The Bilevel Road Pricing Problem

S. Dempe
TU Bergakademie Freiberg, Germany

S. Franke
TU Bergakademie Freiberg, Germany

Abstract

We consider a bilevel road pricing problem. The lower level models a traffic assignment problem, whereas the upper level represents a toll problem which has impact on the distribution of the users on the network. We investigate the structure of this problem and consider a special lower level objective function. We show that a solution algorithm [10] can be applied to the optimal value reformulation of the bilevel programming problem using an outer approximation of the feasible set of the former problem. Furthermore, we formulate optimality conditions and show which real-life transport problems fit into this framework and can be solved using the proposed technique.

Mathematics Subject Classification: 90C30; 91A65

Keywords: bilevel optimization; nonsmooth optimization; optimal value function; solution algorithm

1 Introduction

1.1 The bilevel programming problem

Bilevel programming problems are hierarchical optimization problems between two participants, the leader and the follower. The leader first selects a vector
from his/her feasible set. This vector is a parameter in a second optimization problem which is subsequently solved by the follower. The follower selects a response from his/her set of optimal solutions and communicates it to the leader.

A common strategy to solve the bilevel programming problem is its transformation into a singlelevel problem. There exist three standard approaches.

The first one is the Karush-Kuhn-Tucker (KKT) reformulation where the lower level is replaced by its KKT conditions. This procedure provides necessary and sufficient optimality conditions if we assume that the lower level fulfills a certain constraint qualification and that the involved functions are continuously differentiable and convex. Results on this programming problem can be found in [13, 28]. Necessary and sufficient optimality conditions are formulated in [33, 32]. Furthermore, [27] is devoted to the investigation of stationary concepts and a sensitivity analysis. The resulting reformulation is however not fully equivalent to our original problem, see [9].

The second approach is the usage of the lower level optimal value function in an estimation constraint which is included in the singlelevel programming problem and forces the feasible points of this problem to be optimal for the lower level. Since the optimal value function is not given explicitly, this procedure complicates the feasible set of the resulting problem. The reformulation was introduced in [23]. Corresponding optimality conditions are formulated in [34].

As a third possibility to create a singlelevel programming problem, the lower level can be replaced by a generalized equation. Under certain assumptions, the resulting problem is fully equivalent to the bilevel programming problem, see [13, 24]. Investigations on this approach can be found in the former paper.

1.2 Formulation of the toll problem

It is our aim to model the following situation: The owner of a network wants to maximize his/her revenue arising from tolls he/she can determine on a singlecommodity network. The travel demand is fixed, and the user of the network tries to find a shortest path with respect to certain travel costs.

In Section 3.3, we will expand the problem to multicommodity networks.

The notation is standard. \( \mathbb{R}^n_+ \) stands for the \( n \)-dimensional nonnegative real-valued space.

We begin with the definition of the network. Consider a set of nodes \( N \) and a set of directed arcs \( A \) forming the graph \( G = (N, A) \) which represents a network. Let \( W \subseteq N \times N \) denote the set of origin-destination pairs. For every such pair \( w \in W \), there exists a set of routes \( P_w \) that connects the origin with the respective destination. Let \( P := \bigcup_{w \in W} P_w \) denote the set of all routes of the
network. Throughout this paper, we assume that for every origin-destination pair, there exists at least one route, i.e. that $P_w$ is nonempty for every $w \in W$.

Now, consider an origin-destination pair $w \in W$, a route $p \in P$ and an arc $a \in A$. In the following, we want to introduce two matrices. First, let

$$\Lambda := \Lambda_{wp} = \begin{cases}
1 & p \in P_w \\
0 & p \not\in P_w,
\end{cases}$$

$\Lambda \in \{0, 1\}^{|W| \times |P|}$, denote the origin-destination matrix which indicates whether the route $p$ connects the origin of $w$ with its destination. Second, the matrix

$$\Delta := \Delta_{ap} = (\delta_{ap}) = \begin{cases}
1 & a \text{ belongs to } p \\
0 & a \text{ does not belong to } p,
\end{cases}$$

$\Delta \in \{0, 1\}^{|A| \times |P|}$, shows the affiliation of the arc $a$ to the route $p$.

Let $d := d_w \in \mathbb{R}_+^{|W|}$ denote the travel demand, $q := q_p \in \mathbb{R}_+^{|P|}$ denote the route flow, and let $v := v_a \in \mathbb{R}_+^{|A|}$ denote the arc flow. Furthermore, the maximal possible arc capacity $\bar{c} := \bar{c}_a \in \mathbb{R}_+^{|A|}$. Finally, the arc toll is denoted by $\tau := \tau_a \in \mathbb{R}_+^{|A|}$. Note that this model allows toll-free arcs if the respective arc toll is set to zero.

The route capacity $c := c_p \in \mathbb{R}_+^{|P|}$ equals the minimum of the arc capacities of every arc that is used by the respective route, i.e. $c_p := \min \{ (\delta_{ap}) \bar{c} : a \in A \}$ for $p \in P$.

We want to have a closer look at the feasibility of a route flow. A flow $q$ is feasible if it fulfills the capacity restriction, i.e. $q \leq c$, and if it satisfies $\Lambda q = d$, i.e. the route flow of every route corresponding to an origin-destination pair satisfies its travel demand. We set

$$Q := \left\{ q \in \mathbb{R}_+^{|P|} : q \leq c, \ \Lambda q = d \right\}.$$  

Furthermore, an arc flow $v$ is feasible if there exists a route flow $q \in Q$ such that $\Delta q = v$, i.e. the sum of every route flow for which the corresponding route passes arc $a$ equals the respective arc flow. We set

$$V := \left\{ v \in \mathbb{R}_+^{|A|} : \exists \ q \in Q \text{ with } \Delta q = v \right\}.$$ 

We are ready to formulate the lower level of our bilevel road pricing problem. Therefore, let $f : A \times V \rightarrow \mathbb{R}_+$ denote the route cost function of the follower for every route $p$ under the arc flow $v$ and the arc toll $\tau$. (Note that the costs do not depend on the route flow on the other arcs.) The user of the network has to solve the parametric optimization problem

$$\min_v \ f(\tau, v)$$

s.t. \quad $v \in V$  

\[(5)\]
for $V$ as depicted in (4) and for some arc toll $\tau$.

