The Network Pricing Problem with congestion

Desirée Rigonat[a]*, Lorenzo Castelli[a], Martine Labbé[b]

[a] Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, Italy

[b]Départment d'Informatique, Université Libre de Bruxelles, Belgium

August 15, 2014

Abstract

The Network Pricing Problem (NPP) is a bilevel network optimisation problem where prices have to be set on the arcs of a network in order to maximise the profit of the arcs' owner. The bilevel structure of the NPP implies that these prices will be influenced by the distribution on the network of one or more users that want to travel on it at the minimum cost. The NPP usually assumes that arc costs are independent of flows. When arc costs do depend on flows, the network is usually referred to as *congested*. In the present work, we illustrate two asymptotically converging algorithms to solve the NPP in the case of congested networks, hereafter referred to as the Congested Network Pricing Problem (CNPP). In particular, we propose to identify an equilibrium point for the CNPP using the Frank-Wolfe algorithm by reformulating the bilevel CNPP into a sequence of approximating single level linear problems. One of the algorithms uses these linear approximations to solve only the second level problem (that is, the problem of the users) while the other applies the linearisation procedure to the whole CNPP.

1 Introduction

Bilevel programs belong to a class of Stackelberg sequential games with two players, where a *leader* plays first, taking into account the possible reactions of the second player, called the *follower*. Bilevel programs first appeared in an article by Bracken & McGill (1973), while the complete formulation was first introduced by Shimizu & Aiyoshi (1981). An annotated bibliography containing more than one hundred references on bilevel programming has been compiled by Vicente & Calamai (1994), while the books by Shimizu et al. (1997) and Luo et al. (1996) are devoted, in full or in part, to this subject. Generically non differentiable and non convex, bilevel problems are, by nature, hard. Even the linear bilevel problem, where the objective functions and the constraints are linear, was proved to be $\mathcal{NP}$-hard by Jeroslow (1985). Hansen et al. (1992) prove strong $\mathcal{NP}$-hardness. Vicente et al. (1994) strengthen these results and prove that merely checking strict or local optimality is strongly $\mathcal{NP}$-hard.

The Network Pricing Problem (NPP) describes the interaction between the owner of a subset of a network (i.e. a highway system) who wants to maximise the profit from his business by imposing tolls on his arcs and the users of the network (i.e. people who want to travel on it) who want to minimise the cost for using the network. As demonstrated by Labbé et al. (1998) this problem in its general form is strongly $\mathcal{NP}$-hard. However a linearisation scheme illustrated in the same work leads to a mixed integer programming formulation that involves a small number of binary variables. This formulation can be solved using standard algorithms such as branch and bound that will lead to an exact solution in reasonable time for small instances of the problem, or allows for efficient heuristic procedures to be developed, as shown by Brotcorne et al. (2000, 2001). Such algorithms lead to far better results on large network since they are able to exploit the particular structure of the network in order to quicken the solving procedure. Heilporn (2008) in her Ph.D. thesis gives a detailed analysis of the geometric structure of the problem and the particular cases that have so far been proved to be easier to solve. A recent comprehensive survey on the NPP is available in Labbé & Violin (2013).

NPP usually assumes that arc costs are independent of flows. The present work instead introduces a variant to NPP where arc costs do depend on flows. We will then refer to this problem as the Congested Network Pricing Prooblem (CNPP). We show that by taking congestion into account, the first level problem remains a profit maximisation problem, while the second level problem becomes a user equilibrium problem known as *Traffic Assignment Problem*.

*Corresponding author. Ph.D. candidate

The Traffic Assignment Problem has been covered by a vast amount of literature over the past decades and many resolutive algorithms have been developed for it. The most widely known is probably the Frank-Wolfe (FW) algorithm (Frank & Wolfe 1956), also known as Conditional Gradient method (Bertsekas 1995), which is a generic algorithm for solving convex problems with linear constraints. It has been the preferred algorithm for Traffic Assignment resolution for decades, mostly because of its ease of implementation and low memory consumption, despite the drawback of a slow convergence rate when approaching the optimal solution. Even though over the years several more efficient algorithms were developed specifically for solving the Traffic Assignment Problem, such as column generation (Larsson & Patriksson 1992, Damberg et al. 1996), gradient projection (Jayakrishnan et al. 1994), and origin/destination based methods (Bar-Gera 2002, Boyce et al. 2004, Bar-Gera & Boyce 2006, Pro 2009, Gentile & Noekel 2009), this work relies on a FW approach since FW is still the reference algorithm for performance comparison for all recently proposed algorithms. The FW linearisation scheme was also used by Brotcorne et al. (2001) in the design of a primal-dual heuristic to solve the NPP. In their work however arc costs are still supposed to be independent of flows.

In particular, in this work we propose and test two heuristic algorithms for the CNPP resolution. The first algorithm, called *One Step algorithm*, applies a Frank-Wolfe-like procedure to the whole CNPP, thus generating a succession of linear approximations through which both flows and tolls are obtained at the same time, while the objective still aims to converge to a user equilibrium. The second, which we call *Two-Steps algorithm*, is a Gauss-Seidel-decomposition-type algorithm. It solves the CNPP by iteratively solving first the followers' problem (through Frank-Wolfe) with respect to arc flows and then the leader's problem with respect to the toll vector. A similar algorithm was used by Julsain (1998) to find an equilibrium point on a congested telecommunication network.

The present work is structured as follows. Section 2 introduces the original uncongested Network Pricing Problem, whereas Section 3 presents the bilevel bilinear formulation of the Congested NPP, together with the description of the procedure to determine a single level mixed-integer non-linear formulation. In section 4 two heuristic algorithms for the CNPP are illustrated. Preliminary experimental results are provided in Section 5. Finally, in Section 6 we draw some conclusions.

2 The Network Pricing Problem

As described in Labbé et al. (1998), in the *Network Pricing Problem* (NPP) the context is that of a transport network (e.g. a road network), where the leader is the owner of a subset of the network (e.g. the highways), who wants to maximise his or her profit by imposing tolls on the arcs he owns, while the followers are the users of the network, that is, the people who want to travel on it with the minimum travel cost. We assume that for every user there always exists at least one alternative route that does not pass through any of the arcs owned by the leader (a *toll-free path*). This is to avoid the situation where a user has no alternatives but to travel on one of the tolled arcs, thus allowing the leader to impose an arbitrarily high toll on it. Furthermore, as in Labbé et al. (1998), we assume a favourable scenario for the leader, that is, when a user is faced with two alternative routes of equal cost, he always chooses the one that gives the higher profit for the leader.

Notation

- $G = (\mathcal{N}, \mathcal{A} \cup \mathcal{B})$ defines a transport network where $\mathcal{N}$ denotes the set of nodes and $\mathcal{A} \cup \mathcal{B}$ is the set of arcs; $\mathcal{A}$ is the subset of tolled arcs and $\mathcal{B}$ the subset of toll-free arcs.
- Each arc of $\mathcal{A}$ has a travel cost composed of a fixed part c_a and an unknown toll t_a. Vector t whose components are $t_a, a \in \mathcal{A}$ represents leader's decision variables.
- Each arc of $\mathcal{B}$ bears only a fixed travel cost, identified by d_a.
- $\mathcal{K}$ denotes the set of commodities, where each commodity k is associated with an origin/destination pair (o^k, d^k);
- b^k is defined as:

$$b_i^k = \begin{cases} 1 & \text{if } i = o^k \\ -1 & \text{if } i = d^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{1}$$

- η^k represents the number of users of commodity k (demand vector).
- x_a^k represents follower's decision variables and denotes whether arc $a \in \mathcal{A} \cup \mathcal{B}$ has been chosen to connect the origin/destination pair (o^k, d^k) or not.

