The demand adjustment problem via inexact restoration method

Jorgelina Walpen · Pablo A. Lotito · Elina M. Mancinelli · Lisandro Parente

Received: 30 October 2018 / Revised: 26 November 2019 / Accepted: 2 December 2019 / Published online: 8 July 2020
© SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional 2020

Abstract
In this work, the demand adjustment problem (DAP) associated with urban traffic planning is studied. The framework for the formulation of the DAP is mathematical programming with equilibrium constraints. In particular, if the optimization program associated with the equilibrium constraint is considered, the DAP results in a bilevel optimization problem. In this approach, the DAP via the inexact restoration method is treated.

Keywords Traffic · Origin–destination matrix adjustment · Inexact restoration method · Bilevel problem

Mathematics Subject Classification 90B20 · 90C26

1 Introduction
Transport planning is a developing science and many interesting problems arise from it. Most of them require to have information about the needs of the community to travel from some point to another. Usually this information is stored in a matrix called the origin–destination matrix (OD matrix) or demand matrix. Different points over the transport network are considered as the origin–destination pairs (od pairs) and the OD matrix saves the number of trips originating and terminating in each of them (demands).
Od pairs may be connected by paths, flows on each path of the network, for a deterministic and stationary state of it, are obtained as a result of affecting the demands to the specific network. That is to say, trips are assigned to different available paths so that all of them can be carried out.

The availability of the OD matrix is not always guaranteed. In general, it is expensive and difficult to obtain as it cannot be built by simple observation, in fact, direct interviews of measurements of high cost must be carried out to build the OD matrix.

The Demand Adjustment Problem (DAP) consists in the estimation of the origin-destination matrix (OD matrix) of a congested transport network, based on the information of some observed flows in the network and an available target OD matrix (probably an outdated one).

The problem of adjusting the OD matrix can be modelled as an optimization problem with equilibrium constraints and reformulated as a bilevel problem. Among its drawbacks it has bad mathematical properties which make it difficult to solve it. Some of them are: non-convexity, non-differentiability, huge dimensions of real size problems and the fact that the point-set mapping which gives the equilibrium flows for a fixed demand is not explicitly known.

This version of the problem has been treated by many authors, some whose works are Nguyen (1977), Spiess (1990), Chen and Florian (1996), Yang et al. (1992), Codina and Barceló (2004), Codina and Montero (2006), Lundgren and Peterson (2008), Lotito and Parente (2014) and Walpen et al. (2015).

It is of remarkable importance that, with the only exception of Lotito and Parente (2014), all the methods proposed in the mentioned papers are heuristics. In general, no convergence proofs are given due to the fact that no appropriate characterization of optimality points is available. These methods have another characteristic in common: they all generate sequences of feasible points through their iterations and test the descent of the objective function of the problem.

In Lotito and Parente (2014), instead, the DAP is formulated as a general mathematical program with complementarity constraints (MPCC). Applying a lifting method (see Stein 2012; Izmailov et al. 2012), a necessary optimality condition is obtained in terms of a large non-linear semismooth system which is solved with a Newton-type method. However, the price to pay is the increase of the numerical problem size and the fact that the lower level structure is missed.

This work puts towards an approach to treat the DAP via inexact restoration. This method, originally proposed by Martínez in Martínez (1998) and Martínez (2001) to solve optimization problems with no linear constraints, has been adapted to solve bilevel problems by Andreani et. al. in Andreani et al. (2009).

The inexact restoration method deals separately with feasibility and optimality at each iteration. In the feasibility stage, called restoration phase, it seeks a feasible point (perhaps inexactly), considering the original objective function and constraints.

In the optimality phase, it looks for a trial point that sufficiently reduces the value of a Lagrangian defined by the original data in a tangent set that approximates the feasible region, within a trust region centered at the point obtained in the feasibility phase. Sufficient decrease of a merit function which balances feasibility and optimality determines the acceptance of the trial point obtained in the optimization phase. If the trial point is not accepted, the size of the trust region is reduced.

The purpose of this work is to offer an innovative alternative to solve DAP and a tangible application of the inexact restoration method.
This paper is organized as follows. In Sect. 2, a model of the problem is presented as well as the assumptions made over the network. In Sect. 3, the inexact restoration method and its adaptation to bilevel problems are presented. In Sect. 4, there is a complete description of the application of the inexact restoration to the DAP. A detailed presentation of every step of the algorithm is given and each subproblem is specifically treated in each subsection. Finally, in Sect. 5, some numerical tests are presented. Conclusions are drawn in Sect. 6.

2 Model

In this work, the DAP is considered as a mathematical program with equilibrium constraints (MPEC). The objective is the adjustment of an OD matrix by the minimization of some function that measures both the deviation from a target matrix and the distance to observed flows over the network. For each demand, the flows are constrained to satisfy a deterministic Wardrop’s user equilibrium (DUE).

The transport network is represented as a directed graph \( G = (\mathcal{N}, \mathcal{A}) \), where \( \mathcal{N} \) is the set of nodes and \( \mathcal{A} \) is the set of directed links. \( \mathcal{C} \) represents the set of origin–destination pairs (od pairs) and \( d \in \mathbb{R}_{+}^{\left| \mathcal{C} \right|} \) (from now on the demand vector), where \( d_i \) is the number of trips for the od pair \( i \). \( v \in \mathbb{R}_{+}^{\left| \mathcal{A} \right|} \) is the link flow vector.

Then the DAP is formulated as

\[
\text{(DAP)} \quad \min \ F(v, d) = \eta_1 F_1(v) + \eta_2 F_2(d) \\
\text{s. t.} \quad t(v)^T (v' - v) \geq 0, \quad \forall (v', d) \in \Omega,
\]

where \( \Omega \) is the closed convex cone of pairs \( (v, d) \) with \( v \) a feasible link flow for \( d \), i.e. a non-negative flow which satisfies the demand \( d \). The involved functions are

\[
F_1 : \mathbb{R}^{\left| \mathcal{A} \right|} \to \mathbb{R}_{+}, \quad F_2 : \mathbb{R}^{\left| \mathcal{C} \right|} \to \mathbb{R}_{+}, \quad t : \mathbb{R}^{\left| \mathcal{A} \right|} \to \mathbb{R}^{\left| \mathcal{A} \right|}.
\]

Particularly, the function \( F_1 \) measures the deviation between the assigned flow for the demand \( d \) and the observed flow \( \tilde{v} \), in some links of the network (\( \tilde{\mathcal{A}} \subset \mathcal{A} \)). The function \( F_2 \) measures the distance between \( d \) and a target matrix (usually an outdated OD-matrix \( \tilde{d} \)). The usually used metrics are those of minimum squares, maximum entropy and maximum likelihood (see Chen and Florian 1996). The parameters \( \eta_1 \) and \( \eta_2 \) reflect the confidence of the data \( \tilde{d} \) and \( \tilde{v} \), respectively. Finally, Wardrop equilibrium condition is expressed in terms of a variational inequality for the associated link cost vector \( t(v) \).

