4OR-Q J Oper Res (2013) 11:1–30
DOI 10.1007/s10288-012-0213-0

INVITED SURVEY

Bilevel programming and price setting problems

Martine Labbé · Alessia Violin

Received: 1 June 2012 / Revised: 29 August 2012 / Published online: 5 October 2012

Abstract This paper is devoted to pricing optimization problems which can be modeled as bilevel programs. We present the main concepts, models and solution methods for this class of optimization problems.

Keywords Bilevel programming · Pricing · Networks · Combinatorial optimization · Stackelberg game

MSC classification 90-01 · 90B06 · 90B10 · 90C11 · 90C35 · 90C57 · 90C90 · 91A65 · 91A80

1 Introduction

Several real-world problems involve a hierarchical relationship between two decision levels, for example in management (facility location, environmental regulation, credit allocation, energy policy, hazardous material), in economic planning (social and agricultural policies, electric power pricing, oil production), or in engineering (optimal design, structures and shape). In a sequential model therefore, the upper level may represent decision-makers whose decisions lead to some reaction within a particular market or social entity, which corresponds to the lower level of the problem (Colson et al. 2007). Transportation planning is also a typical domain in which examples of such hierarchical structures appear: the upper level corresponds to the network

M. Labbé (✉) · A. Violin
Départment d'Informatique, Université Libre de Bruxelles, Boulevard du Triomphe,
CP 210/01, 1050 Brussels, Belgium
e-mail: mlabbe@ulb.ac.be

A. Violin
e-mail: aviolin@ulb.ac.be

operator seeking to improve the performance of the network, and the lower level corresponds to network users making their travel choices. Migdalas (1995) provides a review of bilevel programming problems that arise in this context, for instance network design, signal setting and origin/destination matrix adjustment problems. Those problems usually contain the so-called network equilibrium problem as the second level.

Pricing problems are widely present in the economic literature. Consider the problem of a company that wants to determine the price of a set of products in order to maximize its revenue. The reaction of potential clients has to be taken into account: if prices are too high, they may decide not to buy the products.

In this tutorial paper we consider how these problems can be modeled: the relationship between the choices of the company and the purchase decision of its customers can be represented as a bilevel optimization problem. This framework presents a hierarchical structure with a first optimization problem (the firm maximizing its revenue) which contains certain constraints specifying that the solution must be optimal to another optimization problem (customers minimizing their cost of buying). The values of the decision variables of the first problem influence the optimal solution of the second problem and viceversa. In game theory an equivalent problem is known as the Stackelberg game (Stackelberg 1952).

This paper is organized as follows. In Sect. 2 we briefly present the general Bilevel Programming framework, and then pricing problems are introduced in Sect. 3, as well as some mathematical models and properties for these schemes. As surveys already exist on these subjects, we only provide a short introduction to them, necessary for understanding the remainder of the paper. Next, we focus on pricing problems based on networks (Sect. 4). We present in detail models, solution methods and various properties developed in the literature as well as some numerical results to confirm their efficiency. In Sect. 5 we describe a variant of the problem involving pricing on paths instead of arcs of the network, and finally a further application to product pricing is described in Sect. 6. Section 7 concludes the paper.

2 Bilevel programming

A bilevel programming (BP) problem is a hierarchical optimization problem in which part of the constraints translate the fact that some of the variables constitute an optimal solution to a second optimization problem. These problems were introduced by Bracken and McGill (1973) as mathematical programs with optimization problems in the constraints, whereas the terms bilevel and multilevel were later introduced by Candler and Norton (1977).

In this setting the first objective function and its proper constraints, which are not related to the second optimization problem, usually refer to the so-called leader or first level, whilst the second optimization problem (objective function and constraints) refers to the follower or second level. This terminology reflects the sequentiality of the problem: the follower chooses his/her optimal solution once the leader's choice is known, and the leader will therefore optimize his/her choice taking into account that the follower always reacts optimally to it.

Table 1 Stackelberg versus Nash: payoff matrix

	Player 2	
	Strategy A	Strategy B
Player 1		
Strategy A	2, 2	4, 1
Strategy B	1, 0	3, 1

For given values of the first level decision variables, the second level problem may have multiple optimal solutions. In this case, different modeling approaches can be proposed depending on the follower's behavior. A cooperative behavior leads to an optimistic solution, so that when there are multiple solutions the leader assumes that the follower's choice is always the one most favorable to him/her. On the contrary, an aggressive behavior leads to a pessimistic solution, where the leader protects himself against the follower's worst possible reaction (Colson et al. 2007). A more complete discussions of these issues can be found in Loridan and Morgan (1996) and Dempe (2002). In this paper, we only consider cooperative behavior.

Let x and y denote decision vectors, f and g objective functions, and X and Y the feasible solution sets of the leader and the follower respectively. The general BP problem can be formulated as:

$$\max_{x,y} \quad f(x, y), \tag{1a}$$

$$\text{s.t.} \quad (x, y) \in X, \tag{1b}$$

$$y \in S(x), \tag{1c}$$

$$\text{where} \quad S(x) = \arg\min_{y} g(x, y), \tag{1d}$$

$$\text{s.t. } (x, y) \in Y. \tag{1e}$$

In game theory, the BP problem has been introduced under the name of Stackelberg game (Stackelberg 1952). In such a game there are two players, a leader L and a follower F. The leader plays first and decides his/her best strategy, taking into account that the follower reacts in an optimal way to his/her choice. Player F plays second, and so already knows L's choice of strategy when choosing his/her own.

We present an example in Table 1, with two players, called 1 and 2. They both have two strategies, denoted by A and B, and they want to maximize their own gain. Gains for all players and strategies, also often called payoffs, are reported in the table. If Player 1 is the leader and Player 2 is the follower, the Stackelberg solution consists of strategy B for both players, leading to a gain of 3 for Player 1 and of 1 for Player 2. On the contrary, if Player 2 is the leader and Player 1 the follower, the Stackelberg solution consists of strategy A for both players, leading to a gain of 2 for both of them.

Another well known solution concept in game theory is the Nash equilibrium (see e.g. Owen 1968), which is appropriate for a simultaneous game (i.e. players play at the same time, without any hierarchical structure) or games that are repeated many times. Specifically, a Nash equilibrium consists in a solution in which no player has an interest in changing his/her choice of strategy, as it cannot lead to a better gain. For the game described by the payoff matrix in Table 1, the Nash equilibrium is unique and given by strategy A for both players, leading to a gain of 2 for both of them. It is therefore interesting to underline the conceptual difference between a Stackelberg solution and a Nash equilibrium: they correspond to different assumptions in the game rules, as the former is sequential with a precise hierarchical structure. They therefore generally lead to different solutions. One can also point out that, under mild conditions, there is always a Stackelberg solution. On the contrary, a Nash equilibrium may not exist, as it could occur that, for all possible pairs of strategies, at least one player would have an interest in changing his/her choice.

From a computational point of view, bilevel problems are intrinsically difficult. Even the simple version of BP problem where the objective functions and the constraints are linear has been shown to be $\mathcal{NP}$-hard by Jeroslow (1985). Furthermore, Hansen et al. (1992) prove the strong $\mathcal{NP}$-hardness of the problem. Vicente et al. (1994) strengthen these results and proved that merely checking strict local optimality and checking local optimality in linear BP problems are $\mathcal{NP}$-hard problems.

Due to the difficulty of BP, solution methods and algorithms generally focus on particular cases where functions have convenient properties, such as linearity or convexity, in order to exploit their structure to develop efficient solution methods. References can be found in Vicente and Calamai (1994); Dempe (2002); Colson et al. (2005, 2007).

We now concentrate on the application of the BP paradigm for price setting problems, which can be seen as sequential games between a company and potential customers.

3 Price setting problem

In a price setting problem the leader (first level) sets some taxes or prices for some activities, and the followers (second level) select activities from among taxed and untaxed ones to minimize operating costs. We assume that there are n_1 taxed and n_2 untaxed activities. By setting $T \in \mathbb{R}^{n_1}$ as the tax vector, $x \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$ as the vectors associated with taxed and untaxed activities respectively, f and g as the objective functions of the leader and the follower respectively, and $\Pi \subset \mathbb{R}^{n_1+n_2}$ as the feasible solution set, the price setting problem can be formulated as:

$$\max_{T} \quad f(T, x, y), \tag{2a}$$

$$\text{s.t.} \quad (x, y) \in \arg\min_{x,y} g(T, x, y), \tag{2b}$$

$$(x, y) \in \Pi. \tag{2c}$$

This framework fits many applications, such as for example the toll optimization of highways (this has been carried out for France and Spain), of truck toll systems (in Germany), the pricing of express mail delivery, passenger transportation systems (train, airlines), and various other pricing schemes (hotel rooms, car rental, travel and tourism packages and telecommunications packages). An overview of Stackelberg pricing problems applied to networks has been carried out by Van Hoesel (2008).

Toll optimization schemes for highways using bilevel programs have been studied by Labbé et al. (1998); Dewez et al. (2008) and Heilporn et al. (2010b, 2011), with a leader owning the highway and setting tolls on its arcs, and the followers traveling either on the highway or on national routes. We discuss this problem in more depth in Sects. 4 and 5.

Applications of this kind of pricing models to telecommunications can be found in Bouhtou et al. (2007a,b). In the first paper, the authors approach the pricing problem of a telecommunications operator owning part of a network in which clients want to route their flows at minimum cost, whereas in the second paper they consider a restriction of it, considering that each client utilizes at most one of the operator's arcs (as if the operator would own bridges over a river and wanted to optimally price them). They prove that this restricted problem is APX-hard. They also propose some preprocessing methodologies to reduce the size of the network on which the model is solved. We discuss these reduction methods in Sects. 4.6 and 4.7.

Brotcorne et al. (2000) consider a bilevel model for the freight tariff-setting problem, where the follower is a shipping company willing to send a prescribed quantity of goods from origin nodes to customers at minimum cost, and the leader is a carrier seeking to maximize its revenue by setting optimal tariffs on the subset of arcs it controls. They assume that the leader is not a dominant player of the market, implying that the total demand is not influenced by the leader's prices. Competitors (i.e. other carriers) do not react in the short term to the leader's prices. Another domain in which this assumption holds is the trucking industry. For given freight rates set by the leader, the shipper's distribution problem is a transshipment problem, whose solution is an assignment of flow on some subtree of the graph. The authors first show how the bilevel model can be reformulated as a single level bilinear program. They then propose metaheuristics that explicitly take into account the structure of the network. In particular, they describe four primal-dual heuristics. The efficiency of these algorithms is calculated by comparing their solutions to optimal ones obtained from a mixed integer reformulation of the model, using a commercial solver. Extensions of these heuristic algorithms have been proposed by Brotcorne et al. (2001). They present and test an algorithmic scheme that can solve toll-setting problems of significant sizes to near optimality, within reasonable computing times.