The NPP can thus be formulated as a bilevel program with bilinear objective functions and linear constraints, where the variables x_a^k denote the optimal solution of the second level problem parametrised by the upper level toll vector t.

$$\max_{t,x} \sum_{k\in\mathcal{K}} \sum_{a\in\mathcal{A}} t_a \cdot x_a^k \cdot \eta^k \tag{2}$$

$$\text{s.t.} \min_{x} \sum_{k\in\mathcal{K}} (\sum_{a\in\mathcal{A}} (c_a + t_a) \cdot x_a^k + \sum_{a\in\mathcal{B}} d_a \cdot x_a^k) \tag{3}$$

$$\sum_{a\in i^-\cap A} x_a^k + \sum_{a\in i^-\cap B} x_a^k - \sum_{a\in i^+\cap A} x_a^k - \sum_{a\in i^+\cap B} x_a^k = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{4}$$

$$0 \le x_a^k \le 1 \qquad \forall k \in \mathcal{K}, \forall a \in \mathcal{A}\cup\mathcal{B} \tag{5}$$

where $i^- \in \mathcal{N}$ and $i^+ \in \mathcal{N}$ are respectively the entering and exiting arcs for node $i \in \mathcal{N}$.

The structure of the follower's problem is that of a *shortest path problem*, a well known optimisation problem for which many efficient solving algorithms exist (one is the well known *Dijkstra's algorithm*). Note that no constraint is given on the toll vector t. As discussed in Labbé et al. (1998) negative tolls can be seen as incentives that the leader offers to the users to gain more traffic on his arcs. In the same work the authors illustrate that, on certain network configurations, allowing for negative tolls ultimately increases the final revenue for the leader.

3 The NPP with Congestion

In this section we will describe a variant to NPP, which takes congestion levels into account. We will refer to this as the Congested Network Pricing Problem (CNPP). The peculiarity of considering congestion lays in the fact that arc costs are no longer considered as constants. Instead, they depend upon arc flows: they get higher as flow levels approach the capacity of the arcs (which are fixed) and get lower as flow is moved to other arcs. This implies that where the NPP had fixed arc costs c_a and d_a, the CNPP has *arc cost functions* and here lies the difficulty of the model. The problem of user travel cost minimisation in this context becomes what is referred to in literature as *User Equilibrium*, a condition under which no individual is able to reduce their costs by choosing an alternative route over the network (as stated by Wardrop's first principle, Wardrop (1952)).

Our particular case deals with a road transport network, such as a highway system, so that the first level problem remains a profit maximisation problem and the second level problem becomes a *Traffic Assignment Problem.*

3.1 Cost functions

Cost functions express the cost of a path or arc based on the performance of the network. Assuming separable functions and the absence of non-additive path costs, we define a generic arc cost function such that each arc of the network has a cost that is a function only of the flow on the arc itself (originally proposed in Wardrop (1952)):

$$\mathcal{C}_a(f_a) = tr_{0a} \cdot \left[1 + \alpha \cdot \left(\frac{f_a}{q_a}\right)^{\beta}\right] \quad \forall a \in \mathcal{A}\cup\mathcal{B} \tag{6}$$

where

$$f_a = \sum_{k\in K} x_a^k \cdot \eta^k \quad \forall a \in \mathcal{A}\cup\mathcal{B}, \tag{7}$$

and $\forall a \in \mathcal{A}\cup\mathcal{B}$, tr_{0a} is the travel time in unconstrained conditions (i.e. optimal traffic and weather conditions), q_a is the capacity of the arc, and α and β are parameters that have to be calibrated. Note that for every arc $a \in \mathcal{A}$ we will also have to consider the toll t_a; such toll is added to the cost $\mathcal{C}_a$ resulting in the total arc-cost $\mathcal{C}_a(f_a) + t_a$.

The function defined by Equation (6) is not linear, continuous, strictly positive and strictly increasing.

3.2 Second level of the CNPP: traffic assignment

Introducing arc-cost functions in the followers' problem leads to a formulation known as *minimum cost multi-commodity flow convex problem* (LeBlanc et al. (1975)) or *Traffic Assignment problem* (Petersen (1975), Patriksson (1994)):

$$\min_x \left\{ \sum_{a\in A} \int_0^{f_a} (\mathcal{C}_a(\omega_a) + t_a) d\omega_a + \sum_{a\in B} \int_0^{f_a} \mathcal{C}_a(\omega_a) d\omega_a \right\} \tag{8}$$

$$\text{s.t.} \sum_{a\in i^-\cap A} x_a^k + \sum_{a\in i^-\cap B} x_a^k - \sum_{a\in i^+\cap A} x_a^k - \sum_{a\in i^+\cap B} x_a^k = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{9}$$

$$f_a = \sum_{k\in K} x_a^k \cdot \eta^k \qquad \forall a \in \mathcal{A}\cup\mathcal{B} \tag{10}$$

$$0 \le x_a^k \le 1 \qquad \forall a \in \mathcal{A}\cup\mathcal{B}, \forall k \in \mathcal{K} \tag{11}$$

where $i^- \in \mathcal{N}$ and $i^+ \in \mathcal{N}$ are respectively the entering and exiting arcs for node $i \in \mathcal{N}$. Equation (8) expresses the objective function of the problem, which is nonlinear because of the cost function adopted. Equation (9) imposes a number of linear constraints which is equal to the number of network nodes in order to express, together with Equations (10 and (11), conservation of flows at the nodes.

3.3 First level of the CNPP: leader profits

The first level of the CNPP deals with how the *leader* maximises his or her profit by imposing fees on the toll arcs and dependently on the distribution of the *followers* on the network at the equilibrium.

For the uncongested NPP the objective of the leader's problem is the following:

$$\max_{t,x} \sum_{k\in K} \sum_{a\in A\cup B} t_a \cdot x_a^k \cdot \eta^k \tag{12}$$

This is a non-linear objective function, since it contains a bilinear term. Since only arc flows are involved in the leader's objective, and no arc costs, the leader's objective for the CNPP is the same of the NPP, the only meaningful difference being the following non-negativity constraint on the toll vector, which has to be imposed to ensure that the follower's objective remains strictly non-decreasing and positive. This is necessary to ensure convergence of the solving algorithms (see Sheffi (1985) and Bertsekas (1995)):

$$t_a \ge 0 \quad \forall a \in \mathcal{A} \tag{13}$$

The resulting bilevel problem is thus the following:

$$\max_{t,x} \sum_{k\in K} \sum_{a\in A\cup B} t_a \cdot x_a^k \cdot \eta^k \tag{14}$$

$$\min_x \left\{ \sum_{a\in A} \int_0^{f_a} (\mathcal{C}_a(\omega_a) + t_a) d\omega_a + \sum_{a\in B} \int_0^{f_a} \mathcal{C}_a(\omega_a) d\omega_a \right\} \tag{15}$$

$$\text{s.t.} \sum_{a\in i^-\cap A} x_a^k + \sum_{a\in i^-\cap B} x_a^k - \sum_{a\in i^+\cap A} x_a^k - \sum_{a\in i^+\cap B} x_a^k = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{16}$$

$$f_a = \sum_{k\in K} x_a^k \cdot \eta^k \qquad \forall a \in \mathcal{A}\cup\mathcal{B} \tag{17}$$

$$0 \le x_a^k \le 1 \qquad \forall a \in \mathcal{A}\cup\mathcal{B}, \forall k \in \mathcal{K} \tag{18}$$

$$t_a \ge 0 \qquad \forall a \in \mathcal{A} \tag{19}$$

With cost/flow interdependence as expressed by Equation (6) and vector b^k defined by Equation (1).