For a general version of DAP, Chen and Florian proved in Chen and Florian (1996), under minor hypotheses of continuity of the functions \( F_1, F_2 \) and \( t \), that the problem admits at least one solution. In this work, the mapping \( d \mapsto v^*(d) \), which assigns the equilibrium flows to a given demand \( d \), is considered to be single valued (i.e. the DUE admits only one solution) and it is possible to write \( F(d) = F(v^*(d)) \). For the problem to fit in this context, it is necessary to make some assumptions over the traffic network:

- the network is strongly connected, i.e. there exists at least one route for each od pair;
- the route cost functions are additive, i.e. they are the sum of the link costs which constitute the route;
- the link costs are separable, i.e. the flow in each link is independent of the flow of all other links in the network;
• the demand \(d_{pq}\) is positive for each \((p, q) \in C\);
• the link cost function \(t_a : \mathbb{R} \mapsto \mathbb{R}\) is positive, continuous and non-decreasing for each \(a \in A\).

These hypotheses guarantee the existence of equilibrium (both in the link and route flow variables) and uniqueness of equilibrium times for each od pair. If each link cost function \(t_a\) is assumed to be strictly increasing, there is uniqueness of the equilibrium link flow solution.

3 Inexact restoration and its adaptation to solve bilevel problems

The inexact restoration method (IRM) is motivated by the bad behavior of feasible methods in the presence of non-linear constraints. To face these difficulties, the algorithms presented by Martínez et al. in Martínez (1998), Martínez (2001) and Martínez and Pilotta (2000) keep feasibility under control and are tolerant when the iterations are far from the solution. In Martínez and Pilotta (2005), there is an interesting overview of these algorithms and their main characteristics. In addition, in Gomes and Friedlander (2011) another example of application of IRM to solve bilevel problems can be found.

Originally, the IRM was designed to solve the problem

\[
\begin{align*}
\min_{x} & \quad f(x) \\
\text{s. t.} & \quad C(x) = 0, \\
& \quad x \in \Omega,
\end{align*}
\tag{1}
\]

where \(f : \mathbb{R}^n \to \mathbb{R}\) and \(C : \mathbb{R}^n \to \mathbb{R}^p\) are continuously differentiable functions and \(\Omega \subset \mathbb{R}^n\) is a closed convex set.

The algorithm consists of two well-distinguished stages: feasibility (or restoration stage) and optimality. It is an iterative method which generates a sequence \(x^k\) of feasible iterates with respect to \(\Omega\) but which not necessarily verifies \(C(x) = 0\). Precisely, the restoration phase has the objective of moving the sequence in a direction which generates a reduction of \(||C(x)||\) and an auxiliary sequence \(y^k\), is built. In the second phase, the optimality of \(y^k\) is improved by a minimization of a Lagrangian over a space tangent to \(\{C(x) = 0\}\) at \(y^k\).

The innovative use of the Lagrangian in the optimality phase has to do with the fact that it behaves similarly both in the tangent space and the feasible region. This may not be the case of the non-linear objective function.

The acceptance of a candidate \(\tilde{y}^k\) obtained in the optimization phase depends on the value of a merit function which combines feasibility and optimality.

Andreani et al. in Andreani et al. (2009) studied the possibility of adapting the inexact restoration method to solve bilevel problems. The attractiveness of IRM had to do with the fact that this method may allow solving these problems without reformulating them as single level ones as most approaches for bilevel problems do. What is more, the restoration phase gives the possibility of freely choosing a method which improves feasibility. Consequently, if any globally convergent algorithm is available to efficiently solve the lower level problem, its structure could be exploited.
3.1 IRM adaptation for bilevel problems (IRMbi)

Given a bilevel problem of the type

$$\begin{align*}
\min & \ F(x, y) \\
\text{s. t.} & \ x \in X \\
& \ y = \arg \min_y f(x, y), \\
& \ h(x, y) = 0 \\
& \ y \geq 0,
\end{align*}$$

(2)

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, to adapt the method which originally solves (1), the Karush–Kuhn–Tucker optimality conditions of the lower level problem are considered. In fact, they play the role of the constraint $C(x) = 0,$

$$C(x, y, \alpha, \beta) = 0, \ y \geq 0, \ \beta \geq 0,$$

with

$$C(x, y, \alpha, \beta) = \begin{pmatrix} \nabla_y f(x, y) + \nabla_y h(x, y) & -\beta \\ h(x, y) \\ \beta_1 y_1 \\ \vdots \\ \beta_m y_m \end{pmatrix}.$$

The Lagrangian for the optimality phase is given by

$$L(x, y, \alpha, \beta, \mu) = F(x, y) + C(x, y, \alpha, \beta)^T \mu.$$ 

(3)

The restoration phase searches for a point $z^k = (x^k, \tilde{y}, \tilde{\alpha}, \tilde{\beta})$ “more feasible” than the one built in the previous iteration $s^k = (x^k, y^k, \alpha^k, \beta^k)$. To reach that goal the lower level problem, parameterized in the variable $x^k$ is solved. That is to say, a minimizer $\tilde{y}$ and associated multipliers $(\tilde{\alpha}, \tilde{\beta})$ for the problem

$$\begin{align*}
\min & \ f(x^k, y) \\
\text{s. t.} & \ h(x^k, y) = 0 \\
& \ y \geq 0
\end{align*}$$

(4)

must be found. The point $z^k$ is defined as an intermediate one. Then a linear approximation, around $z^k$, of the feasible region of the simplified problem

$$\begin{align*}
\min & \ F(x, y) \\
C(x, y, \alpha, \beta) = 0 \\
(x, y, \alpha, \beta) & \in \Omega \times \Delta
\end{align*}$$

(5)

is built, where $\Omega \times \Delta \subset \mathbb{R}^{n+3m}$ represents the constraints $x \in X$, $y \geq 0$, $\alpha \in \mathbb{R}^m$, $\beta \geq 0$.

The linear approximation of $C$ in $z^k$ is the tangent space

$$\pi(z^k) = \{ s = (x, y, \alpha, \beta) \in \Omega \times \Delta : C'(z^k)(s - z^k) = 0 \}$$

and the Cauchy tangent direction $r^k_\text{tan} = r_\text{tan}(z^k)$ is

$$r^k_\text{tan} = P_k[z^k - \eta \nabla_s L(z^k, \mu^k)] - z^k,$$

where $P_k[\cdot]$ is the orthogonal projection over the space $\pi_k = \pi(z^k)$ and $L$ the Lagrangian presented above (3). $r^k_\text{tan}$ is a feasible descent direction for $L$ over $\pi_k$. 
For the optimization phase, a trust region centered in $z^k$ is defined

$$\mathbb{B}_{k,i} = \{ s \in \Omega \times \Delta : ||s - z^k|| \leq \delta_{k,i} \},$$

and a candidate $v_{k,i} \in \mathbb{B}_{k,i} \cap \pi_k$ that reduces $L(\cdot, \mu^k)$ is sought. The acceptance of $v_{k,i}$ depends on the value of a merit function. If it is rejected the trust radius is reduced and the scheme moves to an iteration $k, i + 1$ until it finds the minimizer $z_{k,i}^*$. The merit function used is

$$\Psi(s, \mu, \theta) = \theta L(s, \mu) + (1 - \theta)||C(s)||,$$

where $\theta \in (0, 1]$ is a penalty parameter that gives different weights to the Lagrangian function and the feasibility.