Brotcorne et al. (2008) consider the problem of pricing in a network together with the design issue: this situation is realistic where the network design can be changed in the short term together with the pricing, for example in telecommunications networks. In this case the leader is a telecommunications operator which wants to simultaneously determine the connections to be opened and the tariff to be applied on them. The followers are users sending flows along cheapest paths joining their respective origins and destinations. Brotcorne et al. (2008) propose a bilevel formulation for this problem and discuss its properties. In particular, they show how the capacity constraints (ensuring that positive flows are routed only on open links, and that they respect capacities) present at the lower level can be moved to the upper level without affecting the optimal solution. From an economic point of view this behavior can be interpreted as follows: those capacity constraints imposed on the users can be enforced through a suitable and finite tariff schedule, and this can be achieved without affecting the leader's revenue. The authors therefore use this property to develop an efficient solution algorithm, as the lower-level problem reduces to a set of independent shortest path problems. Finally they provide some numerical results on both randomly generated and real data.

Recently, a bilevel model to price the transportation of hazardous materials has been proposed by Amaldi et al. (2011). They consider a road network where the authority can block some arcs for transportation of hazardous materials (e.g. by imposing a high toll) and carriers choose minimum cost routes on the road network. The authority wants to minimize the overall risk. Amaldi et al. (2011) propose a bilevel mixed integer programming formulation that guarantees stability, meaning that for each commodity multiple minimum cost paths having different risk values do no exist. In general the optimal solution of this problem does not coincide with the union over all commodities of their relative minimum risk path. They prove the $\mathcal{NP}$-hardness of the problem, even when there is only a single commodity to be shipped. Finally they derive a single level formulation and solve it using a commercial solver, providing numerical results from real data of an Italian city.

In Castelli et al. (2012), a bilevel model has been proposed to determine en route charges in the context of air traffic management (ATM). En route charges are tolls applied by national ATM agencies to all flights passing over European countries, as specified by European Commission regulations. The problem is described as bilevel involving one national agency imposing tolls on its airspace and seeking to gain the maximum possible revenue, and flights traveling between a predetermined pair of airports, choosing the cheapest route. In European airspace, flights are not entirely free to choose their route, as they must pass through a certain number of specific points (latitude, longitude and altitude) prescribed by the national ATM agencies. In practice, a given flight between a pair of airports can typically choose between six to eight different flight paths. These paths may go through different countries, and may involve different distances within each country. Moreover, these en route charges are calculated proportionally to the distance flown in each country and the type of aircraft. Each European national ATM agency chooses one toll unit value to charge all air traffic traveling through its airspace. Castelli et al. (2012) propose a sequential solution procedure and test it on some real air traffic data. They show that a national ATM agency could realistically use this approach to determine the best en route charge to apply in its airspace. This approach could be extended to other cases of proportional tolls (for instance a per kilometer toll on a road network).

Cardinal et al. (2011) consider a pricing problem where the follower is looking for the minimum spanning tree on a graph, and the leader owns a subset of arcs and prices them, maximizing his/her revenue. This framework applies in telecommunications problems for example, where a company owns and sells several point-to-point connections between locations, and a customer wants to buy a network connecting his/her locations in the form of a spanning tree. The market is composed of the company and its competitors, which own the other connections. Cardinal et al. (2011) prove that this problem is APX-hard even if there are only two different cost values on the arcs owned by the competitors. They give an approximation algorithm, provide an integer linear formulation of the problem and study the relation between them.

Finally, this pricing paradigm appears to be an adequate framework for the principal/agent problem (see Van Ackere 1993). In this problem, widely studied in economics, there is a principal who wants to delegate some task to an agent against reward. The principal optimizes his/her utility (e.g. by minimizing the reward offered to the agent) whilst ensuring that the agent accepts the task and performs it in a satisfactory way. In the classical example of agency theory, the principal is interpreted as being the owner of a company and the agent its manager, who should be motivated to act in the interest of the principal even when the principal cannot observe the action. In the basic model, the outcome of the task performed by the agent depends on two elements: the level of effort exerted by the agent, which is a disutility for the agent and cannot be observed by the principal, and a random factor, which the agent learns after selecting his/her level of effort and which the principal does not observe. If the agent refuses the task, he/she obtains a certain utility level, which must be smaller than the expected utility level in case of acceptance. For more details on how these elements can be computed and on variations of this basic scheme, we refer the interested reader to Van Ackere (1993). She considers the model with different risk scenarios and different levels of knowledge for the players involved, and also describes some applications of this model in accounting (for budgetary control systems and variable cost allocation), in industrial organization (for capital structure, disciplining of the product, labor and capital markets and the role of supervision), in marketing (for the relationship between a sales manager and a salesman) and in finance (for the relationship between the shareholders and the manager of a company). Van Ackere (1993) also presents some management science problems in which this framework arises, such as centralized versus decentralized production planning, scheduling of rare resources and selection of batch size. Aside from all the details inherent to each individual real situation to be considered, the principal/agent problem could be modeled as a BP pricing problem, where the principal is the leader who wants to determine the minimum reward to offer to one or more agents (followers), such that tasks are guaranteed to be carried out with good results.

3.1 The linear price setting problem

We now focus on price setting problems in which objective functions and constraints are linear. The second level objective function has coefficient vectors $c_1 \in \mathbb{R}^{n_1}$ and $c_2 \in \mathbb{R}^{n_2}$, which represent the fixed costs of taxed and untaxed activities respectively. Matrices $A_1 \in \mathbb{R}^{m \times n_1}$ and $A_2 \in \mathbb{R}^{m \times n_2}$, and vector $b \in \mathbb{R}^m$, represent the coefficients of the m constraints of the follower's feasible solution set. The linear price setting problem can be modeled as follows:

$$\max_{T} \quad Tx, \tag{3a}$$

$$s.t. (x, y) \in \arg\min_{x,y} \quad (c_1 + T)x + c_2 y, \tag{3b}$$

$$s.t. \quad A_1 x + A_2 y = b, \tag{3c}$$

$$x, y \geq 0. \tag{3d}$$

To guarantee the existence of a bounded solution, we assume that the follower's feasible solution set $\Pi_1 = \{(x, y) : A_1 x + A_2 y = b, x, y \geq 0\}$ is non-empty and bounded, and that the follower set of feasible solutions using only untaxed activities $\Pi_2 = \{y : A_2 y = b, y \geq 0\}$ is non-empty. In fact, if Π_1 is non-empty and bounded, the second

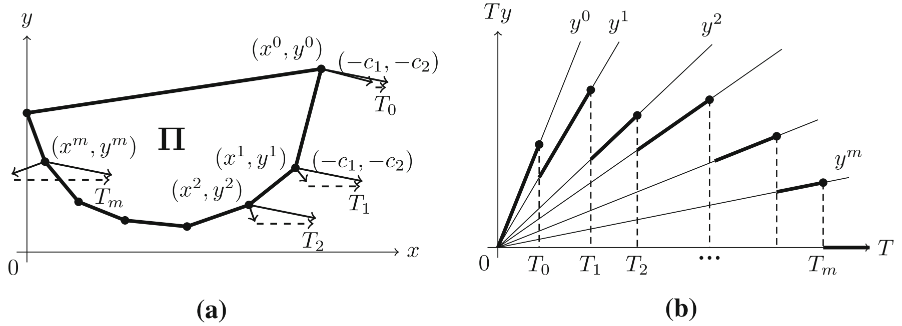


Fig. 1 Graphical example of objective functions in two-dimensional case (Labbé et al. 1998). **a** Second level—feasible solutions, **b** first level—objective function (*in bold*)

level problem will always have a finite solution. The non-emptiness of Π_2 guarantees the existence of a tax-free solution for the follower, which is necessary to prevent the leader from imposing an infinite tax on his/her activities, leading to an infinite revenue.

To illustrate the concepts introduced so far, we consider the particular case where the second level has only two decision variables, meaning that the followers can choose between a taxed activity and a free one, and the leader has one tax value T to determine. The formulation is the same as described above, with decision variables $T, x, y \in \mathbb{R}$, parameters $c_1, c_2 \in \mathbb{R}$ and vectors $A_1, A_2, b \in \mathbb{R}^m$. In such a case a graphical representation of the problem can be provided and the optimal solution can be found using a relatively straightforward procedure.

In Fig. 1a the second level objective function is represented, with the set of feasible solutions Π. Each vertex of Π represents a potential optimal solution for the follower.

From linear programming theory, one can easily conclude that a vertex of Π is optimal if the opposite of the objective function coefficient vector $(-(c_1 + T), -c_2)$ belongs to the cone generated by the coefficient vectors of the active constraints at that vertex. This allows one to determine, for each vertex, the values of T for which it is optimal. For instance, vertex (x^0, y^0) is optimal for $T \in [0, T_0]$, (x^1, y^1) is optimal for $T \in [T_0, T_1]$, and so on. The first level objective function Ty is depicted in terms of T in Fig. 1b. One can observe that this function is discontinuous and piecewise linear with slopes y^0, y^1, etc. The optimal solution in this simple example is given by T_1 and (x^1, y^1).

For further details on this case, we refer the interested reader to Labbé et al. (1998).

4 The network pricing problem

The network pricing problem (NPP) is a pricing problem on a network, with an authority which owns a subset of arcs and imposes tolls on them, and users who travel on the network. The authority is the leader who wants to maximize his/her revenue, and network users are the followers who want to minimize their costs, and so will always travel on the minimum cost path.

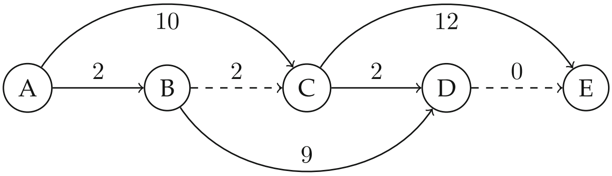
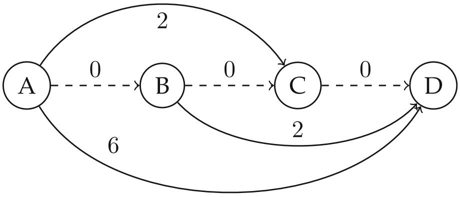


Fig. 2 Example of a NPP (from Dewez 2004)

Fig. 3 Example of a NPP with negative tolls (from Labbé et al. 1998)



The transportation network is defined as a set of nodes linked by a set of (directed) arcs. We define $\mathcal{N}$ as the set of nodes i, $\mathcal{A}_1$ as the set of toll arcs (i.e. arcs owned by the leader on which he/she can impose tolls) and $\mathcal{A}_2$ as the set of other or toll free arcs. We denote as a the generic arc. The set $\mathcal{K}$ represents the commodities k, which are groups of network users traveling from an origin to a destination. We assume that for each commodity there exists a toll free path, i.e. a path which does not pass through any of the arcs owned by the authority, as otherwise the authority could impose an infinite toll on his/her arcs to obtain infinite revenues.

