O–D MATRIX ADJUSTMENT FOR TRANSIT NETWORKS BY CONJUGATE GRADIENT ITERATIONS

L. Héctor Juárez and M. Victoria Chávez
Universidad Autónoma Metropolitana–Iztapalapa
Av. San Rafael Atlixco 186 Col. Vicentina, Mexico D.F., CP 09340
hect@xanum.uam.mx, vicky.fis@gmail.com

ABSTRACT
The adjustment of an obsolete demand matrix, from some given known data, is an important issue for transport research. In this article we introduce a penalized model, based on volume counts on a given set of arcs or segments, to update the demand matrix. Also, we propose a multiplicative conjugate gradient algorithm to solve the resultant convex optimization problem. This algorithm has been programmed with the macro language of EMME and tested with a synthetic scenario from the Winnipeg network. The numerical results show that the proposed algorithm improves the performance of the traditional multiplicative steepest descent algorithm, introduced by Spiess.

KEYWORDS: O-D matrix, demand models, transit assignment, convex optimization, conjugate gradient method, bilevel programming.

MSC:

RESUMEN
El ajuste de una matriz de demanda obsoleta, cuando se conocen ciertos datos, es un tema importante en la investigación del transporte. En este artículo introducimos un modelo penalizado, basado en el conteo de volúmenes sobre ciertos arcos o segmentos, para actualizar la matriz de demanda. También, proponemos un algoritmo multiplicativo de gradiente conjugado para resolver el problema resultante de optimización convexa. Este algoritmo ha sido programado utilizando el macro lenguage de EMME y se ha aplicado a un escenario sintético, obtenido de la red de Winnipeg. Los resultados numéricos muestran que el algoritmo propuesto mejora el desempeño del método tradicional de descenso máximo, introducido por Spiess.

1. INTRODUCTION

Public transport is becoming more relevant in modern societies, especially in large cities, where a good transportation planning is extremely important for many obvious reasons. Therefore, a good knowledge of the transit network and of the operation of the transportation system is necessary. In particular, mathematical models for transit assignment are very useful to help understanding how users travel from their different origins to their diverse destinations. These models must replicate realistic scenarios as close as possible, and for this purpose it is necessary to collect field data. Data may be obtained based on surveys and other complex and expensive studies, but unfortunately they are useful only for a limited short time, due to growth in demand and change in infrastructure around big cities. To avoid making new comprehensive studies, there are some techniques that allow approximations to the most recent data, using previously known data and only a small (but significant enough) amount of new data.

One of the most important elements on the transportation planning process is the O–D matrix, which represents the flow between each origin node to each destination node of the transportation network. Generating or updating these data is not trivial, since people travel to satisfy many diverse needs, and it clearly depends on many factors, such as day and time. Additionally, due to its nature, the O–D matrix is very difficult and costly to estimate requiring a great amount of resources, while the resultant accuracy is frequently very low. One way to estimate demand may be using ticket counting machines at bus stations, or using sensors; but this information, if it is available, may not be entirely realistic. Another more efficient and less expensive way to estimate, or improve O–D matrices, is from volume counts at some specific important links of the network. A large amount of research has been carried out in this direction and many models has been proposed in the past such as those in [2], [3], [5], [11], [13], [14], [15], [17], among many others.

Most of those traditional approaches can be formulated as convex optimization problems for which the objective function corresponds to some distance function between the obsolete demand matrix and the unknown demand. Some constraints are then added in order to force the assigned volumes to be close to the observed volumes on the correspondent arcs (or segments). These problems can be reformulated as least-squares models, where the volume constraints are relaxed and incorporated as additional terms in the objective function. Least-squares models has been studied and extended over the years, as in [2] and [13], and they are part of the well known bilevel programs, [5], [11]. An important issue on these programs is finding a good iterative algorithm that solves efficiently the corresponding model for large scale networks. A more general overview of O–D matrix estimation can be found in reference [1].

