then, we describe a basic branch-and-bound enumeration procedure. Both concepts are used for complexity issues and the development of algorithms: establishing NP-hardness, approximability, and polynomially solvable cases, and an efficient exact branch-and-bound algorithm.

1 Introduction

Combinatorial optimization problems on networks generally involve costs on the arcs. The issue is then, to find the cheapest subset structure of the arcs (such as a path, a tree, or a matching). Thus, the decisions are whether or not to include each arc in the structure. In the problems discussed in this paper there is an additional decision to make, namely, the costs of a given subset of the arcs. This introduces two levels of decisions to be made. At the first level (top) the costs or prices of some edges are determined by an operator

*Faculty of Economics and Business Administration, University Maastricht, P.O. Box 616, 6200 MD Maastricht, The Netherlands, Email: s.vanhoesel@ke.unimaas.nl

or agent, the so-called leader. Once this is done at the second level (lower), customers must decide which edges they use for their optimal structure. This set-up is known as a Stackelberg game. The customer’s structure in this paper is a path between two specified vertices. The game for the leader is then to determine prices of the arcs controlled by him, such that the collected revenues on the shortest paths of the customers are as high as possible.

Applications with a natural network structure can be found in different transportation sectors: toll optimization on roads such as the French highway system, but also the German truck toll system; long-distance freight transportation overseas, passenger transportation in trains, and finally information transportation in telecom networks. Note that an essential ingredient is that the customers must have alternatives, either the market should be oligopolistic or there should be different alternatives, such as the choice between cargo transportation with trains or trucks.

The problem is most naturally formulated as a bilinear bilevel program (see section 2). An integer linear program, has been describe in (Labbé, Marcotte, and Savard 1998). This ILP is, however, not necessary in the description of techniques, and therefore we did not incorporate it in the paper. We introduce the shortest path graph (SPG) in the next section 3. This graph has been introduced in (Bouhtou et al. 2002) to compactify the network representation. It serves also as a very helpful tool for both solution methods and complexity proofs. With the SPG, we develop a basic branch-and-bound scheme in section 4. In section 5, we describe a series of results related to complexity: NP-hardness proofs, (in)approximability, and polynomial-time solvability. In section 6 a series of variants and extensions is described. This section contains some interesting open problems.

2 Problem definition and model

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.

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 given by d_k . The source and destination of commodity k are given by the pair (s_k, t_k) . The set of paths that connect s_k and t_k is given by 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 consecutive arcs: an arc with fixed cost c_a and an arc with tariff t_a .

To ensure that the problem is bounded, we assume that for each commodity there is an upper bound on the amount the customer is willing to pay, or there exists a path from source to destination which uses only fixed cost arcs.

The following formulation of the arc pricing problem is a direct translation of the above description.

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

The formulation given by (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. Notice that the bilevel program given by (1) is not polynomial in its input data, since the set of all possible paths for each client $k \in K$ may be exponential in the size of the problem instance.

Example 1 Consider the following network.

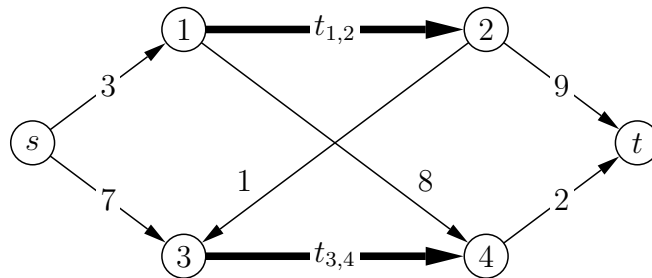


Figure 1: 1-commodity network with two tariff arcs.

<i>Path</i>	<i>Length</i>	<i>Profit</i>
<i>s1234t</i>	$6 + t_{12} + t_{34}$	$t_{12} + t_{34}$
<i>s12t</i>	$9 + t_{12}$	t_{12}
<i>s34t</i>	$12 + t_{34}$	t_{34}
<i>s14t</i>	13	0

In example 1 there are 4 paths connecting s and t . Each of these paths has a fixed cost component and a tariff part (possibly empty). Though the path lengths are linear in the tariffs, the objective is not even continuous in the tariffs. This is illustrated in the example as follows. Let $t_{12} = 0$, and start to increase t_{34} from 0 to an arbitrary large value. Then up till value 3 the path $s1234t$ is optimal, and tariff and profit are equal, and after 3 the profit drops to 0, since the path $s12t$ becomes the most attractive path for the customer. Note that the optimal solution for the leader is to set $t_{12} = 3$ $t_{34} = 3$, with a profit of 6 on the shortest path $s1234t$ per unit demand.