As an example, we consider the network depicted in Fig. 2 in which a single commodity wants to travel from A to E. The authority owns arcs B–C and D–E (dashed arcs), and fixed costs on arcs are reported on the graph. The toll free path is A–C–E with cost 22, so this is an upper bound for the total travel cost the commodity is willing to pay. A corresponding lower bound is given by the cost of the shortest path on the network if the authority imposes zero tolls. This path is A–B–C–D–E with a cost of 6. Hence, an upper bound for the authority's revenue can be calculated as the cost of the shortest toll free path minus the cost of the shortest path if all tolls are set to zero (in this example: $22 - 6 = 16$). The toll free path cost can be computed as the cost of the shortest path on the network in which all tolls are set to infinity. By denoting by $\gamma^k(\mathbf{T})$ the cost of the shortest path for commodity k and toll vector $\mathbf{T}$, this upper bound on the authority's revenue can be written as $UB^k = \gamma^k(\infty) - \gamma^k(0)$. This bound is not always reached, as can been seen in this example: the toll on arc B- C must be at most 5, and on arc D- E at most 10. So the authority's revenue will never exceed 15. In fact, these values provide an optimal solution.

If tolls are allowed to be negative, the model can deal with subsidies. Even if this situation seems unlikely to happen in a real case, the following example shows that an optimal solution may require negative tolls on certain arcs. In Fig. 3 we report the graph in which a single user wants to travel from A to D. The leader owns arcs A–B, B–C and C–D which have zero fixed cost, and then there are toll free arcs A–C, A–D and B–D. As previously described an upper bound on leader's revenue is 6. With some simple calculations, an optimal solution is $T_{A-B} = T_{C-D} = 4$ and $T_{B-C} = -2$, for a total revenue of 6 (as it is equal to the upper bound, the optimality is guaranteed).

In this paper we will suppose tolls to be always non-negative, but most of the results can be generalized to negative tolls.

There are also some cases where the follower has multiple optimal solutions for a given toll vector. Here we consider that in such cases the commodities take the choice which is most profitable for the leader. This assumption is not restrictive, because the leader could decrease some tolls of his/her most profitable solution by ϵ, making that solution the only optimal one for the follower.

4.1 Arc pricing

In this subsection we introduce the definition of "arc pricing" in constrast to the "path pricing" that will be introduced later, meaning that the leader imposes a toll on each of his/her arcs, and these values can be different.

We first introduce some notation. For each commodity $k \in \mathcal{K}$, let η^k be its demand and o^k and d^k be its origin and destination respectively. Moreover, we define c_a as the travel cost on arc $a \in \mathcal{A}_1 \cup \mathcal{A}_2$. The leader wants to set a toll T_a on each toll arc $a \in \mathcal{A}_1$, such that his/her total revenue is maximum, and followers will seek their minimum cost path on the network, fixing flow variables x_a^k on toll arcs, and y_a^k on toll free arcs (these variables are equal to 1 if commodity k uses arc a, 0 otherwise). Later in the description, when needed for the sake of clarity, we will also use the notation (i, j) to indicate an arc a of the network, where i is the tail node and j the head node of the arc.

The NPP for the arc pricing can therefore be modeled as follows:

$$\max_{T \geq 0} \sum_{a \in \mathcal{A}_1} T_a \sum_{k \in \mathcal{K}} \eta^k x_a^k, \tag{4a}$$

$$s.t.\ (x, y) \in \arg\min_{x,y} \sum_{k \in \mathcal{K}} \left(\sum_{a \in \mathcal{A}_1} (c_a + T_a) x_a^k + \sum_{a \in \mathcal{A}_2} c_a y_a^k \right), \tag{4b}$$

$$s.t.\ \sum_{a \in i^+} (x_a^k + y_a^k) - \sum_{a \in i^-} (x_a^k + y_a^k) = b_i^k \qquad \forall k \in \mathcal{K}, \forall i \in \mathcal{N}, \tag{4c}$$

$$x_a^k, y_a^k \geq 0 \qquad \forall k \in \mathcal{K},\ \forall a \in \mathcal{A}_1 \cup \mathcal{A}_2, \tag{4d}$$

where i^- and i^+ denote the sets of arcs with i as head or tail respectively, and b_i^k is equal to -1 if i is the origin node of commodity k, 1 if it is the destination node, and 0 otherwise. As the second level is a shortest path problem (Eqs. 4b, 4c, 4d), whose linear programming formulation has the total unimodularity property, there is no need for integrality constraints on the decision variables. This bilevel NPP for a multicommodity network was first introduced by Labbé et al. (1998).

4.2 The case of a single toll arc

The case of a NPP where the authority owns only one arc a is relatively straightforward, and can be solved in polynomial time (Labbé et al. 1998). We define T as the toll value the leader can impose on arc a, and $\gamma^k(T)$ as the cost of the shortest path for the commodity k for a given value of T. We set the upper bound of the toll that can be imposed by the leader for commodity k as $\pi_k = \gamma^k(\infty) - \gamma^k(0)$. Then we sort all π_k quantities for all commodities in decreasing order. We assume that the order is $\pi_1 \geq \pi_2 \geq \cdot \geq \pi_{|\mathcal{K}|}$, where $|\mathcal{K}|$ is the number of commodities. For any toll value T which is not equal to one of the values in this π_k sequence, we can increase the toll with $\epsilon > 0$ and achieve a higher revenue. Thus, the optimal value of T is equal to one of the π_k values. Moreover, for a toll value π_i ($i \in \{1, \ldots, |\mathcal{K}|\}$) only commodities $k \leq i$ (for which $\pi_k \geq \pi_i$) will choose the toll arc. The leader revenue function is:

$$\mathcal{R}(\pi_i) = \sum_{k \leq i} \pi_i \eta^k. \tag{5}$$

The leader will choose the toll value that maximizes his/her revenue, so the optimal solution will be:

$$T^\star = \pi_{i^\star}\text{ , such as } i^\star = \arg \max_{i \in \{1,\ldots,|\mathcal{K}|\}} \mathcal{R}(\pi_i). \tag{6}$$

The leader revenue function is shown in Fig. 4. It is a piecewise linear function, with discontinuities at π_i values. In each interval the function is described by a straight line whose slope is given by the cumulative demand of commodities which will choose the toll arc for that π_i value.

4.3 Complexity

In Labbé et al. (1998), the authors prove that the general problem is $\mathcal{NP}$-complete, while some particular instances are polynomially solvable, e.g. the single toll arc case described above. Furthermore Roch et al. (2005) prove that the NPP with lower bound constraints on tolls is strongly $\mathcal{NP}$-hard, even for one single commodity and/or when negative tolls are allowed. We will now analyze this proof, for a network with one commodity.

First of all let us define the decision problem: given an instance of the NPP and a constant $\mathcal{R}$, does a toll vector T exist such that $Tx \geq \mathcal{R}$, and such that (x, y) is an optimal flow for the second level in reaction to T?

Given a fixed toll vector T, optimal flow variables can easily be determined with a shortest path algorithm. We can check in polynomial time if the toll vector satisfies the above condition. The NPP problem is therefore in $\mathcal{NP}$.

Roch et al. (2005) consider a reduction from 3-SAT (see for example Garey and Johnson 1979) to the NPP. Consider a 3-SAT formula $F = \bigwedge_{i=1}^m (l_{i,1} \vee l_{i,2} \vee l_{i,3})$, with m clauses and three literals per clause, $l_{i,j}$ for $j = 1, 2, 3$. Each literal corresponds to a variable $x_1, \ldots, x_n$ or to its negation. Each clause of a 3-SAT formula is represented

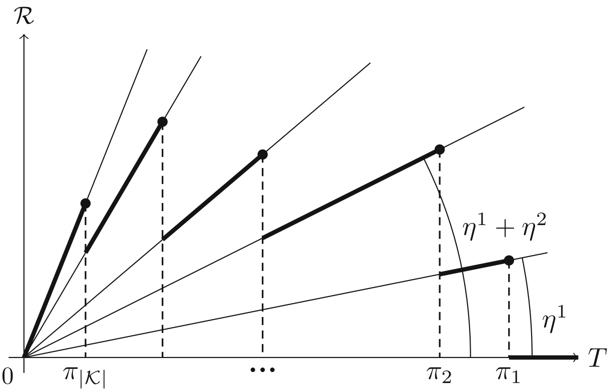
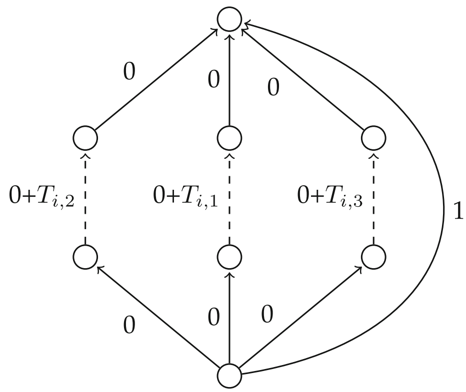


Fig. 4 Leader revenue (*in bold*) in the case of a single toll arc

Fig. 5 Subnetwork for one clause: $(l_{i,1} \vee l_{i,2} \vee l_{i,3})$

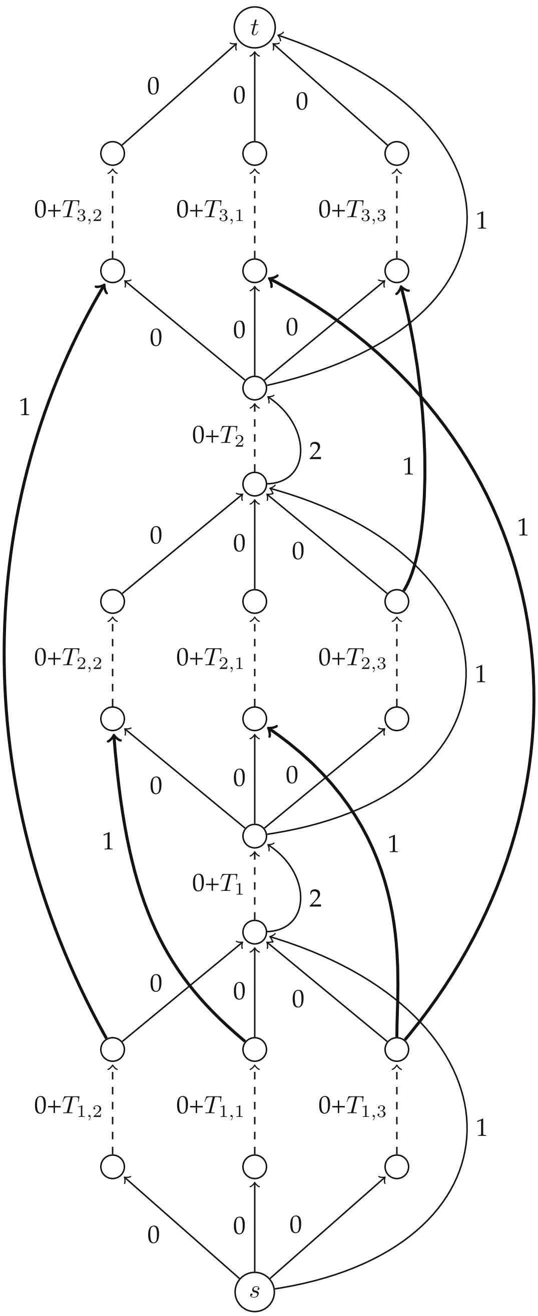


as a subnetwork, where there are three toll arcs representing the three literals of the clause and one toll free arc of cost 1 (see Fig. 5). Subnetworks are then connected by two arcs, one toll arc and one toll free arc of cost 2 (see the example of Fig. 6). All toll arcs have zero fixed cost. There is one single user who wants to travel from node s to node t.