3.4 From bilevel CNPP to single level CNPP

Similarly to the procedure described in Labbé et al. (1998)) for the uncongested NPP, it is possible to reformulate the bilevel CNPP into a single level problem by exploiting *strong duality*, and more specifically by replacing the second level (followers') problem with its *Karush Kuhn Tucker Conditions (KKT)*. This is a legitimate operation, assuming that objective and constraints are $\in C^1$ and have the characteristics described in Section 3.2 (see Sheffi (1985) and Bertsekas (1995)). Be such the case, a vector $\overline{x}$ that satisfies the KKT conditions, is known to be optimal for the generic convex program:

$$\min f(x) \tag{20}$$
$$\text{s.t. } h(x) = 0 \tag{21}$$
$$g(x) \geq 0 \tag{22}$$

where $f(x)$ and $h(x)$ are convex and $g(x)$ are linear. Its Lagrangian is defined as:

$$L(x, \lambda, \mu) = f(x) + \lambda^T h(x) - \mu^T g(x) \tag{23}$$

At a minimum point $\overline{x}$, we have $\nabla L(\overline{x}, \lambda, \mu) = 0$ which implies that the following conditions hold:

$$\nabla_x L : \nabla f(\overline{x}) + \lambda^T \nabla h(\overline{x}) - \mu^T \nabla g(\overline{x}) = 0 \tag{24}$$
$$\nabla_\lambda L : h(\overline{x}) = 0 \tag{25}$$
$$\nabla_\mu L : g(\overline{x}) \geq 0 \tag{26}$$

In addition, the following complementarity condition must hold:

$$\mu^T g(\overline{x}) = 0 \tag{27}$$

In a typical instance of a *Traffic Assignment* problem, which is our second level problem, the corresponding constraints are:

$$h(\overline{x}) = 0 \Rightarrow \sum_{a \in i^- \cap A} \overline{x}_a^k + \sum_{a \in i^- \cap B} \overline{x}_a^k - \sum_{a \in i^+ \cap A} \overline{x}_a^k - \sum_{a \in i^+ \cap B} \overline{x}_a^k = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{28}$$
$$g(\overline{x}) \geq 0 \Rightarrow 0 \leq \overline{x}_a^k \leq 1 \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{29}$$

Consequently, the partial derivatives are:

$$\nabla_x L : \mathcal{C}_a(\overline{f}_a) + t_a + \lambda_i^k - \lambda_j^k - \mu_a^k = 0 \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{30}$$
$$\text{where } \overline{f}_a = \sum_{k \in K} \overline{x}_a^k \cdot \eta^k \qquad \forall a \in \mathcal{A} \cup \mathcal{B} \tag{31}$$
$$\nabla_\lambda L : \sum_{a \in i^- \cap A} \overline{x}_a^k + \sum_{a \in i^- \cap B} \overline{x}_a^k - \sum_{a \in i^+ \cap A} \overline{x}_a^k - \sum_{a \in i^+ \cap B} \overline{x}_a^k = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{32}$$
$$\nabla_\mu L : 0 \leq \overline{x}_a^k \leq 1 \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K}, \tag{33}$$

and complementarity condition is:

$$\mu_a^T \overline{x}_a^k = 0. \tag{34}$$

Hence, we obtain the following one-level reformulation for the CNPP:

$$\max_{t,x} \sum_{a,k} t_a \cdot x_a^k \cdot \eta^k \tag{35}$$
$$\text{s.t. } \sum_{a \in i^- \cap A} x_a^k + \sum_{a \in i^- \cap B} x_a^k - \sum_{a \in i^+ \cap A} x_a^k - \sum_{a \in i^+ \cap B} x_a^k = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{36}$$
$$\mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k - \mu_a^k = 0 \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{37}$$
$$\mu_a^k \cdot x_a^k = 0 \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{38}$$
$$f_a = \sum_{k \in K} x_a^k \cdot \eta^k \qquad \forall a \in \mathcal{A} \cup \mathcal{B} \tag{39}$$
$$0 \leq x_a^k \leq 1 \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{40}$$
$$\mu_a^k \geq 0 \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{41}$$
$$\lambda_i^k \quad \text{free} \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{42}$$

where $\mathcal{C}_a(f_a)$ is the derivative of the cost function expressed by Equation (6).

3.5 From bilinear CNPP to Mixed Integer non-linear CNPP

The problem obtained so far still contains bilinear terms in both the objective and the complementary constraint. What we want to obtain in this section is a single level non-linear problem with a mixed-integer formulation.

3.5.1 Simplification of the objective ($t_a \cdot x_a^k$ term)

In the present section we substitute the $t_a \cdot x_a^k$ term in the objective with terms derived from the equality constraints of the problem in order to obtain an equivalent formulation that will be non-linear in only one variable.

From the first KKT condition (Equation 37) we have that:

$$\mu_a^k = \mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k \qquad \forall a = (i,j) \in \mathcal{A}, \forall k \in \mathcal{K} \tag{43}$$

$$\mu_a^k = \mathcal{C}_a(f_a) + \lambda_i^k - \lambda_j^k \qquad \forall a = (i,j) \in \mathcal{B}, \forall k \in \mathcal{K} \tag{44}$$

Thus the second KKT condition (Equation 38) can be formulated as:

$$(\mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k) \cdot x_a^k = 0 \qquad \forall a = (i,j) \in \mathcal{A}, \forall k \in \mathcal{K} \tag{45}$$

$$(\mathcal{C}_a(f_a) + \lambda_i^k - \lambda_j^k) \cdot x_a^k = 0 \qquad \forall a = (i,j) \in \mathcal{B}, \forall k \in \mathcal{K} \tag{46}$$

by substituting μ_a^k with Equations (43) and (44). Hence,

$$t_a \cdot x_a^k = -\mathcal{C}_a(f_a) \cdot x_a^k + (\lambda_j^k - \lambda_i^k) \cdot x_a^k \quad \forall a = (i,j) \in \mathcal{A}, \forall k \in \mathcal{K} \tag{47}$$

So the objective (Equation 35) becomes:

$$\max_{\lambda,x} \sum_{a \in A} \sum_{k \in K} -(\mathcal{C}_a(f_a) \cdot x_a^k + (\lambda_j^k - \lambda_i^k) \cdot x_a^k) \cdot \eta^k \tag{48}$$

To eliminate the bilinear term $(\lambda_j^k - \lambda_i^k) \cdot x_a^k$, we notice that the following equality holds:

$$\sum_{a \in A} \sum_{k \in K} (\lambda_j^k - \lambda_i^k) \cdot x_a^k = \sum_{k \in K} \sum_{i \in N} \lambda_i^k \cdot (\sum_{a = i^- \in A} x_a^k - \sum_{a = i^+ \in A} x_a^k) \tag{49}$$

From the flow conservation constraint (Equation 36) we have that

$$\sum_{a = i^- \cap A} x_a^k - \sum_{a = i^+ \cap A} x_a^k = b_i^k - \sum_{a = i^- \cap B} x_a^k + \sum_{a = i^+ \cap B} x_a^k \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}. \tag{50}$$

