

Penalization and augmented Lagrangian for O-D demand matrix estimation from transit segment counts

María Victoria Chávez Hernández, Lorenzo Héctor Juárez Valencia & Yasmín Águeda Ríos Solís

To cite this article: María Victoria Chávez Hernández, Lorenzo Héctor Juárez Valencia & Yasmín Águeda Ríos Solís (2018): Penalization and augmented Lagrangian for O-D demand matrix estimation from transit segment counts, *Transportmetrica A: Transport Science*, DOI: [10.1080/23249935.2018.1546780](https://doi.org/10.1080/23249935.2018.1546780)

To link to this article: <https://doi.org/10.1080/23249935.2018.1546780>



Accepted author version posted online: 13 Nov 2018.
Published online: 16 Nov 2018.



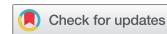
Submit your article to this journal 



Article views: 6



View Crossmark data 



Penalization and augmented Lagrangian for O-D demand matrix estimation from transit segment counts

María Victoria Chávez Hernández^a, Lorenzo Héctor Juárez Valencia^a and Yasmín Águeda Ríos Solís^{b,c}

^aMathematics Department, Universidad Autónoma Metropolitana Unidad Iztapalapa, Mexico City, Mexico;

^bSystems Engineering, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, NL, Mexico;

^cLaboratoire d'Informatique de Paris Nord, Université Paris 13, France

ABSTRACT

In this paper we consider some penalized quadratic models to update Origin-Destination (O-D) matrices in transit networks from observed flows. These models look for the closest O-D matrix to an outdated one, which reproduces some observed segment flows. We demonstrate that the solution of these penalized models converges to the solution of the Spiess model when the penalty parameter increases to infinity. Another contribution is the introduction of an augmented Lagrangian model and its iterative solution by a dual ascent technique and the method of multipliers. This approach yields high-quality solutions with low CPU time and it is tested with two networks: the Winnipeg transit network, which has 23716 O-D pairs; and the transit network of the metropolitan area of the Valley of Mexico with more than 2 million of O-D pairs. For some instances, extracting the null coefficients from the old O-D matrix reduces the computational cost even further.

ARTICLE HISTORY

Received 8 March 2018

Accepted 5 November 2018

KEYWORDS

O-D matrix; transit assignment; penalization; augmented Lagrangian; conjugate gradient method

1. Introduction

The flow of passengers between each pair of zones in a transit network is a matrix known as origin-destination demand matrix (O-D matrix). During the last decades, the origin-destination demand estimation problem has represented an important challenge in the context of transportation planning. While most of the research has been oriented to road networks, here we focus on public transit networks such as buses or subway. O-D matrices are usually obtained from surveys every certain period of time (typically, 10 or more years), depending on the population dynamics (Bera and Rao 2011). Since it is expensive to carry out those surveys and it may take a long time to process all the information, an alternative is to estimate (or update) the O-D matrix from easy to obtain data, such as the observed flow in some strategic places of the network and other available information like an out of date or seed O-D matrix, zone information, mobile phone data or smart card fare data (Alsgar et al. 2016; Bierlaire 1995; Doblas and Benítez 2005; Heidari, Moayedi, and Ali Abbaspour 2017; Kumar et al. 2016; Nuzzolo and Comi 2016). A different but related

problem is about determining the number and locations of counting stations for improving O-D trip table estimation. In Chootinan, Chen, and Yang (2005), the aim is to maximize the coverage while minimizing resource utilization criteria via genetic algorithms. In Chen et al. (2007), the authors develop strategies in the screen-line-based traffic counting location model for selecting additional traffic counts. Once the O-D matrix is updated, then the assignment problem determines the flow volume (number of persons) traveling on each link (segment) of the network. Of course, with a good estimate of the O-D matrix, it is more likely to assimilate observed data, improve the transit models and the corresponding forecasts.

The O-D matrix estimation problem may be divided into the static and dynamic categories. In the static case, a certain period of the day is studied, as the morning rush hour. In the dynamic problem (Antoniou et al. 2016; Ashok and Ben-Akiva 2002; Cascetta et al. 2013; Cipriani et al. 2011; Frederix, Viti, and Tampère 2013; Hu et al. 2017; Shafiei, Saberi, and Sarvi 2016; Verbas, Mahmassani, and Zhang 2011), several periods of the day are considered so the rate at which the flow change along the day must also be modeled. In this study, we concentrate on the static ones since the models, in this case, are the basis of the dynamic ones (Etemadnia and Abdelghany 2009). Among the formulations and methods studied in the literature, several approaches seek to minimize the Mahalanobis distance between the observed data and the estimated values (Bell 1991; Cascetta and Nguyen 1988). Others, like Noriega and Florian (2009), Verbas, Mahmassani, and Zhang (2011) and Shafiei, Saberi, and Sarvi (2016) consider a weighted average of the square of the Euclidean distances between the observed volumes and estimated ones and the squared difference of the seed matrix and the new estimated one. Statistical approaches have been studied to estimate the actual O-D matrix: maximum likelihood, generalized least squares, and Bayes estimators. In Cascetta (1984) and Cascetta and Nguyen (1988), it is claimed that the least squares estimator with a linear assignment procedure is the best-unbiased estimator if both the seed matrix and the observed link volumes are unbiased. The robustness of generalized least squares has been exploited to estimate static or dynamic O-D matrices since it permits the combination of survey data and flow count data, allowing the incorporation of the relative accuracy of the two data sources (Bell 1991; Cascetta et al. 2013; Fujita, Yamada, and Murakami 2016; Malapert and Kuusinen 2017).

In this work, the problem of updating an O-D matrix from transit segments counts is formulated as a constrained convex optimization problem for which the objective function corresponds to a distance function between an *a priori* demand $\hat{\mathbf{g}} = \{\hat{g}_{pq}\}_{pq \in \mathcal{PQ}}$ and the resulting one $\mathbf{g} = \{g_{pq}\}_{pq \in \mathcal{PQ}}$, where $\hat{\mathbf{g}}$ and \mathbf{g} are *a priori* and the resulting demand vectors, respectively. The constraints force the assigned volumes $\mathbf{v} = \{v_a\}_{a \in A}$ to correspond to the observed volumes $\hat{\mathbf{v}} = \{\hat{v}_a\}_{a \in \hat{A}}$ on the count post segments or links $a \in \hat{A} \subset A$. Here, $pq \in \mathcal{PQ}$ denotes an O-D pair, A is the set of all segments on the transit network. The number of segments in A is usually much larger than the number of segments in \hat{A} . In particular, the penalized model (introduced in Chávez and Juárez 2014, 2016; Juárez and Chávez 2014) belongs to the general programming problem

$$\min_{\mathbf{g}} F_1(\mathbf{g}; \hat{\mathbf{g}}) + F_2(\mathbf{v}; \hat{\mathbf{v}}), \quad (1)$$

where F_1 and F_2 are distance functions and $\mathbf{v} = \text{Assign}(\mathbf{g})$ is understood as an equilibrium transit assignment. For instance, the linear model based on optimal strategies introduced

by Spiess and Florian (1989), which is used in this work. After a linear transit assignment, we get the path proportions π_{pq}^a ; each one represents the probability that a person uses link a to go from p to q . These probabilities can be arranged in a matrix $P = \{\pi_{a,i}\}$ where now i represents the O-D pair pq , this means that P has dimensions $m \times n$ with m the number of transit segments and n the number of O-D pairs. Thereby, a transit assignment can be represented as a product of a matrix P and a demand vector \mathbf{g} to get the transit flow $\mathbf{v} = Pg$ over the segments in the network. The problem (1) can be seen as an inverse problem (Michau et al. 2017), where the direct problem consists in computing the segment flows when the O-D matrix is known, and the inverse problem looks for the estimation of the O-D matrix when measurements of flow are known for some transit segments. In this work, we focus on the inverse problem for large size networks.

In the past, such models were of little practical relevance because of ‘the immense computation time and storage requirements that arise in practical implementations and which limit these approaches to very small problem sizes only’, according to Spiess (1990). So, trying to overcome this shortcoming, he proposed the following simple model:

$$\min_{\mathbf{g}} Z(\mathbf{g}) = \frac{1}{2} \sum_{a \in \hat{A}} (v_a - \hat{v}_a)^2 = \frac{1}{2} \|Pg - \hat{\mathbf{v}}\|^2, \quad (2a)$$

$$\text{s.t. } g_{pq} \geq 0 \quad \text{for all } pq \in \mathcal{PQ}, \quad (2b)$$

which is highly underdetermined and has an infinite number of solutions, this means that there are infinite many O-D matrices which recover equally well the observed volumes $\hat{\mathbf{v}}$. To overcome this degeneration, Spiess proposed to choose the closest adjusted matrix to the old one and which also keeps the same structure. Thus, he introduced a multiplicative steepest descent (MSD) method:

$$g_{pq}^{\ell+1} = \begin{cases} \hat{g}_{pq}, & \text{for } \ell = 0, \\ g_{pq}^\ell \left(1 - \gamma_\ell \frac{\partial Z(\mathbf{g}^\ell)}{\partial g_{pq}} \right), & \text{for } \ell = 1, 2, \dots \end{cases} \quad \text{s.t. } \gamma_\ell \frac{\partial Z(\mathbf{g}^\ell)}{\partial g_{pq}} \leq 1, \quad (3)$$

for every $pq \in \mathcal{PQ}$ where ℓ is the iteration counter and γ_ℓ is the step size length at iteration ℓ . This multiplicative algorithm starts with a seed matrix (that can be chosen as the obsolete matrix) $\hat{\mathbf{g}}$ and keeps its structure during the iterations. That is, if $g_{pq}^\ell = 0$, then $g_{pq}^{\ell+1} = 0$. The simplicity of the method makes it applicable to large-scale networks. In Chávez and Juárez (2014), we adapted a multiplicative conjugate gradient (MCG) algorithm in order to improve the Spiess solution with a quadratic model of the type (1), which considers the old O-D matrix in the objective function. The multiplicative methods are inspired to keep the structure of the seed or outdated O-D matrix, which is an assumption for short-term planning, so big changes in the demand are not expected. Some authors have relaxed this condition and formulate other models with adequate solution algorithms (see Bierlaire and Toint 1995; Codina and Barceló 2000; Codina, García, and Marín 2006; Cipriani et al. 2011; Florian and Chen 1995; Lundgren and Peterson 2008; Shafiei, Saberi, and Sarvi 2016; Shen and Winter 2012; Xie, Kockelman, and Travis 2011). In this work, we propose a better estimation of the O-D demand matrix obtained with an augmented Lagrangian approach and the method of multipliers (Boyd et al. 2010; Nocedal and Wright 2006) to force the non-negativity of the O-D matrix coefficients. This approach allows us to avoid

iterative algorithms with multiplicative structure and obtain better results with more or less the same reduced computing time. Related articles where an augmented Lagrangian method is employed in a different way or context, include those of Balakrishnan, Magnanti, and Wong (1989); Bierlaire and Toint (1995) and Doblas and Benitez (2005), for instance.

Other authors have formulated the problem as a linear program in order to obtain more effective algorithms for large-scale networks (Hu et al. 2017; Michau et al. 2017; Pitombeira-Neto, Grangeiro, and Carvalho 2016; Sherali, Sivanandan, and Hobeika 1994). For instance, Hu et al. (2017) claimed that ‘due to the linear structure, their model is more computationally effective and solvable on large real-life networks compared with the commonly seen least-square formulation, which is computationally difficult and inefficient for large networks’. However, they apply their methodology only to a small size network; with 80 traffic zones, 830 links and 395 nodes.

Quadratic models, with suitable solutions algorithms, have demonstrated to be efficient and very competitive for large problems in many different applications. Nevertheless, with the increase on computational power along the years and the success of quadratic models in transit networks (Verbas, Mahmassani, and Zhang 2011), there is much work to do in order to develop effective algorithms to help estimate or update O-D demand matrices for large-scale transit and road networks with quadratic models. Our goal in this work is to study some penalized quadratic models to improve the accuracy of solutions and reduce the computational cost.

First, we show that the set of solutions of the penalized models converge to the solution of Spiess model. Then, we incorporate the non-negativity conditions directly in the model with an augmented Lagrangian approach and the method of multipliers. Also, we employ a direct reduction of the problem to improve even more the computational time. We show the effectiveness of the proposed models and algorithms with two networks: the Winnipeg transit network and the transit network based on the metropolitan area of Valley of Mexico (MAVM), which is constituted by Mexico City itself composed of 16 municipalities and 59 adjacent municipalities of the state of Mexico and 1 of the state of Hidalgo.

The rest of this paper is organized as follows. In Section 2 we formulate our penalized model to update an O-D matrix, show its convergence to the Spiess problem when the penalty parameter tends to infinity and describe the MCG algorithm to solve the penalized model. In Section 3, we introduce an augmented Lagrangian model, discuss the properties of this approach, like the relationship between the two penalty parameters implicated in the augmented Lagrangian, and describe the solution algorithm of the dual ascent technique and the method of multipliers (DAMM). In Section 4, we consider a reduction of the problem to make easier the implementation for large networks and include constraints about the productions and attractions to the Lagrangian formulation. The corresponding results for two networks are shown in Section 5. Finally, in Section 6 we include some conclusions.

2. The penalized model and its MCG solution

2.1. The penalized model

We begin with the original idea of Spiess:

Find a demand matrix \mathbf{g} that fits the volume data $\hat{\mathbf{v}}$ and is closest to the seed matrix $\hat{\mathbf{g}}$.