With all these considerations the inexact restoration method for bilevel problems (IRMbi) was introduced. What is more, it was proved that there is global convergence to points which satisfy the approximate gradient projection optimality condition (AGP points). Briefly, a feasible point $w$ will satisfy the AGP optimality condition if there exists a sequence $\{w_k\}$ that converges to $w$ and verifies that $r_{\text{van}}(w_k) \to 0$. Introduced in Martínez and Svaiter (2003), AGP is a sequential optimality condition that all local minimizers satisfy without the necessity of checking any constraint qualification (in contrast to KKT) and involves first order differentiability, only. Under additional conditions, namely strict constraint qualifications, AGP points are also KKT, see (Andreani et al. 2018).

### 4 IRM for DAP

#### 4.1 DAP as a bilevel problem

The DAP was presented as a mathematical problem with equilibrium constraints. However, Wardrop’s user equilibrium can be obtained as a solution to an optimization problem, the Traffic Assignment Problem (TAP). The hypotheses under which this is true can be read in Patriksson (1994) or Walpen (2015).

In this case, DAP results in

$$\begin{align*}
\min \quad F(v_d, d) = \eta_1 F_1(v_d) + \eta_2 F_2(d) \\
\text{s. t.} \quad v_d = \arg\min_v \sum_{a \in A} \int_0^{v_a} t_a(s)ds \\
\text{s. t.} \quad \sum_{r \in \mathcal{R}_{pq}} h_{pqr} = d_{pq}, \forall \ (p, q) \in \mathcal{C}, \\
\quad h_{pqr} \geq 0, \forall \ r \in \mathcal{R}_{pq}, \forall \ (p, q) \in \mathcal{C}, \\
\quad \sum_{(p, q) \in \mathcal{C} \cap \mathcal{R}_{pq}} \sum_{(p, q) \in \mathcal{C} \cap \mathcal{R}_{pq}} \delta_{pqr} a h_{pqr} = v_a, \forall; a \in A.
\end{align*}$$

where $\mathcal{R}_{pq}$ is the set of routes $r$ that link the origin $p$ with the destination $q$ and $h_{pqr}$ the flow on such route $r$.

With this reformulation, DAP has bilevel structure. Consequently, there exists the possibility of applying IRMbi to solve DAP. What is more, for the lower level problem TAP, there exist globally convergent methods to obtain the solution and in contrast to most of the available methods for DAP, the complex structure of the traffic assignment problem could be exploited.
4.2 Change of variables for Karush–Kuhn–Tucker (KKT) optimality condition calculation

It would be desirable to have KKT optimality conditions associated with the lower level problem which are easy to handle. However, the original version of TAP has a complex structure of the feasible set due to the presence of two flow variables \( v \) and \( h \). To overcome this difficulty, the TAP is reformulated in the node–arc version presented in Patriksson (1994), as it is done in the non-heuristical approach in Lotito and Parente (2014).

The new flow variable \( X = (x_{ia}^d)_{a \in A, i \in C} \) represents the arc flow disaggregated by demand. \( X \in \mathbb{R}^{|A||C|} \) is a column vector.

In this context, Wardrop’s user equilibrium condition is rewritten as

\[
T(X^*)^T(X - X^*) \geq 0, \forall X \in \tilde{\Omega}(d),
\]

where \( \tilde{\Omega}(d) = \{ X \geq 0 : \Gamma d - MX = 0 \} \).

The function \( T \) and the matrices \( \Gamma \) and \( M \) verify:

\[
T(X) = R^T_\Gamma (RX)
\]

with \( R \in \mathbb{R}^{|A| \times |A||C|} \) defined as \( R = (I_{|A|}, \ldots, I_{|A|}) \), and \( I_{|A|} \) the identity matrix in \( \mathbb{R}^{|A|} \),

\[
\Gamma = \begin{pmatrix}
\gamma^1 & 0 & \cdots & 0 \\
0 & \gamma^2 & \cdots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \gamma^{|C|}
\end{pmatrix} \in \mathbb{R}^{C|A| \times |C|},
\]

with \( \gamma^i = (\gamma^i_k)_{k \in N} \) such that \( \gamma^i_k = \begin{cases} -1 & \text{if } k \text{ is the origin node for the demand } i, \\
1 & \text{if } k \text{ is the destination node for the demand } i, \\
0 & \text{otherwise}, \end{cases} \)

\[
M = \begin{pmatrix}
A & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & A
\end{pmatrix} \in \mathbb{R}^{C|N| \times C|A|},
\]

with \( A \in \mathbb{R}^{|A| \times |A|} \) being the node–arc incidence matrix.

Finally, the KKT system for this reformulation of the lower level problem results in

\[
\begin{cases}
T(X) + M^T \alpha - \beta = 0, \\
\Gamma d - MX = 0, \\
IX \beta = 0, \\
\beta \geq 0, X \geq 0,
\end{cases}
\]

Here, \( \alpha \) is the multiplier vector associated with the equality constraints, \( \beta \) the multiplier vector associated with the inequality constraints and \( IX \) is a matrix of zeros which in its diagonal has the entries of vector \( X \).
4.3 IRMbi for DAP

Having done the change of variables presented above (Sect. 4.2), the goal of applying IRMbi to solve DAP is reestablished.

First, simplified problem (5) for DAP, together with the KKT system obtained, is written:

\[
\begin{align*}
\text{min } F(d, X) &= \eta_1 F_1(RX) + \eta_2 F_2(d) \\
\text{s. t. } T(X) + M^T \alpha - \beta &= 0 \\
\Gamma d - MX &= 0 \\
I_X \beta &= 0 \\
\beta &\geq 0, X \geq 0, d \geq 0
\end{align*}
\]  

(7a)

(7b)

(7c)

(7d)

(7e)

Then choosing

\[
C(d, X, \alpha, \beta) = \left( \begin{array}{c} T(X) + M^T \alpha - \beta \\ \Gamma d - MX \\ I_X \beta \end{array} \right)
\]

and \(\Omega \times \Delta = \{ s = (d, X, \alpha, \beta) \in \mathbb{R}^{|C|} \times \mathbb{R}^{|C||A|} \times \mathbb{R}^{|C||A'|} \times \mathbb{R}^{|C||A|} : d \geq 0 \wedge X \geq 0 \wedge \beta \geq 0 \} \), it results in

\[
\begin{align*}
\text{min } F(d, X) &= \eta_1 F_1(RX) + \eta_2 F_2(d) \\
\text{s. t. } C(s) &= 0 \\
\end{align*}
\]

(8)

Problem (8) plays the role, for DAP, of (5) in the general version of IRMbi.

A linear approximation of the feasible set defined by the constraints of (8) must be considered. For a point \(z = (d^*, X^*, \alpha^* \beta^*)\) it results:

\[
\pi_z = \{ s \in \Omega \times \Delta : C'(z)(s - z) = 0 \},
\]

where

\[
C'(z) = \left( \begin{array}{cccc}
0 & T'(X^*) & M^T & -I_{|A||C|} \\
\Gamma & -M & 0 & 0 \\
0 & I_{\beta^*} & 0 & I_{X^*}
\end{array} \right)
\]

\(I_{|A||C|}\) is the identity matrix of dimensions \(|A||C|\).