Given the amount of computational resources needed to solve those programs, specially for large scale networks, with hundreds of transit zones and thousands of network links, in 1990 Spiess proposed a new approach [15]. Instead of defining the objective function with the distance between the obsolete matrix, $\mathbf{G} = \{G_{pq}\}$, and the resulting demand, $\mathbf{g} = \{g_{pq}\}$ (where $pq \in PQ$ denote O–D pairs), he defined the objective function as a measure of the distance between observed and assigned volumes:

$$\min_{\mathbf{g}} Z(\mathbf{g}) = \frac{1}{2} \sum_{a \in \bar{A}} (v(\mathbf{g})_a - V_a)^2 \tag{1}$$

$$v(\mathbf{g}) = assign(\mathbf{g}), \tag{2}$$

where $\bar{A}$ is the subset of links where counts are available, V_a are the measured volumes, and $v(\mathbf{g}) = assign(\mathbf{g})$ indicates the volumes resulting from an assignment of the demand matrix $\mathbf{g}$. This assignment procedure must correspond to a convex optimization problem and it is understood as an equilibrium assignment to ensure the convexity of the model. However, problem (1)–(2) is an ill posed problem, since it usually admits an infinite number of optimal solutions (it is highly underdetermined), i.e. there are infinite many possible demand matrices, and each of them reflects the observed volumes equally well. To overcome this degeneracy, Spiess proposed the application of the method of steepest descent, which eventually would find a solution close to the starting point (the obsolete demand matrix $\mathbf{G}$). Thus, he introduced a multiplicative steepest descent method:

$$g_{pq}^{l+1} = \begin{cases} G_{pq} & \text{for } l = 0, \\ g_{pq}^{l} \left(1 - \lambda_l \frac{\partial Z(\mathbf{g}^{\mathbf{l}})}{\partial g_{pq}}\right) & \text{for } l = 1, 2, ..., L, \end{cases} \tag{3}$$

This multiplicative algorithm keeps the structure of the obsolete matrix, and its simplicity allows its application to large scale networks. This algorithm is implemented in the EMME transportation planning system [9].

In this work we want to to explore the previous problems and models in the context of transit networks instead of traffic networks. Thus, a transit assignment based on optimal strategies, which was introduced in [16], will be used instead of a traffic equilibrium assignment. Also, we follow another approach to a bilevel model. Here we think the problem as a approximated control problem where we want to drive the system (transit network) to a state in which the assigned volumes are as close as possible to the measured volumes, by adjusting the demand matrix. This kind of formulation give rise to an ill posed problem, which not always has solution, but it can be regularized, by penalizing the difference between the measured and assigned volumes, and then adding the square of this term to the objective function. With this approach, the model of Spiess can be thought as a limiting case when the penalization parameter goes to infinity.
Also, we believe that the natural iterative method to solve these quadratic convex problems is the conjugate gradient algorithm, which has proved to be more efficient than the steepest descent method for many models and in a great variety of contexts. Thus, we introduce a multiplicative conjugate gradient algorithm which has a comparable computational cost as steepest descent but with the advantage of requiring much less iterations to converge. To carry on this study we programmed these iterative algorithms using the computer macro language of the EMME4 software. This program can handle both the model of Spiess and bilevel optimization problems.
The organization of the article is as follows. In Section 2. we derive the O–D matrix adjustment model, where the difference of volumes is penalized. The multiplicative conjugate gradient algorithm to solve the problem is shown in Section 3. Then, in Section 4 we show some numerical experiments with the Winnipeg transit network. Finally, in Section 5 we write some conclusions.

2. THE O–D MATRIX ADJUSTMENT PROBLEM

Consider a transit network where an obsolete O–D matrix is known, and where volume counts are available on a subset of links (or segments). Let us denote by A the set of network links, and by $\bar{A}$ its subset where volume counts are available, and let $\{V_a\}_{a\in\bar{A}}$ be the set of measured volumes. We want to find a new demand matrix $\mathbf{g}$, close to the obsolete matrix $\mathbf{G}$, such that its assigned volumes v_a are equal (or close) to the observed volumes V_a, for each $a \in \bar{A}$. This problem can be formulated as follows.