The proposed model of Spiess (2) may be formulated using vector-matrix notation as the minimization of

$$\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2 \quad \text{over the set } \mathcal{U} = \overline{\mathbb{R}_+^n}, \quad (4)$$

where $P \in \mathbb{R}^{m \times n}$ is the matrix associated to a linear assignment model for the given transit network, m is the number of coefficients of $\hat{\mathbf{v}}$, n is the number of variable coefficients of \mathbf{g} (viewed as a vector), $\|\cdot\|_m$ indicates the norm in \mathbb{R}^m , \mathbb{R}_+^n indicates the subset of vectors of \mathbb{R}^n with positive entries and $\overline{\mathbb{R}_+^n}$ is the closure of \mathbb{R}_+^n , that is, the set of vectors with positive or null entries. It is supposed that $m \ll n$, that is to say, it is much cheaper to get volume measurements in some transit segments of the network than actually estimate \mathbf{g} by direct surveys. Notice that the model (4) is an ill-posed problem since it is highly under-determined, allowing infinite many solutions. However, due to its algorithmic structure, the MSD method (3), with starting guess $\mathbf{g}^0 = \hat{\mathbf{g}}$ to find a solution of (4), gives an approximation to the solution that is closest to $\hat{\mathbf{g}}$ and keeps the zeros in the process. A variant of the above model was introduced in Chávez and Juárez (2014), where the closeness condition between the no longer in use O-D matrix $\hat{\mathbf{g}}$ and the adjusted one \mathbf{g} is included directly in the objective function:

Given $\hat{\mathbf{g}} \in \overline{\mathbb{R}_+^n}$ and $\hat{\mathbf{v}} \in \overline{\mathbb{R}_+^m}$, find $\mathbf{g} \in \overline{\mathbb{R}_+^n}$ that minimizes

$$J(\mathbf{g}) = \frac{1}{2} \|\mathbf{g} - \hat{\mathbf{g}}\|_n^2 \quad \text{over the set } \mathcal{V} = \left\{ \mathbf{g} \in \overline{\mathbb{R}_+^n} : P\mathbf{g} = \hat{\mathbf{v}} \right\}. \quad (5)$$

This formulation of the problem allows the incorporation of the restriction $P\mathbf{g} = \hat{\mathbf{v}}$ into the programming model, obtaining the following penalized one:

Given $\hat{\mathbf{g}} \in \mathcal{U}$ and $\hat{\mathbf{v}} \in \overline{\mathbb{R}_+^m}$, find $\mathbf{g} \in \mathcal{U}$ that minimizes

$$J_k(\mathbf{g}) = \frac{1}{2} \|\mathbf{g} - \hat{\mathbf{g}}\|_n^2 + \frac{k}{2} \|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2, \quad (6)$$

where $k > 0$ is the penalty parameter. The larger is parameter k the more will be penalized the squared norm of the difference $P\mathbf{g} - \hat{\mathbf{v}}$. This is a particular case of model formulation (1).

2.2. Convergence of the penalized model

Given the matrix P , the probabilities obtained from the solution of the assignment problem, suppose that P is a full range matrix and let us consider the particular solution $\bar{\mathbf{g}}$ of problem (5). For each $k > 0$, let \mathbf{g}_k be the unique solution of the minimization problem

$$\begin{aligned} \mathbf{g}_k &\in \mathcal{U}, \\ J_k(\mathbf{g}_k) &\leq J_k(\mathbf{g}) \quad \text{for all } \mathbf{g} \in \mathcal{U}. \end{aligned} \quad (7)$$

Then, by (5) and (7) we obtain the following inequality:

$$\frac{1}{2} \|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 + \frac{k}{2} \|P\mathbf{g}_k - \hat{\mathbf{v}}\|_m^2 \leq \frac{1}{2} \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2, \quad \text{for every } k > 0,$$

which in turn implies the following two inequalities

$$\|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 \leq \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2 \quad \forall k > 0, \quad (8)$$

$$\|P\mathbf{g}_k - \hat{\mathbf{v}}\|_m^2 \leq \frac{1}{k} \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2 \quad \forall k > 0. \quad (9)$$

From (9) we obtain

$$\lim_{k \rightarrow \infty} P\mathbf{g}_k = \hat{\mathbf{v}} = P\bar{\mathbf{g}}.$$

According to (8) the sequence $\{\mathbf{g}_k\}_{k>0}$ is bounded above. Thus, there is a convergent subsequence¹, also called $\{\mathbf{g}_k\}$, with limit \mathbf{g}' , that satisfies

$$\|\mathbf{g}' - \hat{\mathbf{g}}\|_n^2 \leq \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2. \quad (10)$$

Since $\bar{\mathbf{g}}$ is the minimum norm solution, the solution of problem (5), then in (10) the equality holds. Also, the application $\mathbf{g} \rightarrow \|\mathbf{g} - \hat{\mathbf{g}}\|_n^2$ is lower semi-continuous, (actually continuous) so

$$\|\mathbf{g}' - \hat{\mathbf{g}}\|_n^2 \leq \liminf_{k \rightarrow \infty} \|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 \leq \limsup_{k \rightarrow \infty} \|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 \leq \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2.$$

Therefore

$$\lim_{k \rightarrow \infty} \mathbf{g}_k = \bar{\mathbf{g}}.$$

The previous analysis shows that when k tends to infinity the solution \mathbf{g}_k of the penalized problem (7) converges to the solution of problem (5). An important observation is that problem (7) can be reformulated as a Tikhonov regularization of problem (4), and the equivalent problem is

$$\begin{aligned} \mathbf{g}_\alpha &\in \mathcal{U}, \\ J_\alpha(\mathbf{g}_\alpha) &\leq J_\alpha(\mathbf{g}) \quad \text{for all } \mathbf{g} \in \mathcal{U}, \end{aligned}$$

where now the minimization function J_α is defined as

$$J_\alpha(\mathbf{g}) = \frac{\alpha}{2} \|\mathbf{g} - \hat{\mathbf{g}}\|_n^2 + \frac{1}{2} \|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2, \quad (11)$$

with the regularization parameter $\alpha > 0$ small. This problem is related to (7) by choosing $\alpha = 1/k$ and we can immediately check that $J_\alpha(\mathbf{g}) \rightarrow (1/2) \|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2$ when $\alpha \rightarrow 0$. Thus, from the previous analysis, we conclude that

$$\lim_{\alpha \rightarrow 0} \mathbf{g}_\alpha = \lim_{k \rightarrow \infty} \mathbf{g}_k = \bar{\mathbf{g}}$$

and $\bar{\mathbf{g}}$ is the solution of problems (4) and/or (5). We conclude that Spiess model can be seen as the limit of the penalized (regularized) models. From the theory of inverse problems (Tarantola 2005), it is well known that regularization often produces stable solutions with respect to perturbation on the given data $\hat{\mathbf{v}}$. We finally want to mention that some authors, like Noriega and Florian (2009) and Verbas, Mahmassani, and Zhang (2011), prefer quadratic

models with weighted averages, of the following form

$$J_\beta(\mathbf{g}) = \frac{\beta}{2} \|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2 + \frac{1-\beta}{2} \|\mathbf{g} - \hat{\mathbf{g}}\|_n^2,$$

where $0 \leq \beta \leq 1$. These models are also equivalent to our penalized model (6), and their relationship is obtained by choosing

$$k = \frac{\beta}{1-\beta} \iff \beta = \frac{k}{k+1}. \quad (12)$$

Thus, we can claim that

$$\lim_{\beta \rightarrow 1} \mathbf{g}_\beta = \lim_{k \rightarrow \infty} \mathbf{g}_k = \bar{\mathbf{g}} \quad \text{and} \quad \lim_{\beta \rightarrow 1} P\mathbf{g}_\beta = \lim_{k \rightarrow \infty} P\mathbf{g}_k = \hat{\mathbf{v}}.$$

2.3. The MCG algorithm

This algorithm, introduced in Chávez and Juárez (2014) and Juárez and Chávez (2014), finds the O-D demand matrix iteratively by the following recursion formula

$$g_{pq}^{\ell+1} = \begin{cases} \hat{g}_{pq} & \text{for } \ell = 0, \\ g_{pq}^\ell (1 + \gamma_\ell d_{pq}^\ell) & \text{for } \ell \geq 1 \end{cases} \quad \forall pq \in \mathcal{PQ},$$

where d_{pq}^ℓ is the direction of descent and γ_ℓ is the step length. Knowing the descent direction vector $\mathbf{d}^\ell = \{d_{pq}^\ell\}_{pq \in \mathcal{PQ}}$ at iteration ℓ , the step length γ_ℓ is computed as the minimum of the scalar function

$$\phi(\gamma) = J_k(\mathbf{g}^\ell + \gamma \mathbf{d}^\ell). \quad (13)$$

The first descent direction is chosen as the negative of the gradient evaluated at the initial guess, $\mathbf{d}^0 = -\nabla J_k(\hat{\mathbf{g}})$ and it is updated at each iteration with the following formula

$$d_{pq}^{\ell+1} = -g_{pq}^{\ell+1} \frac{\partial J_k(\mathbf{g}^{\ell+1})}{\partial g_{pq}} + \beta_\ell d_{pq}^\ell, \quad \forall pq \in \mathcal{PQ},$$

where the constant β_ℓ is computed to ensure that the two directions \mathbf{d}^ℓ and $\mathbf{d}^{\ell+1}$ are conjugate to each other.

Notice that the quadratic cost function of problem (6) has the following gradient

$$\nabla J_k(\mathbf{g}) = \mathbf{g} - \hat{\mathbf{g}} + kP^T(P\mathbf{g} - \hat{\mathbf{v}}) = Q_k \mathbf{g} - \mathbf{b}_k,$$

where $Q_k = I + kP^T P$ with I the identity matrix, and $\mathbf{b}_k = \hat{\mathbf{g}} + kP^T \hat{\mathbf{v}}$. Matrix Q_k is positive definite for every constant $k > 0$, so its inverse exists. From now on we use the notation $\mathbf{x} \odot \mathbf{y}$ to indicate the element-wise multiplication of vectors in \mathbb{R}^n , known as the Hadamard product, this means the vector $\mathbf{z} = \mathbf{x} \odot \mathbf{y}$ has components $z_i = x_i y_i$. The MCG algorithm, step by step is the following:

Algorithm 1 Multiplicative Conjugate Gradient (MCG).**Require:** $\hat{\mathbf{g}}, k, Q_k, \mathbf{b}_k$ and $\varepsilon > 0$ (a small tolerance).**Ensure:** An updated O-D matrix \mathbf{g} .

-
- ```

1: Choose $\mathbf{g}^0 = \hat{\mathbf{g}}$. ▷ Initialization
2: Compute $\mathbf{r}^0 \doteq \nabla J_k(\mathbf{g}^0) \equiv Q_k(\mathbf{g}) - \mathbf{b}_k$.
3: Set $\mathbf{r}_M^0 = \mathbf{g}^0 \odot \mathbf{r}^0$.
4: Set $\mathbf{d}_M^0 = -\mathbf{r}_M^0$.
5: for $\ell \geq 0$, do ▷ Descent.
6: Compute γ_ℓ solution of (13) such that $g_{pq}^\ell + \gamma (d_M^\ell)_{pq} \geq 0$ for all $pq \in \mathcal{PQ}$.
7: Update $\mathbf{g}^{\ell+1} = \mathbf{g}^\ell + \gamma_\ell \mathbf{d}_M^\ell$.
8: Update $\mathbf{r}^{\ell+1} = \mathbf{r}^\ell + \gamma_\ell Q_k \mathbf{d}_M^\ell$.
9: Compute $\mathbf{r}_M^{\ell+1} = \mathbf{g}^{\ell+1} \odot \mathbf{r}^{\ell+1}$.
10: if $\|\mathbf{r}^{\ell+1}\|_n \leq \varepsilon \|\mathbf{r}^0\|_n$ then ▷ Test of convergence
11: take $\mathbf{g}_k = \mathbf{g}^{\ell+1}$ and stop,
12: else
13: Compute $\beta_\ell = \frac{\mathbf{r}_M^{\ell+1} Q_k \mathbf{d}_M^\ell}{(\mathbf{d}_M^\ell)^T Q_k \mathbf{d}_M^\ell}$.
14: $\mathbf{d}_M^{\ell+1} = -\mathbf{r}_M^{\ell+1} + \beta_\ell \mathbf{d}_M^\ell$. ▷ New direction
15: Set $\ell = \ell + 1$ and go back to step 5.

```
- 

In the above algorithm, we employ the notation  $\mathbf{r}^\ell = \nabla J_k(\mathbf{g}^\ell)$ . We want to emphasize that the only difference of this MCG algorithm with respect to the standard conjugate gradient (CG) algorithm is the calculation of the multiplicative gradient by the Hadamard product in steps 3 and 9 at each iteration, so the additional computational cost is marginal. Nevertheless, we need to be careful with the initialization step 1, the multiplicative structure of the algorithm requires that the initial guess  $\mathbf{g}^0 = \hat{\mathbf{g}}$  be different to  $\mathbf{0} \in \mathbb{R}^n$ , otherwise  $\mathbf{g}^\ell$  will remain  $\mathbf{0}$  at each iteration. Notice that the formula in step 8 is obtained from the formula in step 7 multiplying by  $Q_k$  and subtracting  $\mathbf{b}_k$  in both sides. To compute  $\gamma_\ell$  in step 6, we first calculate the value of  $\gamma$  such that  $\phi'_\ell(\gamma) = \nabla J_k(\mathbf{g}^\ell + \gamma \mathbf{d}_M^\ell)^T \mathbf{d}_M^\ell = (\mathbf{r}^{\ell+1})^T \mathbf{d}_M^\ell = 0$ ; this value is

$$\gamma = -(\mathbf{r}^\ell)^T \mathbf{d}_M^\ell / (\mathbf{d}_M^\ell)^T Q_k \mathbf{d}_M^\ell,$$

which very frequently satisfies the given restriction  $g_{pq}^\ell + \lambda(d_M^\ell)_{pq} \geq 0$  for any  $pq \in \mathcal{PQ}$ , and if it does not then we set  $g_{pq}^{\ell+1} = 0$  (see Vollebregt 2014). The value for  $\beta_\ell$  in step 13 is such that  $\mathbf{d}_M^{\ell+1}$  and  $\mathbf{d}_M^\ell$  are  $Q_k$ -conjugate. We can observe that the algorithm can be rewritten in order to compute  $Q_k \mathbf{d}_M^\ell$  and  $(\mathbf{d}_M^\ell)^T Q_k \mathbf{d}_M^\ell$  only once at each iteration. Equality in step 8 implies that  $\beta_\ell$  is also equal to

$$\beta_\ell = \frac{(\mathbf{r}_M^{\ell+1})^T (\mathbf{r}^{\ell+1} - \mathbf{r}^\ell)}{(\mathbf{d}_M^\ell)^T (\mathbf{r}^{\ell+1} - \mathbf{r}^\ell)}, \quad (14)$$

which resembles the Hestenes–Stiefel formula for the CG algorithm (Nocedal and Wright 2006).