This arc toll is the basis for the formulation of the upper level. It is the leader’s variable. The road authority, who corresponds to the leader, tries to influence the traffic flow of the network by imposing tolls. The bilevel programming problem reads

$$\min_{\tau} \quad F(\tau, v)$$

subject to

$$\begin{cases} 
\tau \in T \\
v \in \Psi(\tau) := \text{Argmin}_{v' \in V} \{ f(\tau, v') : v' \in V \} 
\end{cases}$$

(6)

with $T$ being the set of feasible tolls and $\Psi$ being the solution set mapping of the lower level problem (5), i.e. the set of optimal arc flows for some $\tau \in T$.

For the tolls, it is useful to assume that there exist upper and lower bounds $0 \leq l_a \leq \tau_a \leq u_a$ for all $a \in A$. This condition can be reflected in the formulation of $T$.

The upper level objective function $F$ is the disutility function and can for example model the total travel time or the toll charges or their weighted sum.

Since $\Psi$ is a point-to-set mapping, the response of the follower may not be clear. In order to avoid this ambiguity, we presuppose a cooperation between the two players. This means that for every toll pattern $\tau \in T$, the follower chooses an arc flow within the lower level optimal set mapping that minimizes the upper level objective function. The approach corresponds to the optimistic bilevel programming problem

$$\min_{\tau, v} \quad F(\tau, v)$$

subject to

$$\begin{cases} 
\tau \in T \\
v \in \Psi(\tau). 
\end{cases}$$

(7)

**Remark 1.1.** In real-life application, this is indeed a realistic assumption: The leader sets the tolls in order to encourage the follower to choose routes such that certain arcs of the network are discharged.

**Remark 1.2.** The definition and investigations regarding the pessimistic bilevel programming problem can for example be found in [7].

Throughout this paper, we suppose that $F$ and $f$ are twice continuously differentiable and, being cost functions (in some sense) that are minimized, positive.

In the literature, the bilevel road pricing problem was mainly investigated using the KKT reformulation or the selection function $v(\tau)$, see for example [17].

The KKT approach however is locally not equivalent to the bilevel programming problem, see [8]. Another disadvantage becomes clear by having a closer look at the structure of our bilevel road pricing problem. The constraints are linear, hence the second derivatives vanish. But since we already
need the first order derivative of the lower level problem for the formulation
of the KKT problem, useful information will get lost in the process.

For the reduction to only one variable \( \tau \), we either need to determine the
selection function \( v(\tau) \) or we have to presuppose uniqueness of the lower level
solution for every \( \tau \in T \) which is a quite strong assumption.

These are the main reasons for us to use the optimal value reformulation in
this paper. In order to reformulate (7) as a singlelevel programming problem,
we use the lower level parameter-dependent optimal value function

\[
\varphi(\tau) := \min_{v \in V} f(\tau, v).
\]

The resulting problem

\[
\min_{\tau, v} F(\tau, v) \\
\text{s.t. } \begin{cases} 
\tau \in T \\
v \in V \\
f(\tau, v) \leq \varphi(\tau)
\end{cases}
\] (8)

is both globally and locally equivalent to (7), see [23].

First results on optimality conditions for the bilevel road pricing problem
can be found in [12]. We want to extend the work that is done there.

We investigate the problem (9) using the lower level objective function
\( f(\tau, v) := \tau^T v \), hence our problem reads

\[
\min_{\tau, v} F(\tau, v) \\
\text{s.t. } \begin{cases} 
\tau \in T \\
v \in V \\
\tau^T v \leq \varphi(\tau)
\end{cases}
\]

(9)

This is indeed realistic if we take into account that such an objective function models the direct connection of the toll with the respective utilization of
the arc. Other possibilities for a choice of the lower level objective function
can be found for example in [12].

Remark 1.3. It is possible to use the lower level objective function
\( f(\tau, v) = (\tau_1 + \tau_2)^T v \) with \( \tau_1 \) being the arc toll as explained above and \( \tau_2 \) being some fixed
minimal unit travel cost which is independent of the toll and might arise from
personnel or operating costs and, hence, is heavily influenced by the length
of the arc and the respective travel time. Then, the upper level objective function
only has to be minimized with respect to \( \tau_1 \) since the second term in \( f \) is fixed. The algorithm we discuss later is still applicable since this change
has no influence on the curvature of \( f \) which is the crucial requirement for the
algorithm to work, see Section 2.1.
A first observation that is easy to prove is the existence of solutions of the optimal value reformulation. As long as the upper level feasible set $T$ is bounded, there exists at least one solution of (9): The feasible set of the lower level problem is nonempty, fixed and bounded. This means that the optimal set mapping is nonempty and upper semicontinuous [2]. The existence then follows for example from [7].

Our model corresponds to the case of a network without congestion. Investigations on a situation with congestion can be found for example in [31]. Their work, being limited to very small networks, has been extended in [15]. The impact of tolls on congestion was investigated in [16]. A stochastic user equilibrium approach in the lower level is considered in [21].

1.3 Basic material

It can be easily shown that the lower level optimal value function is generally nondifferentiable (see Section 2.1). However, it was shown in [12] that $\varphi$ is Lipschitz continuous. This means that, under reasonable assumptions, the optimal value reformulation (9) is a Lipschitz optimization problem. In this case, the Clarke subdifferential and normal cone are the proper basic tools for the formulation of optimality conditions. They will shortly be introduced in the following.