Hence from Equations (49) and (50) it follows that:

$$\sum_{k \in K} \sum_{i \in N} \lambda_i^k \cdot (\sum_{a = i^- \cap A} x_a^k - \sum_{a = i^+ \cap A} x_a^k) = \tag{51}$$

$$\sum_{k \in K} \sum_{i \in N} \lambda_i^k \cdot b_i^k - \sum_{k \in K} \sum_{i \in N} \lambda_i^k \cdot (\sum_{a = i^- \cap B} x_a^k + \sum_{a = i^+ \cap B} x_a^k) \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{52}$$

Finally, since Equation (46) states that:

$$\mathcal{C}_a(f_a) \cdot x_a^k = (\lambda_j^k - \lambda_i^k) \cdot x_a^k \quad \forall a = (i,j) \in \mathcal{B}, \forall k \in \mathcal{K}, \tag{53}$$

we can thus substitute the $\sum_{a \in A} \sum_{k \in K} (\lambda_j^k - \lambda_i^k) \cdot x_a^k$ term in the objective function, which becomes:

$$\max_{\lambda,x} \sum_{k \in K} \left(\sum_{i \in N} \lambda_i^k \cdot b_i^k - \sum_{a \in A} \mathcal{C}_a(f_a) \cdot x_a^k - \sum_{b \in B} \mathcal{C}_a(f_a) \cdot x_a^k \right) \cdot \eta^k. \tag{54}$$

3.5.2 Linearisation of the $\mu_a^k \cdot x_a^k$ term

The problem obtained so far still contains the nonlinear constraint $\mu_a^k \cdot x_a^k = 0$ deriving from the complementarity slackness condition. We linearise it by introducing a binary variable z_a^k, defined as:

$$z_a^k = \begin{cases} 1 & \text{if} \quad \mu_a^k \neq 0 \\ 0 & \text{if} \quad \mu_a^k = 0 \end{cases} \tag{55}$$

Since we need to impose that μ_a^k is zero if $x_a^k \neq 0$ and vice versa, we can write the KKT conditions as,

$$\mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k \leq M \cdot z_a^k \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{56}$$

$$x_a^k \leq M \cdot (1 - z_a^k) \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K}, \tag{57}$$

where M is an arbitrary *Big-M* constant. However, for the problem not to be unbounded it also has to hold that:

$$\mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k \geq 0 \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{58}$$

$$x_a^k \geq 0 \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K}. \tag{59}$$

Thus the final one-level mixed-integer non-linear reformulation for the CNPP is:

$$\max_{\lambda,x} \sum_{k \in K} \Big(\sum_{i \in N} \lambda_i^k \cdot b_i^k - \sum_{a \in A} \mathcal{C}_a(f_a) \cdot x_a^k - \sum_{b \in B} \mathcal{C}_a(f_a) \cdot x_a^k \Big) \cdot \eta^k \tag{60}$$

$$\text{s.t.} \sum_{a \in i^- \cap A} x_a^k + \sum_{a \in i^- \cap B} x_a^k - \sum_{a \in i^+ \cap A} x_a^k - \sum_{a \in i^+ \cap B} x_a^k = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{61}$$

$$\mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k \leq M \cdot z_a^k \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{62}$$

$$\mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k \geq 0 \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{63}$$

$$x_a^k \leq M \cdot (1 - z_a^k) \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{64}$$

$$f_a = \sum_{k \in K} x_a^k \cdot \eta^k \qquad \forall a \in \mathcal{A} \cup \mathcal{B} \tag{65}$$

$$0 \leq x_a^k \leq 1 \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{66}$$

$$z_a^k \in \{0;1\} \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{67}$$

$$t_a \geq 0 \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{68}$$

$$\lambda_i^k \quad \text{free} \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{69}$$

3.6 Complexity of the CNPP

The complexity of the CNPP, as formulated above, depends on the chosen cost function. In particular, for the cost function introduced in Section 3.1, this complexity relies on the choice of the positive real parameters α and β. We can distinguish three situations that can occur and that are of some interest:

1. $\alpha = 0$

   In this case the congestion-dependent term $\alpha \cdot \left(\frac{f_a}{q_a}\right)^\beta$ equals zero, and the CNPP instance becomes a standard NPP one. In fact, while the constraints and leader's objective face no change, the follower's objective becomes:

$$\min_x \sum_{a \in A \cup B} \int_0^{f_a} (\mathcal{C}_a(\omega_a) + t_a) d\omega_a = \tag{70}$$

$$= \min_x \sum_{a \in A \cup B} \int_0^{f_a} (tr_{0a} + t_a) d\omega_a = \tag{71}$$

$$= \min_x \sum_{a \in A \cup B} (tr_{0a} + t_a) \cdot x_a \tag{72}$$

   This leads to a formulation that is identical to the one illustrated in Section 2 for the uncongested NPP. The case $\beta = 0$ is similar and does not offer any interesting insight.

2. $\alpha > 0$ and $\beta = 1$

   In this case, the CNPP reduces to a quadratic problem (note that the constraints are all linear).

3. $\alpha > 0$ and $\beta > 1$

   In this case, the CNPP is heavily non linear and we can expect it to be much harder to solve than the uncongested NPP. Several solving approaches are possible, such as penalising the nonlinear constraint in the objective (thus obtaining a convex polyhedron as feasible region) or designing heuristic procedures that use local approximations of the nonlinear term to generate a succession of approximating subproblems. Two procedures of this latter type will be presented in the next Section.

4 Solving algorithms

4.1 One step algorithm (OS)

The algorithm we propose in the present section involves the linearisation scheme introduced in Labbé et al. (1998) for the standard NPP formulation. The idea is to use the *conditional gradient (Frank-Wolfe) algorithm* to solve the first and second level problem at the same time, by means of a succession of linear approximations of the CNPP that are aimed to converge to the *User Equilibrium*. In order to do so, both sets of problem variables, that is, the flows x_a^k and the tolls t_a, are determined at the same time. We want the tolls to effectively influence the flows distribution at the second level across the iterations (and vice versa), so that the final result will not subordinate one variable to the optimality of the other (which is instead the case of the Two Steps algorithm, illustrated in section 4.2).

4.1.1 Changes to the cost functions

In order to formulate a linear approximating problem for the CNPP, we need to modify the cost function previously defined by Equation (6). We decompose the toll as follows: $t_a^{(j)} = (t_a^{(j-1)} + \Delta t_a^{(j)})$, where $t_a^{(j-1)}$ is the toll resulting from the previous iterations and $\Delta t_a^{(j)}$ is the variation to that toll that is calculated at the current iteration (j). The cost function used in the algorithm will then be:

$$\mathcal{C}_a^{(j)}(f_a^{(j-1)}) = \begin{cases} tr_{0a} \cdot \left[1 + \alpha \cdot \left(\frac{f_a^{(j-1)}}{q_a}\right)^{\beta}\right] + t_a^{(j-1)} & \text{if } a \in \mathcal{A} \\ tr_{0a} \cdot \left[1 + \alpha \cdot \left(\frac{f_a^{(j-1)}}{q_a}\right)^{\beta}\right] & \text{if } a \in \mathcal{B} \end{cases} \tag{73}$$

The $\Delta t_a^{(j)}$ component of the toll is the decision variable that we effectively calculate at each iteration.