The idea is that if the optimal path goes through a toll arc, then the corresponding literal is true. If F is satisfiable, the optimal path has to go through exactly one toll arc per subnetwork, meaning that one literal per clause is true. Moreover the assignment of variables has to be consistent, such that the optimal path does not include a variable and its negation. To prevent this from occuring, Roch et al. (2005) add an interclause toll free arc with fixed cost of 1 between the toll arcs of each pair of literals corresponding to a variable and its negation. These arcs guarantee that inconsistent paths are suboptimal. In Fig. 6 we show the network constructed with these rules for the formula $F = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3 \vee x_4)$.

The length of the shortest toll free path in such a network is equal to $m + 2(m-1) = 3m - 2$, and the length of the shortest path with zero tolls is equal to 0. The leader's

Fig. 6 Network for the formula: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3 \vee x_4)$ (from Roch et al. (2005))



revenue is therefore bounded from above by $3m - 2$. Roch et al. (2005) prove that F is satisfiable if and only if the optimal solution value of the NPP is equal to that bound.

Let us initially assume that the NPP has an optimal solution value of $3m - 2$. To reach this bound, the optimal path has to go through one toll arc per subnetwork, and tolls have to be set to 1 on the corresponding literals, to $C + 1$ on other literals (where C is the sum of all fixed arc costs of the network, large enough that the commodity does not take these arcs) and to 2 on toll arcs linking subnetworks. The optimal path cannot include a variable and its negation because, if this were the case, the interclause toll free arc would impose an upper bound of 1 on tolls of arcs linking subnetworks, which is in contradiction with the optimal value of 2 for tolls on these arcs. Therefore, the optimal path corresponds to a consistent assignment and F is satisfiable.

Consider now that F is satisfiable. This means that at least one literal per clause is true. It is then possible to take into account the path going through these literals. Moreover, as the assignment is consistent, there are no interclause toll free arcs limiting the revenue. Thus, the upper bound of $3m - 2$ is reached on this path.

Finally, there are at most $10m + 2(m - 1) + (3m)^2$ arcs in the network constructed with these rules, such that every 3-SAT instance is reducible to an NPP instance in polynomial time.

For the case of a single commodity, Roch et al. (2005) also provide a polynomial approximation algorithm, with an approximation factor of $\alpha = \frac{1}{2} log_2 |\mathcal{A}_1| + 1$, where $|\mathcal{A}_1|$ denotes the number of toll arcs in the network. This means that such an algorithm is guaranteed to compute a feasible solution with an objective value of at least OPT/α, where OPT is the optimal leader's revenue.

Recently, Joret (2011) shows that the single commodity NPP is APX-hard, with a reduction from 3-SAT-5 (i.e. 3-SAT where each variable appears in exactly 5 clauses) and a network structure similar to the one developed by Roch et al. (2005) for the $\mathcal{NP}$-hardness proof.

4.4 One level MIP formulation

Labbé et al. (1998) show how the lower level optimization problem can be replaced by its primal and dual constraints (Eqs. 7b, 7c, 7d), and its optimality conditions (Eq. 7e), stating that the primal and dual objective functions of each commodity must be equal. This yields the following single-level optimization problem (λ_i, $\forall i \in \mathcal{N}$, are the dual variables):

$$\max_{T,x,y,\lambda} \sum_{a \in \mathcal{A}_1} T_a \sum_{k \in \mathcal{K}} \eta^k x_a^k, \tag{7a}$$

$$s.t. \sum_{a \in i^+} (x_a^k + y_a^k) - \sum_{a \in i^-} (x_a^k + y_a^k) = b_i^k \qquad \forall k \in \mathcal{K}, \forall i \in \mathcal{N}, \tag{7b}$$

$$\lambda_i^k - \lambda_j^k \leq c_a + T_a \qquad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}_1, \forall i, j \in \mathcal{N}, \tag{7c}$$

$$\lambda_i^k - \lambda_j^k \leq c_a \qquad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}_2, \forall i, j \in \mathcal{N}, \tag{7d}$$

$$\sum_{a\in\mathcal{A}_1}(c_a+T_a)x_a^k+\sum_{a\in\mathcal{A}_2}c_a y_a^k=\lambda_{o_k}^k-\lambda_{d_k}^k \qquad \forall k\in\mathcal{K}, \tag{7e}$$

$$x_a^k, y_a^k\geq 0 \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1\cup\mathcal{A}_2, \tag{7f}$$

$$T_a\geq 0 \qquad \forall a\in\mathcal{A}_1. \tag{7g}$$

This problem is non-linear, as the objective function (7a) and constraints (7e) contain the bilinear term $T_a x_a^k$. A linearization can be done by introducing the set of decision variables $T_a^k=T_a x_a^k$, which represent the effective price paid by commodity k on arc a (equal to T_a if the commodity is using the arc, or 0 otherwise). The necessity of linking new variables with the old ones induces the addition of new constraints (8f, 8g, 8h), where M_a^k and N_a are so-called "big M" constants. In particular, constraints (8f) impose that $T_a^k=0$ if the arc a is not used by commodity k, and constraints (8g) and (8h) set $T_a^k=T_a$ if the arc is used, $\forall k\in\mathcal{K}$, $\forall a\in\mathcal{A}_1$. The linearized model is a mixed integer problem (MIP), whose formulation is as follows:

$$\max_{T,x,y,\lambda}\ \sum_{a\in\mathcal{A}_1}\sum_{k\in\mathcal{K}}\eta^k T_a^k, \tag{8a}$$

$$s.t.\ \sum_{a\in i^+}(x_a^k+y_a^k)-\sum_{a\in i^-}(x_a^k+y_a^k)=b_i^k \qquad \forall k\in\mathcal{K},\forall i\in\mathcal{N}, \tag{8b}$$

$$\lambda_i^k-\lambda_j^k\leq c_a+T_a \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \forall i,j\in\mathcal{N}, \tag{8c}$$

$$\lambda_i^k-\lambda_j^k\leq c_a \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_2, \forall i,j\in\mathcal{N}, \tag{8d}$$

$$\sum_{a\in\mathcal{A}_1}(c_a x_a^k+T_a^k)+\sum_{a\in\mathcal{A}_2}c_a y_a^k=\lambda_{o_k}^k-\lambda_{d_k}^k \qquad \forall k\in\mathcal{K}, \tag{8e}$$

$$T_a^k\leq M_a^k x_a^k \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \tag{8f}$$

$$T_a-T_a^k\leq N_a(1-x_a^k) \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \tag{8g}$$

$$T_a^k\leq T_a \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \tag{8h}$$

$$x_a^k\in\{0,1\} \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \tag{8i}$$

$$y_a^k\geq 0 \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_2, \tag{8j}$$

$$T_a^k\geq 0 \qquad \forall k\in\mathcal{K}, \forall a\in\mathcal{A}_1, \tag{8k}$$

$$T_a\geq 0 \qquad \forall a\in\mathcal{A}_1. \tag{8l}$$

Note that variables x_a^k must be binary for this reformulation to be valid.

Dewez et al. (2008) present several families of valid inequalities in order to reinforce the linear relaxation.

4.5 Tight values for "big M and N"

In order to make the linear relaxation of the previous problem as tight as possible, M_a^k and N_a should be set to the smallest values that ensure a valid formulation. As we will see later from the numerical results, a good choice of these constants can

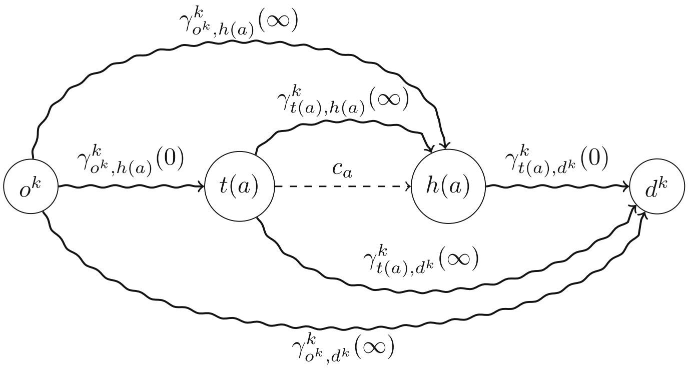


Fig. 7 Paths that occur in the calculation of valid values for "big M"

significantly improve the performance of solution methods. Dewez et al. (2008) show how to calculate valid values for them. In practice, the constant M_a^k represents an upper bound on the toll value that commodity k is willing to pay to use arc a. The constant N_a represents the maximum value of T_a that can be imposed on toll arc a so that at least one commodity uses that arc.

Let us extend the meaning of the shortest path cost function by introducing origin and destination indices: $\gamma_{i,j}^k(T)$ represents the cost of the shortest path for commodity k from node i to node j, with tolls set to T. In particular, $\gamma_{i,j}^k(\infty)$ represents the cost of the shortest toll free path from i to j, and $\gamma_{i,j}^k(0)$ the cost of the shortest path if tolls are all set to zero. Consider now one toll arc a, with fixed cost c_a, and let us denote its tail and head nodes by $t(a)$ and $h(a)$ respectively. Recall also that o^k and d^k are the origin and destination nodes of commodity k. One feasible value for M_a^k can be calculated as the cost of the shortest toll free path from $t(a)$ to $h(a)$ minus the fixed cost c_a. Other feasible values can be similarly obtained using the paths going from the origin node o^k to the head node $h(a)$, the paths going from the tail node $t(a)$ to the destination node d^k and finally the paths going from the origin node o^k to the destination node d^k. Obviously the best value for M_a^k will be the smallest one among the four. In Fig. 7, we show all the paths that occur in the calculation of the four bounds for one toll arc (the dashed arc) and one commodity. Wavy arcs represent shortest paths on the original network.