Definition 1.4. The normal cone at $\bar{x} \in M$ with respect to $M$ in the sense of Clarke is given by

$$N^C_M(\bar{x}) := \{ w \in \mathbb{R}^n : w^\top r \leq 0 \ \forall \ r \in T^C_M(\bar{x}) \}$$

with

$$T^C_M(\bar{x}) := \{ v \in \mathbb{R}^n : \forall \ x_k \xrightarrow{k\to\infty} \bar{x}, t_k \downarrow 0 \ \exists \ v_k \xrightarrow{k\to\infty} v \text{ with } x_k + t_k v_k \in M \}$$

being the Clarke tangent cone at $\bar{x} \in M$ with respect to $M$.

Definition 1.5. Let $\zeta : \mathbb{R}^n \to \mathbb{R}$ be lower semicontinuous at $\bar{x}$. Then, the subdifferential in the sense of Clarke of $\zeta$ at $\bar{x}$ is given by

$$\partial^C \zeta(\bar{x}) := \{ v \in \mathbb{R}^n : (v, -1) \in N^C_{\text{epi}} \zeta(\bar{x}, f(\bar{x})) \}.$$ 

Here, $\text{epi} \ z := \{(x, \alpha) \in \mathbb{R}^{n+1} : z(x) \leq \alpha \}$ denotes the epigraph of $z$.

It follows from [22] that $\partial^C \zeta(\bar{x})$ is nonempty, convex and compact as long as $\zeta$ is locally Lipschitz continuous around $\bar{x}$. 
2 Solving the toll problem

2.1 A solution algorithm

The following discussion is based on the research done in [10]. There, a detailed theoretical discussion of the upcoming summary can be found.

First, we have a closer look at the problem (9). It is obviously a nondifferentiable, nonconvex optimization problem. Even for our case with \( f(\tau, v) = \tau^T v \) and \( V \) being polyhedral (see (4)), the lower level optimal value function \( \varphi \) is not differentiable. We do however get the following important property.

**Theorem 2.1.** [3] The lower level optimal value function \( \varphi \) is a piecewise linear (hence nonsmooth) concave function over the set \( Q := \{ \tau \in T : |\varphi(\tau)| < \infty \} \). Furthermore, \( Q \) is a convex polyhedron.

This is the basis for the functionality of our solution approach because it allows the following estimation of the optimal function value of (9).

Let \( \zeta_i \in \partial^C \varphi(\tau^i) \) \((i = 1, \ldots, t)\) be supergradients of the concave function \( \varphi \) with

\[
\partial^C \varphi(\tau^i) := \left\{ \alpha \in \mathbb{R}^{|A|} : \varphi(\tau) \leq \varphi(\tau^i) + \alpha^T (\tau - \tau^i) \quad \forall \tau \in \mathbb{R}^+ \right\}.
\tag{10}
\]

Then,

\[
\left\{ (\tau, v) \in \mathbb{R}^{|A|}_+ : \tau \in T, \ v \in V, \ \tau^T v \leq \varphi(\tau) \right\} \\
\subseteq \left\{ (\tau, v) \in \mathbb{R}^{|A|}_+ : \tau \in T, \ v \in V, \ \tau^T v \leq \varphi(\tau^i) + \zeta^i (\tau - \tau^i) \ (i = 1, \ldots, t) \right\}
\tag{11}
\]

for the vertices \( \tau^i \) \((i = 1, \ldots, t)\) of the polyhedron \( T \).

We need to find a proper way to calculate the supergradients of \( \varphi \) for some \( \tau^i \). As it turns out, those supergradients correspond to lower level optimal solutions (see also [10]). This means that \( \varphi(\tau^i) = \zeta^i \tau^i \), hence part of the last right-hand side of (11) can be simplified to \( \varphi(\tau^i) + \zeta^i (\tau - \tau^i) = \zeta^i \tau \).

Using the outer approximation (11) of the feasible set of (9), we will now relax the latter problem by replacing the restriction \( \tau^T v \leq \varphi(\tau) \). (Remember that \( \varphi(\tau) \) is not given explicitly, hence this is indeed a useful procedure.) Therefore, let \( Z = \{ z_0, \ldots, z_k \} \), \( k \in \mathbb{Z}_+ \), be a subset of the lower level feasible set \( V \). It should contain supergradients of the lower level optimal value function, as it was explained above. We consider the following relaxation

\[
\min_{\tau, v} \ F(\tau, v) \\
\text{s.t.} \ \\
\begin{cases}
\tau \in T \\
v \in V \\
\tau^T v \leq \tau^T z \quad \forall \ z \in Z \subseteq V.
\end{cases}
\tag{12}
\]
It is the aim of the algorithm to sequentially improve the approximation of the feasible set of (9) by the one of (12) with the help of gradually enlarging $Z$. As soon as a solution of (12) is feasible for (9), it must be an optimal solution of the latter problem as well, see (11).

**Algorithm 2.2. Global/local optimality for (9) by an outer approximation**

**Initialization** Choose a vector for $Z \subseteq V$. Set $k := 1$.

**Step 1** Solve problem (12) globally/locally. The optimal solution is $(\tau_k, v_k)$.

**Step 2** If $(\tau_k, v_k)$ is feasible for (9), stop. The solution is globally/locally optimal for this problem. Otherwise, compute an optimal solution $z_k$ of the lower level problem with the parameter $\tau_k$. Set $Z := Z \cup \{z_k\}$, $k := k + 1$ and go to Step 1.

**Remark 2.3.** Concerning Step 2 of Algorithm 2.2, the lower level problem is solved in order to compute a supergradient of the optimal value function $\varphi$. This step also ensures that $Z$ is a subset of $V$.

For globally optimal solutions, the following result was proven.