4.1.2 Changes to the leader's objective

For each iteration (j), the approximating subproblem is a linear approximations of the CNPP where arc costs are fixed and therefore independent of flows, Flows are assigned according to an *all-or-nothing* criterion, where all demand from the same commodity is routed on the cheapest path. The flow variables for the approximating subproblem, $r_a^{k,(j)}$ are binary variables defined as follows:

$$r_a^{k,(j)} = \begin{cases} 1 & \text{if commodity } k \text{ uses arc } a \text{ in the } (j)\text{-th approximating subproblem} \\ 0 & \text{otherwise} \end{cases} \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{74}$$

According to the FW algorithm, at each iteration j, flows $x_a^{k,(j)}$ are calculated as a convex combination of the flows from the previous iteration ($x_a^{k,(j-1)}$) and the flows obtained from solving the $j-th$ approximating subproblem ($r_a^{k,(j)}$).

$$x_a^{k,(j)} = x_a^{k,(j-1)} + \mu^{(j)} \cdot (r_a^{k,(j)} - x_a^{k,(j-1)}) = \mu^{(j)} \cdot r_a^{k,(j)} + (1 - \mu^{(j)}) \cdot x_a^{k,(j-1)} \tag{75}$$

Since the $\mu^{(j)}$ parameter is calculated only after $r_a^{k,(j)}$ are obtained, it is necessary to replace $x^{(j)}$ in the leader's objective. From Equation (75), the objective function of the leader for the j-th problem is equal to:

$$\sum_{k\in K}\sum_{a\in A\cup B}\eta^k\cdot t_a^{(j)}\cdot x_a^{k,(j)} = \tag{76}$$

$$\sum_{k\in K}\sum_{a\in A}\eta^k\cdot\left[\mu^{(j)}\cdot t_a^{(j)}\cdot r_a^{k,(j)}+(1-\mu^{(j)})\cdot t_a^{(j)}\cdot x_a^{k,(j-1)}\right]\leq \tag{77}$$

$$\sum_{k\in K}\sum_{a\in A}\eta^k\cdot\left[t_a^{(j)}\cdot r_a^{k,(j)}+t_a^{(j)}\cdot x_a^{k,(j-1)}\right] \tag{78}$$

We thus have obtained an upper bound for the leader's objective that does not contain the $\mu^{(j)}$ parameter. We will use this as objective of our linear approximating problem.

By applying the classical FW procedure to the CNPP as described in Frank & Wolfe (1956), along with the modified cost function introduced in Equation (73), and the modified leader's objective function in Equation 78, the formulation of the j-th approximating subproblem is (see also Rigonat (2012) for further details):

$$\max_{t,x}\sum_{k\in K}\sum_{a\in A}\eta^k\left[(t_a^{(j-1)}+\Delta t_a^{(j)})\cdot r_a^{k,(j)}+(t_a^{(j-1)}+\Delta t_a^{(j)})\cdot x_a^{k,(j-1)}\right] \tag{79}$$

$$\min_x\left\{\sum_{a\in A}(\mathcal{C}_a^{(j)}(f_a^{(j-1)})+\Delta t_a^{(j)})\cdot r_a^{(j)}+\sum_{a\in B}\mathcal{C}_a^{(j)}(f_a^{(j-1)})\cdot r_a^{(j)}\right\} \tag{80}$$

$$\sum_{a=i^-\cap A}r_a^{k,(j)}+\sum_{a=i^-\cap B}r_a^{k,(j)}-\sum_{a=i^+\cap A}r_a^{k,(j)}-\sum_{a=i^+\cap B}r_a^{k,(j)}=b_i^k \qquad \forall i\in\mathcal{N},\forall k\in\mathcal{K} \tag{81}$$

$$f_a^{(j-1)}=\sum_{k\in K}x_a^{k,(j-1)}\cdot\eta^k \qquad \forall a\in\mathcal{A}\cup\mathcal{B} \tag{82}$$

$$0\leq x_a^{k,(j-1)}\leq 1 \qquad \forall a=(i,j)\in\mathcal{A}\cup\mathcal{B},\forall k\in\mathcal{K} \tag{83}$$

$$r_a^{k,(j)}\in\{0,1\} \qquad \forall a\in\mathcal{A}\cup\mathcal{B},\forall k\in\mathcal{K} \tag{84}$$

$$\Delta t_a^{(j)} \quad \text{free} \qquad \forall a\in\mathcal{A} \tag{85}$$

$$t_a^{(j-1)}+\Delta t_a^{(j)}\geq 0 \qquad \forall a\in\mathcal{A} \tag{86}$$

where $t_a^{(j)}=(t_a^{(j-1)}+\Delta t_a^{(j)})$ and the relation between $x_a^{k,(j)}$ and $r_a^{(j)}$ is expressed by Equation (75).

To solve this problem, however, it is first necessary to reformulate it into a single level linear problem.

4.1.3 From bilevel to single level linear CNPP

Parallelling the procedure introduced in Section 3.4, it is possible to reformulate the bilevel CNPP linear approximation into a single level problem by exploiting strong duality:

$$\max_{t,x} \sum_{k\in K}\sum_{a\in A} \eta^k \left[(t_a^{(j-1)} + \Delta t_a^{(j)}) \cdot r_a^{k,(j)} + (t_a^{(j-1)} + \Delta t_a^{(j)}) \cdot x_a^{k,(j-1)}\right] \tag{87}$$

$$\text{s.t.} \sum_{a\in A} (\mathcal{C}_a^{(j)}(f_a^{(j-1)}) + \Delta t_a^{(j)}) \cdot r_a^{k,(j)} + \sum_{a\in B} \mathcal{C}_a^{(j)}(f_a^{k,(j-1)}) \cdot r_a^{k,(j)} =$$

$$= \sum_{n\in N} \lambda_i^{k,(j)} \cdot b_i^k \qquad \forall k \in \mathcal{K} \tag{88}$$

$$\sum_{a=i^-\cap A} r_a^{k,(j)} + \sum_{a=i^-\cap B} r_a^{k,(j)} - \sum_{a=i^+\cap A} r_a^{k,(j)} - \sum_{a=i^+\cap B} r_a^{k,(j)} = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{89}$$

$$\lambda_i^{k,(j)} - \lambda_j^{k,(j)} \leq \mathcal{C}_a^{(j)}(f_a^{(j-1)}) + \Delta t_a^{(j)} \qquad a = (i,j) \in \mathcal{A}, \forall k \in \mathcal{K} \tag{90}$$

$$\lambda_i^{k,(j)} - \lambda_j^{k,(j)} \leq \mathcal{C}_a^{(j)}(f_a^{(j-1)}) \qquad \forall a = (i,j) \in \mathcal{B}, \forall k \in \mathcal{K} \tag{91}$$

$$f_a^{(j-1)} = \sum_{k\in K} x_a^{k,(j-1)} \cdot \eta^k \qquad \forall a \in \mathcal{A} \cup \mathcal{B} \tag{92}$$

$$0 \leq x_a^{k,(j-1)} \leq 1 \qquad \forall a = (i,j) \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{93}$$

$$\Delta t_a^{(j)} \quad \text{free} \qquad \forall a \in \mathcal{A} \tag{94}$$

$$t_a^{(j-1)} + \Delta t_a^{(j)} \geq 0 \qquad \forall a \in \mathcal{A} \tag{95}$$

$$r_a^{k,(j)} \in \{0,1\} \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{96}$$

$$\lambda_i^{k,(j)} \quad \text{free} \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{97}$$

Note that the problem is still non-linear because of the bilinear term $\Delta t_a^{(j)} \cdot r_a^{k,(j)}$ in both the leader's objective and in Equation (88).