The following expression, which translates the reasoning just explained, gives a valid value for M_a^k:

$$\begin{aligned} \max \quad \{0, \quad \min \quad \{ & \gamma^k_{t(a),h(a)}(\infty) - c_a, \\ & \gamma^k_{o^k,h(a)}(\infty) - \gamma^k_{o^k,t(a)}(0) - c_a, \\ & \gamma^k_{o^k,d^k}(\infty) - \gamma^k_{o^k,t(a)}(0) - c_a - \gamma^k_{h(a),d^k}(0), \\ & \gamma^k_{t(a),d^k}(\infty) - \gamma^k_{h(a),d^k}(0) - c_a\}\}. \end{aligned} \tag{9}$$

Note that none of the feasible values dominates any other, as shown by the four networks in Fig. 8. In each network there is one commodity traveling from node 1 to node 5.

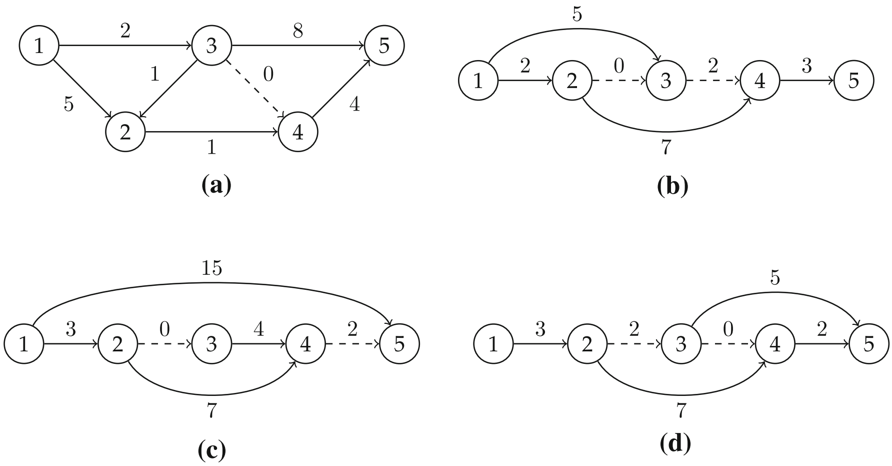


Fig. 8 Example: networks for calculation of valid values on "big M" (from Dewez 2004). **a** Network in which the first bound is the tightest; **b** network in which the second bound is the tightest; **c** network in which the third bound is the tightest; **d** network in which the fourth bound is the tightest

Table 2 Feasible values for M_a for the four networks in Fig. 8 (best values in bold)

	Network (a) Toll arc (3,4)	Network (b) Toll arc (2,3)	Network (c) Toll arc (2,3)	Network (d) Toll arc (3,4)
$M1 = \gamma^k_{t(a),h(a)}(\infty) - c_a$	**2**	∞	∞	∞
$M2 = \gamma^k_{o^k,h(a)}(\infty) - \gamma^k_{o^k,t(a)}(0) - c_a$	4	**3**	∞	5
$M3 = \gamma^k_{o^k,d^k}(\infty) - \gamma^k_{o^k,t(a)}(0) - c_a - \gamma^k_{h(a),d^k}(0)$	4	5	**6**	5
$M4 = \gamma^k_{t(a),d^k}(\infty) - \gamma^k_{h(a),d^k}(0) - c_a$	4	5	∞	**3**

Table 2 reports all these feasible values of M_a for the four networks in Fig. 8. The best value for each network is in bold. Since we assume the existence of a toll free path from the origin to the destination for each commodity, at least the third bound $M3$ is always finite.

Furthermore, for each toll arc a, it is easy to see that a valid value for N_a is the largest M_a^k for all commodities, i.e. $N_a = \max_{k \in \mathcal{K}} M_a^k$ (see Dewez et al. 2008).

Computational experiments (see Sect. 4.8) show that using these sharp values for M_a^k and N_a allows us to halve the size of the gap between the values of the optimal solution and the solution of the linear relaxation. Furthermore, some instances were not solvable with arbitrarily large values for M_a^k and N_a whilst solutions were found using the sharp values presented above.

4.6 The shortest path graph model

The shortest path graph model (SPGM), proposed by Bouhtou et al. (2007b), is a reformulation of the NPP that reduces the practical size of the original network. In theory,

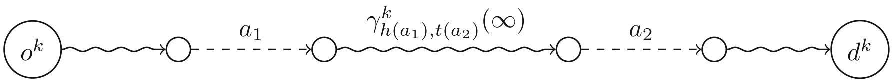


Fig. 9 Example of the structure of possible paths

this reformulation does not necessarily yield a smaller network, but Bouhtou et al. (2007b) conduct numerical experiments showing improvements to the resolution of their models. From a computational time point of view, the SPGM generation can be rather time consuming, especially for large networks, but algorithms can subsequently solve the instances more quickly than would otherwise be the case. Details of numerical experiments for both realistic and random networks are reported in Bouhtou et al. (2007b).

The SPGM exploits the following two observations:

1. the (shortest) path selected by each commodity constitutes an alternating sequence of toll arcs and subpaths containing only toll free arcs—see Fig. 9;
2. each subpath linking the head $h(a_i)$ and the tail $t(a_{i+1})$ of two consecutive toll arcs a_i and a_{i+1} of a path selected by a commodity k has a minimum cost given by $\gamma^k_{h(a_i),t(a_{i+1})}(\infty)$.

By following this reasoning we can construct a SPGM for each commodity. Let $\mathcal{G} = (\mathcal{N}, \mathcal{A}_1 \cup \mathcal{A}_2)$ be the original graph, we then define the new SPGM graph as $\mathcal{G}^k = (\mathcal{N}^k, \mathcal{A}^k_1 \cup \mathcal{A}^k_2)$, $\forall k \in \mathcal{K}$. The set of toll arcs $\mathcal{A}^k_1$ is equal to $\mathcal{A}_1$. Then the set of toll free arcs $\mathcal{A}^k_2$ is constructed as follows for each commodity k. First, $\mathcal{A}^k_2$ contains a toll free arc between the origin node o^k and the destination node d^k. Then, for two toll arcs $a_1 = (i_1, j_1)$ and $a_2 = (i_2, j_2)$, a toll free arc from j_1 to i_2 is added to $\mathcal{A}^k_2$ if there is a toll free path in $\mathcal{G}$ between them. Finally, a toll free arc from the origin node to the tail of a toll arc is also added to $\mathcal{A}^k_2$ if there is a toll free path in $\mathcal{G}$ between them, and similarly, a toll free arc from the head of a toll arc to the destination node is added to $\mathcal{A}^k_2$ if there is a toll free path in $\mathcal{G}$ between them. The cost of a toll free arc $a = (i, j)$ is given by $\gamma^k_{i,j}(\infty)$, which is calculated in the original graph. The NPP in the SPGM constructed with these rules is equivalent to the original NPP in the sense that the optimal solution and its value are equal. We can observe that toll free arcs between toll arcs can be computed once for all commodities. Only toll free arcs linking origin and destination nodes to toll arcs need to be calculated separately for each commodity. An example of the SPGM graph for one commodity is depicted in Fig. 10, where there is one commodity traveling from o to d.

4.7 Preprocessing

The SPGM allows us to work on a reduced network for each commodity. It is possible to further reduce the SPGM graph by removing arcs that will not be taken by the commodity regardless of the values of the tolls. Bouhtou et al. (2007b) propose several graph reductions, which are valid for both the SPGM and the original graph. They can therefore also be used as a general preprocessing to reduce the size of the original

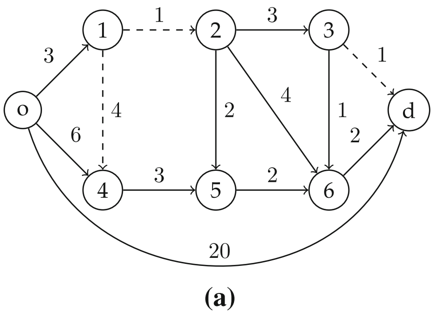


Fig. 10 Example of shortest path graph reduction (from Dewez 2004). **a** Original graph; **b** SPGM graph

network. We will only provide definitions and examples of reduction, for the proofs the interested reader is refered to Bouhtou et al. (2007b).

Consider the SPGM with the reduced graph $\mathcal{G}^k = (\mathcal{N}^k, \mathcal{A}_1^k \cup \mathcal{A}_2^k)$ for one commodity $k \in \mathcal{K}$. For simplicity, in the remainder of this subsection we will omit the commodity index, as we will always consider networks with only one commodity. Note that $\gamma_{i,j}(\infty)$ is an upper bound and $\gamma_{i,j}(0)$ a lower bound on the cost for the path from i to j.

There are six cases in which one or more arcs can be canceled from a network:

1. If $\gamma_{j,t}(0) = \gamma_{j,t}(\infty)$, then if an optimal path from s to t passes through node j, it can use arc (j, t). Hence all other arcs leaving j can be removed.
2. If $\gamma_{s,i}(0) = \gamma_{s,i}(\infty)$, then, similarly, any optimal path from s to t using node i will use arc (s, i) and all other arcs entering i can be removed.

Consider now two toll arcs, (i_1, j_1) and (i_2, j_2):

3. If $\gamma_{j_1,t}(\infty) \leq \gamma_{j_1,i_2}(\infty) + c_{i_2,j_2} + \gamma_{j_2,t}(0)$, then we can delete arc (j_1, i_2).
4. If $\gamma_{s,i_1}(\infty) \leq \gamma_{j_2,i_1}(\infty) + c_{i_2,j_2} + \gamma_{s,i_2}(0)$, then we can delete arc (j_2, i_1).
5. If $\gamma_{s,t}(\infty) \leq \gamma_{s,i_1}(0) + c_{i_1,j_1} + \gamma_{j_1,t}(0)$, then we can delete toll arc (i_1, j_1).
6. If $\gamma_{s,t}(\infty) \leq \gamma_{s,i_1}(0) + c_{i_1,j_1} + \gamma_{j_1,i_2}(\infty) + c_{i_2,j_2} + \gamma_{j_2,t}(0)$, then we can delete arc (j_1, i_2).

In Fig. 11, we report an example of these reductions, in which, for simplicity, toll arcs have zero fixed cost. For instance, we can remove the toll free arc (4, 1) using the fourth reduction: take the two toll arcs (1, 2) and (3, 4), we have $\gamma_{o,1}(\infty) = 2 \leq 4+1 = \gamma_{4,1}(\infty)+\gamma_{o,3}(0)$. Similarly, we can remove arcs (2, 3) and (4, 5). Moreover, we can remove the toll arc (5, 6) using the fifth reduction: $\gamma_{o,d}(\infty) = 11 \leq 10+2 = \gamma_{o,5}(0) + \gamma_{6,d}(0)$.

Bouhtou et al. (2007b) also propose methods to eliminate paths which would never be selected whatever the values of the tolls. Finally, the authors describe a two-phase branch-and-bound algorithm using the SPGM and their preprocessing methods.