**Theorem 2.4.** [10] Assume that the set $P := \{((\tau, v) \in \mathbb{R}^{|A|} : \tau \in T, v \in V \} is bounded. Then, every accumulation point $(\tau, v)$ of the sequence $\{(\tau_k, v_k)\}_{k=1}^{\infty}$ produced by Algorithm 2.2 (with the global solution of (12) in Step 1) is globally optimal for (9).

In order to prove the convergence of the aforementioned sequence to a locally optimal solution, we first need to introduce a suitable cone of feasible directions.

**Definition 2.5.** Let $M \subseteq \mathbb{R}^n$. The tangent/Bouligand cone at $x \in M$ with respect to $M$ is defined by

$$T_M(x) := \left\{ r \in \mathbb{R}^n : \exists \{x_k\}_{k=1}^{\infty} \subseteq M, t_k \downarrow 0 \text{ with } x_k \xrightarrow{k \to \infty} x, \frac{x_k - x}{t_k} \xrightarrow{k \to \infty} r \right\}.$$

Since the feasible set of (12) depends on the choice of $Z = \{z_0, ..., z_k\}$, we denote its feasible set in the $k$-th iteration of Algorithm 2.2 by $M_k$.

Presupposing a first order optimality condition, we get the following convergence result.

**Theorem 2.6.** [10] Assume that the set $P$ as introduced in Theorem 2.4 is bounded. Then, every accumulation point $(\tau, v)$ of the sequence $\{(\tau_k, v_k)\}_{k=1}^{\infty}$ produced by Algorithm 2.2 (with the local solution of (12) in Step 1) is locally optimal for (9) provided that $\nabla F(\tau_k, v_k)^T r > 0$ for all $r \in T_{M_k}(\tau_k, v_k)$ in the $k$-th iteration.
2.2 Reformulating the lower level problem

We consider the lower level problem (5)

$$\min_v \quad \tau^T v$$

s.t. \quad v \in V.

It is our aim to find a suitable reformulation of this problem using the route information instead of the arc information such that Algorithm 2.2 is still applicable.

**Lemma 2.7.** The optimal value mapping $\tau \mapsto \varphi(\tau)$ of the problem

$$\min_q \quad \tau^T (\Delta q)$$

s.t. \quad \begin{align*}
0 & \leq q \leq c \\
\Lambda q & = d
\end{align*}

with $\Delta$ being a given matrix as in (2) is concave.

*Proof.* Taking the definition of $V$ and $Q$ into account (see (3) and (4)), we see that now $Q$ forms the lower level feasible set. $V$ consists of every $v$ for which there exists $q \in Q$ such that $\Delta q = v$. This is expressed in the objective function. $\varphi$ is concave since $Q$ is convex and the mapping $q \mapsto \Delta q$ is linear. \qed

We replace the lower level in the optimal value reformulation and arrive at

$$\min_{\tau,q} \quad F(\tau, \Delta q)$$

s.t. \quad \begin{align*}
\tau & \in T \\
0 & \leq q \leq c \\
\Lambda q & = d \\
\tau^T (\Delta q) & \leq \varphi(\tau)
\end{align*}

with

$$\varphi(\tau) := \min_q \left\{ \tau^T (\Delta q) : 0 \leq q \leq c, \ \Lambda q = d \right\}.$$

2.3 Optimality conditions

A sufficient condition for an optimal solution of the bilevel programming problem was formulated in [34]. For our optimization problem which most likely has a discrete feasible set due to the structure of the set $V$ (representing for example the amount of users of a network), we need to adapt the result. The condition in the original formulation that the point of interest is in the interior of the feasible set is no longer satisfiable.

Let $N(\cdot)$ denote the normal cone in the usual sense of convex analysis.
Theorem 2.8. Consider \((\tau, \nu) \in T \times V\). If there exist \(\mu \geq 0\), a positive constant \(\kappa \in \mathbb{Z}\), \(\lambda_i \geq 0\), \(\sum_{i=1}^{\kappa} \lambda_i = 1\), \(v_i \in \Psi(\tau)\) such that
\[
0 \in \nabla_\tau F(\tau, \nu) + \mu \left( \nabla_\tau f(\tau, \nu) - \sum_{i=1}^{\kappa} \nabla_\tau f(\tau, v_i) \right) + N_T(\tau)
\]
\[
0 \in \nabla_\nu F(\tau, \nu) + \mu \nabla_\nu f(\tau, \nu) + N_V(\nu)
\]
and the matrix
\[
\frac{\partial^2}{\partial (\tau, \nu)^2} \left( F(\tau, \nu) + \mu (f(\tau, \nu) - \sum_{i=1}^{\kappa} \lambda_i f(\tau, v_i)) \right)
\]
is positive semidefinite for arbitrary \(\tau\) and \(\nu\), then \((\tau, \nu)\) is an optimal solution of (7).

Proof. The proof can be formulated as in [34, Theorem 6.1] with the addition of the normal cones at the necessary places which is straightforward. \(\square\)

We want to formulate optimality conditions for the optimal value reformulation (9). Therefore, we need suitable constraint qualifications. Note that at every feasible point of this problem, the constraint \(\tau^\top \nu \leq \varphi(\tau)\) is fulfilled as equality. Hence, usual constraint qualifications like the (nonsmooth) MFCQ are violated, see also [34].

Even if we want to formulate optimality conditions for the relaxed problem (12), we have to take into account that we are actually interested in optimal solutions of this problem that are feasible for (9). At those points, the same problem described above arises for the relaxation (12). Hence, we have to formulate such conditions using other assumptions.

In the following section, we investigate strong sufficient optimality condition of second order using feasible directions. The subsequent section is devoted to the concept of partial calmness as a constraint qualification.

2.3.1 Strong sufficient conditions of second order

Consider the relaxed problem (12). Remember that \(M_k\) denotes the feasible set of (12) in the \(k\)-th iteration of Algorithm 2.2 for \(k \in \mathbb{Z}_+\). Let \(M\) denote the feasible set of (9).