The MCG algorithm ensures that two consecutive directions,  $\mathbf{d}_M^\ell$  and  $\mathbf{d}_M^{\ell+1}$ , are  $Q_k$ -conjugate, but there is no guaranty that all the generated directions during the iteration process are conjugate to each other.

### 3. Augmented Lagrangian approach

One of the main drawbacks of the MCG algorithm is that it does not allow the evolution from null coefficients in the seed O-D matrix to non-zero values in the updated demand matrix; this may be an unwanted feature for some instances. Also, from the computational point of view, with the MCG algorithm, not all descent directions  $\{\mathbf{d}_M^\ell\}_\ell$  are  $Q_k$ -conjugate to each other, which may degrade the performance of the method and the accuracy of the numerical solution. An alternative is the explicit inclusion of non-negativity conditions,  $g_{pq} \geq 0$ , into the cost function, avoiding the multiplicative structure of the iterative algorithm and thus enabling the use of the standard conjugate gradient (CG) algorithm.

Our departure point here is the penalized model (6), for which we may employ a Lagrangian approach to deal with the non-negativity constraints on the coefficients of the O-D matrix. We introduce a new variable vector  $\mathbf{y} \in \mathbb{R}^n$  that satisfies

$$y_{pq}^2 = g_{pq} \quad \text{for all } pq \in \mathcal{PQ}. \quad (15)$$

With (15) we convert the inequality constraints  $g_{pq} \geq 0$  to equality constraints. Thus the minimization problem (6)–(7) is equivalent to minimize the quadratic cost functional (6) subject to

$$\mathbf{g} = \mathbf{y} \odot \mathbf{y} \quad \text{and} \quad \mathbf{y} \in \mathbb{R}^n.$$

A common way to deal with equality constraints (Nocedal and Wright 2006) is by introducing a variable called the Lagrange multiplier, that we denote by a vector  $\boldsymbol{\mu} \in \mathbb{R}^n$ , and form the following Lagrangian function

$$\mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = J_k(\mathbf{g}) - \boldsymbol{\mu}^\top (\mathbf{g} - \mathbf{y} \odot \mathbf{y}). \quad (16)$$

Thus, a critical point  $(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu})$  satisfies the Karush–Kuhn–Tucker (KKT) conditions:

$$\nabla_{\mathbf{g}} \mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = (\mathbf{g} - \hat{\mathbf{g}}) + kP^\top(P\mathbf{g} - \hat{\mathbf{v}}) - \boldsymbol{\mu} = Q_k\mathbf{g} - \mathbf{b}_k - \boldsymbol{\mu} = \mathbf{0}, \quad (17)$$

$$\nabla_{\mathbf{y}} \mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = 2\boldsymbol{\mu} \odot \mathbf{y} = \mathbf{0}, \quad (18)$$

$$\nabla_{\boldsymbol{\mu}} \mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = \mathbf{y} \odot \mathbf{y} - \mathbf{g} = \mathbf{0}, \quad (19)$$

where

$$Q_k = I + kP^\top P \quad \text{and} \quad \mathbf{b}_k = \hat{\mathbf{g}} + kP^\top \hat{\mathbf{v}}. \quad (20)$$

The non-linear system (17)–(19) may be solved by an iterative process, but unfortunately, the starting point given by  $\mathbf{g}^0 = \hat{\mathbf{g}}$ ,  $\mathbf{y}^0 = \sqrt{\hat{\mathbf{g}}}$  (component-wise square root), may yield large values for the multiplier at the first iteration (and at the next few iterations), since  $\boldsymbol{\mu}^J = Q_k\mathbf{g}^J - \mathbf{b}_k = \nabla J_k(\mathbf{g}^J)$ , especially for big values of the penalty parameter  $k$ . To avoid

this instability, we convexify the Lagrangian (16) introducing the following augmented Lagrangian function

$$\mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = J_k(\mathbf{g}) - \boldsymbol{\mu}^T(\mathbf{g} - \mathbf{y} \odot \mathbf{y}) + \frac{\rho}{2} \|\mathbf{g} - \mathbf{y} \odot \mathbf{y}\|_n^2, \quad (21)$$

where  $k$  and  $\rho$  are positive constants. Now the KKT conditions are

$$\begin{aligned} \nabla_{\mathbf{g}} \mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) &= \mathbf{g} - \hat{\mathbf{g}} + kP^T(P\mathbf{g} - \hat{\mathbf{v}}) - \boldsymbol{\mu} + \rho(\mathbf{g} - \mathbf{y} \odot \mathbf{y}) \\ &= Q_{k,\rho}\mathbf{g} - \mathbf{b}_k - \boldsymbol{\mu} - \rho\mathbf{y} \odot \mathbf{y} = \mathbf{0}, \end{aligned} \quad (22)$$

$$\nabla_{\mathbf{y}} \mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = 2[\boldsymbol{\mu} + \rho(\mathbf{y} \odot \mathbf{y} - \mathbf{g})] \odot \mathbf{y} = \mathbf{0}, \quad (23)$$

$$\nabla_{\boldsymbol{\mu}} \mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = \mathbf{y} \odot \mathbf{y} - \mathbf{g} = \mathbf{0}, \quad (24)$$

where  $\mathbf{b}_k$  is defined in (20) and

$$Q_{k,\rho} = (1 + \rho)I + kP^TP.$$

Note that, if we substitute (24) in (22) and (23), we recover the system (17)–(19), thus both systems are equivalent. However, some algorithmic and computational properties of the system (22)–(24) that may be advantageous over the system (17)–(19) are the following.

- (1)  $Q_{k,\rho}$  is a symmetric and positive definite matrix with better condition number than  $Q_k$  for the same value of  $k$ , since  $\rho > 0$ .
- (2) The value of  $\rho$  in (21) does not need to be large since the restriction  $\mathbf{g} = \mathbf{y} \odot \mathbf{y}$  has already been relaxed with the Lagrange multiplier. This parameter is chosen based on the constant value of  $k/(1 + \rho)$ , as is shown in the numerical examples.
- (3) When an iterative process is applied to get a solution, we have the option to update  $\boldsymbol{\mu}$  using Equation (23) as is shown in (27) below; while in (17)–(19) the only option is using (17).

### 3.1. The DAMM algorithm

According to the observations in the previous section, we propose a dual ascent and method of multipliers (DAMM) algorithm (Boyd et al. 2010).

The problem (25) involves the minimization of a strictly convex quadratic function over  $\mathbb{R}^n$  with constant Hessian equal to  $Q_{k,\rho}$ , thus the unique minimum  $\mathbf{g}^J$  satisfies the linear system

$$Q_{k,\rho}\mathbf{g} = \mathbf{b}_k + \boldsymbol{\mu}^{J-1} + \rho\mathbf{y}^{J-1} \odot \mathbf{y}^{J-1}, \quad (28)$$

which can be approximated by the CG algorithm with initial guess  $\mathbf{g}^0 = \mathbf{g}^{J-1}$ . However, to avoid excessive negative values in the solution  $\mathbf{g}^J$  of the CG, we stop at iteration  $\ell$  if we obtain  $\min(\mathbf{g}^\ell) \leq -0.25$  and continue with step 4 of the DAMM algorithm. In other words, the DAMM algorithm can be more efficient if CG (step 3) performs only a few iterations instead of achieving high accuracy solutions and continue with step 4 to update the multiplier; especially when the CG algorithm (or any other) yields a demand matrix with negative

**Algorithm 2** Dual Ascent and Method of Multipliers (DAMM).**Require:**  $\hat{\mathbf{g}}$  and  $\varepsilon > 0$  (a small tolerance).**Ensure:** An updated O-D matrix  $\mathbf{g}$ .

- 1: Take the initial guess  $\mu^0 = 0$  and  $\mathbf{y}^0 \odot \mathbf{y}^0 = \hat{\mathbf{g}}$ . ▷ Initialization
  - 2: **for**  $j \geq 1$ , given  $\mu^{j-1}$  and  $\mathbf{y}^{j-1}$ , to compute  $\mathbf{g}^j$  and  $\mathbf{y}^j$  **do** ▷ Iteration.
  - 3:  $\mathbf{g}^j = \arg \min_{\mathbf{g}} \mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}^{j-1}, \mu^{j-1})$ ,
- (25)

- 4:  $\mathbf{y}^j = \arg \min_{\mathbf{y}} \mathcal{L}_{k,\rho}(\mathbf{g}^j, \mathbf{y}, \mu^{j-1})$ .

(26)

- 5:     **if**  $\|\mathbf{y}^j \odot \mathbf{y}^j - \mathbf{g}^j\|_n \leq \varepsilon \|\hat{\mathbf{g}}\|_n$ , **then** ▷ Convergence test
  - 6:         take  $\mathbf{g} = \mathbf{g}^j$  and **stop**,
  - 7:     **else** ▷ Update
  - 8:          $\mu^j = \mu^{j-1} + \rho (\mathbf{y}^j \odot \mathbf{y}^j - \mathbf{g}^j)$ .
- (27)

- 9:     Set  $j = j + 1$  and go back to step 2.

---

coefficients. Thus, the CG algorithm stops at iteration  $\ell$  if the residual is less or equal to  $\varepsilon \|Q\mathbf{g}^\ell - \mathbf{b}_k - \mu^{\ell-1} - \rho \mathbf{y}^{\ell-1} \odot \mathbf{y}^{\ell-1}\|$  (like in the MCG algorithm) or if  $\min(\mathbf{g}^\ell) \leq -0.25$ .

The problem (26) involves the minimization of

$$f(\mathbf{y}) = \frac{\rho}{2} \|\mathbf{g}^j - \mathbf{y} \odot \mathbf{y}\|_n^2 - (\mu^{j-1})^\top (\mathbf{g}^j - \mathbf{y} \odot \mathbf{y}) + J_k(\mathbf{g}^j) \quad (29)$$

with respect to  $\mathbf{y}$ . A critical point  $\mathbf{y} = \{y_i\}_{i=1}^n$  (here,  $i$  denotes an O-D pair  $pq \in \mathcal{PQ}$ ) satisfies

$$\frac{\partial f}{\partial y_i}(\mathbf{y}) = 2 \left[ \rho (y_i^2 - g_i^j) + \mu_i^{j-1} \right] y_i = 0 \quad \text{for } i = 1, \dots, n,$$

this means,  $y_i = 0$  or  $y_i^2 = g_i^j - \mu_i^{j-1}/\rho$ . So, the critical point  $\mathbf{y}^j$  is built up in the following manner.

For each  $i = 1, \dots, n$ : **if**  $g_i^j - \mu_i^{j-1}/\rho > 0$  **then**  $(y_i^j)^2 = g_i^j - \mu_i^{j-1}/\rho$ , **else**  $y_i^j = 0$ . (30)

This time, the Hessian of  $f(\mathbf{y})$  is a diagonal matrix  $D$  with entries

$$D_{ii}(\mathbf{y}) = \frac{\partial^2 f}{\partial y_i^2} = 2 \left[ \rho (y_i^2 - g_i^j) + \mu_i^{j-1} + 2\rho y_i^2 \right],$$

and its evaluation at the critical point in (30) gives

$$D_{ii}(\mathbf{y}^j) = \begin{cases} 4\rho (y_i^j)^2, & \text{if } (y_i^j)^2 = g_i^j - \mu_i^{j-1}/\rho > 0, \\ 2\rho(\mu_i^{j-1}/\rho - g_i^j), & \text{if } y_i^j = 0. \end{cases}$$

Thus, the Hessian has positive entries along its diagonal except when  $g_i^j - \mu_i^{j-1}/\rho = 0$ , for which the correspondent entry in the diagonal becomes zero at iteration  $j$ . Thus the Hessian is positive semi-definite in general. However, the critical point (30) still yields a good

estimation of a minimum in (26), and a way to see this is looking at the quadratic function for  $\mathbf{z} = \mathbf{y} \odot \mathbf{y}$ :

$$f(\mathbf{z}) = \frac{\rho}{2} \|\mathbf{g}^J - \mathbf{z}\|_n^2 - (\boldsymbol{\mu}^{J-1})^\top (\mathbf{g}^J - \mathbf{z}) + J_k(\mathbf{g}^J).$$

So, the original quartic function  $f(\mathbf{y})$  in (29) is converted to a new quadratic function on the variable  $\mathbf{z}$  with non-negative coefficients. For this new function, the gradient and Hessian are, respectively

$$\nabla f(\mathbf{z}) = \rho (\mathbf{z} - \mathbf{g}^J) + \boldsymbol{\mu}^{J-1} \quad \text{and} \quad H_f(\mathbf{z}) = \rho I,$$

where  $I$  is the identity matrix of size  $n \times n$ . So, this time the Hessian is constant and positive definite, and the gradient vanishes if

$$\mathbf{z} = \mathbf{g}^J - \boldsymbol{\mu}^{J-1}/\rho \quad \text{and} \quad \mathbf{z} \geq \mathbf{0}. \quad (31)$$

Observe that (31), for the variable  $\mathbf{z}$ , is related to (30) for the variable  $\mathbf{y}$ .

### 3.2. Relation between the penalty parameters $k$ and $\rho$

These two parameters have an important influence on the performance of the iterative Algorithm 2 and, in particular, on the sub-problems (25)–(27); mainly in the solution of the linear system (28) and the minimization of the quadratic function (29). Concerning the sub-problem (28), the larger the value  $k/(1 + \rho)$  is the larger the condition number of matrix  $Q_{k,\rho}$ . For the penalized model (6) we obtain good results with  $k \geq 10^2$ , and if we want to have a similar computational performance on the solution of (28) we set  $k/(1 + \rho) \geq 10^2$ . Notice that step 3 is the most expensive part of the iterative Algorithm 2, however it has the advantage that it can be solved by the CG algorithm. Finally, observe that if we set  $k = \infty$  (or  $\alpha = 0$  in (11)), we still get a symmetric and positive definite matrix  $Q_\rho = \rho I + P^\top P$ , which implies that this variant of the augmented Lagrangian has a regularizing effect for the original problem, and in this case, it suffices to take any positive value for  $\rho$ . The numerical examples in Section 5.3 corroborate this property.