(Labbé, Marcotte, and Savard 1998) considered the following arc oriented bilevel model. Let the vector b^k be the demand/supply vector for each commodity where each element of the vector represents the demand/supply for a commodity at each node in the graph.

$$\begin{aligned}
& \max_{t \geq 0} \sum_{k \in K} \sum_{a \in T} t_a x_a^k \\
& \min_{x^k \geq 0} \sum_{k \in K} \left\{ \sum_{a \in T} t_a x_a^k + \sum_{a \in F} c_a x_a^k \right\} \\
& s.t. \quad \sum_{a \in A_i^+} x_a^k - \sum_{a \in A_i^-} x_a^k = \begin{cases} d_k & i = s_k \\ -d_k & i = t_k \\ 0 & otherwise \end{cases} \tag{2}
\end{aligned}$$

Here A_i^+ is the set of arcs leaving i , and A_i^- is the set of arcs entering i . In this bilevel model, $x^k \in \mathbb{R}^{|A|}$ represents the flow on the arcs, in vector notation of commodity k . Furthermore, A represents the node-arc incidence matrix of the network. This model is a bilinear bilevel program, since the upper level is linear in the tariff variables and the lower level is linear in the arc choice variables. Clearly, the formulation is not linear in the combination of these variables.

Formulation 2 has been used in (Brotcorne et al. 2000) and (Brotcorne et al. 2001) for the development of primal-dual heuristics in case of a single-commodity and multiple commodities, respectively. (Labbé, Marcotte, and Savard 1998) developed an integer linear programming formulation from the bilevel program as follows. The lower level problem is a set of shortest path problems each of which can be described as the linear program given in the lower level problem of 2. The optimal solution can now be characterized using duality theory: add the dual and set the two objectives of dual and primal equal to

one another. Finally, the constraints and objective contain products of tariff and design variables of the arcs. These must be linearized.

3 The Shortest Path Graph

If a client selects a shortest path, say p , then the subpaths of p are also shortest paths. This holds specifically for the subpaths between two consecutive tariff arcs. Consider two such arcs $a_1 = (i_1, j_1) \in T$ and $a_2 = (i_2, j_2) \in T$. Then the subpath between j_1 and i_2 is a shortest path that contains only fixed arcs.

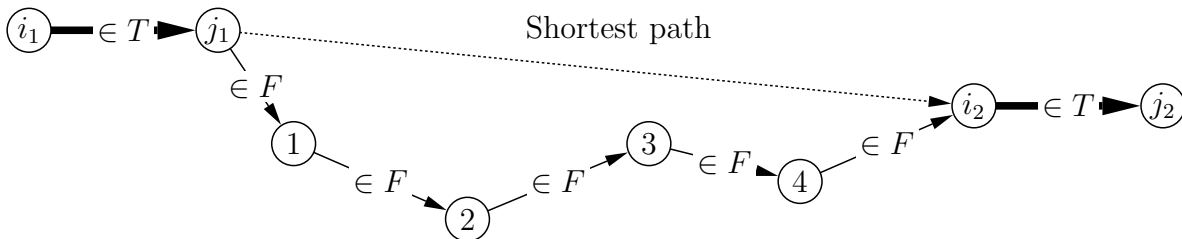


Figure 2: Shortest path of fixed cost arcs between j_1 and i_2 .

Thus, we can restrict the client's choice to paths p of the following structure:

$$p = \{sp_1, a_1, sp_2, a_2, \dots, sp_k, a_k, sp_{k+1}\}, \quad (3)$$

where sp_i , $i \in \{1, \dots, k+1\}$ are shortest subpaths using only fixed cost arcs from a tariff arc a_i to a tariff arc a_{i+1} on the path. Since these subpaths are part of the subgraph using the arcs from F , their length can be computed without determining the tariffs. We can therefore construct a new graph model, in which this is actually done: the shortest path graph.

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 a 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 shortest path graph (SPG).