4.1.4 Linearisation

The linearisation scheme of the bilinear term $\Delta t_a^{(j)} \cdot r_a^{k,(j)}$ follows the one illustrated in Labbé et al. (1998)) for the uncongested NPP. A slack variable, different for each commodity, is used to re-define the continuous toll variable:

$$p_a^{k,(j)} = \begin{cases} \Delta t_a^{(j)} & \text{if } r_a^{k,(j)} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{98}$$

The following constraints need to be added:

$$p_a^{k,(j)} - \Delta t_a^{(j)} \leq M \cdot (1 - r_a^{k,(j)}) \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{99}$$

$$-p_a^{k,(j)} + \Delta t_a^{(j)} \leq M \cdot (1 - r_a^{k,(j)}) \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{100}$$

$$p_a^{k,(j)} - N \cdot r_a^{k,(j)} \leq 0 \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{101}$$

$$-p_a^{k,(j)} - N \cdot r_a^{k,(j)} \leq 0 \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{102}$$

where M e N are arbitrary $big-M$ parameters. In fact, Equations (99) and (100) are *Big-M* type constraints. If r_a^k is one, p_a^k is equal to Δt_a; if r_a^k is zero, the constraints are relaxed. Equations (101) and(102) denote an *upper bound* for p_a^k. If r_a^k is equal to one, p_a^k is less than N_a. If r_a^k is zero, p_a^k is zero too.

it follws that the transformation carried out, the formulation for the single level linear j-th approximating CNPP subproblem is:

$$\max_{t,x} \sum_{k\in K} \sum_{a\in A\cup B} \eta^k \cdot (t_a^{(j-1)} \cdot r_a^{k,(j)} + p_a^{k,(j)}) \tag{103}$$

$$\text{s.t.} \sum_{a\in A} (\mathcal{C}_a^{(j)}(f_a^{k,(j-1)}) \cdot r_a^{k,(j)} + p_a^{k,(j)}) + \sum_{a\in B} \mathcal{C}_a^{(j)}(f_a^{k,(j-1)}) \cdot r_a^{k,(j)} = \sum_{n\in N} \lambda_i^{k,(j)} \cdot b_i^k \qquad \forall k \in \mathcal{K} \tag{104}$$

$$\sum_{a=i^-\in A\cup B} r_a^{k,(j)} - \sum_{a=i^+\in A\cup B} r_a^{k,(j)} = b_i^k \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{105}$$

$$\lambda_i^{k,(j)} - \lambda_j^{k,(j)} \leq \mathcal{C}_a^{(j)}(f_a^{(j-1)}) + \Delta t_a^{(j)} \qquad \forall a = (i,j) \in \mathcal{A}, \forall k \in \mathcal{K} \tag{106}$$

$$\lambda_i^{k,(j)} - \lambda_j^{k,(j)} \leq \mathcal{C}_a^{(j)}(f_a^{(j-1)}) \qquad \forall a = (i,j) \in \mathcal{B}, \forall k \in \mathcal{K} \tag{107}$$

$$p_a^{k,(j)} - \Delta t_a^{(j)} \leq M \cdot (1 - r_a^{k,(j)}) \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{108}$$

$$\Delta t_a^{(j)} - p_a^{k,(j)} \leq M \cdot (1 - r_a^{k,(j)}) \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{109}$$

$$p_a^{k,(j)} \leq N \cdot r_a^{k,(j)} \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{110}$$

$$-p_a^{k,(j)} \leq N \cdot r_a^{k,(j)} \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{111}$$

$$f_a^{(j-1)} = \sum_{k\in K} x_a^{k,(j-1)} \cdot \eta^k \qquad \forall a \in \mathcal{A} \cup \mathcal{B} \tag{112}$$

$$\Delta t_a^{(j)} \quad \text{free} \qquad \forall a \in \mathcal{A} \tag{113}$$

$$t_a^{(j-1)} + \Delta t_a^{(j)} \geq 0 \qquad \forall a \in \mathcal{A} \tag{114}$$

$$p_a^{k,(j)} \quad \text{free} \qquad \forall a \in \mathcal{A}, \forall k \in \mathcal{K} \tag{115}$$

$$r_a^{k,(j)} \in \{0;1\} \qquad \forall a \in \mathcal{A} \cup \mathcal{B}, \forall k \in \mathcal{K} \tag{116}$$

$$\lambda_i^{k,(j)} \quad \text{free} \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \tag{117}$$

4.1.5 OS Algorithm

Relying on the formulation (103)-(117), we generate a succession of approximating subproblems that will converge to the solution of the CNPP.

Initialization A threshold ϵ is chosen for the stopping criterion and the *Big M* parameters M and N are set;

Starting solution An admissible starting solution, $\bar{x}^{(0)}$ and $\Delta t^{(0)}$, is found;

Initial costs Arc costs are adjusted according to the initial admissible flows vector.

j-th iteration Given the flow vector $\bar{x}^{(j-1)}$ and the toll vector $t^{(j-1)}$, solution of the problem at the $(j-1)$-th iteration:

1. The cost vector $\mathcal{C}_a^{(j)}$ is determined as function of the flow vector and the toll vector:

$$\mathcal{C}_a^{(j)}(f_a^{(j-1)}) = \begin{cases} tr_{0a} \cdot \left[1 + \alpha \cdot \left(\frac{f_a^{(j-1)}}{q_a}\right)^\beta\right] + t_a^{(j-1)} & \text{if } a \in \mathcal{A} \\ tr_{0a} \cdot \left[1 + \alpha \cdot \left(\frac{f_a^{(j-1)}}{q_a}\right)^\beta\right] & \text{if } a \in \mathcal{B} \end{cases} \tag{118}$$

where $f_a^{(j-1)} = \sum_{k\in K} x_a^{k,(j-1)} \cdot \eta^k$.

2. The linear approximating problem (103) - (117) is solved in order to obtain the *All-Or-Nothing* flow vector $r^{k,(j)}$ and the toll difference vector $\Delta t^{(j)}$ for the current iteration;
3. The FW *descent step* is executed in order to determine the flows for the j-th iteration. It consists of the following one-dimensional non-linear search problem:

$$\mu^{(j)} \in \text{argmin}_{\mu\in[0,1]}\psi(\mu) = z\left[\bar{x}^{(j-1)} + \mu\cdot(\bar{r}^{(j)}\cdot\eta - \bar{x}^{(j-1)})\right] \tag{119}$$

where μ is a scalar variable. This problem can be solved through the bisection algorithm.
4. The values of $r_a^{k,(j)}$ are used to determine the flows by means of the following equation:

$$x_a^{k,(j)} = x_a^{k,(j-1)} + \mu^{(j)}\cdot\left(\eta^k\cdot r_a^{k,(j)} - x_a^{k,(j-1)}\right) \quad \forall a\in\mathcal{A}\cup\mathcal{B}, \forall k\in\mathcal{K} \tag{120}$$

5. Tolls are calculated for the current iteration as:

$$t_a^{(j)} = t_a^{(j-1)} + \Delta t_a^{(j)} \tag{121}$$

6. The value of the *leader*'s objective is calculated as:

$$Z_l = \sum_{k\in K}\sum_{a\in A} t_a^{(j)}\cdot x_a^{k,(j)} \tag{122}$$

Stopping criterion The stopping test is executed against the threshold ϵ.

$$\frac{|z(\bar{x}^{(j)}) - z(\bar{x}^{(j-1)})|}{|z(\bar{x}^{(j-1)})|} < \epsilon \tag{123}$$

Equation (123) is the *relative gap* function of the follower's objective $z(\bar{x}^{(j)})$ at iteration j. If the test fails, a new iteration is executed with vector $\bar{x}^{(j)}$ as the starting solution; otherwise the algorithm is stopped and $\bar{x}^{(j)}$ is the deterministic flows vector and $\bar{t}^{(j)}$ the optimum toll vector associated with it.