Summarizing, the iterative Algorithm 2 is efficient because the additional work (which includes solving the problem (26) and updating the multipliers in (27)) is marginal compared to the solution of the problem (25). Notice that, the numerical computation of (25) with the CG algorithm is quicker than the MCG algorithm to solve the penalized problem (7), because the matrix  $Q_{k,\rho}$  has a lower condition number than  $Q_k$  for the same value of  $k$ . Also, all the descent directions generated with the CG algorithm are guaranteed to be  $Q_{k,\rho}$ -conjugate. In addition, this new approach is more robust because the null coefficients of the O-D matrix are not enforced to remain null.

## 4. Size reduction of the problem

In this section, we discuss a reduction of the problem, when it is convenient and/or possible. When this operation is used we obtain two benefits: memory saving and additional reduction of the computational cost.

One of the properties of the MSD and MCG algorithms is that they preserve the structure of the seed O-D matrix  $\hat{\mathbf{g}}$  when updating the new matrix  $\mathbf{g}$ : every null entry in the first matrix

yields a null entry in the updated one. Indeed, it is unlikely that all null entries will evolve to a positive value, especially when updating within a short time on a large network. Then extracting the null coefficients from the seed O-D matrix  $\hat{\mathbf{g}}$ , we obtain a new seed O-D matrix  $\hat{\mathbf{g}}_r$  of size  $n_r \leq n$ , which has only positive coefficients. Then, we reformulate the problem (5) in the following way:

Given  $\hat{\mathbf{g}}_r \in \mathbb{R}_+^{n_r}$  and  $\hat{\mathbf{v}} \in \mathbb{R}_+^m$ , find  $\mathbf{g}_r \in \mathbb{R}_+^{n_r}$  that minimizes

$$J(\mathbf{g}_r) = \frac{1}{2} \|\mathbf{g}_r - \hat{\mathbf{g}}_r\|_{n_r}^2 \quad \text{over the set } \mathcal{V}_r = \{\mathbf{g}_r \in \mathbb{R}_+^{n_r} : P_r \mathbf{g}_r = \hat{\mathbf{v}}\},$$

where matrix  $P_r \in \mathbb{R}^{m \times n_r}$  is obtained from  $P$  after extracting those columns which correspond to null entries of the original seed O-D matrix  $\hat{\mathbf{g}}$ . Now, the penalized model (7) and the MCG algorithm have the following additional benefits:

- (1) The null space of matrix  $P_r$  has a lower dimension than the null space of  $P$  and they share the same positive coefficients. Thus, there is a significant saving of memory, especially when  $\hat{\mathbf{g}}$  has a big proportion of null coefficients, which is common for large-scale networks.
- (2) Since there are no null coefficients in  $\hat{\mathbf{g}}_r$ , then it is more likely that  $\gamma_\ell$  satisfies the restriction of non-negativity in step 6 of the algorithm 1.

Analogously, with this reduction, the augmented Lagrangian function becomes

$$\mathcal{L}_{k,\rho}(\mathbf{g}_r, \mathbf{y}_r, \boldsymbol{\mu}_r) = J_k(\mathbf{g}_r) - \boldsymbol{\mu}_r^\top (\mathbf{g}_r - \mathbf{y}_r \odot \mathbf{y}_r) + \frac{\rho}{2} \|\mathbf{g}_r - \mathbf{y}_r \odot \mathbf{y}_r\|_{n_r}^2,$$

where  $\mathbf{y}_r, \boldsymbol{\mu}_r \in \mathbb{R}^{n_r}$ . Thus, the KKT conditions yield

$$\begin{aligned} ((1 + \rho)I_{n_r} + kP_r^\top P_r)\mathbf{g}_r &= \hat{\mathbf{g}}_r + kP_r^\top \hat{\mathbf{v}} + \boldsymbol{\mu}_r + \rho \mathbf{y}_r \odot \mathbf{y}_r, \\ 2[\boldsymbol{\mu}_r + \rho(\mathbf{y}_r \odot \mathbf{y}_r - \mathbf{g}_r)] \odot \mathbf{y}_r &= \mathbf{0}, \\ \mathbf{g}_r - \mathbf{y}_r \odot \mathbf{y}_r &= \mathbf{0}, \end{aligned}$$

where  $I_{n_r}$  is the identity matrix of  $n_r \times n_r$ . The DAMM algorithm for this reduced problem is applied in the same way as for the complete problem.

## 5. Numerical experiments

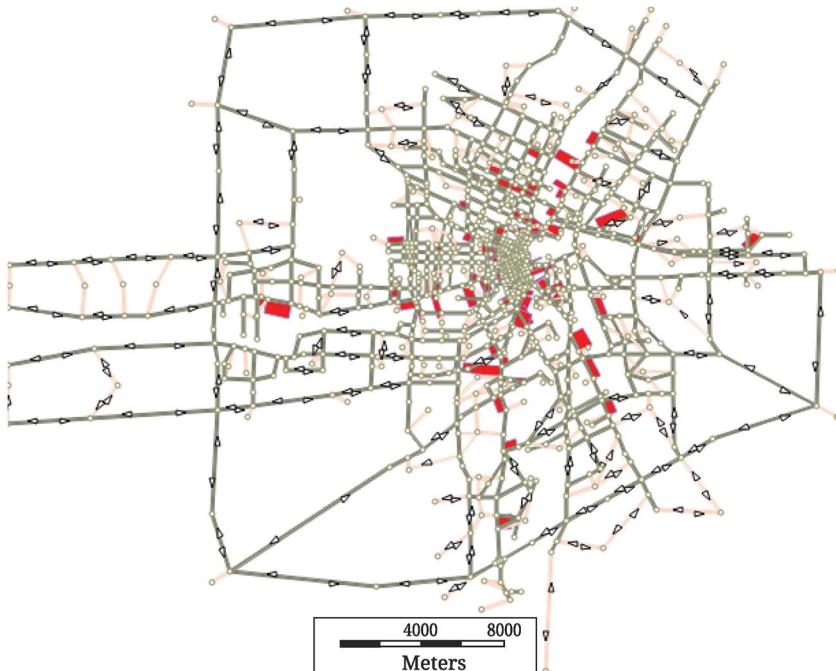
The algorithms presented in this work were coded in Matlab and implemented in an HP-Pavilion Dm4 computer with an Intel(R) Core(TM) i5 processor and 8 GB RAM.

### 5.1. Study cases

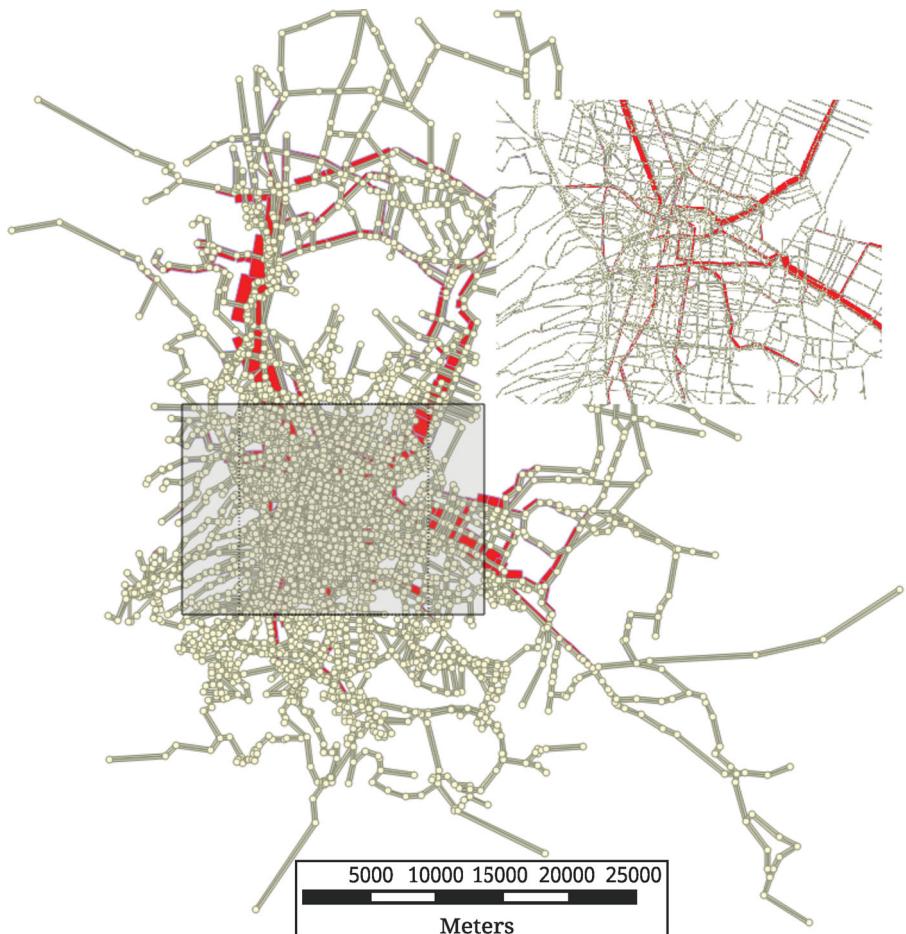
For the numerical experiments, we considered synthetic data (segment flow volumes) from two transit networks, the Winnipeg network from Canada, included in the demonstration database of EMME (INRO 2018), and the network based on the metropolitan area of the Valley of Mexico (MAVM), Mexico, provided by Torres (2013). These data were generated in the following way: given a demand matrix  $\bar{\mathbf{g}}$  (which may represent the demand at the peak hour in the morning), we apply a linear transit assignment and extracted

**Table 1.** Characteristics of the Winnipeg and the MAVM networks.

| Attributes                           | Winnipeg      | MAVM          |
|--------------------------------------|---------------|---------------|
| Zones                                | 154           | 1705          |
| O-D pairs                            | 23,716        | 2,907,025     |
| No. of pairs with $\hat{g}_{pq} > 0$ | 5,394 (22.7%) | 20,278 (0.7%) |
| Regular nodes                        | 906           | 7241          |
| Links                                | 3,005         | 31,720        |
| Modes                                | 5             | 18            |
| Types of vehicles                    | 4             | 11            |
| Transit lines                        | 133           | 845           |
| Transit segments                     | 4,347         | 46,981        |
| Segments with counts                 | 136 (3.1%)    | 1,470 (3.1%)  |

**Figure 1.** Segments with available counts in red. Winnipeg transit network.

some of the segment flows which play the role of measured volumes  $\hat{\mathbf{v}}$ . Next, we generated a seed O-D matrix  $\hat{\mathbf{g}}$  by doing a random uniform perturbation of 20% of the matrix  $\bar{\mathbf{g}}$ . Then, with this information, we applied our methods to evaluate their performances. Table 1 shows the general characteristics of the networks; where ‘regular nodes’ represent the intersection of two or more links. Also, we are considering all the transit modes of the network (e.g. bus, trolley bus and subway). In this context, links may represent the streets whereas the segments represent the routes of the transit lines because for each link we can have more than one transit segment (or none). Figures 1 and 2 show the segments with flow counts, highlighted in red for Winnipeg and MAVM networks, respectively.



**Figure 2.** Segments with available counts in red. Network based on the metropolitan area of Valley of Mexico (MAVM).

### 5.2. Performance of the MSD and MCG algorithms

In this section, we compare the performance of the MSD algorithm proposed by Spiess (part of the EMME/4 transportation planning system INRO 2018) with our MCG algorithm, described in Section 2. Tables 2 and 3 summarize the numerical results obtained with the penalized model (6) for the Winnipeg and the MAVM transit networks, respectively. In both tables  $k$  is the penalty parameter, CPU is the computing time in seconds,  $l_{ters.}$  is the number of iterations to achieve convergence for each method within the given tolerance (set to  $\varepsilon = 10^{-3}$  for these experiments),  $RMSE_v$  is the root mean square error for the segment flows,  $\|P\mathbf{g} - \hat{\mathbf{v}}\|_m$  is the distance between the estimated segment flows and the observed ones,  $RMSE_g$  is the root mean square error for the demand and  $\|\mathbf{g} - \hat{\mathbf{g}}\|_n$  is the distance between the estimated demand matrix and the old one. The case  $k = \infty$  ( $\alpha = 0$  in (11)) in both tables must be regarded as the case where the objective function is  $(1/2)\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2$  (Spiess model).