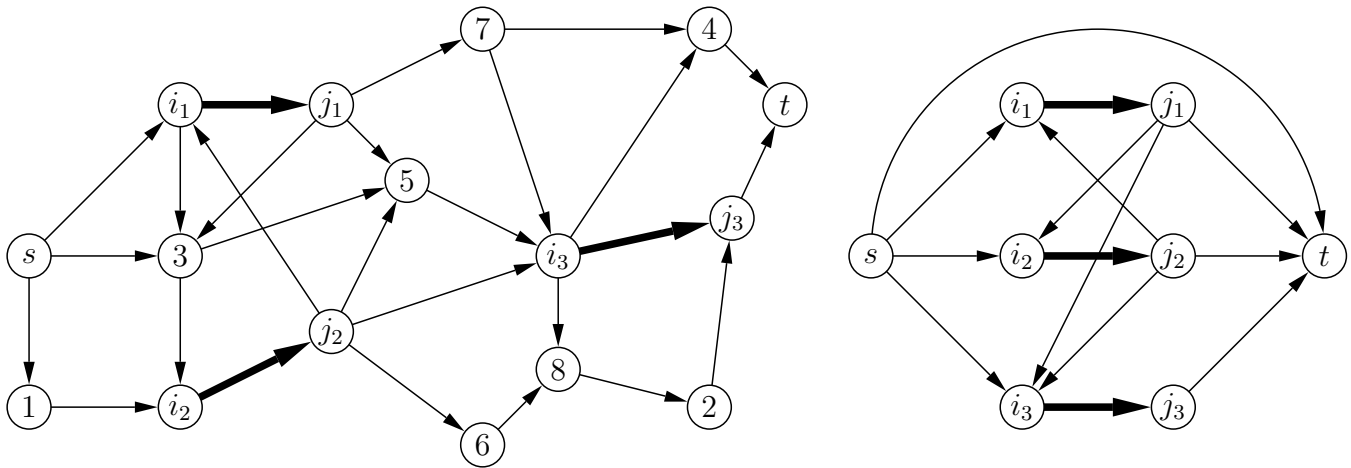


Figure 3: Shortest path graph.

Example 2 Figure 3 shows the SPG of a network containing three tariff arcs for a customer with demand from s to t . The tariff arcs are given by 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. The shortest path graph need not contain all possible arcs: if there is no path between two tariff arcs, then the connecting arc is missing in the SPG is also missing. In the example the arcs (j_3, i_1) and (j_3, i_2) are missing.

In case of multiple customers, we create an SPG for each of them. The *inner graph* (consisting of the end vertices of the tariff arcs, and the arcs between them) is equal for all customers and hence needs to be determined only once. Additional shortest path calculations are necessary only for the arcs leaving the source and/or entering the target of each customer. The shortest path graph model is equivalent to the original graph in the sense that both have an optimal solution of the same value: each path in the original graph is represented by a path in the SPG which is at least as good. Alternatively, a path in the SPG, has exactly the same fixed costs as the shortest path in the original graph connecting the tariff arcs of the first in the same order, and thus it contains the same tariff arcs.

We can further reduce the SPG by removing arcs that will not be taken for any set of tariffs. This is done by use of (path) dominance criteria. Here, nonnegativity of the tariffs is vital.

Definition 1 *If, for any set of tariffs, the path p is at least as short as path q , then path p dominates path q .*

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 1 *Consider two paths p and q . If $T_p \subseteq T_q$ and $c_q \geq c_p$, then path p dominates path q for all tariffs.*

By the nonnegativity of the tariffs and $T_p \subseteq T_q$, the total tariff on p is at most the total tariff on q . Since this is also the case for the total fixed costs, q will never be shorter than p .

Example 3 *An instance where dominance of paths occurs is given in figure 4. The tariff arcs are the arcs (i_1, j_1) , and (i_2, j_2) . 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 fact, the arc (j_1, i_2) is never used and can therefore be removed*

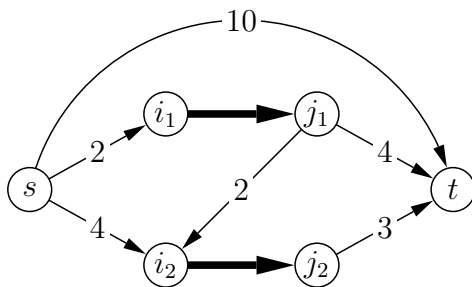


Figure 4: Path dominance.

Arcs can be removed under various circumstances. We will mention a few, a longer list is given in (Bouhtou et al. 2002). Denote by u_{ij} , 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 customer, where node s represents the source node and node t the destination node. In figure 5 we depict the values defined here: the u_{ij} are arc values, and the l_{ij} are node values.