The matrix \(C'(z), C'_s\) for simplicity, always exists and can be easily obtained. Consequently, it is possible to obtain the tangent space \(\pi_z\).

4.4 Algorithm

In this section, the complete scheme adapted for DAP is presented. The details of implementation are given in Subsection 4.5.

The following constants are fixed. \(\eta > 0, K > 0, \theta_{-1} \in (0, 1), \delta_{\min} > 0, \tau_1 > 0, \tau_2 > 0, k = 0\).

Let \(s^0 = (d^0, X^0, \alpha^0, \beta^0) \in \mathbb{R}^{|C|} \times \mathbb{R}^{|C||A|} \times \mathbb{R}^{|C||A'|} \times \mathbb{R}^{|C||A|} \) be an initial approximation, \(\mu^0\) an initial approximation of the multiplier and \(\omega^j\) a sequence of positive numbers such that \(\sum_{i=0}^{\infty} \omega^j < \infty\).
Step 1 Penalty parameter initialization
\[ \theta_{k}^{\text{min}} = \min\{1, \theta_{k-1}, \ldots, \theta_{-1}\}, \]
\[ \theta_{k}^{\text{large}} = \min\{1, \theta_{k}^{\text{min}} + \omega_{k}\}, \]
\[ \theta_{k-1} = \theta_{k}^{\text{large}}. \]

Step 2 Restoration Phase

Solve the traffic assignment problem for \( d = d^{k} \) and get the Lagrange multipliers associated with the obtained equilibrium.

Let \( X^* \) be the equilibrium solution and \( \alpha^*, \beta^* \) the associated multipliers.

Define \( z^{k} = (d^{k}, X^{*}, \alpha^{*}, \beta^{*}) \).

Step 3 Cauchy tangent direction

Calculate \( r_{k}^{\text{tan}} = P_{k}[z^{k} - \eta \nabla_{s} L(z^{k}, \mu^{k})] - z^{k} \),

where \( P_{k} \) is the projection onto \( \pi_{k} = \{ s \in \Omega \times \Delta : C'(z^{k})(s - z^{k}) = 0 \} \).

If \( z^{k} = s^{k} \) and \( r_{k}^{\text{tan}} = 0 \) then finish: \( (d^{k}, X^{k}) \) is the solution to DAP;

else, \( i = 0, \delta_{k,0} \geq \delta_{\text{min}} \) and move to Step 4.

Step 4 Optimization Phase in \( \pi_{k} \)

If \( r_{k}^{\text{tan}} = 0 \) set \( v^{k,i} = z^{k} \);

else, calculate \( t_{\text{break}}^{k,i} = \min\{1, \delta_{k,i}/||r_{k}^{\text{tan}}||\} \) and get \( v^{k,i} \) such that

- \( v^{k,i} \in \pi_{k} \),
- \( ||v^{k,i} - z^{k}||_{\infty} < \delta_{k,i} \),
- for some \( t \in (0, t_{\text{break}}^{k,i}] \),
\[ L(v^{k,i}, \mu^{k}) \leq \max\{L(z^{k} + tr_{k}^{\text{tan}}, \mu^{k}), L(z^{k}, \mu^{k}) - \tau_{1}\delta_{k,i}, L(z^{k}, \mu^{k}) - \tau_{2}\}. \]

Step 5 Trial multipliers

If \( r_{k}^{\text{tan}} = 0 \) set \( \mu_{k}^{\text{trial}} = \mu^{k} \);

else, calculate \( \mu_{\text{trial}} \in \mathbb{R}^{2|C||A| + |C'||A'|} \) such that \( |\mu_{k}^{\text{trial}}| \leq K \).

Step 6 Predicted reduction

Define \( \forall \theta \in [0, 1], \)
\[ \text{Pred}_{k,i}(\theta) = \theta[L(s^{k}, \mu^{k}) - L(v^{k,i}, \mu^{k}) - C(z^{k})^{T}(\mu_{k}^{\text{trial}} - \mu^{k})] + (1 - \theta)||C(s^{k})|| - ||C(z^{k})||. \]

Compute \( \theta_{k,i} \) as the maximum \( \theta \in [0, \theta_{k,i-1}] \) which verifies

\[ \text{Pred}_{k,i}(\theta) \geq \frac{1}{2}[||C(s^{k})|| - ||C(z^{k})||]. \]

Define \( \text{Pred}_{k,i} = \text{Pred}_{k,i}(\theta_{k,i}) \).

Step 7 Compare actual and predicted reduction

Calculate \( \text{Ared}_{k,i} = \theta_{k,i}[L(s^{k}, \mu^{k}) - L(v^{k,i}, \mu_{k}^{\text{trial}})] + (1 - \theta_{k,i})||C(s^{k})|| - ||C(v^{k,i})||. \)

If \( \text{Ared}_{k,i} \geq 0.1 \text{Pred}_{k,i} \) UPDATE: \( s^{k+1} = v^{k,i}, \mu^{k+1} = \mu_{k}^{\text{trial}}, \theta_{k} = \theta_{k,i} \), \( \delta_{k} = \delta_{k,i}, k = k + 1 \), and go back to Step 1;

else, choose \( \delta_{k,i+1} \in [0.1\delta_{k,i}, 0.9\delta_{k,i}] \), \( i = i + 1 \), and move to step 4.

4.5 Implementation issues

So far in this work, both phases of the algorithm IRMbi have been revised and analyzed for the DAP. In this section, details of implementation for each step of the scheme are given.
4.5.1 Solving the traffic assignment problem: step 2

For the bilevel problem DAP, there exist algorithms which efficiently solve the lower level problem: TAP. In this work, the Disaggregated Simplicial Decomposition (DSD) algorithm is chosen. Particularly, the version implemented in Lotito et al. (2003) (Scilab Toolbox [5], Lotito et al. 2003) is used, as it gives the possibility of working with the flow variable disaggregated by demand. Precisely, this variable is an auxiliary one and it is available without any modifications to the code of the DSD, except for the output.

Even though the DSD algorithm solves TAP for a fixed demand \(d^k\) providing a solution \(X^k\), the associated multipliers \(\alpha^k\) and \(\beta^k\) must be obtained to build the intermediate point \(z^k = (d^k, X^k, \alpha^k, \beta^k)\), and this cannot be done through the DSD.

However, the KKT system associated with TAP always admits solutions \(\alpha^k\) and \(\beta^k\). That is to say, for a given demand \(d^k\) and the associated equilibrium vector \(X^k\), there exist \(\alpha^k\) and \(\beta^k\) which satisfy the system:

\[
\begin{align*}
T(X^k) + M^T \alpha - \beta &= 0, \\
\Gamma d^k - MX^k &= 0, \\
\beta^T \cdot X^k &= 0, \\
\beta &\geq 0, X^k \geq 0.
\end{align*}
\] (9)

The above assertion is possible due to the linearity of the problem’s constraints and the fact that there is a solution existence proof for TAP.