**Table 2.** Comparison of MSD and MCG algorithms for the Winnipeg network.

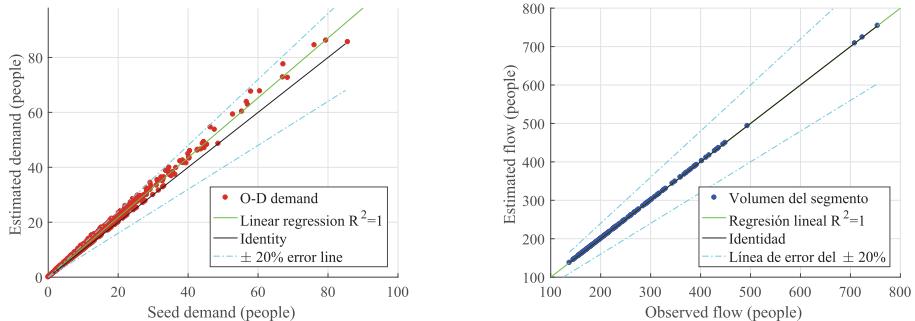
| $k$      | Method      | CPU    | <i>Iters.</i> | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g - \hat{g}\ _n$ |
|----------|-------------|--------|---------------|-----------------|----------------------|-----------------|---------------------|
| Initial  | ( $I = 0$ ) |        |               | 28.79           | 335.7                |                 |                     |
| 100      | MSD         | 1.1 s. | 80            | 0.16            | 1.9                  | 0.33            | 50.3                |
|          | MCG         | 0.3 s. | 21            | 0.11            | 1.3                  | 0.33            | 50.2                |
| 1000     | MSD         | 0.8 s. | 78            | 0.17            | 1.9                  | 0.33            | 50.9                |
|          | MCG         | 0.3 s. | 21            | 0.11            | 1.3                  | 0.33            | 51.0                |
| 10000    | MSD         | 0.9 s. | 78            | 0.17            | 1.9                  | 0.33            | 50.9                |
|          | MCG         | 0.3 s. | 21            | 0.11            | 1.3                  | 0.33            | 51.1                |
| $\infty$ | MSD         | 1.0 s. | 78            | 0.16            | 1.9                  | 0.33            | 50.9                |
|          | MCG         | 0.3 s. | 21            | 0.11            | 1.3                  | 0.33            | 51.1                |

**Table 3.** Comparison of MSD and MCG algorithms for the MAVM network.

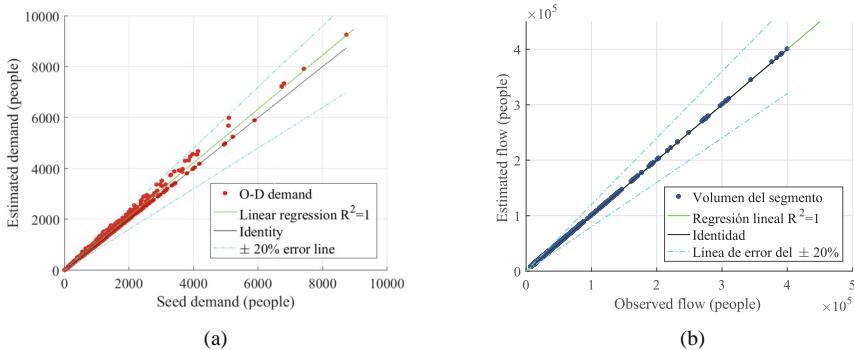
| $k$      | Method      | CPU    | <i>Iters.</i> | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g - \hat{g}\ _n$ |
|----------|-------------|--------|---------------|-----------------|----------------------|-----------------|---------------------|
| Initial  | ( $I = 0$ ) |        |               | 5797.04         | 222262.0             |                 |                     |
| 100      | MSD         | 23.5 s | 147           | 78.94           | 3026.8               | 2.54            | 4323.0              |
|          | MCG         | 5.4 s  | 30            | 52.34           | 2006.8               | 2.64            | 4495.7              |
| 1000     | MSD         | 23.6 s | 147           | 78.92           | 3025.9               | 2.54            | 4324.9              |
|          | MCG         | 5.6 s  | 30            | 52.34           | 2006.7               | 2.64            | 4499.3              |
| 10000    | MSD         | 23.6 s | 147           | 78.92           | 3025.9               | 2.54            | 4325.1              |
|          | MCG         | 5.5 s  | 30            | 52.34           | 2006.6               | 2.64            | 4499.7              |
| $\infty$ | MSD         | 24.0 s | 147           | 78.92           | 3025.8               | 2.54            | 4325.1              |
|          | MCG         | 5.7 s  | 30            | 52.34           | 2006.6               | 2.64            | 4499.7              |

As expected, when the penalty parameter  $k$  approaches infinity the results converge to the solution of the Spiess' model. Actually, for values of  $k > 100$ , we obtain similar results, including the number of iterations and the CPU time, and there is no practical change when  $k \geq 1000$ . This may explain why the numerical results obtained by some authors (like Noriega and Florian 2009; Verbas, Mahmassani, and Zhang 2011) are better, in their respective cases, for values of  $\beta$  in the interval [0.999, 1). Thus, a value of  $k = 100$  or larger is sufficient in practice for both networks. The most remarkable difference is the performance (number of iterations and CPU time) of MCG algorithm with respect to the MSD algorithm. For the Winnipeg transit network, the initial  $\text{RMSE}_v$  is reduced 262 (180) times with the MCG (MSD) algorithm and the computing time is improved about three times with MCG with respect to MSD. For the case of the MAVM transit network, the initial  $\text{RMSE}_v$  is reduced 111 (73) times with MCG (MSD), while the CPU time is reduced about 4.3 times for the MCG with respect to MSD. Thus, the MCG algorithm not only is faster, but it also reduces the  $\text{RMSE}_v$  for both networks.

Figure 3 depicts the seed demand (x-axis) against the estimated one (y-axis), part (a), and the observed volumes (x-axis) against the estimated ones (y-axis), part (b), for the Winnipeg transit network. All dots on the identity line indicate perfect fit between the corresponding estimated values and the data, while the dots that fall between the two dotted lines have a relative error less than 20%. The regression lines depicted in green, with coefficient correlation  $R^2 = 1$  for both figures. The respective slope and y-intercept for the demand are  $m_g = 1.09$  and  $b_g = 0.00$ . The respective slope and y-intercept for the segment flows are  $m_v = 1.00$  and  $b_v = -0.02$ . These parameters indicate that the estimated values are highly correlated with the data. Analogously, Figure 4 plots the corresponding scatter plots for the MAVM network with slope  $m_g = 1.06$  and y-intercept  $b_g = 0$  for the demand and slope  $m_v = 1.00$  and y-intercept  $b_v = -0.39$  for the segment flows. These two figures show that



**Figure 3.** Scatter plots for the estimations obtained with  $k = 1000$  and MCG in the Winnipeg transit network. (a) Results for the estimated demand at each O-D pair and (b) Results for the estimated flow at each transit segment with available counts.



**Figure 4.** Scatter plots for the estimations obtained with  $k = 1000$  and MCG in the MAVM transit network. (a) Results for the estimated demand at each O-D pair and (b) Results for the estimated flow at each transit segment with available counts.

most of the points fall into the area bounded by the  $\pm 20\%$  error lines, which indicates that the proposed model provides accurate results. Notice that the observed volumes are almost perfectly estimated, meanwhile the estimated demand is not necessarily close to the old one.

Considering the reduced problem discussed in Section 4, we tested the MSD and MCG algorithms obtaining the results shown in Tables 4 and 5 for each network. Here, we also present the mean percentage error (MPE) for the demand matrix; this metric is usually used to measure a bias in the estimation. Comparing these results with those shown in Tables 2 and 3, the most significant difference is the CPU time, which is reduced about three times for the Winnipeg and about six times for the MAVM networks. Since the number of iterations in the reduced problem is similar to the number of iterations in the complete problem, the improvement of CPU time is mainly related to memory saving and communication, since 77.3% of all coefficients are null in the O-D matrix of Winnipeg, while 99.3% of the coefficients are null in the matrix of the MAVM. We do not include the corresponding scatter plots since the quality of the solution is similar to those shown in Figures 3 and 4.

According to the numerical experiments shown in this section, at most three consecutive directions are  $Q_k$ -conjugate. However, more consecutive directions are linearly

**Table 4.** Comparison of MSD and MCG algorithms for the Winnipeg network (reduced problem).

| $k$      | Method      | CPU    | <i>Iters.</i> | $\text{RMSE}_v$ | $\ P_r g_r - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g_r - \hat{g}_r\ _n$ | MPE   |
|----------|-------------|--------|---------------|-----------------|---------------------------|-----------------|-------------------------|-------|
| Initial  | ( $I = 0$ ) |        |               | 5797.04         | 222262.0                  |                 |                         |       |
| 100      | MSD         | 0.3 s. | 74            | 0.18            | 2.1                       | 0.33            | 50.3                    | -8.23 |
|          | MCG         | 0.1 s. | 21            | 0.11            | 1.3                       | 0.33            | 50.2                    | -8.30 |
| 1000     | MSD         | 0.3 s. | 72            | 0.18            | 2.1                       | 0.33            | 50.9                    | -8.16 |
|          | MCG         | 0.1 s. | 21            | 0.11            | 1.3                       | 0.33            | 51.0                    | -8.20 |
| 10000    | MSD         | 0.3 s. | 70            | 0.18            | 2.2                       | 0.33            | 50.9                    | -8.15 |
|          | MCG         | 0.1 s. | 21            | 0.11            | 1.3                       | 0.33            | 51.1                    | -8.19 |
| $\infty$ | MSD         | 0.3 s. | 70            | 0.18            | 2.1                       | 0.33            | 50.9                    | -8.15 |
|          | MCG         | 0.1 s. | 21            | 0.11            | 1.3                       | 0.33            | 51.1                    | -8.19 |

**Table 5.** Comparison of MSD and MCG algorithms for the MAVM network (reduced problem).

| $k$      | Method | CPU    | <i>Iters.</i> | $\text{RMSE}_v$ | $\ P_r g_r - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g_r - \hat{g}_r\ _n$ | MPE   |
|----------|--------|--------|---------------|-----------------|---------------------------|-----------------|-------------------------|-------|
| 100      | MSD    | 5.7 s. | 147           | 78.94           | 3026.8                    | 2.54            | 4323.0                  | -5.24 |
|          | MCG    | 1.2 s. | 30            | 52.34           | 2006.8                    | 2.64            | 4495.7                  | -5.40 |
| 1000     | MSD    | 5.6 s. | 147           | 78.92           | 3025.9                    | 2.54            | 4324.9                  | -5.24 |
|          | MCG    | 1.2 s. | 30            | 52.34           | 2006.7                    | 2.64            | 4499.3                  | -5.40 |
| 10000    | MSD    | 5.6 s. | 147           | 78.92           | 3025.9                    | 2.54            | 4325.1                  | -5.24 |
|          | MCG    | 1.2 s. | 30            | 52.34           | 2006.6                    | 2.64            | 4499.7                  | -5.40 |
| $\infty$ | MSD    | 5.6 s. | 147           | 78.92           | 3025.8                    | 2.54            | 4325.1                  | -5.24 |
|          | MCG    | 1.2 s. | 30            | 52.34           | 2006.6                    | 2.64            | 4499.7                  | -5.40 |

independent, and the more directions are linearly independent the larger is the subspace where the cost function is minimized, so the faster the minimum is reached within the given tolerance. For instance, the intuition about the steepest descent method is that it has a low convergence rate because it moves in orthogonal steps and a ‘zig-zag’ phenomenon occurs, especially with ill-conditioned problems. In Algorithm 1 the descent directions do not move in orthogonal steps and according to the formula in step 14, the new multiplicative descent direction is a combination of the new multiplicative gradient and the previous multiplicative direction. Nevertheless, our numerical experiments indicate that the penalized model combined with the MCG algorithm does not only reproduce the results obtained with the model of Spiess combined with MSD algorithm, but also improves the computing time for both the complete and the reduced problem.

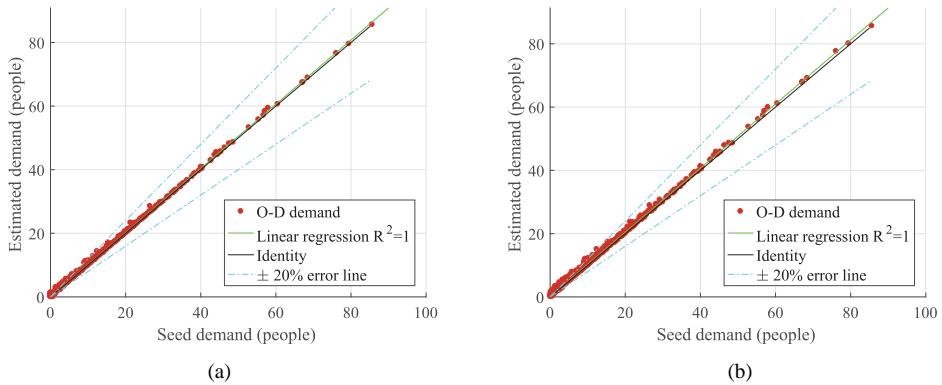
Looking at the MPE in Tables 4 and 5 as well as in Figures 3 (a) and 4 (a), we observe that the multiplicative algorithms tend to overestimate the O-D demand with respect to the seed matrix. Given the multiplicative nature of the algorithm, these increases are more noticeable in the zones with higher initial demand.

### 5.3. Numerical results with the DAMM algorithm

In this section, we show the results obtained by applying the methodology described in Section 3 considering the complete problem and its reduction (where null coefficients are extracted from the O-D matrices) discussed in Section 4, for both networks. Table 6 shows the numerical results obtained with the augmented Lagrangian model and Algorithm 2 (DAMM) for the Winnipeg transit network. The first three columns of the table include the values of the parameters  $k$ ,  $\rho$  and  $k/(1 + \rho)$ , common for both problems. From the fourth

**Table 6.** Results for the Winnipeg network obtained with the augmented Lagrangian model and the DAMM algorithm. From the fourth to the ninth column, results of the complete (reduced) problem are shown on the left (right) of the dash.