4.8 Computational results

In Table 3 we report numerical results taken from Dewez et al. (2008): the arc pricing problem has been tested on random grid networks involving 60 nodes and 208 arcs,

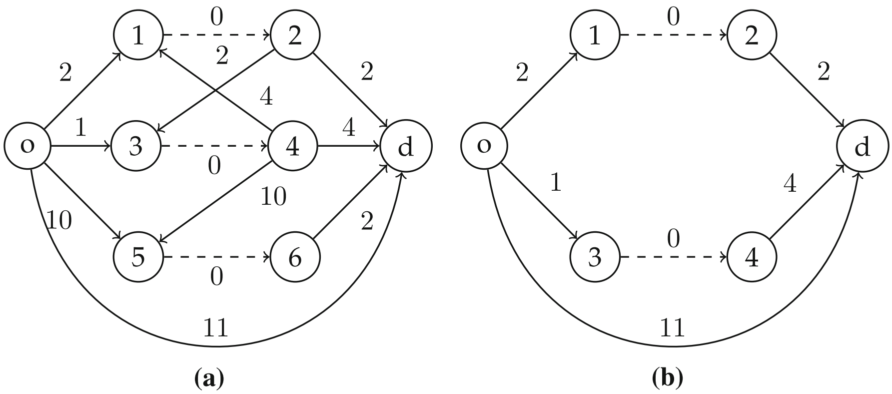


Fig. 11 Example of arc reduction (from Bouhtou et al. 2007b). **a** Original graph; **b** graph after the arc reduction

Table 3 Numerical results for arc pricing (from Dewez et al. 2008)

		20 Commodities			30 Commodities			40 Commodities		
		10 %	15 %	20 %	10 %	15 %	20 %	5 %	10 %	15 %
Arc(G)	Gap (%)	22.8	16.5	16.8	26.2	20.2	22.4	23	24.5	23.6
BM	Cpu	23	349	–	99	4,182	–	23	158	–
	Nodes	140	5,943	–	447	47,522	–	48	1,053	–
Arc(G)	Gap (%)	10.2	9.5	10.4	12	11	13.1	7.6	11.1	13.2
SM	Cpu	13	34	221	59	180	720	13	43	1,341
	Nodes	33	161	2,119	173	839	4,391	10	61	8,235
Arc(G)	Gap (%)	10.1	9.3	10	11.9	10.7	12.8	7.6	11	13
SM+C	Cpu	11	39	133	38	171	788	9	40	1,491
	Nodes	26	135	687	77	511	3,179	12	53	6,257
Arc(SPGM)	Gap (%)	22.8	16.5	16.8	26.2	20.2	22.4	23	24.5	23.6
BM	Cpu	<1	12	621	2	160	–	<1	3	–
	Nodes	60	1,264	20,193	314	6,550	–	27	305	–
Arc(SPGM)	Gap (%)	10.2	9.5	10.3	12	11	12.9	7.6	11.1	13.2
SM	Cpu	<1	4	66	1	18	226	<1	1	73
	nodes	19	199	1, 818	75	535	2,493	7	39	2,590
Arc(SPGM)	Gap (%)	10.2	9.4	10.2	12	10.7	12.8	7.6	11	13.1
SM+C	Cpu	<1	6	101	2	26	231	<1	2	156
	Nodes	21	110	1, 613	58	405	2,563	7	55	3,621

designed to promote interactions between commodities, and made for problems that are combinatorially challenging. Fixed costs on arcs are a random number in the interval [2, 20], for 20, 30 and 40 commodities and 5, 10, 15, 20 % of toll arcs on the total number of arcs in the network. Experiments have been run using CPLEX 8.1 with default values on a Pentium III (500 MHz). The arc pricing has been tested on the original network (noted as G in the table) and on the reduced network with SPGM and reductions previously described in Sects. 4.6 and 4.7 (noted as SPGM in the table). The "big M and N" have been first set to arbitrarily high values (BM in the table), and then to the sharp values described in Sect. 4.5 (SM in the table). Moreover some families of valid inequalities (see Dewez et al. 2008) have been added to strengthen the formulation (+ C in the table).

The gap represents the difference in percent between the optimal solution Z_{opt} and the linear relaxation Z_{lp}: gap = $(Z_{lp} - Z_{opt}) / Z_{opt}$. The "cpu" row gives the cpu times in seconds and the "nodes" row represents the number of nodes in the branch-and-bound tree.

The gaps are similar for the formulation on the original network and the SPGM network, supporting the idea that the reduction of the network does not change the combinatorial complexity very much. However, the SPGM reduces the number of nodes and cpu times. On the contrary, using the sharp values for "big M and N" reduces the gap by a factor of two, and allows the formulation to find the optimal solution for complex instances. Finally cuts have little impact on the gap, but allow the reduction of the number of nodes in the branch-and-bound tree.

5 Path pricing

In contrast to the previous problem, where tolls are set on each arc and the total toll paid by a commodity is the sum of the tolls on the toll arcs of its shortest path, we will now consider a problem of path pricing involving tolls associated to paths. Paths could now be even priced differently for each commodity, introducing unfairness. In such a case $\mathcal{P}$ represents the set of toll paths whose number is polynomial or fixed. We shall denote as $\mathcal{P}_k \subseteq \mathcal{P}$ the set of toll paths that can be taken by commodity k. For each commodity the shortest toll free path is then considered, with cost c^k_{od} and associated binary decision variable y^k. Using this notation, this path NPP can be described as follows:

$$\max_{T\geq 0} \sum_{k\in\mathcal{K}} \eta^k \sum_{p\in\mathcal{P}_k} T_p x^k_p, \tag{10a}$$

$$s.t.\ (x, y) \in \arg\min_{x,y} \sum_{k\in\mathcal{K}} \left(\sum_{p\in\mathcal{P}_k} (c_p + T_p) x^k_p + c^k_{od} y^k_{od} \right), \tag{10b}$$

$$s.t.\ \sum_{p\in\mathcal{P}_k} x^k_p + y^k_{od} = 1 \qquad \forall k \in \mathcal{K}, \tag{10c}$$

$$x^k_p, y^k_{od} \in \{0, 1\} \qquad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k, \tag{10d}$$

where constraint (10c) allows one path choice for each commodity. In this formulation variables need to be binary (it can be easily seen that this condition could be relaxed for y variables).

For a general network with n nodes, the maximum number of paths is exponential. For this reason we consider the path pricing problem only for particular situations or networks where the number of paths is polynomial or fixed, as we will see in the example presented in Sect. 5.2.

The multicommodity path NPP has been proved to be strongly $\mathcal{NP}$-hard by Heilporn et al. (2010b), whether toll arcs are single or bidirectional, based on a reduction from the problem 3-SAT (the proof is similar to the one described in Sect. 4.3 for the the arc NPP). The path NPP with only one toll path is equivalent to the single toll arc NPP introduced in Sect. 4.2, and so is polynomial. Moreover, and contrary to the arc version, the path NPP with only one commodity is also polynomial (Dewez 2004), and Heilporn et al. (2010b) provide a complete description of the convex hull of solutions in this case. We report more details in Sect. 5.4.

5.1 Path pricing versus arc pricing

Consider an authority owning a set $\mathcal{A}_1$ of toll arcs on a network $(\mathcal{N}, \mathcal{A}_1 \cup \mathcal{A}_2)$. Solving the arc NPP provides an optimal toll for each arc of $\mathcal{A}_1$.

However, the authority may instead decide to set tolls on paths containing toll arcs. Since the authority controls only the toll arcs, setting a price for a path is in fact equivalent to setting a price for using the subset of toll arcs of that path, and all paths containing the same subset of toll arcs should have the same toll. Hence, when several paths contain the same subset of toll arcs, only the path with the smallest fixed cost will be considered for pricing. Further, the authority may decide to price only some subset of arcs, e.g. those constituting a simple path in the network (i.e. visiting each node at most once). This special path NPP may yield an optimal revenue higher than its arc version.

First of all, consider the network reported in Fig. 2 of Sect. 4: for the arc pricing we saw that the leader's optimal revenue is 15. Consider now the problem of path pricing: there are four toll paths, plus one toll free path of cost 22 (path A–C–E). The first toll path is A–B–C–D–E, with a fixed cost of 6. The second one is A–C–D–E, with a fixed cost of 12. The third one is A–B–D–E, with a fixed cost of 11. Finally, the fourth one is A–B–C–E, with a fixed cost of 16. One can see that the second toll path should not be considered as, whatever toll the leader chooses, this path will always be more expensive than the third toll path (the paths have the same set of toll arcs and the third one has a smaller fixed cost). This is an example of path domination briefly introduced above. In this simple example the leader may choose a toll of 16 for the first toll path (the one with the smallest fixed cost), and a large toll on all other paths, such that the shortest path for the commodity will be the first toll path. For this network, the leader's optimal revenue for the path pricing model is thus larger than the one for the arc pricing model.

Consider now the network in Fig. 12, where there are three commodities, each of them with unit demand and traveling from o_1 to d_1, from o_2 to d_2 and from o_3 to d_3 respectively. Toll arcs 1–2 and 2–3 are connected as in the case of a highway (more details are reported in the next subsection). Using arc pricing gives an optimal solution value of 10 fot the leader's revenue, with a toll of 3 on arc 1–2 and of 2 on arc 2–3. On the other hand, using path pricing yields a leader's revenue of 11, with a toll of 3 on path o_1-1-2-d_1, a toll of 2 on path o_3-2-3-d_3 and a toll of 6 on path o_2-1-2-3-d_2.

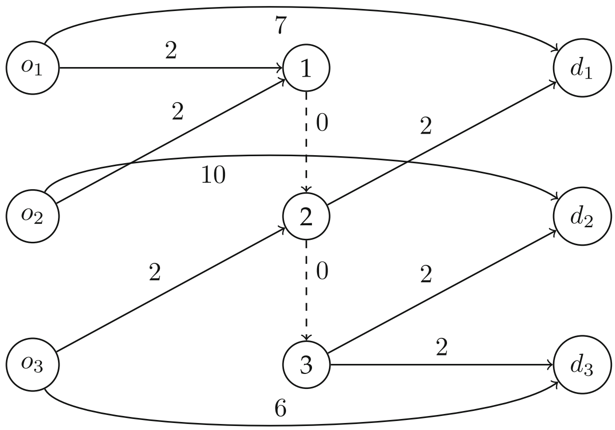


Fig. 12 Example of arc versus path pricing: connected toll arcs

Note that the toll on this last path is larger than the sum of the toll on arcs composing it. Moreover, the leader's revenue for the path pricing problem is again larger than for the arc pricing one.