To obtain a pair \((\alpha, \beta)\) compatible with \((d^k, X^k)\), the square norm of the equality constraints in (9) is minimized in \((\alpha, \beta)\) subject to \(\beta \geq 0\), i.e.,

\[
\min_{\alpha, \beta} \left\| \begin{bmatrix}
M^T & -I \\
0 & I_{X^k}
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta
\end{bmatrix}
+ \begin{bmatrix}
T(X^k) \\
0
\end{bmatrix}
\right\|^2, \quad \text{s. t. } \beta \geq 0,
\] (10)

which leads to a quadratic program which is the problem to be solved. Taking advantage of the structure of the involved matrices, instead of solving (10) directly, the \(|C|\) smaller problems

\[
\min_{\alpha_d, \beta_d} \left\| \begin{bmatrix}
A^T & -I \\
0 & I_{X^k}
\end{bmatrix}
\begin{bmatrix}
\alpha_d \\
\beta_d
\end{bmatrix}
+ \begin{bmatrix}
t(X^k) \\
0
\end{bmatrix}
\right\|^2, \quad \text{s. t. } \beta_d \geq 0
\] (11)

are actually solved. The subroutine `quapro` from ScicosLab 4.4.1 which solves quadratic programs with linear constraints is chosen to solve (11) numerically.

4.5.2 Building the Cauchy tangent direction: step 3

In this step, a projection problem must be solved. Precisely, the projection of a vector \(z - v\) over the tangent space \(\pi_{\bar{z}}\) is needed. To calculate it, the following optimization problem is solved:

\[
\min_s \frac{1}{2} \|z - v - s\|^2, \quad \text{s. t. } C_s'(s - z) = 0.
\] (12)

Here, the vector \(v = -\eta \nabla_s L(z^k, \mu^k)\).

The optimality conditions of the problem are studied. Under appropriate hypotheses which state non-singularity of the matrix \(C_s'\) (see Lemma 4 from Walpen 2015 in the Appendix), the existence of the Cauchy tangent direction is proved.
Again, subroutine \texttt{quapro} is used and the structure of the matrices is exploited, obtaining $|C|$ smaller quadratic problems that are, in fact, the ones to be solved.

Step 3 also includes the stopping condition. This is satisfied by any AGP point. See (Andrea et al. 2009) for more details.

To check the stopping condition for the candidate $z^k$ numerically, the following test is carried out: if

$$||z^k - s^k|| < \varepsilon_1 \text{ and } ||r^k_{\text{tan}}|| < \varepsilon_2,$$

for $\varepsilon_1$ and $\varepsilon_2$ small, the algorithm is stopped.

4.5.3 Finding the candidate $v^{k,i}$ which improves optimality: step 4

The original version of IRMbi gives freedom to choose the method to find $v^{k,i}$ which satisfies all the conditions stated.

The optimization phase is carried out to make a descent of the Lagrangian. The Cauchy tangent direction is always a descent direction for such Lagrangian as it is proved in Martínez (1998). However, in Martínez (2001), it is stated that moving in such direction may not be the best choice.

The candidate $v^{k,i}$ is asked to satisfy simultaneously:

- $v^{k,i} \in \pi_k$,
- $||v^{k,i} - z^k||_{\infty} < \delta^{k,i}$,
- for some $t \in (0, t_{\text{break}}]$,

$$L(v^{k,i}, \mu^k) \leq \max\{L(z^k + tr^k_{\text{tan}}, \mu^k), L(z^k, \mu^k) - \tau_1\delta^{k,i}, L(z^k, \mu^k) - \tau_2\}.$$

The first two conditions guarantee that the candidate $v^{k,i}$ is in the tangent space $\pi_k$ and not so far from $z^k$ (the point that improves factibility in the previous step).

The last condition is a descent one. The three values considered are lower than $L(z^k, \mu^k)$, as it is illustrated in Fig. 1, where max refers to the maximum function, depicted in magenta.

![Fig. 1 Descent condition over the candidate $v^{k,i}$, $L(z) = L(z, \cdot)$, max refers to the maximum function](image-url)
To find such $v^{k,i}$ an algorithm (from now on: Algorithm 2) proposed by Martínez in Martínez (1998) is used. For that approach, the following auxiliary problem is considered:

$$\begin{align*}
\min_v L(v, \mu^k) \\
\text{s. t. } C'_z(v - z^k) &= 0, \\
\|v - z^k\|_\infty &\leq \delta_{k,i}.
\end{align*}$$ (13)

The solution to this linearly constrained problem is undoubtedly a candidate for $v^{k,i}$. However, it is not necessary to solve the problem to find an appropriate $v^{k,i}$. The successive iterates generated by Algorithm 2 which solves (13) are tested and the scheme is stopped as soon as there is one approximation which verifies all the conditions for descent direction for $L(v, \mu^k)$.

Martínez (1998) is used. For that approach, the following auxiliary problem is considered:

**Lemma 1** Let $r_{\pi_k}(v)$ calculated. Precisely, $r_{\pi_k}(v)\in S$ is the feasible set which the constraints in (14) describe, is a feasible direction in $S$. What is more, $r_{\pi_k}(v)$ is a descent direction for $L(v, \mu^k)$.

**Proof** To see that $r_{\pi_k}(v)$ is a feasible direction it is checked that there exists $\varepsilon > 0$ such that $v + \alpha r_{\pi_k}(v) \in S \forall \alpha \in [0, \varepsilon)$. Let $u = v + \alpha r_{\pi_k}(v) = v + \alpha(w^* - v)$, where $w^*$ is a solution to (14) and consequently verifies $w^* \in S$. Re-writing, $u = (1 - \alpha)v + \alpha w^*$ with $S$ convex, it results in $u \in S$ if $\alpha \in [0, 1]$.

To see that $r_{\pi_k}(v)$ is a descent direction for $L(v, \mu^k)$, it is first proved that it is a descent direction for $F(v), r_{\pi_k} \neq 0$ is assumed. Then $w^* \neq v$ and due to the fact that $w^* \in S$, it results in $||w^* - (v - \nabla F(v))||_2^2 < ||v - (v - \nabla F(v))||_2^2$, then $||w^* - v||_2^2 + 2(w^* - v, \nabla F(v)) + ||\nabla F(v)||_2^2 < ||\nabla F(v)||_2^2$, and consequently

$$\langle r_{\pi_k}, \nabla F(v) \rangle < 0.$$ (14)

Taking into account that $r_{\pi_k}$ belongs to $Ker(C'_z)$, in fact,

$$C'_z r_{\pi_k} = C'_z (w^* - v) = C'_z (w^* - z^k + z^k - v) = 0,$$

classifying that $w^*$ and $v$ are both in $\pi_{k,i}$, it results in

$$\langle r_{\pi_k}, \nabla L(v, \mu) \rangle = \langle r_{\pi_k}, \nabla F(v) \rangle < 0$$

as it was desired to prove.