4.2 Two steps algorithm (TS)

The procedure described in this section is a Gauss-Seidel-decomposition-type algorithm. It iteratively solves first the followers' problem with respect to arc flows $\bar{x}^{(j)}$ and then the leader's problem with respect to the toll vector $t^{(j)}$. Since in this algorithm tolls are calculated only when the flows are already distributed, they will be at most as large as the slack between the cost of the used paths and the cost of the toll-free path. A similar algorithm was indeed used by Julsain (1998) to find an equilibrium point on a congested telecommunication network.

We solve the followers' problem through the FW algorithm. In the case of the Traffic Assignment Problem the sequence of linear approximating subproblems results in a succession of shortest path problem instances. Note that, however, any traffic assignment algorithm that provides an arc-flow configuration can be used. The main points of the Two Steps algorithm are the following:

1. **Initialisation:** Relevant parameters such as α, β of the cost function and threshold ϵ for the stopping criterion of the first and the second step are chosen;
2. **First step: traffic assignment** At iteration j: given the flow vector $\bar{x}^{(j-1)}$ and the toll vector $t^{(j-1)}$, solution of the problem at the $(j-1)$-th iteration, the Traffic Assignment problem described by Equations (8)-(10) is solved, and a new flow configuration $\bar{x}^{(j)}$ is thus obtained.
3. **Second step: toll maximisation** Given the flow vector $\bar{x}^{(j)}$, the first level problem is solved in order to calculate the toll vector $t^{(j)}$, which is optimal for the current iteration. This is carried out by solving a single level bilinear reformulation of the CNPP as described in Section 3.4. Note that since at this point flows x_a^k are known, the problem here is linear, and the flow conservation constraint (Equation 36) and flow non-negativity constraint (Equation 39) are not necessary here:

$$\max_{t,x}\sum_{a,k} t_a\cdot x_a^k\cdot\eta^k \tag{124}$$

$$\text{s.t.} \quad \mathcal{C}_a(f_a) + t_a + \lambda_i^k - \lambda_j^k - \mu_a^k = 0 \qquad \forall a\in\mathcal{A}\cup\mathcal{B}, \forall k\in\mathcal{K} \tag{125}$$

$$\mu_a^k\cdot x_a^k = 0 \qquad \forall a\in\mathcal{A}\cup\mathcal{B}, \forall k\in\mathcal{K} \tag{126}$$

$$f_a = \sum_{k\in K} x_a^k\cdot\eta^k \qquad \forall a\in\mathcal{A}\cup\mathcal{B} \tag{127}$$

$$\mu_a^k \geq 0 \qquad \forall a\in\mathcal{A}\cup\mathcal{B}, \forall k\in\mathcal{K} \tag{128}$$

$$\lambda_i^k \quad \text{free} \qquad \forall i\in\mathcal{N}, \forall k\in\mathcal{K} \tag{129}$$

4. **Stopping test**

   Let ϵ be a chosen value for an acceptable relative objective error, and $Z(\bar{x}^{(j)})$ the value of the leader's objective at the $j-th$ iteration, then the stopping condition (*relative gap function*) is as follows:

$$\frac{|Z(\bar{x}^{(j)}) - Z(\bar{x}^{(j-1)})|}{|Z(\bar{x}^{(j-1)})|} < \epsilon \tag{130}$$

   If the stopping test fails, a new iteration is executed with vector $\bar{x}^{(j)}$ as the starting solution for the First step; otherwise the algorithm is stopped and $\bar{x}^{(j)}$ is the deterministic flows vector and $\bar{t}^{(j)}$ the optimum toll vector associated with it.

5 Numerical experiments

We tested both algorithms on randomly generated networks, varying in size and percentage of toll arcs:

- 50 nodes, 250 arcs, number of users: 5; toll arcs ratio: 5% , 10%, 25%, 50%.
- 100 nodes, 500 arcs, number of users: 10; toll arcs ratio: 5% , 10%, 25%, 50%.
- 150 nodes, 800 arcs, number of users: 30; toll arcs ratio: 5% , 10%, 25%.

All tests were run on 50 networks per each of the above combinations. Arc costs was randomly set between 1 and 100, commodity demand between 1 and 20 and arc capacity between 1 and 2*(avg. demand on all K commodities); all tests were run with a stopping threshold of $\epsilon = 0.001$.

The testing system is an Intel Xeon quad core machine running at 2.27 GHz, with 16 GB of Ram, running a 64 bit Ubuntu Linux 10.4 operating system and the FICO Xpress optimisation suite v.7.3. Algorithms were coded in the C programming language; optimisation problems were coded in the Mosel language and solved through Xpress.

Test results are illustrated in Tables 1-3. The OS algorithm proves to be superior in both results quality and running time. We consider leader's revenue as a relevant parameter for results quality, since both algorithms will stop once the follower's problem has reached the equilibrium. OS delivers better results on an average 50-65% of smaller networks with fewer congested arcs (5%) and average leader's revenue is around 4 times higher than TS. As the percentage of toll arcs increases (25%, 50%) however, OS delivers better results for more than 90% of the testing instances, leading to an average leader's profit that is 10 times higher than TS. This proves the full-CNPP linearisation approach adopted in OS to be far superior to the problem decomposition approach of TS.

Where execution time is concerned, OS generally performs better than TS. We expect this as an outcome of the fact that for every main execution in TS, many FW iterations are required to solve the second level problem. This implies that a different choice of algorithm for solving the user equilibrium at the second level is likely to improve running time for TS by a considerable amount.

The parameter that appears to be most influential on performance for both algorithms is the number of commodities. This is consistent with the respective formulations, where commodities $\mathcal{K}$ are the sole dimension that is shared among all problem constraints (whereas node number $\mathcal{N}$ and arc subsets $\mathcal{A}$, $\mathcal{B}$ only influence a subset of constraints each). On the other end, the variation of percentage of toll arcs appears to be less influent. In fact it leaves execution time mostly constant on networks with the same size for both algorithms, impacting running time in a significant way only on bigger networks and with toll arcs percentage above 25%.

Network size	**Alg.**	**5%**	**10%**	**25%**	**50% toll arcs**
50 nodes, 200 arcs, 5 users	OS	442.22	5518.44	8986.20	21793.59
	TS	158.04	420.81	1065.33	2206.80
100 nodes, 500 arcs, 10 users	OS	876.66	8985.48	18544.12	37881.13
	TS	223.81	626.25	1396.19	3428.34
150 nodes, 800 arcs, 30 users	OS	4739.74	20063.63	101542.69	
	TS	470.45	1730.80	5235.94	

Table 1: Average revenue

Network size	Alg.	5%	10%	25%	50% toll arcs
50 nodes, 200 arcs, 5 users	OS	66%	74%	92%	96%
	TS	34%	26%	8%	4%
100 nodes, 500 arcs, 10 users	OS	52%	85%	98%	96%
	TS	48%	15%	2%	4%
150 nodes, 800 arcs, 30 users	OS	75%	63%	94%	
	TS	25%	37%	6%	

Table 2: Percentage of best results from all instances

Network size	Alg.	5%	10%	25%	50% toll arcs
50 nodes, 200 arcs, 5 users	OS	3.405s	2.181s	1.402s	3.23s
	TS	30.770s	35.981s	30.154s	41.462s
100 nodes, 500 arcs, 10 users	OS	15.993s	14.286s	13.5s	47.427s
	TS	1m 47.198s	2m 42.621s	2m 35.275s	2m 31.605s
150 nodes, 800 arcs, 30 users	OS	3m 55.928s	3m 13.324s	11m 46.587s	
	TS	11m 49.234s	10m 51.747s	12m 4.177s	

Table 3: Average running time

6 Conclusions

In this paper we present a particular type of Network Pricing Problem that takes congestion into consideration and we develop two heuristics to solve it, namely One Step (OS) and Two Steps (TS) algorithms. These algorithms are then implemented and their performances are tested against different sets of networks.