Definition 2.9. The strong second order sufficient condition (SSOSC) holds at \((\tau, \nu) \in M_k\) if \(r^\top \nabla^2 F(\tau, \nu)r > 0\) for all \(r \in T_{M_k}(\tau, \nu), r \neq 0\).

We just mentioned that the (nonsmooth) MFCQ does not hold at any point that is feasible for the optimal value reformulation (9). Hence, the set of Lagrangian multipliers related to the Lagrangian function of (12) may be empty or at least not compact. In order to formulate optimality conditions, it is therefore more useful to employ the following approach.
Theorem 2.10. Let $M_k$ be compact and let SSOSC hold at $(\tau, \nu) \in M_k$. Furthermore, suppose that

$$\nabla F(\tau, \nu)^\top r \geq 0 \quad \forall \ r \in T_{M_k}(\tau, \nu).$$

(14)

Then, $(\tau, \nu)$ is a locally optimal solution of (12).

Proof. Suppose the opposite. Then, there exists a sequence $\{(\tau_l, \nu_l)\}_{l=1}^{\infty} \in M_k$ that fulfills $(\tau_l, \nu_l) \xrightarrow{l \to \infty} (\tau, \nu)$ and $F(\tau_l, \nu_l) \geq F(\tau_l, \nu_l)$ for all $l \in \mathbb{N}$. Combining this inequality with

$$F(\tau_l, \nu_l) = F(\tau, \nu) + \nabla F(\tau, \nu)((\tau_l, \nu_l) - (\tau, \nu)) + \frac{1}{2}((\tau_l, \nu_l) - (\tau, \nu))\nabla^2 F(\tau, \nu)((\tau_l, \nu_l) - (\tau, \nu)) + o(||((\tau_l, \nu_l) - (\tau, \nu)||^2),$$

we get

$$0 \geq \frac{F(\tau_l, \nu_l) - F(\tau, \nu)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2)} \geq \frac{\nabla F(\tau, \nu)^\top (\tau_l, \nu_l) - (\tau, \nu)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2)} + \frac{1}{2} \frac{||\nabla^2 F(\tau, \nu)((\tau_l, \nu_l) - (\tau, \nu))||^2}{||((\tau_l, \nu_l) - (\tau, \nu)||^2)} + \frac{o(||((\tau_l, \nu_l) - (\tau, \nu)||^2)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2}. \tag{15}$$

Here, it holds that $\frac{o(||((\tau_l, \nu_l) - (\tau, \nu)||^2)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2} \downarrow 0$ for $l \to \infty$. By assumption (14), we have

$$\frac{\nabla F(\tau, \nu)^\top (\tau_l, \nu_l) - (\tau, \nu)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2} \geq 0 \quad \forall \ t,$$

hence

$$\lim_{l \to \infty} \frac{\nabla F(\tau, \nu)^\top (\tau_l, \nu_l) - (\tau, \nu)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2} \geq 0.$$ 

Furthermore, note that

$$\frac{(\tau_l, \nu_l) - (\tau, \nu)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2} \xrightarrow{l \to \infty} r \in T_{M_k}(\tau, \nu).$$

Two cases occur.

First case: $\nabla F(\tau, \nu)^\top r > 0$. Since $||((\tau_l, \nu_l) - (\tau, \nu)||^2 \downarrow 0$ and $\nabla F(\tau, \nu)^\top ((\tau_l, \nu_l) - (\tau, \nu)) \geq 0$ for $l \to \infty$, it follows that

$$\frac{\nabla F(\tau, \nu)^\top r}{||((\tau_l, \nu_l) - (\tau, \nu)||^2} \xrightarrow{l \to \infty} 0,$$

which is an immediate contradiction to (15).

Second case: $\nabla F(\tau, \nu)^\top r = 0$. Then, (15) reduces to

$$0 \geq \frac{1}{2} \frac{||((\tau_l, \nu_l) - (\tau, \nu)||^2}{||((\tau_l, \nu_l) - (\tau, \nu)||^2} \geq \frac{1}{2} \frac{||\nabla^2 F(\tau, \nu)((\tau_l, \nu_l) - (\tau, \nu))||^2}{||((\tau_l, \nu_l) - (\tau, \nu)||^2} + \frac{o(||((\tau_l, \nu_l) - (\tau, \nu)||^2)}{||((\tau_l, \nu_l) - (\tau, \nu)||^2}. \tag{16}$$

We get a contradiction to SSOSC. The proof is complete. \qed
Remark 2.11. A similar theorem for a general nonlinear optimization programming problem with another proof using the Lagrangian function can be found in [26].

We are ready to formulate a sufficient optimality condition of second order.

Theorem 2.12. Assume that the set

$$P := \{(\tau, v) \in \mathbb{R}_+^{2|A|} : \tau \in T, v \in V\}$$

is bounded. Furthermore, let \(\{(\tau_k, v_k)\}_{k=1}^\infty\) be the sequence the algorithm produces. Then, every accumulation point \((\tau, v)\) of \(\{(\tau_k, v_k)\}_{k=1}^\infty\) is a local optimum of problem (9) provided that (14) and SSOSC are fulfilled at \((\tau_k, v_k)\) in the k-th iteration.

Proof. The existence of accumulation points follows from the concavity, hence continuity of \(\varphi\) which ensures, together with the assumed boundedness of \(P\), the compactness of the feasible set \(M\).

It was shown in [10, Theorem 3.7] that every accumulation point \((\tau, v)\) of the above mentioned sequence is a locally optimal solution of the relaxed problem (12) and the optimal value reformulation (9).