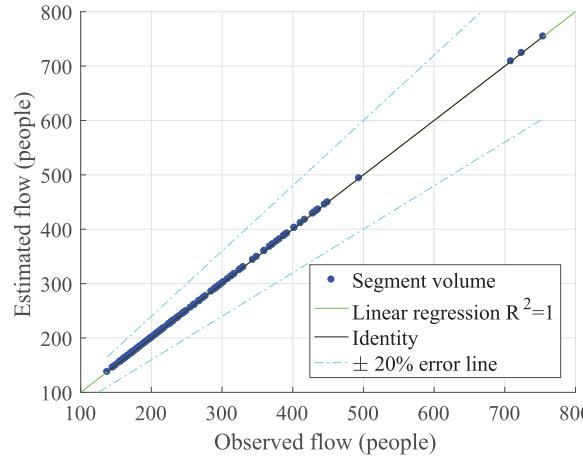
| $\rho$ | $k$             | $\frac{k}{1+\rho}$ | CPU (sec) | $(J, \bar{\ell})$ | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g - \hat{g}\ _n$ | MPE    |
|--------|-----------------|--------------------|-----------|-------------------|-----------------|----------------------|-----------------|---------------------|--------|
| 1      | $2 \times 10^2$ | $10^2$             | 2.5–0.6   | (7,28)–(3,55)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.83 |
|        | $2 \times 10^3$ | $10^3$             | 2.3–0.4   | (7,28)–(3,53)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |
|        | $2 \times 10^4$ | $10^4$             | 2.2–0.4   | (7,28)–(3,53)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |
|        | $\infty$        | –                  | 2.1–0.4   | (7,29)–(3,52)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |
| 19     | $2 \times 10^2$ | 10                 | 2.3–0.3   | (7,28)–(2,44)     | 0.09–0.09       | 1.1–1.1              | 0.12–0.18       | 19.2–27.5           | -40.98 |
|        | $2 \times 10^3$ | $10^2$             | 2.3–0.5   | (7,28)–(3,53)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.82 |
|        | $2 \times 10^4$ | $10^3$             | 2.2–0.4   | (7,28)–(3,53)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |
|        | $\infty$        | –                  | 2.2–0.4   | (7,28)–(3,53)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |
| 199    | $2 \times 10^3$ | 10                 | 2.2–0.4   | (7,29)–(2,44)     | 0.09–0.09       | 1.1–1.1              | 0.12–0.18       | 19.2–27.5           | -40.99 |
|        | $2 \times 10^4$ | $10^2$             | 2.1–0.4   | (7,28)–(3,52)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.82 |
|        | $2 \times 10^5$ | $10^3$             | 2.2–0.4   | (7,28)–(3,51)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |
|        | $\infty$        | –                  | 2.1–0.4   | (7,28)–(3,51)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |
| 1999   | $2 \times 10^3$ | 1                  | 2.4–0.1   | (7,30)–(1,17)     | 0.10–0.58       | 1.2–6.8              | 0.12–0.17       | 19.0–25.8           | -41.57 |
|        | $2 \times 10^4$ | 10                 | 2.2–0.2   | (7,29)–(2,44)     | 0.09–0.09       | 1.1–1.1              | 0.12–0.18       | 19.2–27.5           | -40.99 |
|        | $2 \times 10^5$ | $10^2$             | 2.2–0.4   | (7,28)–(3,52)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.82 |
|        | $\infty$        | –                  | 2.1–0.4   | (7,28)–(3,51)     | 0.09–0.01       | 1.1–0.1              | 0.12–0.18       | 19.2–27.8           | -40.84 |



**Figure 5.** Scatter plots for the updated O-D matrix obtained with  $k/(\rho + 1) = 10^3$  for the Winnipeg transit network. (a) Complete model and (b) Reduced model.

to the ninth column, we show together the numerical results for the complete problem (left) and the reduced problem (right), separated by a dash in the middle. Finally, the tenth column shows the MPE for the reduced problem. In this table,  $J$  denotes the number of iterations performed by the DAMM algorithm to reach convergence up to the desired tolerance ( $\varepsilon = 10^{-3}$ ), while  $\bar{\ell}$  is the average of CG iterations to find the minimum in step 3 of Algorithm 2. Thus, the total number of CG-iterations to solve the problem is  $J \times \bar{\ell}$ .

The qualitative behavior of the updated O-D matrices for the complete and reduced problem is shown in Figure 5. The slope and  $y$ -intercept of the corresponding regression lines shown in this figure are  $m_g = 1.01$ ,  $b_g = 0.05$  and  $m_{g_r} = 1.01$ ,  $b_{g_r} = 0.21$ , for the complete problem and the reduced one, respectively. These results show a significant improvement for demand estimation with respect to those obtained previously with the penalized model and the MCG algorithm (see Tables 2, 4 and Figure 3). Figure 6 shows the fit in the segment flows which are similar to those obtained with the penalized model.



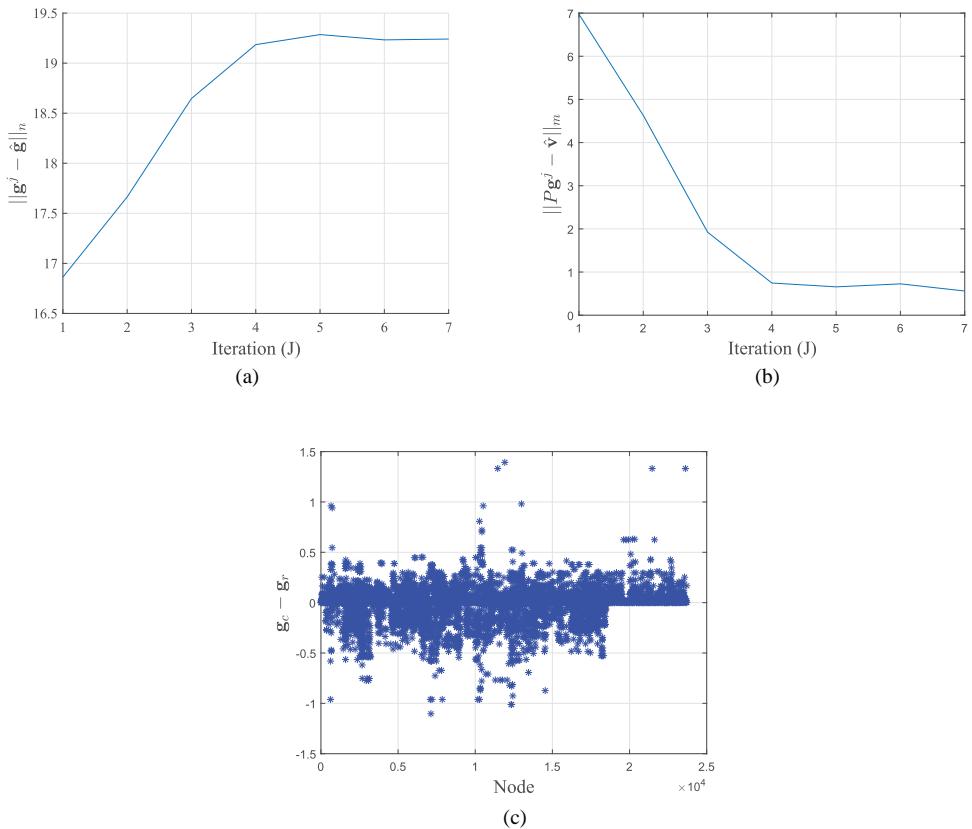
**Figure 6.** Scatter plots for the segment flows obtained with  $k/(\rho + 1) = 10^3$  for the Winnipeg transit network.

**Table 7.** Results for the MAVM network obtained with the augmented Lagrangian model and the DAMM algorithm. From the fourth to the ninth column the results of the complete (reduced) problem are shown on the left (right) of the dash.

| $\rho$ | $k$             | $\frac{k}{(1+\rho)}$ | CPU (s)  | $(J, \bar{\ell})$ | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g - \hat{g}\ _n$ | MPE   |
|--------|-----------------|----------------------|----------|-------------------|-----------------|----------------------|-----------------|---------------------|-------|
| 1      | $2 \times 10^2$ | $10^2$               | 9.5–1.1  | (1,28)–(1,28)     | 69.05–69.04     | 2647.2–2647.1        | 2.07–2.07       | 3528.7–3528.8       | -7.94 |
|        | $2 \times 10^3$ | $10^3$               | 9.2–0.8  | (1,28)–(1,28)     | 69.05–69.04     | 2647.3–2647.1        | 2.07–2.07       | 3529.1–3529.1       | -7.95 |
|        | $2 \times 10^4$ | $10^4$               | 9.3–0.9  | (1,28)–(1,28)     | 69.04–69.04     | 2647.1–2647.1        | 2.07–2.07       | 3529.2–3529.1       | -7.95 |
|        | $\infty$        | —                    | 9.6–0.7  | (1,28)–(1,28)     | 69.04–69.05     | 2647.1–2647.3        | 2.07–2.07       | 3529.2–3529.2       | -7.95 |
| 19     | $2 \times 10^2$ | 10                   | 9.0–0.8  | (1,27)–(1,27)     | 73.27–73.27     | 2809.2–2809.2        | 2.06–2.06       | 3508.7–3508.7       | -7.95 |
|        | $2 \times 10^3$ | $10^2$               | 9.3–0.9  | (1,28)–(1,28)     | 69.05–69.04     | 2647.3–2647.1        | 2.07–2.07       | 3528.7–3528.8       | -7.95 |
|        | $2 \times 10^4$ | $10^3$               | 9.2–0.8  | (1,28)–(1,28)     | 69.05–69.04     | 2647.5–2647.1        | 2.07–2.07       | 3529.0–3529.1       | -7.95 |
|        | $\infty$        | —                    | 10.3–0.8 | (1,28)–(1,28)     | 69.04–69.04     | 2647.1–2647.2        | 2.07–2.08       | 3529.2–3529.2       | -7.95 |
| 199    | $2 \times 10^3$ | 10                   | 9.3–0.9  | (1,27)–(1,27)     | 73.27–73.27     | 2809.2–2809.2        | 2.06–2.06       | 3508.7–3508.7       | -7.95 |
|        | $2 \times 10^4$ | $10^2$               | 9.7–0.8  | (1,28)–(1,28)     | 69.04–69.04     | 2647.1–2647.1        | 2.07–2.07       | 3528.8–3528.8       | -7.95 |
|        | $2 \times 10^5$ | $10^3$               | 9.9–0.8  | (1,28)–(1,28)     | 69.04–69.04     | 2647.1–2647.1        | 2.07–2.07       | 3529.1–3529.1       | -7.95 |
|        | $\infty$        | —                    | 10.0–1.0 | (1,28)–(1,28)     | 69.04–69.05     | 2647.1–2647.4        | 2.07–2.07       | 3529.2–3529.1       | -7.95 |
| 1999   | $2 \times 10^3$ | 1                    | 9.7–0.8  | (1,27)–(1,27)     | 73.82–73.82     | 2830.2–2830.2        | 2.04–2.04       | 3477.5–3477.5       | -7.93 |
|        | $2 \times 10^4$ | 10                   | 9.9–0.8  | (1,27)–(1,27)     | 73.27–73.27     | 2809.2–2809.2        | 2.06–2.06       | 3508.7–3508.7       | -7.95 |
|        | $2 \times 10^5$ | $10^2$               | 9.8–0.9  | (1,28)–(1,28)     | 69.04–69.05     | 2647.1–2647.3        | 2.07–2.07       | 3528.8–3528.7       | -7.95 |
|        | $\infty$        | —                    | 10.2–0.8 | (1,28)–(1,28)     | 69.04–69.04     | 2647.1–2647.2        | 2.07–2.07       | 3529.1–3529.2       | -7.95 |

In Table 6, we observe that the solution does not change for values of  $k/(\rho + 1) > 10^2$ , independently of the values of  $\rho$  and  $k$ . Also, the solution of the reduced problem is computed about five or six times faster than the solution of the complete problem due to memory saving. According to the last four columns in this table, segment flows are calculated more accurately for the reduced problem, meanwhile, the demand adjustment is slightly better for the complete problem. In Figure 7 we show the evolution of  $\|\mathbf{g}^J - \hat{\mathbf{g}}\|_n$  and  $\|Pg^J - \hat{v}\|_m$  along the iterations  $J$  of the DAMM algorithm. For completeness, we also show the point-wise difference between the solution of the complete and reduced models, denoted by  $\mathbf{g}_c - \mathbf{g}_r$ , for the case  $\rho = 19$  and  $k = 20000$ .

The last set of numerical experiments were done for the MAVM network. Table 7 summarizes the results obtained with the augmented Lagrangian model and the DAMM algorithm.

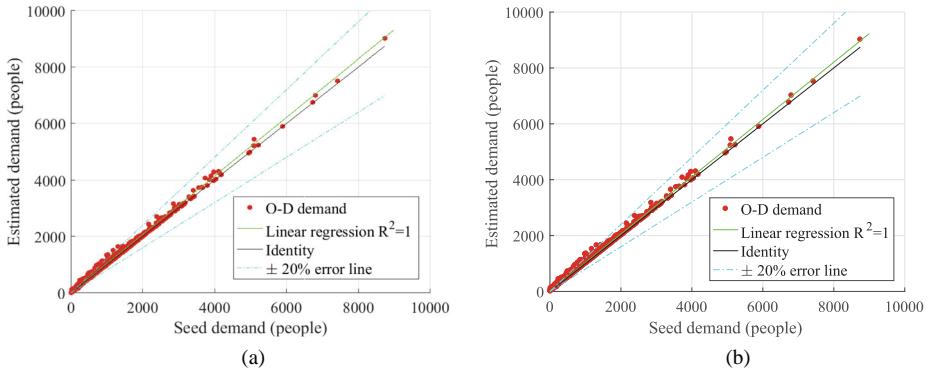


**Figure 7.** Evolution of the distances between the data and the estimated values for  $j = 1, \dots, J$ . Difference between the solution of the complete model and the reduced one for the Winnipeg network. (a) Evolution of  $\|g^j - \hat{g}\|_n$ , (b) Evolution of  $\|Pg^j - \hat{v}\|_m$  and (c) Difference of the solutions.

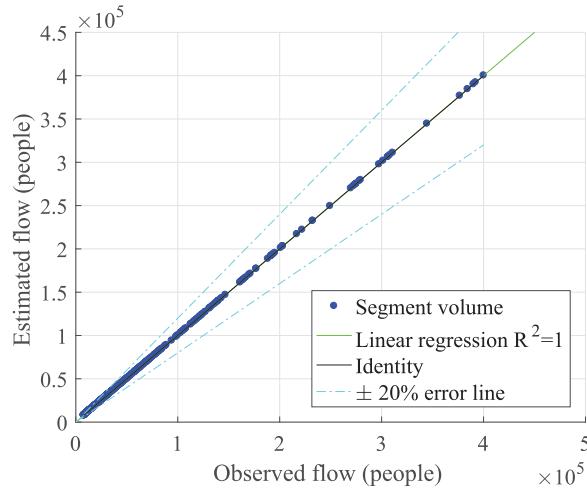
Once more, we include the results for both the complete problem and reduced problem together following the same format as in Table 6.