Find an O–D matrix $\mathbf{g}$ that solves

$$\min_{\mathbf{g}} Z(\mathbf{g}) = \frac{1}{2}\sum_{pq\in PQ}(g_{pq} - G_{pq})^2 \tag{4}$$

such that

$$V_a = v(\mathbf{g})_a, \quad \forall\, a \in \bar{A} \tag{5}$$

$$v(\mathbf{g}) = assign(\mathbf{g}). \tag{6}$$

The assignment $assign(\mathbf{g})$ can be done in many forms. Here we consider the transit assignment, based on optimal strategies, introduced in [16]. The optimal strategy is obtained by the distribution of the demand on the different links, in such a way that the total transit time in the system (travel time + waiting time) is minimum. In the case of a non congested network this yields a convex optimization problem that is solved very efficiently using dynamic programming. This formulation yields a unique distribution of the demand $\mathbf{g}$ on the transit network, represented by the volume flows $\{v_a\}_{a\in A}$. The corresponding program, and variants that considers congestion, are included in the EMME transportation planning system [9]. Let us to say that we have applied with success this assignment procedure to the transit network associated to the metropolitan area that includes Mexico City and neighboring municipalities [7], [10].

The problem given by (4)–(6) is a variant of a control (or inverse) problem, and it is ill posed. So, to find a solution a regularization procedure is needed first. Actually, problem (4)–(6) can be thought as one in which the the transit system want to be driven to a desired state, given by (5), using the demand matrix $\mathbf{g}$ as control variable. To find a solution, equation (5) is relaxed, and therefore we seek a demand matrix $\mathbf{g}$, such that the assigned volume $v_a(\mathbf{g})$ gets close enough to the known value V_a, for each $a \in \bar{A}$. We can include in the model this restirction by penalizing the differences between these two quantities and adding them to the objective function. We obtain the following problem

$$\min_{\mathbf{g}} Z(\mathbf{g}) = \frac{1}{2} \sum_{pq \in PQ} (g_{pq} - G_{pq})^2 + \frac{k}{2} \sum_{a \in \bar{A}} (v(\mathbf{g})_a - V_a)^2, \qquad k > 0 \tag{7}$$

subject to:

$$v(\mathbf{g}) = assign(\mathbf{g}), \text{ and } g_{pq} \geq 0 \ \forall \ pq \in PQ, \tag{8}$$

where k is the penalty coefficient. The effect of this formulation is that, in most cases, the restriction (5) is satisfied very accurately when large values of k are imposed, as it is demonstrated by the numerical results. It is possible that the measured volumes are more reliable or accurate on some links or segments. However, in those cases, different penalty values k_a may be used for each $a \in \bar{A}$. To keep the discussion as simple as possible, in this paper a constant value of k is considered for all links or segments.

The previous formulation is equivalent to the bilevel programs [2], [5], [11], [13]. For instance, the objective function proposed in [13] is

$$Z(\mathbf{g}) = \frac{\alpha}{2} \sum_{a \in \bar{A}} (v(\mathbf{g})_a - V_a)^2 + \frac{1-\alpha}{2} \sum_{pq \in PQ} (g_{pq} - G_{pq})^2, \quad 0 \leq \alpha \leq 1. \tag{9}$$

The equivalence of this objective function with the objective function (7) can be obtained with $k = \alpha/(1-\alpha)$ or $\alpha = k/(k+1)$. Therefore, the model of Spiess (1) is obtained with $k = \infty$ or $\alpha = 1$. For us, it is more intuitive the penalized model (7), since it indicates directly the relative importance given to condition (5), through the value given to the penalization parameter k. Also, this model indicates the level of regularization applied to the ill posed problem, which is $1/k$.

3. A MULTIPLICATIVE CONJUGATE GRADIENT METHOD

The iterative descent methods are distinguished from each other by the descent direction chosen at each iteration, and by the step size to ensure a good decrease of the objective function. One of the simplest and most intuitive method is the method of steepest descent, where the descent direction is taken as the opposite to the gradient direction. The method of steepest descent has been successfully used to solve the O–D matrix adjustment problem (see [13] and [15], for instance). However, it is well known that in practice the method of steepest descent can be inefficient because of the zig–zag phenomenon, which occurs mainly with ill conditioned problems, requiring a large number of iterations to approach the optimum [12]. For those cases, the Newton's method converge much faster, but it becomes very expensive, specially for large scale problems, since it needs the evaluation of Hessians and the solution of a linear algebraic equation at each iteration. The conjugate gradient method does not share those disadvantages, it has a comparable computational cost to the steepest descent, and it is particularly efficient for convex quadratic optimization problems.