\(\Box\)
Numerically, the descent direction is obtained by solving problem (14) in a similar way to what is done for (12). A maximum of 10 iterations are performed and each approximation is tested as a possible candidate $v^{k,i}$.

5 Numerical experiments

In this section, some implementations of the proposed method are shown. The addressed problems involve small-size and medium-size networks, including well-known examples from the literature. The algorithm was coded using ScicosLab 4.4.1 (see www.scilab.org). The reason for this choice is the availability of CiudadSim 5 toolbox, which includes traffic network examples and the DSD solver. The drawbacks are that this environment is not adequate for the implementation of parallel programming (so the multiple quadratic problems on each iteration had to be sequentially solved) and does not allow for the use of sparse matrices in the quadratic programming solver \textit{quadprog}. Nevertheless, since the main purpose of this paper is to test the applicability of the IRM to an specific problem, DAP, the implementation of the method on a better environment together with numerical comparisons with other schemes is left for future work.

The chosen tolerance parameters were $\varepsilon_1 = 10^{-5}\sqrt{N}$ and $\varepsilon_2 = 10^{-4}\sqrt{N}$, where $N$ is a scaling constant according to the network size, precisely, the dimension of vector $s^0$, i.e., $N = |C|(1 + 2|A| + |N|)$. The maximum number of iterations was 1000. The initial multipliers $\mu^0$ were all chosen equal to 1 and the trial multipliers were heuristically updated. Generally, $\mu_{\text{trial}} = \mu - C(z^k)$, but in some experiments, $\mu_{\text{trial}} = \mu - C(v^k)$ was a better option and, if $\|r_{\text{tan}}\|$ increased, it was reinitialized as $\mu_{\text{trial}} = \mu^0$.

Performance tests were ran on a 2.66 GHz, 8GB RAM, Intel Core i5 processor PC.

5.1 Toy example

A toy problem has been chosen as an expository device to illustrate the applicability of the inexact restoration method for bilevel problems to solve DAP. The transport network has 3 nodes, 4 links and 2 demands represented by the pink arrows (Fig. 2).

The link flow variable, disaggregated by demand, is in this case:

$$X = (x_1^1 x_2^1 x_3^1 x_4^1 x_1^2 x_2^2 x_3^2 x_4^2)^T,$$

![Fig. 2 Validation test example](image)
where $x_j^i$ represents the flow in arc $i$ associated with the demand $j$ and consequently $x_i = x_1^i + x_2^i$, $X$ defined in this way verifies $X \in \mathbb{R}^8$. The matrices $R \in \mathbb{R}^{4 \times 8}$, $\Gamma \in \mathbb{R}^{6 \times 2}$ and $M \in \mathbb{R}^{6 \times 8}$ are in this case:

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\Gamma = \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix},$$

and

$$M = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \end{pmatrix}.$$

The link cost function is given by

$$T(X) = R^T t(RX) = (t_1(x_1) t_2(x_2) t_3(x_3) t_4(x_4) t_1(x_1) t_2(x_2) t_3(x_3) t_4(x_4))^T = (x_1 x_2 x_3 x_4 x_1 x_2 x_3 x_4)^T.$$

The numerical tests are carried out considering a known target demand and observed flows in arcs 1 and 2 which correspond to an affectation of such demand. The purpose of this approach is to guarantee that there exists a global minimizer where the objective function of the associated DAP assumes value zero. The constants were fixed as follows: $d_1 = 1.5$, $d_2 = 1.75$, $\bar{x}_1 = 1.5833333$, $\bar{x}_2 = 1.6666667$, $\eta_1 = 0.5$, $\eta_2 = 0.5$, $F_1 = ||x - \bar{x}||^2$, $F_2 = ||d - \bar{d}||^2$. Four experiments were performed ($TE_i$, $i = 1, 2, 3, 4$), with different initial values for the variable $s$ given by

$$s^{0i} = (d^{0,i}, (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0), (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0))^T,$$

where $d^{0,1} = (1, 2)$, $d^{0,2} = (1, 1)$, $d^{0,3} = (1, 1.5)$ and $d^{0,4} = (1.8, 2)$. Details of the experiments and the obtained results are shown in Table 1.

For the total of the four experiments convergence to a global optimum of the problem was registered.

| Table 1: Experiments details |
|-----------------------------|
| Exp | It | $\|r_{tan}\|$ | $\|s - z\|$ | $F(d^*, x^*)$ | Time (s) | Demands |
|-----|----|----------------|----------------|----------------|----------|---------|
| $TE_1$ | 15 | 4.7e−04 | 0.0000 | 4.1e−07 | 0.14 | (1.49909, 1.75000) |
| $TE_2$ | 16 | 3.2e−04 | 0.0000 | 2.6e−07 | 0.13 | (1.49928, 1.74991) |
| $TE_3$ | 16 | 3.1e−04 | 0.0000 | 4.1e−07 | 0.14 | (1.49909, 1.74996) |
| $TE_4$ | 11 | 2.8e−04 | 0.0000 | 1.7e−07 | 0.10 | (1.50058, 1.75006) |
5.2 Codina–Barceló network

This example was introduced in Codina and Barceló (2004) and presents a network with 9 nodes, 12 arcs and 2 od pairs (see Fig. 3).

The link cost function \( t(x) \) is computed as \( t_a(x) = t_{0,a}(1 + 10(x_a/q_a)^4) \), where \( t_{0,a} = (0.10, 0.10, 0.20, 0.70, 0.50, 1.00, 0.10, 0.10, 1.00, 0.08, 0.70) \) and \( q = (200, 200, 100, 100, 100, 100, 200, 100, 100, 200, 100, 100) \).

A Wardrop equilibrium flow \( \bar{x} \) for a demand \( d = (d_1, d_2) = (400, 400) \) is \( \bar{x}_1 = 400.00, \bar{x}_2 = 400.00, \bar{x}_3 = 189.17, \bar{x}_4 = 0.00, \bar{x}_5 = 431.23, \bar{x}_6 = 368.76, \bar{x}_7 = 400.00, \bar{x}_8 = 179.59, \bar{x}_9 = 220.41, \bar{x}_{10} = 179.59, \bar{x}_{11} = 400.00 \) and \( \bar{x}_{12} = 210.82 \). Two objective functions are proposed in Codina and Barceló (2004), given by \( G_1(x) = \frac{1}{2}((x_4 - \bar{x}_4)^2 + (x_8 - \bar{x}_8)^2) \) and \( G_2(x) = \frac{1}{2}((x_4 - \bar{x}_4)^2 + (x_8 - \bar{x}_8)^2) \), where \( \bar{x}_4, \bar{x}_8 \) and \( \bar{x}_{12} \) are the observed flows in arcs 4, 8 and 12, respectively.