We can apply Theorem 2.10: Every point \((\tau_k, v_k)\) is a local optimum of (12) in the respective k-th iteration. Now, suppose that \((\tau, v)\) is not locally optimal, i.e. there exists a sequence \(\{(\tau_l, v_l)\}_{l=1}^\infty\) with \((\tau_l, v_l) \in M\) for all \(l \in \mathbb{N}\) with \(F(\tau, v) > F(\tau_l, v_l)\). But since the inclusion \(M \subseteq M_k\) holds for all \(k\), we get a contradiction as it was created in the proof of Theorem 2.10. Therefore, \((\tau, v)\) is locally optimal for (9). \(\square\)

2.3.2 The partial calmness condition

A suitable concept for a constraint qualification for the problem (9) is the partial calmness condition which was first introduced in [6]. It is defined in terms of the fully perturbed problem. The optimal value reformulation (9) only has one constraint that causes difficulties, namely the one including the lower level optimal value function. We consider the following partially perturbed problem

$$\min_{\tau, v} \quad F(\tau, v)$$

s.t. \[
\begin{cases}
\tau \in T \\
v \in V \\
f(\tau, v) - \varphi(\tau) + u \leq 0
\end{cases}
\]

and denote its feasible set by \(M_u\).

Definition 2.13. Let \((\tau, v)\) be an optimal solution of (9). The problem (9) is partially calm at \((\tau, v)\) if there exist \(\varepsilon, \kappa > 0\) such that

$$F(\tau, v) - F(\tau, v) + \kappa|u| \geq 0$$

for all \((u, \tau, v) \in U_{\varepsilon}(0, \tau, v) \cap (\mathbb{R} \times M_u)\).
We get the following connection between the partial calmness condition and an exact penalty problem.

**Theorem 2.14.** [34, Proposition 3.3] Let \((\tau, \bar{v})\) be an optimal solution of (9). The problem (9) is partially calm at \((\tau, \bar{v})\) if and only if there exists \(\kappa > 0\) such that \((\tau, \bar{v})\) is an optimal solution of

\[
\begin{array}{ll}
\min_{\tau, v} & F(\tau, v) + \kappa(f(\tau, v) - \varphi(\tau)) \\
\text{s.t.} & \tau \in T \\
& v \in V.
\end{array}
\]

This means that under partial calmness of (9), we are allowed to move the constraint that causes difficulties directly to the objective function as a penalty term and significantly simplify the feasible set of (9).

Furthermore, it was shown in [34] that the problem (9) with lower level objective function \(f(\tau, v) = \tau^\top v\) and linear lower level constraints is partially calm.

Now we want to formulate necessary optimality conditions for a locally optimal solution of (9).

**Theorem 2.15.** Let \((\tau, \bar{v})\) be a locally optimal solution of (9). Then, there exists a multiplier \(\kappa \geq 0\) such that

\[
0 \in \nabla_\tau F(\tau, \bar{v}) + \kappa \left( \nabla_\tau f(\tau, \bar{v}) - \partial_C^\tau \varphi(\tau) \right) + N_T(\tau)
\]

\[
0 \in \nabla_v F(\tau, \bar{v}) + \kappa \nabla_v f(\tau, \bar{v}) + N_V(\bar{v}).
\]

**Proof.** The problem (9) is partially calm at \((\tau, \bar{v})\). Hence, it follows from Theorem 2.14 that \((\tau, \bar{v})\) is a locally optimal solution of (17). We deduce from [22] that

\[
0 \in \partial^C (F(\tau, \bar{v}) + \kappa(f(\tau, \bar{v}) - \varphi(\bar{v}))) + N^C_{T \times V}(\tau, \bar{v}).
\]

For Lipschitz continuous functions \(F\) and \(f\) (which was presupposed in Section 1.2) as well as constants \(\alpha, \beta \geq 0\), the well-known rule

\[
\partial^C (\alpha F(\tau, \bar{v}) + \beta f(\tau, \bar{v})) \subseteq \alpha \partial^C F(\tau, \bar{v}) + \beta \partial^C f(\tau, \bar{v})
\]

is applicable. Hence, there exists \(\kappa \geq 0\) such that the following first-order optimality conditions

\[
0 \in \nabla_\tau F(\tau, \bar{v}) + \kappa \left( \nabla_\tau f(\tau, \bar{v}) - \partial_C^\tau \varphi(\tau) \right) + N_T(\tau)
\]

\[
0 \in \nabla_v F(\tau, \bar{v}) + \kappa \nabla_v f(\tau, \bar{v}) + N_V(\bar{v})
\]

are fulfilled. \(\Box\)
Remark 2.16. In [33, Theorem 3.2], a problem containing a variational inequality such that it can model a bilevel programming problem is considered. The validity of conditions as formulated in Theorem 2.15 are then proved under pseudo upper Lipschitz condition (which is sometimes called Lipschitz-like/Aubin property). Keeping in mind that this condition directly implies partial calmness [22], we see that this result holds for our problem as well.

We conclude this section with a remark on the calculation of $\partial^C \varphi$ in general. Remember that $\varphi$ is Lipschitz continuous and that $f$ is continuously differentiable. If

$$
\Psi(\tau) := \text{Argmin}_{v \in V} f(\tau, v)
$$

is nonempty for every $\tau \in T$, then it is possible to derive

$$
\partial^C \varphi(\tau) \subseteq \text{conv} \bigcup_{v \in \Psi(\tau)} \{ \nabla_{\tau} f(\tau, v) + N_T(\tau) \}
$$

from [6] and from [25].

3 Real-life application

3.1 The lower level in reality

We distinguish three cases.

First case: Polyhedral $V$
This is the case the algorithm was written for in [10]. The algorithm can be applied without any additional assumptions.

Second case: Discrete $V$
Considering the application of the toll problem, we see that the set of feasible arc flows $V$ is discrete (in contrast to the upper level feasible set $T$ of arc tolls).

Since the lower level objective function is linear with respect to $v$, we get the following equality

$$
\min_{v \in V} f(\tau, v) = \min_{v \in \text{conv} V} f(\tau, v),
$$

i.e. every solution of the lower level problem is a vertex of the set conv $V$.