From these tests it appears that OS outperforms TS in solution quality: in fact OS leader's revenues are up to ten times higher those produced by TS. TS however appears to have better scalability towards network dimension and complexity than OS: where OS shows an exponential growth in running time for networks with 25% or more toll arcs, TS running time remains more or less constant for all sets of networks with the same number of arcs and nodes. OS on the other hand outperforms our implementation of TS in running time on smaller networks.

This behaviour is most likely due to the particular structure of the two algorithms. In TS the second level problem is solved first: this is a very well known convex problem known as Distributed User Equilibrium for Traffic Assignment for which many solving algorithms exist. The one we chose to implement, the Frank-Wolfe algorithm, is the simplest and most generic (it was first developed for solving quadratic problems, then adapted to convex problems with linear constraints), thus rather inefficient, but it uses a sequence of linear approximations that are each an instance of a Shortest Path problem, which can be solved efficiently. The first level problem is instead a Linear Problem, which presents no "well-known" structure, but the number of times it has to be solved is comparatively smaller than the iterations of Shortest Path problem that are necessary to solve an instance.

On the other hand, the different approach adopted for the OS, that is, finding both flows and tolls at the same time, has proven to be successful. Clearly, a Frank-Wolfe-like procedure based on linear approximations is definitely not the best approach for solving Traffic Equilibrium problems, especially where running time is concerned. The time required to solve the linear problem that represents the current iteration of the algorithm grows at such a high rate that even solving instances of 30 commodities and 800 arcs required hours. The cause for this lies in the fact that this linear problem, that has to be solved a considerable amount of times, has no well-known structure and consequently no well-known efficient and dedicated algorithm, as opposed to the Shortest Path problem used in the TS algorithm.

Despite this, we believe that the OS approach is indeed valuable, considering the improvements in final revenue it delivered in our tests. One direction that could definitely be worth investigating in the future is applying this very approach to other, more recent, Traffic Assignment algorithms, in order to obtain more efficient algorithms for solving CNPP.

References

Bar-Gera, H. (2002), 'Origin-based algorithm for the traffic assignment problem', *Transportation Science* **36**(4), 398–417.

Bar-Gera, H. & Boyce, D. (2006), 'Solving a non-convex combined travel forecasting model by the method of successive averages with constant step sizes', *Transportation Reasearch Part B* **40**, 351–367.

Bertsekas, D. P. (1995), *Nonlinear Programming*, Athena Scientific, Belmont, MA.

Boyce, D., Ralevic-Dekic, B. & Bar-Gera, H. (2004), 'Convergence of traffic allocations: how much is enough?', *Journal of Transportation Engineering* **130**(1), 49–55.

Bracken, J. & McGill, J. (1973), 'Mathematical programs with optimization problems in the constraints', *Operations Research* (21), 37–44.

Brotcorne, L., Labbé, M., Marcotte, P. & Savard, G. (2000), 'A bilevel model and solution algorithm for a freight tariff-setting problem', *Transportation Science* **34**(3), 289–302.

Brotcorne, L., Labbé, M., Marcotte, P. & Savard, G. (2001), 'A bilevel model for toll optimization on a multicommodity transportation network', *Transportation Science* **35**(4), 345–358.

Damberg, O., Lundgren, J. T. & Patriksson, M. (1996), 'An algorithm for the stochastic user equilibrium problem.', *Transportation Reasearch Part B* **30**(2), 115–131.

Frank, M. & Wolfe, P. (1956), 'An algorithm for quadratic programming', *Naval Research Logistic Quarterly 3* **Part 1**, 95–110.

Gentile, G. & Noekel, K. (2009), 'Linear user cost equilibrium: the new algorithm for traffic assignment in visum', VISUM 11 Manual.

Hansen, P., Jaumard, B. & Savard, G. (1992), 'A new branch-and-bound rules for linear bilevel programming', *SIAM Journal on Scientific and Statistical Computing* **5**(13), 1194–1217.

Heilporn, G. (2008), Network pricing problems: complexity, polyhedral study and solution approaches, PhD thesis.

Jayakrishnan, R., Tsai, W., Prashker, J. & Rajadhyaksha, S. (1994), 'A faster path-based algorithm for traffic allocation', *Transportation Research Record (TRR), Journal of the Transportation Research Board* **1443**, 75–83.

Jeroslow, R. (1985), 'The polynomial hierarchy and a simple model for competitive analysis', *Mathematical Programming* **32**, 146–164.

Julsain, H. (1998), Tarification dans les réseaux de télécommunications: une approche par programmation mathématique à deux niveaux, Master's thesis, Ecole Polytechnique, Montréal.

Labbé, M., Marcotte, P. & Savard, G. (1998), 'A bilevel model of taxation and its application to optimal highway pricing', *Management Science* **44**(12), 1608–1622.

Labbé, M. & Violin, A. (2013), 'Bilevel programming and price setting problems', *4OR: A Quarterly Journal of Operations Research* **11**(1), 1–30.

Larsson, T. & Patriksson, M. (1992), 'Simplicial decomposition with disaggregated representation for the traffic allocation problem', *Transportation Science* **26**, 4–17.

LeBlanc, L. J., Morlok, B. & Pierskalla, W. (1975), 'An efficient approach to solving the road network equilibrium traffic allocation problem', *Transportation Research* **9**, 309–318.

Luo, Z., Pang, J. & Ralph, D. (1996), *Mathematical Programs With Equilibrium Constraints*, Cambridge University Press.

Patriksson, M. (1994), *The traffic assignment problem: models and methods*, Topics in transportation, VSP.

Petersen, E. R. (1975), 'A primal-dual traffic assignment algorithm', *Management Science* **22**(1), 87–95.

Pro (2009), *On the accurate convergence of deterministic assignment when comparing scenarios for large networks: investigating the LUCE algorithm*, Vol. 187-193, L. Mussone, U. Crisalli, Maggioli Editore, Rimini, Italy.

Rigonat, D. (2012), Approximate algorithms for the network pricing problem with congestion, Master's thesis, Universitá degli Studi di Trieste, Italy.

Sheffi, Y. (1985), *Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods*, Prentice Hall.

Shimizu, K. & Aiyoshi, E. (1981), 'A new computational method for Stackelberg and min-max problems by use of a penalty method', **26**, 460–466.

Shimizu, K., Ishizuka, Y. & Bard, J. (1997), *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic Publishers.

Vicente, L. & Calamai, P. (1994), 'Bilevel and multilevel programming: A bibliography review', *Journal of Global Optimization* (5), 291–306.

Vicente, L., Savard, G. & Júdice, J. (1994), 'Descent approaches for quadratic bilevel programming', *Journal of Optimization Theory and Applications* **81**(2), 379–399.

Wardrop, J. G. (1952), 'Some theoretical aspects of road traffic research', *ICE Proceedings: Engineering Divisions* **1**(3), 325–362.