Two more objective functions are considered here, given by \( G_3(d, x) = \frac{1}{2}((d_1 - \bar{d}_1)^2 + (d_2 - \bar{d}_2)^2) + G_1(x) \) and \( G_4(d, x) = \frac{1}{2}((d_1 - \bar{d}_1)^2 + (d_2 - \bar{d}_2)^2) + G_2(x) \), which also adjust the demands to the target demand pattern \( \bar{d} = (\bar{d}_1, \bar{d}_2) = (390, 410) \). Note that both \( G_1 \) and \( G_2 \) have no demand target, so it is reasonable to expect that the algorithm could obtain a zero value for these objective functions. For the last two cases, there is no guarantee that the observed flows belong to a Wardrop equilibrium vector for the demand target, so the optimal value could be strictly positive. The numerical results confirm this expectation, as it is illustrated in Table 2, where also the number of iterations, the obtained norms for the stopping criteria and the computational times for each experiment \( CB_i, i = 1, 2, 3, 4 \) are shown.

Finally, the obtained demands and flows are shown in Tables 3 and 4, where the flows of the observed arcs are highlighted in bold for each objective function.

![Network example of Codina–Barcelo](image)

**Fig. 3** Network example of Codina–Barcelo
Table 2 Algorithm outputs for Codina–Barceló network experiments

| Exp. | Iter | $\|\hat{r}\|_2$ | $\|s-z\|$ | $G^1(d^*, x^*)$ | Time (s) |
|------|------|-----------------|-----------|-----------------|---------|
| $CB_1$ | 325  | 2.9e-004 | 0.000000 | 0.00010 | 13.62   |
| $CB_2$ | 334  | 8.0e-004 | 0.000000 | 0.00020 | 13.97   |
| $CB_3$ | 514  | 7.0e-004 | 0.000000 | 26.87730 | 20.38   |
| $CB_4$ | 846  | 6.9e-004 | 0.000000 | 15.64569 | 31.03   |

Table 3 Obtained flows and demands for experiments $CB_1$ and $CB_2$

| Dem | Experiment $CB_1$ | $d_1$ | $d_2$ | Experiment $CB_2$ | $d_1$ | $d_2$ |
|-----|------------------|------|------|------------------|------|------|
|     | $x_a$ | $x_a^1$ | $x_a^2$ | $x_a$ | $x_a^1$ | $x_a^2$ |
| 1   | 399.999 | 399.999 | 0 | 400.004 | 400.004 | 0 |
| 2   | 399.999 | 0 | 399.999 | 400.002 | 0 | 400.002 |
| 3   | 189.181 | 189.181 | 0 | 189.181 | 189.181 | 0 |
| 4   | 0 | 0 | 0 | 0 | 0 | 0 |
| 5   | 431.238 | 210.818 | 220.420 | 431.241 | 210.822 | 220.418 |
| 6   | 368.761 | 189.181 | 179.579 | 368.764 | 189.181 | 179.583 |
| 7   | 399.999 | 399.999 | 0 | 400.004 | 400.004 | 0 |
| 8   | 179.579 | 0 | 179.579 | 179.583 | 0 | 179.58341 |
| 9   | 220.420 | 0 | 220.420 | 220.418 | 0 | 220.418 |
| 10  | 179.579 | 0 | 179.579 | 179.583 | 0 | 179.583 |
| 11  | 399.999 | 0 | 399.999 | 400.002 | 0 | 400.002 |
| 12  | 210.818 | 210.818 | 0 | 210.822 | 210.822 | 0 |

Table 4 Obtained demands and flows for experiments $CB_3$ and $CB_4$

| Dem | Experiment $CB_3$ | $d_1$ | $d_2$ | Experiment $CB_4$ | $d_1$ | $d_2$ |
|-----|------------------|------|------|------------------|------|------|
|     | $x_a$ | $x_a^1$ | $x_a^2$ | $x_a$ | $x_a^1$ | $x_a^2$ |
| 1   | 391.091 | 391.091 | 0 | 390.00 | 390.00 | 0 |
| 2   | 406.961 | 0 | 406.961 | 406.563 | 0 | 406.563 |
| 3   | 183.505 | 183.505 | 0 | 183.034 | 183.034 | 0 |
| 4   | 0 | 0 | 0 | 0 | 0 | 0 |
| 5   | 430.277 | 207.585 | 222.691 | 429.524 | 206.965 | 222.559 |
| 6   | 367.775 | 183.505 | 184.269 | 367.038 | 183.034 | 184.003 |
| 7   | 391.091 | 391.091 | 0 | 390.00 | 390.00 | 0 |
| 8   | 184.269 | 0 | 184.269 | 184.003 | 0 | 184.00385 |
| 9   | 222.691 | 0 | 222.691 | 222.559 | 0 | 222.559 |
| 10  | 184.269 | 0 | 184.269 | 184.003 | 0 | 184.003 |
| 11  | 406.961 | 0 | 406.961 | 406.563 | 0 | 406.563 |
| 12  | 207.585 | 207.585 | 0 | 206.965 | 206.965 | 0 |
Table 5 Outputs for Steenbrink and Sioux Falls experiments

| Exp | Pairs | Iter | $\|r_{n\alpha}\|$ | $\|s - z\|$ | $F(d^\ast, x^\ast)$ | Time (s) |
|-----|-------|------|-----------------|-----------------|------------------------|---------|
| St1 | 12    | 55   | 2.6e−04         | 0.00000         | 0.0000082              | 19.78   |
| St2 | 12    | 126  | 2.2e−04         | 0.00000         | 0.0000084              | 57.86   |
| St3 | 12    | 109  | 1.4e−04         | 0.00000         | 0.0000019              | 43.15   |
| St4 | 12    | 256  | 7.9e−04         | 0.00000         | 15.83405               | 98.39   |
| St5 | 12    | 218  | 8.1e−04         | 0.00000         | 12.72183               | 78.37   |
| St6 | 12    | 214  | 7.2e−04         | 0.00000         | 23.38262               | 81.32   |
| SF1 | 107   | 13   | 1.2e−02         | 0.00004         | 0.00013                | 167.22  |
| SF2 | 171   | 32   | 9.2e−03         | 0.00000         | 0.00001                | 688.36  |
| SF3 | 214   | 29   | 1.7e−02         | 0.00006         | 0.00044                | 782.66  |
| SF4 | 107   | 1000 | 3.5e−01         | 0.00001         | 0.39748                | 14953.75|
| SF5 | 171   | 31   | 1.1e−02         | 0.00000         | 0.01238                | 636.21  |
| SF6 | 214   | 251  | 1.2e−02         | 0.00000         | 0.02381                | 7370.49 |

5.3 Steenbrink and Sioux Falls networks

Two well-known examples from literature are considered here. The first one is the Steenbrink network (Steenbrink 1976), which consists of 9 nodes, 36 arcs and 12 od pairs. The link cost function is $t_a(x) = t_0 + m_a x$ with $t_0$ between 0 and 1 and $m_a$ between $3 \times 10^4$ and $2 \times 10^3$. The second one is the Sioux Falls network that has some variants in the literature (Abdulaal and LeBlanc 1979; Suwansirikul et al. 1987). The analyzed example here consists 24 nodes and 76 links, with different od pairs in each addressed problem (from 107 to 214 od pairs) and it can be considered as a medium-size network. Here, the link cost function is $t_a(x) = t_0 + m_a x^4$ with $t_0$ between 0.02 and 0.1 and $m_a$ between $10^8$ and $2 \times 10^5$.