In principal, an arc can be removed if the fixed costs to reach it from s , or to leave it to t are large enough. Thus, arc (i_3, j_3) can be removed since reaching it costs at least 10 and leaving it costs 2. So, the fixed costs for using (i_3, j_3) are at least 12, which is more than moving from s to t directly. The arc (j_1, i_2) can be removed, since the fixed costs of moving directly to t are 4, and leaving j_1 through (j_1, i_2) has fixed costs of at least 6. Similarly, arc (j_2, i_1) can be removed.

In the SPG, 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. This number is reached in a complete SPG.

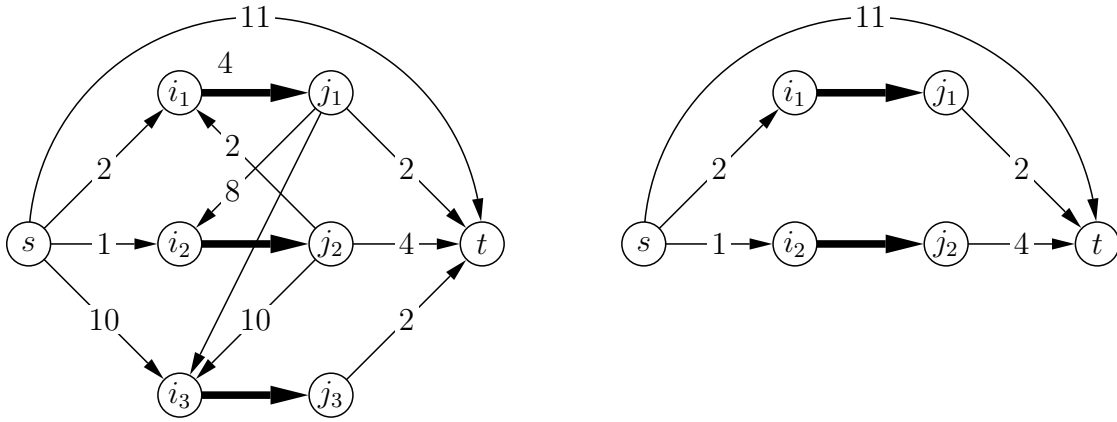


Figure 5: Arc removal.

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. Figure 6 shows that this number of undominated paths can be reached for $|T| = 4$, with an easy extension to arbitrary $|T|$.

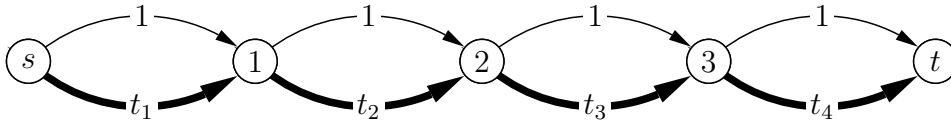


Figure 6: Network with 4 tariff arcs, and 2^4 undominated paths.

4 A basic Branch-and-Bound scheme

In this section we describe a branch-and-bound algorithm for the pricing problem. This algorithm consists of two steps. In the first step we generate for each client his shortest path graph, and we enumerate all feasible undominated paths with their fixed costs and tariffs. In step two we solve the problem to optimality by branching and bounding on the paths.

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 shortest path graph model allow us to determine the set of relevant paths for each customer. We suppose that P_k is reduced to contain the relevant paths only. Recall that the linear function $l_p(t) = c_p + \pi_p(t)$ denotes 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 p_k^u 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.

4.1 Branching

In each node of the branch and bound tree, we select a client, and we 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.

4.2 Bounding

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.

$$\begin{aligned}
 \max \quad & \sum_{k \in K_f} d_k \pi_{p_k^*}(t) \\
 \text{s.t.} \quad & l_p(t) \geq l_{p_k^*}(t) \quad \forall k \in K_f, \forall p \in P_k \\
 & t_a \geq 0 \quad \forall a \in T
 \end{aligned} \tag{4}$$

The linear program described in (4) 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 (4) and then adding the revenues from the tariffs of (4) for the clients in $K \setminus K_f$.