Once more, the numerical solutions obtained with  $k/(\rho + 1) > 10^2$  remain the same, independently of the values of  $\rho$  and  $k$ . This time, the solution of the reduced problem is computed about nine or ten times faster than the solution of the complete problem. It is remarkable that, contrary to what happens with the Winnipeg network, this time we obtain exactly the same results for both the complete and reduced models. The qualitative behavior of the solution for demand is shown in Figure 8, where the scatter plot for the updated O-D matrix is shown for the complete (a) and the reduced models (b). The slope and  $y$ -intercept of the corresponding regression line in (a) are  $m_g = 1.04$ ,  $b_g = 0.03$  and the slope and  $y$ -intercept of the regression line in (b) are  $m_{g_r} = 1.02$ ,  $b_{g_r} = 7.25$ . Scatter plots for the segment flows are shown in Figure 9.

Figure 10 shows the difference between the O-D matrices associated to the solution of the complete and the reduced models, denoted by  $\mathbf{g}_c - \mathbf{g}_r$ , where it is evident that the highest difference is very small and less than 0.025. Thus, the augmented Lagrangian model with the DAMM algorithm yields an accurate solution in an efficient way, at least for the large network of the MAVM.



**Figure 8.** Scatter plots for the updated O-D matrix obtained with  $k/(\rho + 1) = 10^3$  for the MAVM transit network. (a) Complete model and (b) Reduced model.



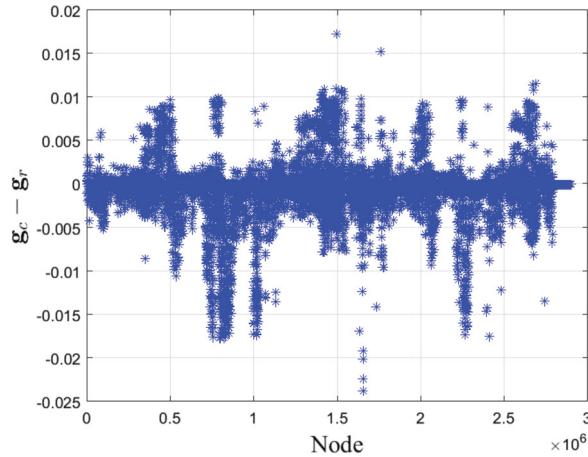
**Figure 9.** Scatter plots for the segment flows obtained with  $k/(\rho + 1) = 10^3$  for the MAVM transit network.

So far, the best results have been obtained with the augmented Lagrangian approach. These results can be improved to get more realistic results by adding more information. For instance, the total productions and attractions for each aggregated transit zone (centroid node). In this case, the following constraints must be added:

$$\sum_{q \in Q} g_{pq} = O_p, \quad \forall p \in P, \quad (32)$$

$$\sum_{p \in P} g_{pq} = D_q, \quad \forall q \in Q, \quad (33)$$

where  $O_p$  are the productions at each zone  $p \in P$  and  $D_q$  are the attractions at each zone  $q \in Q$ . These constraints can be represented using a matrix-vector product, let us say  $A\mathbf{g} = \mathbf{O}$  and  $B\mathbf{g} = \mathbf{D}$ . By penalizing the difference of these quantities and adding them to the



**Figure 10.** Difference between the solution of the complete model and the reduced model for the MAVM network.

**Table 8.** Results for the Winnipeg network obtained with the augmented Lagrangian model and the DAMM algorithm adding constraints (32)–(33).

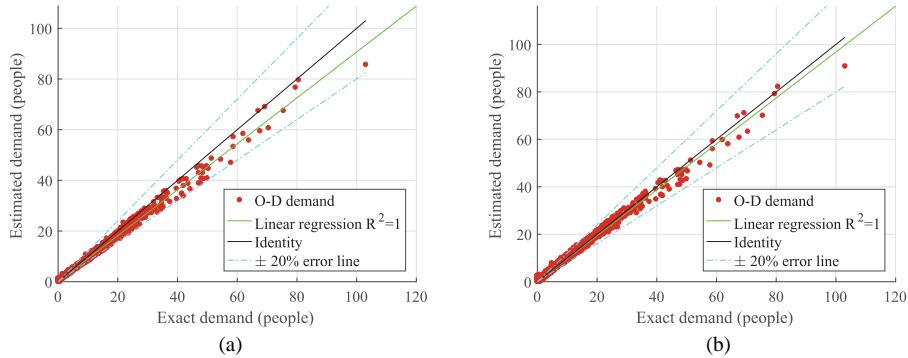
| $\rho$ | $k$             | $\frac{k}{(1 + \rho)}$ | CPU (s) | $(J, \bar{\ell})$ | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ \mathbf{g} - \hat{\mathbf{g}}\ _n$ |
|--------|-----------------|------------------------|---------|-------------------|-----------------|----------------------|-----------------|---------------------------------------|
| 1      | $2 \times 10^2$ | $10^2$                 | 5.8     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $2 \times 10^3$ | $10^3$                 | 5.4     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $2 \times 10^4$ | $10^4$                 | 5.5     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $\infty$        | –                      | 3.3     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
| 19     | $2 \times 10^2$ | 10                     | 5.7     | (26,17)           | 0.36            | 4.2                  | 0.32            | 48.9                                  |
|        | $2 \times 10^3$ | $10^2$                 | 5.8     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $2 \times 10^4$ | $10^3$                 | 5.5     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $\infty$        | –                      | 3.2     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
| 199    | $2 \times 10^3$ | 10                     | 5.4     | (26,17)           | 0.36            | 4.1                  | 0.32            | 48.9                                  |
|        | $2 \times 10^4$ | $10^2$                 | 5.4     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $2 \times 10^5$ | $10^3$                 | 5.6     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $\infty$        | –                      | 3.2     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
| 1999   | $2 \times 10^3$ | 1                      | 5.2     | (25,17)           | 0.38            | 4.5                  | 0.32            | 48.8                                  |
|        | $2 \times 10^4$ | 10                     | 5.7     | (26,17)           | 0.36            | 4.2                  | 0.32            | 48.9                                  |
|        | $2 \times 10^5$ | $10^2$                 | 5.5     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |
|        | $\infty$        | –                      | 3.1     | (26,17)           | 0.37            | 4.3                  | 0.32            | 48.9                                  |

augmented Lagrangian (21) approach we obtain the new Lagrangian:

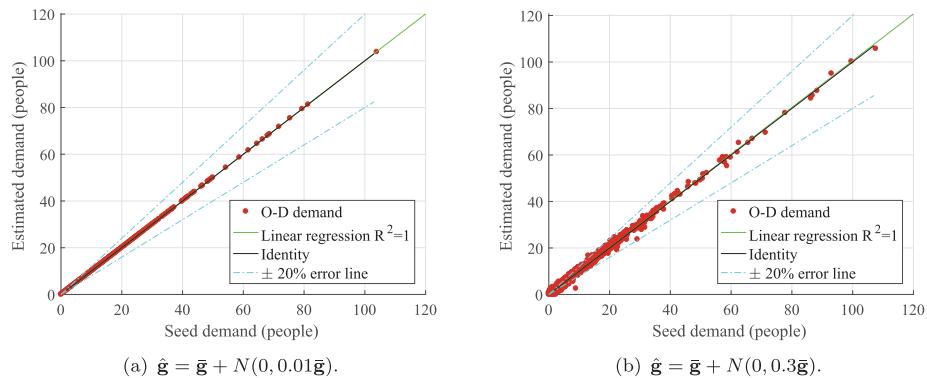
$$\mathcal{L}_{k_1, k_2, k_3, \rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = \mathcal{L}_{k_1, \rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) + \frac{k_2}{2} \|A\mathbf{g} - \mathbf{O}\|^2 + \frac{k_3}{2} \|B\mathbf{g} - \mathbf{D}\|^2.$$

The results obtained with this model and  $k_1 = k_2 = k_3$  for the Winnipeg network are shown in Table 8. Comparing these results with those shown in Table 6 for the complete model, we observe an increase in the number of iterations as well as in the CPU time. Also, the distance between the data and the estimations is larger when the constraints (32)–(33) are considered.

Although  $\|\mathbf{g} - \hat{\mathbf{g}}\|$  is larger when constraints (32)–(33) are incorporated this time the estimated  $\mathbf{g}$  is closer to the ‘exact solution’  $\bar{\mathbf{g}}$ , which in our experiments is available since we are using synthetic data (of course, the exact solution is not available in a real case). More



**Figure 11.** Scatter plots for the updated O-D matrix obtained with  $k/(\rho + 1) = 10^3$  against the O-D ‘exact’ demand  $\bar{\mathbf{g}}$  for the Winnipeg transit network. (a) Without considering constraints (32)–(33) and (b) Considering constraints (32)–(33).



**Figure 12.** Scatter plots for the demand obtained with different values of  $\hat{\mathbf{g}}$  for the Winnipeg transit network.

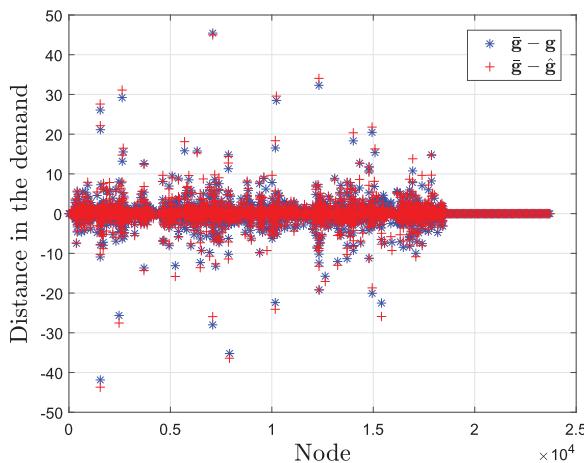
precisely  $\|\mathbf{g} - \bar{\mathbf{g}}\| = 45.3$  when constraints (32)–(33) are used in the model, and  $\|\mathbf{g} - \bar{\mathbf{g}}\| = 63.3$  when those constraints are not used. Therefore, this information confirms that the new results are close to those obtained without constraints (32)–(33), but the numerical results are slightly improved when productions and attraction are incorporated in the optimization model, at least for this particular network. Figure 11 shows the scatter plots for the respective results with  $k/(\rho + 1) = 10^3$ .

To evaluate the performance of the DAMM algorithm when different changes in the demand are expected, let us consider an outdated demand  $\hat{\mathbf{g}} = \bar{\mathbf{g}} + N(0, \delta\bar{\mathbf{g}})$ , where  $N(0, \delta\bar{\mathbf{g}})$  is a vector generated from a normal distribution with mean zero and standard deviation  $\delta\bar{\mathbf{g}}$ . Table 9 shows the numerical results for the Winnipeg network and different values of  $\delta$ . These results show that the distance between the old matrix and the estimated one increases in proportion to  $\delta$ . Figure 12 shows the scatter plots for the demand obtained with different values of  $\hat{\mathbf{g}}$ .

In Figure 13 we show the distance between the exact demand and the outdated one for each O-D pair in blue and the distance between the exact demand and the estimated one for each O-D pair in red. Notice that there is no significant variability in these distances.

**Table 9.** Results for the Winnipeg network obtained with the augmented Lagrangian model and the DAMM algorithm considering the constraints (32)–(33) and different values of  $\delta$ .

| $\delta$ | $\frac{\ \bar{g} - \hat{g}\ }{\ \bar{g}\ }$ | CPU (s) | $(J, \bar{\ell})$ | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g - \hat{g}\ _n$ |
|----------|---------------------------------------------|---------|-------------------|-----------------|----------------------|-----------------|---------------------|
| 0.01     | 0.01                                        | 0.7     | (1,57)            | 0.46            | 5.4                  | 0.01            | 0.8                 |
| 0.05     | 0.05                                        | 2.0     | (5,31)            | 0.48            | 5.5                  | 0.04            | 5.9                 |
| 0.10     | 0.10                                        | 3.0     | (10,24)           | 0.29            | 3.4                  | 0.07            | 11.1                |
| 0.20     | 0.21                                        | 3.8     | (19,18)           | 0.23            | 2.6                  | 0.15            | 22.8                |
| 0.30     | 0.31                                        | 2.5     | (24,14)           | 0.31            | 3.6                  | 0.26            | 39.6                |



**Figure 13.** Distances of  $\hat{g}$  and  $g$  from  $\bar{g}$  for the Winnipeg transit network.

As we mentioned in the introduction of this paper, the methodology presented here is for short-term planning where no drastic changes in the demand are expected.

## 6. Conclusions

We studied two methodologies to estimate O-D matrices in the context of transit networks: a (previously introduced) penalized model combined with the multiplicative conjugate gradient algorithm, and a new methodology, based on an augmented Lagrangian model combined with a dual ascent and method of multipliers algorithm. We show (theoretically and numerically) that the solutions of the penalized model converge to the solution of the problem of Spiess when the penalty parameter approaches infinity. The penalized model is equivalent to a quadratic regularization model and also to some models based on weighted averages, an important property that may help to get stability with respect to perturbations of the input data. These two methodologies are shown to be several times faster than the methodology of Spiess and more accurate when the augmented Lagrangian is employed to enforce non-negativity of the O-D matrix coefficients.