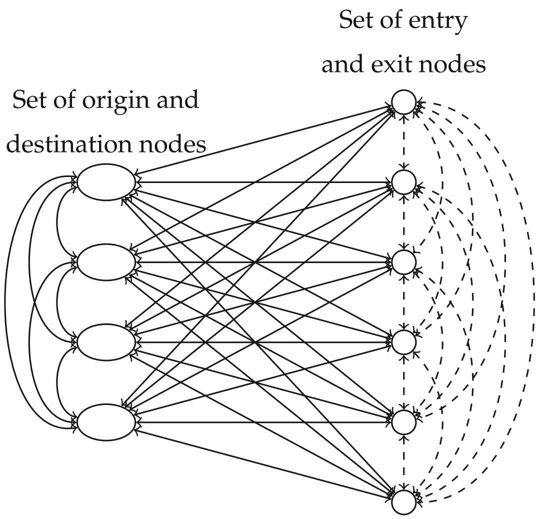
We observe that the path pricing problem always yields a leader's revenue greater or equal to the arc pricing problem, regardless of the network we consider.

5.2 The highway problem

One widely studied system with a polynomial number of paths is the highway system: it is characterized by a network whose connected toll arcs constitute a single path (as in the network in Fig. 12). This variant of the NPP has been considered in Heilporn et al. (2010b, 2011). If we make the assumption that users who have left the highway do not re-enter it, paths considered for toll can be uniquely determined by their entry and exit nodes. In consequence, for n nodes in the highway, the number of total paths will be n^2. Because of the completeness of the toll subgraph, this problem is also called the clique pricing problem (Heilporn et al. 2011).

Figure 13 depicts the so-called Complete Toll NPP, introduced in Heilporn et al. (2010b): toll free arcs are inserted between origin and destination nodes, as well as from the origin and destination nodes to the highway, representing shortest toll free paths between the corresponding nodes. Each pair of entry and exit nodes of the highway is linked by a toll subpath. Each toll subpath can be represented by a single artificial toll arc (dotted arcs in Fig. 13). In this a case additivity conditions are not considered, meaning that the toll of a path might not be equal to the sum of tolls on subpaths composing it. For each of these artificial toll arcs, representing toll subpaths p, and each commodity k with origin o^k and destination d^k, the fixed cost is denoted as c_p^k and is calculated as the sum of the cost of the shortest toll free path from o^k to the tail of the toll arc $t(p)$ and the cost of the shortest toll free path from the head of the toll arc $h(p)$ to d^k.

Fig. 13 Complete toll NPP (from Heilporn et al. 2010b)



Furthermore, computational experiments revealed that triangle inequalities and monotonicity constraints on the toll variables may not be satisfied by the optimal solution if they are not explicitly included. The guarantee of these conditions is important for real applications, as triangle inequalities prevent a commodity being able to pay less using two (subsequent) highway arcs instead of the direct one from the same origin and destination, and monotonicity constraints imply that the toll of a path cannot be smaller than the toll of any of its subpaths. Mathematically, triangle inequalities are expressed as follows:

$$T_p \leq T_q + T_s \quad \forall p, q, s \in \mathcal{P} : t(p) = t(q), h(q) = h(s), h(s) = h(p), \tag{11}$$

whereas monotonicity inequalities are represented by:

$$\begin{aligned} T_p \geq T_q \quad \forall p, q \in \ & \mathcal{P} : t(p) = t(q) < h(p) = h(q) + 1 \\ & \text{or } t(p) = t(q) - 1 < h(p) = h(q) \\ & \text{or } t(p) = t(q) > h(p) = h(q) - 1 \\ & \text{or } t(p) = t(q) + 1 > h(p) = h(q). \end{aligned} \tag{12}$$

The highway problem without triangle inequalities and monotonicity constraints is the path pricing introduced above. Both versions, with and without these constraints, have been proved to be strongly $\mathcal{NP}$-hard (Heilporn et al. 2010b). We will see from the numerical results in Sect. 5.5 that these constraints make the problem more difficult.

In the following subsections we present results for the highway or clique pricing problem which can be extended to the version with triangle inequalities and monotonicity constraints. For this last version further results can be found in Heilporn et al. (2010b, 2011).

5.3 One level MIP formulation

The path pricing problem (Eqs. (10a) to (10d)) can be reformulated as a single level optimization problem. First, the follower objective function can be separated for each commodity, and the lower level optimization problem can be replaced by constraints stating explicitly that the used path is the shortest one (13b). Moreover, the variables y^k, associated to the toll free paths, can be eliminated using constraint (10c). Finally, bilinear terms $T_p x_p^k$ can be replaced by variables T_p^k by adding constraints (13d) to (13g) and introducing "big M" and "big N" constants. Here we present only the last result of the reformulation, for more details we refer to Heilporn et al. (2010b), where the authors propose this MIP formulation of the path NPP:

$$\max_{T,x} \sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}_k} \eta^k T_p^k, \tag{13a}$$

$$\text{s.t.} \sum_{q\in\mathcal{P}_k} (c_q^k x_q^k + T_q^k) + c_{od}^k \Big(1 - \sum_{q\in\mathcal{P}_k} x_q^k\Big) \le c_p^k + T_p \quad \forall k \in \mathcal{K},\ \forall p \in \mathcal{P}_k, \tag{13b}$$

$$\sum_{p\in\mathcal{P}_k} x_p^k \le 1 \qquad \forall k \in \mathcal{K}, \tag{13c}$$

$$T_p^k \le M_p^k x_p^k \qquad \forall k \in \mathcal{K},\quad \forall p \in \mathcal{P}_k, \tag{13d}$$

$$T_p - T_p^k \le N_p(1 - x_p^k) \qquad \forall k \in \mathcal{K},\quad \forall p \in \mathcal{P}, \tag{13e}$$

$$T_p^k \le T_p \qquad \forall k \in \mathcal{K},\quad \forall p \in \mathcal{P}, \tag{13f}$$

$$T_p^k \ge 0 \qquad \forall k \in \mathcal{K},\quad \forall p \in \mathcal{P}_k, \tag{13g}$$

$$T_p \ge 0 \qquad \forall p \in \mathcal{P}, \tag{13h}$$

$$x_p^k \in \{0, 1\} \qquad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k. \tag{13i}$$

Similarly to arc pricing, constants M_p^k represent upper bounds on T_p^k variables, i.e. the highest value that commodity k is willing to pay for path p. Therefore, a valid value is $M_p^k = \max\{0, c_{od}^k - c_p^k\}$, $\forall k \in \mathcal{K}$ and $\forall p \in \mathcal{P}_k$. Constants N_p represent upper bounds on the cost of a path for all commodities, and a valid value is $N_p = \max_{k\in\mathcal{K}:p\in\mathcal{P}^k} M_p^k$, $\forall p \in \mathcal{P}$.

5.4 Valid inequalities

Here we describe a family of valid inequalities that amounts to a strengthening of constraints (13b) and has been introduced by Heilporn et al. (2010b). They are called "strengthened shortest path inequalities" (SSPI) and they can be written as:

$$\begin{aligned} \sum_{q\in\mathcal{P}_k} \left(c_q^k x_q^k + T_q^k\right) + c_{od}^k \left(1 - \sum_{q\in\mathcal{P}_k} x_q^k\right) &\le c_p^k + T_p \\ + \sum_{q\in\mathcal{P}_k\setminus(\mathcal{S}\cup\{p\})} \left(T_q^k + (c_q^k - c_p^k) x_q^k\right) &\qquad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k, \end{aligned} \tag{14}$$

where $\mathcal{S}$ is any subset of $\mathcal{P}_k$ (possibly the empty set).

To prove the validity we need to consider four possible cases:

1. If $x_q^k = 0, \forall q \in \mathcal{P}_k$, then $T_q^k = 0, \forall q \in \mathcal{P}_k$ by (13d) and the inequality becomes $c_{od}^k \leq T_p + c_p^k$, which is valid by (13b).
2. If $x_p^k = 1$, then $x_q^k = 0 = T_q^k, \forall q \in \mathcal{P}_k \setminus \{p\}$ by (13c) and (13b). The inequality becomes $T_p^k \leq T_p$, which is valid by (13f).
3. If $x_q^k = 1$ for $q \in \mathcal{P}_k \setminus (\mathcal{S} \cup \{p\})$, the inequality becomes $0 \leq T_p$, which is valid by (13h).
4. If $x_q^k = 1$ for $q \in \mathcal{S}$, then the inequality becomes $T_q^k + c_q^k \leq T_p + c_p^k$. As $x_q^k = 1$, $T_q^k = T_q$ and the path corresponding to q must be shorter than the path corresponding to p, i.e. the inequality is valid.

For any choice of the set $\mathcal{S}$ to be valid, there is an exponential number of inequalities (14), but Heilporn et al. (2010b) propose an efficient separation procedure with a complexity of $O(n \log n)$.

For the path NPP the single commodity case is polynomially solvable (Dewez 2004). Indeed, the toll path yielding the largest revenue for the leader, i.e. with the largest value of "big M" ($\max_{p \in \mathcal{P}} M_p$), can be found in polynomial time. The toll on this path p is set to M_p, whereas tolls on the other paths are set to sufficiently large values.

Heilporn et al. (2010b) conduct a study of the polyhedral structure for the single commodity case, showing that the formulation reported in Sect. 5.3 with the strengthened shortest path inqualities (14) completely describes the convex hull of solutions.

Furthermore, in the case of several commodities, Heilporn et al. (2011) exploit interactions between pairs of commodities to propose a further strengthening of the shortest path inequalities. These new families define facets of the convex hull of feasible solutions for the two-commodity case.

5.5 Computational results

In Table 4 we report numerical results taken from Heilporn et al. (2011). The path pricing has been tested on scenarios built around the network topology of the Canadian Highway 10 (autoroute des Cantons de l'Est, Québec). The instances have then been randomly generated: commodities are represented by pairs of cities chosen in a set of 5–8 ones (v cities therefore means $v(v-1)$ commodities), and toll paths by pairs of entry/exit nodes of the highway containing between 10 and 15 nodes (again n nodes means $n(n-1)$ toll paths). Five scenarios have been constructed for each size, with the demand for each city pairs randomly set between 10 and 100. Fixed costs on all arcs are also randomly generated, with the intention to provide realistic situations (for instance roads beside the highway are considered slower, so their fixed costs are multiplied by a factor of 1.5). For each commodity, toll paths with a higher fixed cost than the toll free path are clearly irrelevant, as they will never be the shortest path for any value of the tolls. They can therefore be removed reducing the size of the network. For each class of instances, in the "feasible paths" row of Table 4 we report the number of paths per commodity after this reduction, in the form $(Min, Max, Average)$. The problem has been solved for the path formulation presented in Sect. 5.3 (HP in the table),

Table 4 Numerical results for path pricing (from Heilporn et al. 2011)

		5 Cities			8 Cities	
		10 Nodes	12 Nodes	15 Nodes	10 Nodes	12 Nodes
Feasible paths		(1,64,22.6)	(1,69,29.7)	(1,140,41.9)	(1,63,17.6)	(1,92,28.3)
(HP)	Gap (%)	11.85	16.82	13.10	15.98	19.19
	Cpu	20	188	405	3,520	5,272
	Nodes	1,893	4,067	9,797	159,015	147,793
(HP)	Gap (%)	1.52	1.84	1.86	3.56	1.59
+SSPI	Cpu	21	52	241	3,038	1,313
	Nodes	388	103	719	9,722	3,974
(HP)	Gap (%)	18.10	20.13	19.52	30.09	32.04
+TMI	Cpu	2	18	5	262	947
	Nodes	184	1,131	407	27,991	263,875
(HP)	Gap (%)	6.53	3.98	4.55	12.64	9.96
+TMI	Cpu	4	10	8	682	1, 099
+SSPI	Nodes	69	127	101	9,205	23,473

and also with the strengthened shortest path inequalities (added at the root node and whenever they were violated) described in Sect. 5.4 (+SSPI in the table). Next, triangle inequalities and monotonicity constraints have been added to the formulation (+TMI in the table), with and without the SSPI.