Example 4 Consider a problem with 4 customers, with demand and list of undominated paths as follows:

$d_1 = 20$			$d_2 = 10$			$d_3 = 15$			$d_4 = 5$		
1	$+t_1$	$+t_2$	2	$+t_1$	$+t_2$	3	$+t_1$		2	$+t_{12}$	
3		$+t_2$	4	$+t_1$		5			4		$+t_{12}$
5			5		$+t_2$				7		
			7								

The lower and upper bounding are as follows. Suppose that the subproblem to be solved is the one where client 1 has the second path as shortest path (cost $3 + t_2$), and client 2 has the first path as shortest path (cost $2 + t_1 + t_2$). Then the following LP has to be solved:

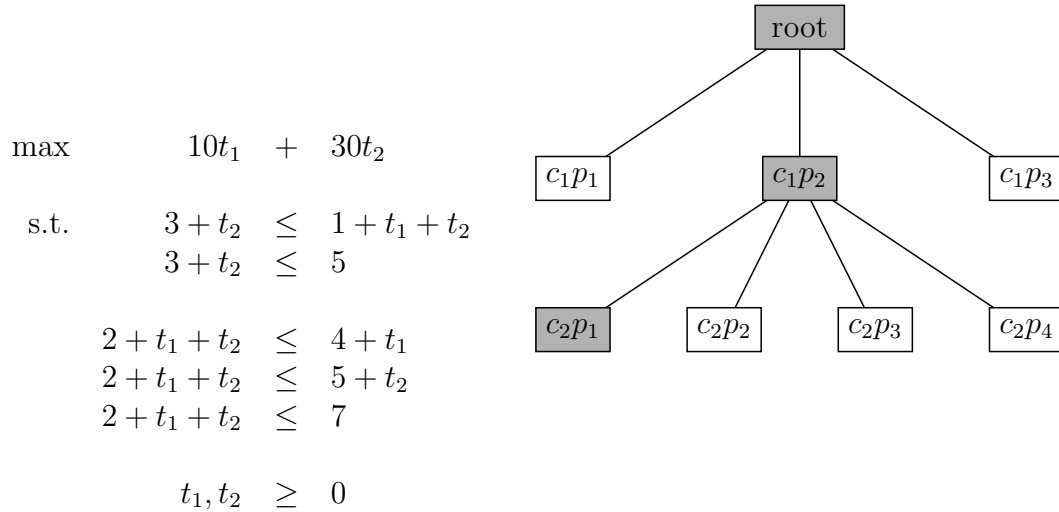


Figure 7: Subproblem with LP.

Note that the subproblems define a linear program to find a lower bound (feasible solution). In the problem above the LP generates the optimal solution $t_1 = 3$, $t_2 = 2$ with value 90. The upper bound is now computed by taking the worst scenario: the gap between the fixed costs of the best path and the fixed cost path, multiplied with the demand of each of the remaining clients. In this case the upper bound is $15*2+5*5=55$ higher than the lower bound. Note that the lower bound is easily increased by taking the contribution of clients 3 and 4 into account. In Bouhtou et al. (Bouhtou et al. 2002) the upper and lower bounds are strengthened furthermore.

For a client $k \in K$ an upper bound for the unit demand revenue generated by that client is given by $c_{p_k^u} - c_{p_k^l}$. As is shown by Labbé et al. (Labbé, Marcotte, and Savard 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 given in figure 8. For a single client with a unit demand from node 1 to node 4, the optimal tariffication 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 given by $7 - 2 = 5$.

(Roch, Savard, and Marcotte 2003) give an example which shows that the relative gap can be logarithmic in the number of tariff arcs. In other words it can be arbitrarily large.

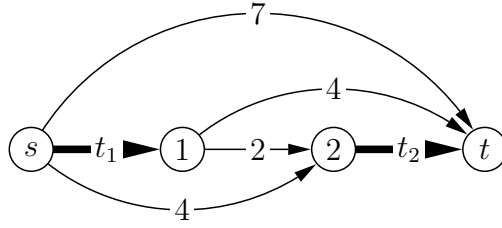


Figure 8: Example where gap can not be used completely.

5 Complexity

5.1 NP-hardness

In Labbe et al. (Labbé, Marcotte, and Savard 1998) the following generalization of the problem has been shown to be NP-hard: general (possibly negative) lower bounds. This is the case already for one customer. The proof uses a reduction from Hamiltonian path. Later, (Roch, Savard, and Marcotte 2003) prove that the problem is strongly NP-hard, also for one customer, in case all lower bounds equal 0. They use a reduction from 3-SAT, which we give below in a slightly modified version. Consider n variables x_1, \dots, x_n (and their negations) and m literals C_1, \dots, C_m . A literal is represented in the tariff network with the following construct.