Six different experiments were performed for each network. In all cases, the values of the observed arcs were taken from a Wardrop equilibrium link flow $\bar{x}$ associated with a demand $d$, choosing random observed arcs (15% of the arcs). Problems $St_i$ correspond to Steenbrink network while problems $SF_i$ correspond to Sioux Falls network, where the demand target $\bar{d} = d$ is considered for $i = 1, 2, 3$ and a random demand target $\bar{d} \neq d$ (a perturbation of $d$) is considered for $i = 4, 5, 6$. So, as in the previous examples, it is expected that the proposed method could obtain zero values of the objective functions for the first cases and positive values when the observed arcs do not necessarily belong to a Wardrop equilibrium flow associated with the demand target $\bar{d}$. The obtained results are shown in Table 5.

Except for experiment $SF_4$, the algorithm obtained points that satisfy the stopping criteria. In experiment $SF_4$, the scheme started descending but, after some iterations, it produced very close iterates, keeping the norm of the Cauchy direction almost constant and far away from zero. This behavior was also observed in some numerical trials for other examples, but minor changes in the algorithm parameters or in the updating strategy for $\mu_{trial}$ allowed to obtain the reported results. In the case of $SF_4$, none of these alternatives worked and the scheme reached the maximum number of iterations without obtaining a solution.
6 Conclusions and future work

In this work, an application of the inexact restoration method for bilevel problems to a real problem, the DAP, has been presented. The advantages of the method have been exploited. Few of the available methods to treat DAP maintain the structure of the lower level problem TAP as IRMbi does. Most methods deal with the single level version of the DAP. What is more, for IRMbi there are proofs of convergence to AGP points while others are just heuristics or descent methods.

In the feasibility phase the TAP was solved exactly through the available software. In the optimality phase, a descent method for the Lagrangian proposed by Martínez was implemented.

Numerical tests over a small- and medium-size networks were carried out and convergence to global optimum was obtained in most of the cases.

When applied to real-size networks, this formulation leads to very large-scale problems. Consequently, the next step will be directed towards developing at least the basic tools of CiudadSim on a more adequate computational environment. This will allow to use better quadratic programming solvers and to implement parallel programming for the multiple quadratic programs on steps 2, 3 and 4 of the scheme.

Acknowledgements This work was partially supported by CONICET (PIP 2012-2014 No. 0286), FONCyT (Pict 2012-2212), SPU (3325/15c Proy.31-65-128) and Universidad Nacional de Rosario (ING 428), Argentina.

Appendix

Lemma 4 in Walpen (2015): sufficient condition for the existence of the Cauchy tangent direction

Under hypothesis A7, A8 and A9 of Andreani et al. (2009), it results: $C'_z$ non-singular. Then the tangent Cauchy direction, $r_{\text{tan}}$, exists and it is unique.

A7. For each $d \geq 0$, the lower level problem’s solution $X^*(d)$ is a regular point (the active constraints in $X^*(d)$ are linearly independent).

A8. The matrices associated with second-order conditions of the lower level problem are positive definite in a particular subspace.

A9. Every solution to $C(s) = 0, s \in \Omega$ verifies:

$$X_i(d) + \bar{\lambda}_i(d) > 0, \quad \forall i = 1, \ldots, |\mathbb{A}| || C ||.$$

References

Abduaal M, LeBlanc LJ (1979) Continuous equilibrium network design models. Transp Res B 13:19–32
Andreani R, Castro SLC, Chela JL, Friedlander A, Santos SA (2009) An inexact-restoration method for nonlinear bilevel programming problems. Comput Optim Appl 43:307–328
Andreani R, Martínez JM, Ramos A, Silva PIS (2018) Strict constraint qualifications and sequential optimality conditions for constrained optimization. Math Oper Res 43(3):693–717
Chen Y, Florian M (1996) O-d demand adjustment problem with congestion: Part i. model analysis and optimality conditions. Adv Methods Transp Anal 1–22
Codina E, Barceló J (2004) Adjustment of O-D trip matrices from observed volumes: an algorithmic approach based on conjugate directions. Eur J Oper Res 155:535–557
Codina E, Montero L (2006) Approximation of the steepest descent direction for the o-d matrix adjustment problem. Ann Oper Res 144:329–362
Gomes F AM, Friedlander A (2011) Solution of a truss topology bilevel programming problem by a means of an inexact restoration method. Comput Appl Math 30.1
Izmailov AF, Pogosyan AL, Solodov MV (2012) Semismooth Newton method for the lifted reformulation of mathematical programs with complementarity constraints. Comput Optim Appl 51(1):199–221
Lotito PA, Mancinelli EM, Quadrat JP, Wynter L (2003) The traffic assignment toolboxes of scilab. INRIA - Rocquencourt
Lotito PA, Parente LA (2014) Estimación de Matrices O-D Mediante Técnicas de elevación y Métodos de Newton Semiusuves. Mecánica Computacional, 33:2457-2467, AMCA
Lundgren JT, Peterson A (2008) A heuristic for the bilevel origin-destination-matrix estimation problem. Transp Res B 42:339–354
Martínez JM, Pilotta EA (2005) Inexact-restoration methods for nonlinear programming: Advances and perspectives. In L. Qi, K. Teo and X. Yang editors, Optimization and Control with Applications, pages 271–292, Springer
Martínez JM (1998) Two-phase model algorithm with global convergence for nonlinear programming. J Optim Theory Appl 96(2):397–436
Martínez JM (2001) Inexact-restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming. J Optim Theory Appl 111(1):39–58
Martínez JM, Pilotta EA (2000) Inexact-restoration algorithm for constrained optimization. J Optim Theory Appl 104(1):135–163
Martínez JM, Svaiter BF (2003) A practical optimality condition without constraint qualifications for nonlinear programming. J Optim Theory Appl 118(1):117–133
Nguyen S (1977) Estimating an o-d matrix from network data: A network equilibrium approach. Publication 87, Centre de recherche sur les transports (CRT), Université de Montréal, Montréal, Canada
Patriksson M (1994) The Traffic Assignment Problem. Models and Methods. Topics in Transportation VSP, The Netherlands
Spiess H (1990) A descent based approach for the od matrix adjustment problem. Publication 693, Centre de recherche sur les transports (CRT), Université de Montréal, Montréal, Canada
Steenbrink PA (1976) Optimization of transport networks. Wiley, Hoboken
Stein O (2012) Lifting mathematical programs with complementarity constraints. Math Program 131:71–94
Suwansirikul C, Friesz TL, Tobin RL (1987) Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. Transp Sci 40:540–542
Walpen J (2015) Sobre la resolución del problema de ajustar la matriz origen destino en una red de tráfico vehicular congestionada. Tesis de Doctorado en Matemática - FCEIA - Universidad Nacional de Rosario,Director: Pablo A. Lotito. Codirectora: Elina M. Mancinelli
Walpen J, Mancinelli EM, Lotito PA (2015) A heuristic for the od matrix adjustment problem in a congested transport network. Eur J Oper Res 242:807–819
Yang H, Sasaki T, Iida Y, Asakura Y (1992) Estimation of origin-destination matrices from link traffic counts on congested networks. Transp Res 26B:417–434

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.