The gap represents the difference in percent between the optimal solution Z_{opt} and the linear relaxation Z_{lp} : gap $= (Z_{lp} - Z_{opt})/Z_{opt}$. The "cpu" row gives the cpu times in seconds and the "nodes" row represents the number of nodes in the branch-and-bound tree. Table 4 reports average values for these elements. The computational time upper bound has been set to 5 h (18,000 s).

The models have been implemented with Mosel language of Xpress-MP Optimizer version 18, run with default values on a Pentium 4.3 GHz processor with 2 Gb of RAM and Linux Kernel.

Results show that the shortest path inequalities have a big impact on the gap and on the number of nodes, supporting the intuition of their efficiency as they are facet defining for the single commodity case. For all but the smallest instances, we observe also a sharp drop in the cpu time. The addition of triangle inequalities and monotonicity constraints to the formulation increases the complexity of the problem, and also for this version of the problem the shortest path inequalities prove to be very efficient.

6 Product pricing

A parallel between the pricing of products and of arcs of a network has been considered by Heilporn et al. (2010a). The product pricing problem considers a company producing and pricing a set of products (singularly or in bundles) and aiming to maximize its revenue, and customers aiming to maximize their total utility or minimize their costs when buying. This problem, which has been widely studied in economics, includes different versions regarding the objective function, constraints and variables used to model different situations.

If we consider the pricing of substitutable products and of arcs of a highway network (with a polynomial number of paths), we can establish a clear parallel: both problems involve revenue maximization for the company and utility maximization for the customers or cost minimization for travelers. More specifically, when modeling these problems, customers correspond to commodities, products to toll arcs, reservation prices (the highest price a buyer is willing to pay for a product) to toll windows (the difference between the toll free path cost and the fixed cost for a toll path, as the space left for tolling), prices to tolls, and product assignment variables to arc flow variables, respectively for product and network pricing. The assignment of a commodity to a toll free arc corresponds to no purchase, which means that the customer is not buying the product (or buying it from a competitor).

In the context of product pricing, we define $\mathcal{K}$ as the set of customers k and $\mathcal{I}$ as the set of products i sold by the company. Product od is added to this set and represents the substitutable product sold by competitors, as a "toll free" alternative. The demand of customer k is then expressed as η^k, and the reservation price of customer k for product i as r_i^k. Decision variables are x_i^k, equal to 1 if customer k is buying product i and 0 otherwise. The price set by the company on product i is represented by π_i. The product pricing problem can therefore be modeled as follows (Heilporn et al. 2010a):

$$\max_{\pi,x} \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} \eta^k \pi_i x_i^k, \tag{15a}$$

$$s.t. \sum_{j\in\mathcal{I}\cup\{od\}} (r_j^k - \pi_j)x_j^k \geq r_i^k - \pi_i \qquad \forall k \in \mathcal{K}, \forall i \in \mathcal{I}, \tag{15b}$$

$$\sum_{i\in\mathcal{I}\cup\{od\}} x_i^k = 1 \qquad \forall k \in \mathcal{K}, \tag{15c}$$

$$\pi_{od} = 0, \tag{15d}$$

$$\pi_i \geq 0 \qquad \forall i \in \mathcal{I}, \tag{15e}$$

$$x_i^k \in \{0,1\} \qquad \forall k \in \mathcal{K}, \forall i \in \mathcal{I}. \tag{15f}$$

Constraints (15b) ensure that each customer is buying the product with the largest utility, as constraints (13b) guarantee the use of the shortest path for each commodity in the highway problem.

This model is analogous to the highway model presented above and can be linearized as described in Sect. 5.3. Valid inequalities introduced in Sect. 5.4 also apply.

Moreover, the triangle and monotonicity constraints introduced for the network problem also make sense for product pricing, when products represent different quantities of the same goods. For quantities X, Y, Z such that $X = Y + Z$, market consistency requires that the triangle inequality on prices holds ($\pi_X \leq \pi_Y + \pi_Z$), and similarly, if $X \leq Y$, one would expect that $\pi_X \leq \pi_Y$ (monotonicity).

Heilporn et al. (2011) test this formulation on a set of product pricing instances proposed by Shioda et al. (2011). Numerical results show that this model clearly outperforms the formulation proposed by Shioda et al. (2011), with or even without the valid inequalities.

7 Conclusion

The bilevel optimization paradigm provides a rich framework for pricing goods and services, in network-based industries among others.

Such models are, both theoretically and computationally challenging and it is very important to exploit the inner structure for each particular problem in order to obtain efficient solution methods.

In this tutorial paper, we provided a detailed analysis of the basic models relevant and useful for attacking real applications, and which may be a starting point for the development of ad-hoc methodologies.

There remains a whole range of interesting and open questions to be addressed, such as how to integrate real-life features into bilevel models (e.g. congestion, market segmentation, dynamics, randomness) or how to extend those models to tackle variants of product pricing (e.g. pricing together bundles of different products rather than single units).

Acknowledgments The first author acknowledges support from the "Ministerio de Ciencia e Innovacíon" through the research project MTM2009-14039-C06. The second author acknowledges support from the Belgian national scientific funding agency "Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture" (FRIA), of which she is a research fellow.

References

Amaldi E, Bruglieri M, Fortz B (2011) On the hazmat transport network design problem. In: Pahl J, Reiners T, Voß S (eds) Network optimization, lecture notes in computer science, vol 6701. Springer, Berlin, pp 327–338

Bouhtou M, Van Hoesel S, Van der Kraaij A, Lutton J (2007a) Tariff optimization in networks. INFORMS J Comput 19(3):458–469

Bouhtou M, Grigoriev A, Van Hoesel S, Van der Kraaij A, Spieksma F, Uetz M (2007b) Pricing bridges to cross a river. Naval Res Logist 54:411–420

Bracken J, McGill J (1973) Mathematical programs with optimization problems in the constraints. Oper Res 21(1):37–44

Brotcorne L, Labbé M, Marcotte P, Savard G (2000) A bilevel model and solution algorithm for a freight tariff-setting problem. Transp Sci 34(3):289–302

Brotcorne L, Labbé M, Marcotte P, Savard G (2001) A bilevel model for toll optimization on a multicommodity transportation network. Transp Sci 35(4):345–358

Brotcorne L, Labbé M, Marcotte P, Savard G (2008) Joint design and pricing on a network. Oper Res 56(5):1104–1115

Candler W, Norton R (1977) Multilevel programming. Technical report 20, World Bank Development Research Center, Washington, DC

Cardinal J, Demaine E, Fiorini S, Joret G, Langerman S, Newman I, Weimann O (2011) The stackelberg minimum spanning tree game. Algorithmica 59:129–144

Castelli L, Labbé M, Violin A (2012) A network pricing formulation for the revenue maximization of european air navigation service providers. Transp Res Part C. doi:10.1016/j.trc.2012.04.013

Colson B, Marcotte P, Savard G (2005) Bilevel programming: a survey. 4OR. Q J Oper Res 3:87–105

Colson B, Marcotte P, Savard G (2007) An overview of bilevel optimization. Ann Oper Res 153:235–256

Dempe S (2002) Foundations of bilevel programming, nonconvex optimization and its applications, vol 61. Kluwer, Dordrecht

Dewez S (2004) On the toll setting problem. PhD thesis, Université Libre de Bruxelles, Brussles
Dewez S, Labbé M, Marcotte P, Savard G (2008) New formulations and valid inequalities for a bilevel pricing problem. Oper Res Lett 36(2):141–149
Garey M, Johnson D (1979) Computers and interactability. W.H. Freeman, San Francisco
Hansen P, Jaumard B, Savard G (1992) A new branch-and-bound rules for linear bilevel programming. SIAM J Sci Stat Comput 5(13):1194–1217
Heilporn G, Labbé M, Marcotte P, Savard G (2010a) A parallel between two classes of pricing problems in transportation and marketing. J Rev Pricing Manage 9(1/2):110–125
Heilporn G, Labbé M, Marcotte P, Savard G (2010b) A polyhedral study of the network pricing problem with connected toll arcs. Networks 3(55):234–246
Heilporn G, Labbé M, Marcotte P, Savard G (2011) Valid inequalities and branch-and-cut for the clique pricing problem. Discret Optimiz 8(3):393–410
Jeroslow R (1985) The polynomial hierarchy and a simple model for competitive analysis. Math Program 32:146–164
Joret G (2011) Stackelberg network pricing is hard to approximate. Networks 57(2):117–120
Labbé M, Marcotte P, Savard G (1998) A bilevel model of taxation and its application to optimal highway pricing. Manage Sci 44(12):1608–1622
Loridan P, Morgan J (1996) Weak via strong stackelberg problem: new results. J Glob Optimiz 8:263–287
Migdalas A (1995) Bilevel programming in traffic planning: models, methods and challenge. J Glob Optimiz 7:381–405
Owen G (1968) Game theory. Emerald Group, Bingley
Roch S, Marcotte P, Savard G (2005) Design and analysis of an approximation algorithm for stackelberg network pricing. Networks 46(1):57–67
Shioda R, Tunçel L, Myklebust T (2011) Maximum utility product pricing models and algorithms based on reservation price. Comput Optimiz Appl 48:157–198
Stackelberg H (1952) The theory of market economy. Oxford University Press, Oxford
Van Ackere A (1993) The principal/agent paradigm: its relevance to various functional fields. Eur J Oper Res 70(1):83–103
Van Hoesel S (2008) An overview of stackelberg pricing in networks. Eur J Oper Res 189:1393–1402
Vicente L, Calamai P (1994) Bilevel and multilevel programming: a bibliography review. J Glob Optimiz 5:291–306
Vicente L, Savard G, Júdice J (1994) Descent approaches for quadratic bilevel programming. J Optimiz Theory Appl 81(2):379–399