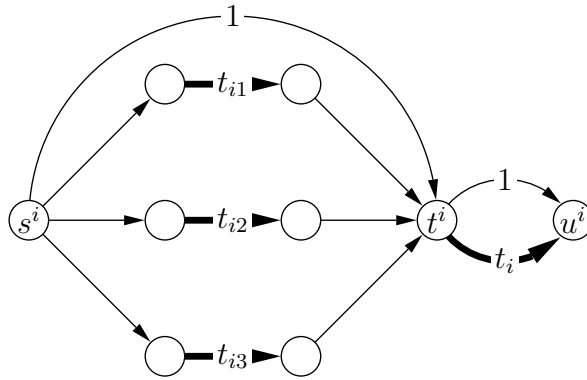


Figure 9: Literal i for three arbitrary variables.

The left three toll arcs correspond to the three variables in the literal. The other arcs are fixed cost arcs, each having fixed cost 0, except the arcs (s^i, t^i) and (t^i, u^i) which have cost 1. The literals are coupled by identifying the nodes u^i and s^{i+1} ($i = 1, \dots, m - 1$). Finally, we add arcs with fixed cost 0 between any pair of tariff arcs that corresponds to a variable and its negation. We connect the head vertex of the earlier literal arc to the tail vertex of the later literal arc with an arc of fixed cost $\frac{1}{2}$. See the picture below where the dashed arcs are of this type. Finally, the source $s = s^1$ and the destination is $t = u^m$.

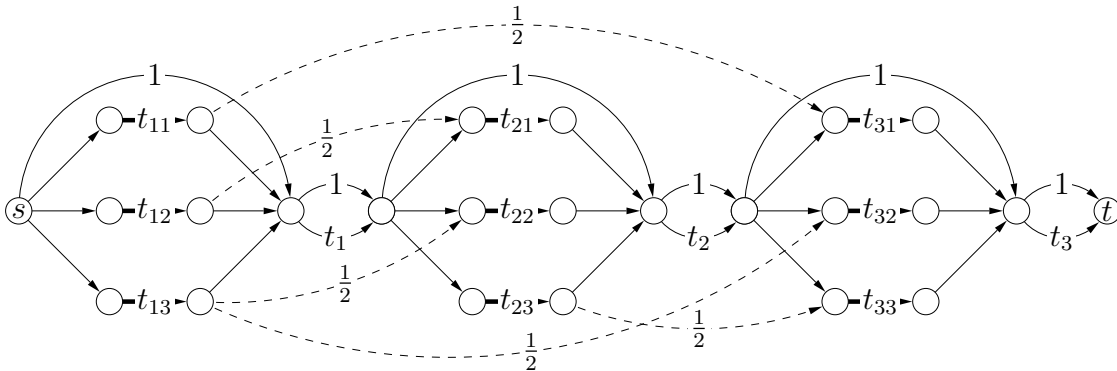


Figure 10: Network for formula $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee x_3 \vee x_4)$.

Note that the only path with fixed cost arcs only has length $2m$, and any path taking two tariff arcs in each literal has fixed cost 0. Moreover, each literal construct can add at most 2 to the value of the problem. The idea is now to let the arcs of a selected (optimal) path correspond with a truth assignment of the variables in the 3-SAT problem.

It is not so hard to see that the tariffication problem has a solution with value $2m$ if and only if there is a truth assignment to the variables of the corresponding 3-SAT instance. A path with value $2m$ can not take any of the dashed arcs, since that would incorporate a (small) fixed cost. Therefore, it must take a tariff arc corresponding to a variable in each of the literals. Putting the variables corresponding to the taken arcs true, gives therefore a valid truth assignment: each literal has a true variable, and the path does not contain a variable and its negation, since in that case there would be a shortcut by using the corresponding dashed arc. On the other hand a valid truth assignment, defines a path where in each literal a true variable can be selected. Setting the tariff on such arcs to 1, and the tariffs on the arcs corresponding to false variables to a sufficiently high value gets us the optimal solution of value $2m$.

5.2 Approximation

(Roch, Savard, and Marcotte 2003) give also an approximation algorithm for the problem with one customer, with a performance guarantee of $O(\log T)$. The idea of the algorithm is fairly easy. It successively tries to find paths with high tariff revenue. It starts with the best path P possible: the one with the smallest amount of fixed costs. After computing the optimal tariffs for P , the tight fixed cost arcs are identified. The one with the smallest cost replaces the subpath in P it connects. For the new path the procedure is repeated until all tariff arcs are removed.