We want to ensure the existence of a solution in (18). Therefore, we first have to assume that $V$ is nonempty. Furthermore, it holds that

$$
V \text{ bounded } \Rightarrow \text{conv } V \text{ compact}.
$$

If $V$ is bounded, there exists a solution of the lower level problem which is feasible with respect to both $V$ and conv $V$ as its feasible set. It follows from
linear optimization that this solution is a vertex of conv $V$. We conclude that we can apply the algorithm as in the first case.

**Third case: General, real $V$**

The feasible set of the lower level problem comprises every arc flow $v$ for which there exists a route flow $q \in Q$ such that the sum of the route flows on this arc is equivalent to the arc flow, i.e.

$$V = \left\{ v \in \mathbb{R}^{[A]}_+ : \exists q \in Q \text{ with } \Delta q = v \right\}.$$

Under lower semicontinuity of $F$, the existence of solutions of the problem (9) is ensured by the compactness of its feasible set

$$M := \left\{ (v, \tau) \in \mathbb{R}^{[A]}_+ : \tau \in T, v \in V, \tau^\top v \leq \varphi(\tau) \right\}.$$

This was proved in [30]. Hence, for the existence of optimal solutions of (9), it is crucial that $M$ is closed. We obviously get this property under the continuity of the lower level optimal value function $\varphi$. The next lemma provides the continuity of $\varphi$ under reasonable assumptions.

**Lemma 3.1.** Consider $\min_{v \in V} \tau^\top v$. Let $V \neq \emptyset$ be compact. Then, $\varphi$ is continuous.

**Proof.** Consider two sequences $\{\tau_k\}_{k=1}^{\infty}$, $\{v_k\}_{k=1}^{\infty}$ with $\tau_k \in T$ and $v_k \in V$ for all $k$. Let $\{\tau_k\}_{k=1}^{\infty}$ converge to $\tau$ with $\tau \in T$. Set $v_k \in \Psi(\tau_k)$ for all $k$ (which is possible without loss of generality since the existence of a lower level solution is ensured by the compactness assumption of $V$ and the structure of the lower level objective function). By compactness of $V$, the sequence $\{v_k\}_{k=1}^{\infty}$ has an accumulation point $\bar{v} \in V$. The first estimate

$$\lim_{k \to \infty} \varphi(\tau_k) = \lim_{k \to \infty} (\tau_k)^\top v_k = \tau^\top \bar{v} \geq \varphi(\tau)$$

provides the lower semicontinuity of $\varphi$ at $\bar{v}$. For upper semicontinuity, we first observe that

$$\lim_{k \to \infty} \varphi(\tau_k) = \lim_{k \to \infty} (\tau_k)^\top v_k \leq \lim_{k \to \infty} \left\{ (\tau_k)^\top v : v \in V \right\}. \quad (19)$$

Using the same argumentation as above, there exists a lower level solution $\bar{v}$ for $\tau$. Choosing exactly this $\bar{v}$, we can continue with (19) and get

$$\lim_{k \to \infty} \varphi(\tau_k) = \lim_{k \to \infty} (\tau_k)^\top v_k \leq \lim_{k \to \infty} \left\{ (\tau_k)^\top v : v \in V \right\} = \tau^\top \bar{v} = \varphi(\tau)$$

$$\limsup_{k \to \infty} \varphi(\tau_k) \leq \varphi(\tau).$$

Since $\tau$ can be chosen arbitrary (as long as it is contained in $T$), the optimal value function $\varphi$ is thereby continuous. \qed
Furthermore, we need to provide the boundedness of $T$ in order to ensure that $M$ is compact.

**Theorem 3.2.** Let $T \neq \emptyset$ be bounded and assume that $V \neq \emptyset$ is compact. Then, $M$ is compact as well and, hence, problem (9) has a solution.

Lemma 3.1 and Theorem 3.2 lead us to the following statement.

**Corollary 3.3.** Let \( \{ (\tau, v) \in \mathbb{R}_{+}^{2|A|} : \tau \in T, \ v \in V \} \neq \emptyset \) be compact. Then, the algorithm can be applied.

### 3.2 The lower level as a classical flow problem

The linear flow problem with capacity restriction $u_a$ on every arc $a$ can be modeled by

\[
\min_{v} \sum_{a \in A} \tau_a v_a \\
\text{s.t.} \quad \left\{ \begin{array}{l}
\sum_{a \in A^s(i)} v_a - \sum_{a \in A^e(i)} v_a = d_i \quad \forall \ i \in N \\
0 \leq v_a \leq u_a \quad \forall \ a \in A.
\end{array} \right.
\]

(20)

Here, $A^s(i)$ and $A^e(i)$ denote the set of (directed) arcs with start and end node $i$, respectively.

We set

\[
A := \begin{cases}
-1 & i \text{ is the start node of arc } a := a_{ij} \\
1 & i \text{ is the end node of arc } a := a_{ij} \\
0 & \text{otherwise}
\end{cases}
\]

(21)

$A \in \{0, 1\}^{N \times |A|}$, with the rows of $A$ depicting the nodes and the columns depicting the arcs of the graph $G$. Then, we can reformulate problem (20) as the standard linear optimization problem

\[
\min_{v} \quad \tau^\top v \\
\text{s.t.} \quad \left\{ \begin{array}{l}
A v = d \\
0 \leq v \leq u.
\end{array} \right.
\]

Hence, Algorithm 2.2 is applicable.

An algorithmic approach as well as complexity investigations regarding this problem can be found in [17].

### 3.3 The lower level as a multicommodity flow problem

We want to generalize the problem and extend the singlecommodity assumption. Now, we consider individual commodities that share arc capacities.
Therefore, we consider again the graph $G = (N, A)$ as introduced in Section 1.2.