The numerical results show that we get the same solution, with the same computing time, for values of the penalty parameter  $k$  greater or equal than 1000. This behavior is consistent with the convergence properties of the penalized model in Section 2.1 and

**Table 10.** Overall comparison for the Winnipeg transit network.

| $k$   | $\rho$ | Method | CPU   | Iters  | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ _m$ | $\text{RMSE}_g$ | $\ g - \hat{g}\ _n$ |
|-------|--------|--------|-------|--------|-----------------|----------------------|-----------------|---------------------|
| 1000  | 0      | MSD    | 0.8 s | 78     | 0.17            | 1.9                  | 0.33            | 50.9                |
| 1000  | 0      | MSD-R  | 0.3 s | 72     | 0.18            | 2.1                  | 0.33            | 50.9                |
| 1000  | 0      | MCG    | 0.3 s | 21     | 0.11            | 1.3                  | 0.33            | 51.1                |
| 1000  | 0      | MCG-R  | 0.1 s | 21     | 0.11            | 1.3                  | 0.33            | 51.1                |
| 20000 | 19     | DAMM   | 2.2 s | (7,28) | 0.09            | 1.1                  | 0.12            | 19.2                |
| 20000 | 19     | DAMM-R | 0.4 s | (3,53) | 0.01            | 0.1                  | 0.18            | 27.8                |

**Table 11.** Overall comparison for the MAVM network.

| $k$   | $\rho$ | Method | CPU   | Iters  | $\text{RMSE}_v$ | $\ Pg - \hat{v}\ $ | $\text{RMSE}_g$ | $\ g - \hat{g}\ $ |
|-------|--------|--------|-------|--------|-----------------|--------------------|-----------------|-------------------|
| 1000  | 0      | MSD    | 23.6s | 147    | 78.92           | 3025.9             | 2.54            | 4324.9            |
| 1000  | 0      | MSD-R  | 5.6s  | 147    | 78.92           | 3025.9             | 2.54            | 4325.1            |
| 1000  | 0      | MCG    | 5.6s  | 30     | 52.34           | 2006.7             | 2.64            | 4499.3            |
| 1000  | 0      | MCG-R  | 1.2s  | 30     | 52.34           | 2006.7             | 2.64            | 4499.7            |
| 20000 | 19     | DAMM   | 9.2s  | (1,28) | 69.05           | 2647.5             | 2.07            | 3529.0            |
| 20000 | 19     | DAMM-R | 0.8s  | (1,28) | 69.04           | 2647.1             | 2.07            | 3529.1            |

with the results obtained with models based on weighted averages, like Noriega and Florian (2009) and Verbas, Mahmassani, and Zhang (2011), where convergent results were obtained around  $\beta = 0.999$  (equivalent to  $k = 1000$ , according to (12)).

Tables 10 and 11 show an overall comparison of the methodologies employed in this work for the Winnipeg and the MAVM networks. In the third column of those tables, MSD-R indicates the multiplicative steepest descent method (Spiess) for the reduced model, MCG-R and DAMM-R have a similar meaning.

The penalized model with MCG gives the same solution than the approach of Spiess with a lower computational cost, while the augmented Lagrangian model with the DAMM algorithm gets more accurate solutions with less computational cost for the larger transit network of the MAVM. The low CPU time of the calculations shows that quadratic models with appropriate solution algorithms are a good option for large-scale transit networks and may be adapted to the dynamic case. Future research lines are the following:

- Incorporate more information in the model. For example, zone information, upper and lower bounds, and travel costs.
- The matrix  $P$  can be reduced even more by deleting those columns which have only null coefficients (i.e. delete column  $j$  if  $P_{ij} = 0$ ,  $\forall i = 1, 2, \dots, m$ ). In this case, the corresponding value of  $g_j$  must be set to  $\hat{g}_j$ .
- The augmented Lagrangian approach can be implemented for other equality constraints and other algorithms that have been effective in other contexts (see Boyd et al. 2010); for instance, Balakrishnan, Magnanti, and Wong (1989) used the alternating direction method of multipliers for network design.

The next step in this research is to consider a nonlinear transit assignment. This case is more difficult since the convexity of the optimization models is lost. However, an iterative approach with linearization at each step, where our approach for linear models may be applied.

## Note

1. The Bolzano-Weierstrass theorem is a fundamental result about convergence in a  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ . It states that each bounded sequence in  $\mathbb{R}^n$  has a convergent subsequence.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

The authors want to acknowledge the financial support of CONACYT (National Council of Science and Technology in Mexico) through the scholarships for the first author, the support of Red de Matemáticas y Desarrollo (Network of Mathematics and Development-CONACYT, in Mexico) and Proyect 1948-Fronteras 2016-01. We acknowledge also the support from the Math Graduate program at the Autonomous Metropolitan University campus Iztapalapa in Mexico City.

## References

- Alsgaer, A., B. Assemi, M. Mesbah, and L. Ferreira. 2016. "Validating and Improving Public Transport Origin-Destination Estimation Algorithm Using Smart Card Fare Data." *Transportation Research Part C: Emerging Technologies* 68: 490–506.
- Antoniou, C., J. Barceló, M. Breen, M. Bullejos, J. Casas, E. Cipriani, and B. Ciuffo, et al. 2016. "Towards a Generic Benchmarking Platform for Origin-Destination Flows Estimation/Updating Algorithms: Design, Demonstration and Validation." *Transportation Research Part C: Emerging Technologies* 66: 79–98.
- Ashok, K., and M. E. Ben-Akiva. 2002. "Estimation and Prediction of Time-Dependent Origin-Destination Flows with a Stochastic Mapping to Path Flows and Link Flows." *Transportation Science* 36 (2): 184–198.
- Balakrishnan, A., T. L. Magnanti, and R. T. Wong. 1989. "A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design." *Operations Research* 37 (5): 716–740.
- Bell, M. G. H. 1991. "The Estimation of Origin-Destination Matrices by Constrained Generalized Least Squares." *Transportation Research Part B: Methodological* 25 (1): 13–22.
- Bera, S., and K. V. K. Rao. 2011. "Estimation of Origin-Destination Matrix from Traffic Counts: The State of the Art." *European Transport* 49: 3–23.
- Bierlaire, M. 1995. "Mathematical Models for Transportation Demand Analysis." Ph.D. Thesis, Facultés Universitaires Notre-Dame de la Paix de Namur, Faculté des Sciences, Département de Mathématique, Namur.
- Bierlaire, M., and L. Toint. 1995. "Meuse: An Origin-Destination Matrix Estimator that Exploits Structure." *Transportation Research, Part B: Methodological* 29 (1): 47–60.
- Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2010. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." *Foundations and Trends in Machine Learning* 3 (1): 1–122.
- Cascetta, E. 1984. "Estimation of Trip Matrices from Traffic Counts and Survey Data: A Generalized Least Squares Estimator." *Transportation Research Part B: Methodological* 18 (4–5): 289–299.
- Cascetta, E., and S. Nguyen. 1988. "A Unified Framework for Estimating or Updating Origin/Destination Matrices from Traffic Counts." *Transportation Research Part B: Methodological* 22 (6): 437–455.
- Cascetta, E., A. Papola, V. Marzano, F. Simonelli, and I. Vitiello. 2013. "Quasi-dynamic Estimation of O-D Flows from Traffic Counts: Formulation, Statistical Validation and Performance Analysis on Real Data." *Transportation Research Part B: Methodological* 55: 171–187.
- Chávez, M. V., and L. H. Juárez. 2014. "A Multiplicative Conjugate Gradient Method for the O-D Adjustment Matrix." In *Recent Advances in Theory, Methods, and Practice of Operations Research*, edited by R. Z. Ríos-Mercado et al., 97–104. Monterrey: UANL – Casa Universitaria del Libro.

- Chávez, M. V., and L. H. Juárez. 2016. "Demand Adjustment for the Transit Network of Mexico City and its Surroundings." Congreso Panamericano de Ingeniería y Tránsito, Transporte y Logística (PAMAM 2016), Mexico City, Mexico, 1338–1352.
- Chen, A., S. Pravinvongvuth, P. Chootinan, M. Lee, and W. Recker. 2007. "Strategies for Selecting Additional Traffic Counts for Improving O-D Trip Table Estimation." *Transportmetrica* 3 (3): 191–211.
- Chootinan, P., A. Chen, and H. Yang. 2005. "A Bi-Objective Traffic Counting Location Problem for Origin-Destination Trip Table Estimation." *Transportmetrica* 1 (1): 65–80.
- Cipriani, E., M. Florian, M. Mahut, and M. Nigro. 2011. "A Gradient Approximation Approach for Adjusting Temporal Origin-Destination Matrices." *Transportation Research Part C: Emerging Technologies* 19 (2): 270–282.
- Codina, E., and J. Barceló. 2000. "Adjustment of O-D Trip Matrices from Traffic Counts: An Algorithmic Approach Based on Conjugate Directions." Proceedings of the 8th Euro Working Group on Transportation, Rome, Italy, 427–432.
- Codina, E., R. García, and A. Marín. 2006. "New Algorithmic Alternatives for the O-D Matrix Adjustment Problem on Traffic Networks." *European Journal of Operation Research* 175 (3): 1484–1500.
- Doblas, J., and F. G. Benitez. 2005. "An Approach to Estimating and Updating Origin-Destination Matrices Based upon Traffic Counts Preserving the Prior Structure of a Survey Matrix." *Transportation Research Part B: Methodological* 39 (7): 565–591.
- Etemadnia, H., and K. Abdelghany. 2009. "Distributed Approach for Estimation of Dynamic Origin-Destination Demand." *Journal of the Transportation Research Board* 2105: 127–134.
- Florian, M., and Y. Chen. 1995. "A Coordinate Descent Method for the Bi-level O/D Matrix Adjustment Problem." *International Transactions on Operations Research* 2 (2): 165–179.
- Frederix, R., F. Viti, and C. M. J. Tampère. 2013. "Dynamic Origin-Destination Estimation in Congested Networks: Theoretical Findings and Implications in Practice." *Transportmetrica A: Transport Science* 9 (6): 494–513.
- Fujita, M., S. Yamada, and S. Murakami. 2016. "Time Coefficient Estimation for Hourly Origin-Destination Demand from Observed Link Flow Based on Semidynamic Traffic Assignment." *Journal of Advanced Transportation* 2017: 1–17.
- Heidari, A. A., A. Moayedi, and R. Ali Abbaspour. 2017. "Estimating Origin-Destination Matrices Using an Efficient Moth Flame-based Spatial Clustering Approach." Tehran's Joint ISPRS Conferences of GI Research, SMPR and EOEC, Tehran, 381–387.
- Hu, X., Y. Chiu, J. A. Villalobos, and E. Nava. 2017. "A Sequential Decomposition Framework and Method for Calibrating Dynamic Origin-Destination Demand in a Congested Network." *IEEE Transactions on intelligent transportation systems* 18 (10): 2790–2797.
- INRO. 2018. The Evolution of Transport Planning. <http://www.inrosoftware.com/en/products/emme/index.php>.
- Juárez, L. H., and M. V. Chávez. 2014. "O-D Matrix Adjustment for Transit Networks by Conjugate Gradient Iterations." *Investigación Operacional* 36 (2): 115–126.
- Kumar, A. A., J. E. Kang, C. Kwon, and A. Nikolaeva. 2016. "Inferring Origin-Destination Pairs and Utility-based Travel Preferences of Shared Mobility System Users in a Multi-modal Environment." *Transportation Research Part B: Methodological* 91: 270–291.
- Lundgren, J., and A. Peterson. 2008. "A Heuristic for the Bilevel Origin–Destination-Matrix Estimation Problem." *Transportation Research Part B: Methodological* 42 (4): 339–354.
- Malapert, A., and J. M. Kuusinen. 2017. "Estimation of Elevator Passenger Traffic Based on the Most Likely Elevator Trip Origin-Destination Matrices." *Building Services Engineering Research and Technology* 38 (5): 563–579.
- Michau, G., N. Pustelnik, P. Borgnat, A. Bhaskar, and E. Chung. 2017. "A Primal-dual Algorithm for Link Dependent Origin Destination Matrix Estimation." *IEEE Transactions on Signal and Information Processing over Networks* 3 (1): 104–113.
- Nocedal, J., and S. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research and Financial Engineering. New York: Springer-Verlag.
- Noriega, Y., and M. Florian. 2009. *Some Enhancements of the Gradient Method for O-D Matrix Adjustment*. Technical Report 4. CIRRELT.

- Nuzzolo, A., and A. Comi. 2016. "Advanced Public Transport and Intelligent Transport Systems: New Modelling Challenges." *Transportmetrica A: Transport Science* 12 (8): 674–699.
- Pitombeira-Neto, A. R., C. F. Grangeiro, and L. E. Carvalho. 2016. "Bayesian Inference on Dynamic Linear Models of Day-to-day Origin-Destination Flows in Transportation Networks." 1–22. Accessed October 2018. <https://arxiv.org/abs/1608.06682>.
- Shafiei, S., M. Saberi, and M. Sarvi. 2016. "Application of an Exact Gradient Method to Estimate Dynamic Origin-Destination Demand for Melbourne Network." IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1945–1950.
- Shen, W., and L. Winter. 2012. "A New One-level Convex Optimization Approach for Estimating Origin-Destination Demand." *Transportation Research Part B* 46 (2012): 1535–1555.
- Sherali, H. D., R. Sivanandan, and A. G. Hobeika. 1994. "A Linear Programming Approach for Synthesizing Origin-Destination Trip Tables from Link Traffic Volumes." *Transportation Research Part B: Methodological* 8 (3): 213–233.
- Spiess, H. 1990. *A Gradient Approach for the O-D Matrix Adjustment Problem*. Switzerland: EMME/2 Support Center. Accessed October 2018. <http://www.spiess.ch/emme2/demadj/demadj.html>.
- Spiess, H., and M. Florian. 1989. "Optimal Strategies: A new Assignment Model for Transit Networks." *Transportation Research Part B: Methodological* 23 (2): 83–102.
- Tarantola, A. 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia: Society for Industrial and Applied Mathematics.
- Torres, P. 2013. Subgerente de planeación estratégica, Sistema de Transporte Colectivo - Metro. Accessed October 2018. <http://www.metro.cdmx.gob.mx>.
- Verbas, I. O., H. S. Mahmassani, and K. Zhang. 2011. "Time-Dependent Origin-Destination Demand Estimation." *Transportation Research Board* 2263: 45–56.
- Vollebregt, E. A. H. 2014. "The Bound-Constrained Conjugate Gradient Method for Non-negative Matrices." *Journal of Optimization Theory and Applications* 162 (3): 931–953.
- Xie, C., K. M. Kockelman, and S. Travis. 2011. "A Maximum Entropy-Least Squares Estimator for Elastic Origin-Destination Trip Matrix Estimation." *Transportation Research Part B: Methodological* 45 (9): 1465–1482.