A subroutine of this procedure is to find the best tariffs given that the path P is optimal. Tariff arcs not in P get sufficiently high tariffs, in order not to be a problem. For the arcs in P a greedy algorithm does the job: in order of appearance in P each arc gets a tariff as high as possible.

The analysis of the algorithm is the complicated part. The bound is tight for this particular algorithm, as shown by an example given in (Roch, Savard, and Marcotte 2003).

(Grigoriev et al. 2004) show that in the special case where the inner graph of the SPG contains only tariff arcs $O(\log T)$ is worst-case even for multiple customers. They also prove that this problem is APX-hard.

5.3 Polynomially solvable cases

In (Labbé, Marcotte, and Savard 1998) many special cases that are polynomially solvable have been identified. One of them is the single-customer case, where the order of used tariff arcs is known. Another is the single tariff-arc problem. (van der Kraaij 2004) proves that even in the case of fixed charge costs this problem is polynomially solvable. We will concentrate here on the problem where the number of tariff arcs is not part of the input, i.e., bounded of fixed beforehand.

The bilevel program defined in (1) is shown to be equivalent to a set of linear programs. Consider the problem where we force for each client a specific relevant path to be shortest. Then, as illustrated in the section on Branch-and-bound, the determination of optimal tariffs, if feasible, is a linear program. Since, in the optimal solution, there is a set of shortest paths for the clients, we can consider any possible set of shortest paths and solve the corresponding LP. However, doing this directly does not result in a running time polynomial in the number of customers. We dig a little deeper in the structure of the constraints, to get the desired result.

For any client $k \in K$, consider two paths $p_1, p_2 \in P_k$. If p_1 is to be the shortest of the two paths, the constraint $l_{p_1}(d_k) \leq l_{p_2}(d_k)$ must hold. Thus:

$$c_{p_1}(d_k) + \pi_{p_1}(d_k) \leq c_{p_2}(d_k) + \pi_{p_2}(d_k) \iff \pi_{p_1}(d_k) - \pi_{p_2}(d_k) \leq c_{p_2}(d_k) - c_{p_1}(d_k)$$

This constraint is of the form:

$$\sum_{a \in T_1} t_a - \sum_{a \in T_2} t_a \leq b^k(p_1, p_2) \tag{5}$$

Here, $b^k(p_1, p_2)$ is a constant and T_1 and T_2 are disjoint subsets of T . Note that T_1 contains the tariff arcs in p_1 not in p_2 , and T_2 contains the tariff arcs in p_2 not in p_1 . The constant $b^k(p_1, p_2)$ is referred to in the remainder as the *switching value* for the pair (p_1, p_2) for a client $k \in K$. The number of different left-hand sides of 5 is $3^{|T|}$, since each variable can have coefficient only in $\{-1, 0, 1\}$. The number of switching-values per client is the number of different pairs of paths, and that is bounded by $(eT!)^2$.

The main idea now, is to collect all possible switching values b_r , and to order them: ($r \in \{1, \dots, R\}$). The next point is that we create our set of LPs as follows. For each

disjoint pair of subsets of T , T_1 and T_2 we fix a consecutive pair of switching values with the index $r(T_1, T_2)$, and we add the following constraint to the LP:

$$b_{r(T_1, T_2)-1} \leq \sum_{a \in T_1} t_a - \sum_{a \in T_2} t_a \leq b_{r(T_1, T_2)} \quad (6)$$

It is not hard to show that the number of LPs that we can create is polynomial in the number of clients, but exponential in the number of tariff arcs. For details, see (van Hoesel et al. 2003).

6 Variants of the Stackelberg pricing problem

6.1 Special cases

The structure of the network can be restricted in the sense that the tariff arcs meet certain properties. Two obvious properties are: the tariff arcs form a cut-set (\square), or the tariff arcs form a path (\square). The first problem is not easier than the original problem: it is still NP-hard, for multiple customers and the best known approximation algorithm does not improve the one given in the previous section. The second problem is under recent investigation, with a slight modification: using multiple arcs in common the total tariff may be smaller than the sum of tariffs of individual arcs. It is not known whether this problem is NP-hard. Its application is found in toll systems with many entrances and exits, such as the French highway structure.

6.2 Extensions

Extensions can be defined in several ways. One way is to incorporate capacities on the arcs. This extension is quite hard to handle. For instance, the path-oriented ideas should be replaced with network oriented ideas. No research on this subject has been reported, as far as known to the author.