Multicommodity networks have a wide variety of application. For example, consider the distribution of different products from a factory to retailers using trucks as the joint capacities on the arcs.

A comprehensive theoretical basis for such problems is given in [20]. For numerous applications and optimality conditions for a singlelevel multicommodity problem, see [1]. Multicommodity networks are also investigated in [4, 5] where heuristic approaches based for example on dualization can be found.

We assume that the goods are homogeneous, i.e. every unit flow of the commodity uses one unit of capacity on every arc.

**Definition 3.4.** [19] $x^{kl} : A \to \mathbb{R}_+$ is called a flow in $G$ from source $k$ to sink $l$ with travel demand $d^{kl}$ if

$$
\sum_{a \in A^c(i)} x^{kl}_a - \sum_{a \in A^c(i)} x^{kl}_a = \begin{cases} 
-d^{kl} & i = k \\
d^{kl} & i = l \\
0 & i \in N \setminus \{k, l\}
\end{cases}
$$

is fulfilled.

It follows directly from this definition that for every flow, there exist exactly one source $k$ and one sink $l$ in $G$ and that the inflow of every other node of the network equals its outflow.

Let $C$ denote the set of commodities. For every commodity $c \in C$, there is an origin-destination pair $w = (w_o, w_d) \in W$ with origin $w_o$ and destination $w_d$. Then, the travel demand for a commodity $c$ is given by

$$
d^c_i := \begin{cases} 
-n^c & i = w_o \\
n^c & i = w_d \\
0 & i \in N \setminus \{w_o, w_d\}
\end{cases}
$$

with $n^c$ denoting the number of users of commodity $c$. As it was defined in Section 1.2, $v^c$ represents the arc flow for every commodity $c$. Then, the lower level problem reads

$$
\begin{align*}
\min_v & \quad \sum_{a \in A} \left( \tau_a \sum_{c \in C} v^c_a \right) \\
\text{s.t.} & \quad \sum_{a \in A^c(i)} v^c_a - \sum_{a \in A^c(i)} v^c_a = d^c_i \quad \forall \ i \in N \ \forall \ c \in C \\
& \quad v^c_a \geq 0 \quad \forall \ a \in A \ \forall \ c \in C.
\end{align*}
$$

Remark 3.5. In order to control the possible competitive behavior in the lower level, the optimal solution has to be a Nash Equilibrium [29].
We obviously have to deal with an integer problem. It is now the question whether Algorithm 2.2 is still applicable. Therefore, we have a look at the optimal value function $\varphi$ of (22).

**Proposition 3.6.** The function

$$
\varphi(\tau) := \min_v \left\{ \sum_{a \in A} \left( \tau_a \sum_{c \in C} v^c_a \right) : \sum_{a \in A^c(i)} v^c_a - \sum_{a \in A^s(i)} v^c_a = d^c_i \quad \forall \ i \in N \ \forall \ c \in C \right. \\
\left. \quad v^c_a \geq 0 \quad \forall \ a \in A \ \forall \ c \in C \right\}
$$

is concave and piecewise linear.

**Proof.** Using the matrix $A^c$ for every commodity $c \in C$ as introduced in (21), we can reformulate (22) as

$$
\min_v \quad \tau^\top \sum_{c \in C} v^c \\
\text{s.t.} \quad \left\{ \begin{array}{l}
A^c v^c = d^c \quad \forall \ c \in C \\
v^c \geq 0 \quad \forall \ c \in C \end{array} \right.
$$

Then, we can apply [3] in order to prove the statement.

With this proposition, it is verified that we can apply Algorithm 2.2.

**Remark 3.7.** Note that the practical realization is more difficult than in the singlecommodity case.

### 4 Numerical results

Algorithm 2.2 was implemented in AMPL [14] using MINLP as a solver for integer programming problems [18]. Two test examples can be found in [15] (see also [11]). There, a heuristic algorithm using a so-called filling function is created for solving bilevel toll problems. The example problems are small-/medium-sized multi-commodity flow problems consisting of seven/twenty-five nodes and twelve/fourty arcs of whom five/twenty are toll-free. The number of origin-destination commodity flows to be optimized are two/three. For each of the examples, five data sets are given.

For comparison, we tested our algorithm using the same examples. The results are depicted in Tables 1 and 2, respectively. There, the deviation of our computed objective function values and the ones achieved with the filled function approach in [15] are given. A positive sign stands for an improvement of the original results, whereas a negative sign indicates that we achieved a worse result. It can be seen that in almost all of the cases, both upper and lower objective functions values of our calculations are in the same range.
as in the results as in the original paper. It may however happen that the
distribution of the commodities sometimes differs. This is due to the different
arcs that connect the same origin-destination pair and the variability of how
to impose tolls. We are able to find solutions within two to four iterations.

Note that Algorithm 2.2 depends on the starting vector (used for solving the
lower level problem) which is an arbitrary point being feasible with respect
to the upper level constraints. We usually chose a boundary point of the
feasible tolls. Remark also that the algorithm in [15] is designed to compute a
global optimum whereas Algorithm 2.2 is shown to converge to a local optimal
solution.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Deviation upper objective</th>
<th>Deviation lower objective</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0%</td>
<td>0%</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4.2%</td>
<td>5.3%</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0%</td>
<td>0%</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>−13.4%</td>
<td>−11.0%</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>0%</td>
<td>0%</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Small sized problem

<table>
<thead>
<tr>
<th>Data set</th>
<th>Deviation upper objective</th>
<th>Deviation lower objective</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0%</td>
<td>0%</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>−3.7%</td>
<td>−3.6%</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>−14.9%</td>
<td>30.0%</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0%</td>
<td>0%</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0%</td>
<td>0%</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2: Medium sized problem

Acknowledgements

The authors’ work has been supported by Deutsche Forschungsgemeinschaft.

References


The bilevel road pricing problem


Received: April 17, 2015; Published: May 20, 2015