A second extension is the pricing mechanism. Instead of linear tariffs, these may be fixed charge, or even just increasing with demand. The PhD thesis of (van der Kraaij 2004) contains an analysis of the problem with different types of cost structures. It shows that the case with one tariff arc is polynomial for the fixed charge costs. Moreover, it shows how the branch-and-bound algorithm can be used.

A third extension is the incorporation of the design of the network. The following formulation of the tariffation problem is a direct translation of the above description. Here, P_k is the set of paths in the network $G = (V, F + T')$ where T' is the subset of T of selected

arcs, for which a cost c_a is involved. In (Brotcorne et al. 2005) the problem is formulated and solved with heuristics and a specialized Lagrangean relaxation approach.

6.3 Related problems

In traffic congestion problems time can be considered as a price. Here, however, the time is dependent on the capacity usage, whereas in our case we have only a linear relation between capacity usage and price. Examples are routing of traffic flows (see (Roughgarden and E 2000) and (Roughgarden 2001)) and IP traffic engineering (see (Fortz and Thorup 2000)).

6.4 General Bilevel Programs

The general linear-linear bilevel program has been shown to be \mathcal{NP} -hard by Jeroslow (Jeroslow 1985). For a reference on bilevel programming, we refer the reader to Vicente and Calamai (Vicente and Calamai 1994) who have compiled an annotated bibliography on this subject containing more than one hundred references.

7 Concluding remarks

The standard Stackelberg pricing problem on networks is well-solved from a practical point of view, and also many important theoretical questions have been answered. Nevertheless, some interesting open problems remain: approximability for special cases within a constant factor, and cutting plane methods for the integer linear programming formulation of the problem.

Moreover, for extensions of the problem, many of the results discussed here have no counterpart in the extensions. For instance, the capacitated case has no good algorithmic methods, and no results on approximability. This, of course, applies also for other network or combinatorial bilevel programs.

In other words, the field is still rich of open interesting questions.

References

- Bouhtou, M., S. van Hoesel, A. F. van der Kraaij, and J.L. Lutton. 2002. “Linear Tarification in Multi-Commodity Telecommunications Networks.” *Meteor Research Memorandum*, vol. RM02012.
- Brotcorne, L., M. Labbé, P. Marcotte, and G. Savard. 2000. “A Bilevel Model and Solution Algorithm for a Freight Tariff-Setting Problem.” *Transportation Science* 34 (3): 289–302.
- . 2001. “A bilevel model for toll optimization on a multicommodity transportation network.” *Transportation Science* 35 (4): 345–358.
- . 2005. “Joint Design and Pricing on a Network.” *Working Paper*.
- Fortz, B., and M. Thorup. 2000. “Internet Traffic Engineering by Optimizing OSPF Weights.” *Proc. 19th IEEE Conf. on Computer Communications (INFOCOM)*. 519–528.
- Grigoriev, A., S. van Hoesel, A. F. van der Kraaij, M. Uetz, and M. Bouhtou. 2004. “Pricing network edges to cross a river.” *Meteor Research Memorandum*, vol. RM04009.
- Jeroslow, R.G. 1985. “The polynomial hierarchy and a simple model for competitive analysis.” *Mathematical Programming* 32:146–164.
- Labbé, M., P. Marcotte, and G. Savard. 1998. “A bilevel model of taxation and its application to optimal highway pricing.” *Management Science* 44:1608–1622.
- Roch, S., G. Savard, and P. Marcotte. 2003. “Design and Analysis of an approximation algorithm for Stackelberg network pricing.” *Optimization Online*.
- Roughgarden, T. 2001. “Designing networks for selfish users is hard.” *Proceedings of the 42nd Annual IEEE Symposium on the Foundations of Computer Science*. 472–482.
- Roughgarden, T., and Tardos E. 2000. “How Bad is Selfish Routing?” *Proceedings of the 41st Annual IEEE Symposium on the Foundations of Computer Science*. 93–102.
- van der Kraaij, A.F. 2004. “Pricing in Networks.” Ph.D. diss., University Maastricht.
- van Hoesel, S., A. F. van der Kraaij, C. Mannino, G. Oriolo, and M. Bouhtou. 2003. “Polynomial cases of the tarification problem.” *Meteor Research Memorandum*, vol. RM03053.
- Vicente, L.N., and P. H. Calamai. 1994. “Bilevel and multilevel programming: A bibliography review.” *Journal of Global Optimization* 5:291–306.