The conjugate gradient algorithm to solve problem (7)–(8) can be formulated as follows:

$$g_{pq}^{l+1} = \begin{cases} G_{pq} & \text{for } l = 0, \\ g_{pq}^{l} + \lambda_l \, d_{pq}^{l} & \text{for } l = 1, 2, \ldots, L, \end{cases} \qquad \forall \; pq \in PQ, \tag{10}$$

where the starting value in the iterative process is the obsolete O–D matrix $\mathbf{G}$, and $\mathbf{d}^l = \{d_{pq}^l\}$ is a conjugate direction vector at iteration l, and finally λ_l is the length step that minimizes the objective function along that direction. We will explain in detail how to compute the last two values, $\mathbf{d}^l = \{d_{pq}^l\}$ and λ_l. But before, let us introduce the multiplicative iterative version of this algorithm, where a change in demand is proportional to the demand in the initial matrix and where, in particular, zeros are preserved in the iterative process:

$$g_{pq}^{l+1} = \begin{cases} G_{pq} & \text{for } l = 0, \\ g_{pq}^{l} \left(1 + \lambda_l \, d_{pq}^{l}\right) & \text{for } l = 1, 2, \ldots, L. \end{cases} \qquad \forall \; pq \in PQ. \tag{11}$$

This is the equivalent version of the multiplicative iteration formula (3) introduced by Spiess [15].
The conjugate direction at each new iteration d_{pq}^{l+1} is generated as a linear combination of the previous conjugate direction and the current gradient. Thus

$$d_{pq}^{l+1} = g_{pq}^{l} \left[-\frac{\partial Z(\mathbf{g}^{l+1})}{\partial g_{pq}} \right] + \beta_l \, d_{pq}^{l}, \quad pq \in PQ, \tag{12}$$

where the constant β_l is computed to ensure that the two directions d_{pq}^l and d_{pq}^{l+1} are conjugate to each other. Notice that in (12) we have multiplied by g_{pq}^l to preserve the multiplicative structure of the algorithm. The gradient in (12) can be computed from (7):

$$\frac{\partial Z(\mathbf{g}^{l+1})}{\partial g_{pq}} = (g_{pq}^{l+1} - G_{pq}) + k \sum_{a \in \bar{A}} \left(v_a(\mathbf{g}^{l+1}) - V_a\right) \frac{\partial v_a(\mathbf{g}^{l+1})}{\partial g_{pq}}, \quad pq \in PQ. \tag{13}$$

The last derivative in (13) can be computed from the relation between link flows and path flows:

$$v_a(\mathbf{g}^{l+1}) = \sum_{pq \in PQ} \sum_{s \in S_{pq}} \delta_{as} h_s, \quad a \in \bar{A}, \text{ and } \delta_{as} := \begin{cases} 0 & if \; a \notin s \\ 1 & if \; a \in s \end{cases} \tag{14}$$

where S_{pq} is the set of used paths in the network to travel from $p \in P$ to $q \in Q$, and h_s denotes the total flow along one path $s \in S_{pq}$. Equation (14) can be rewritten in terms of the path probabilities $\pi_s^{l+1} = h_s / g_{pq}^{l+1}$, $s \in S_{pq}$, $pq \in PQ$:

$$v_a(\mathbf{g}^{l+1}) = \sum_{pq \in PQ} g_{pq}^{l+1} \sum_{s \in S_{pq}} \delta_{as} \, \pi_s^{l+1}, \qquad a \in \bar{A}. \tag{15}$$

Assuming that $\pi_s^{l+1} \approx \pi_s^l$, we get

$$\frac{\partial v_a}{\partial g_{pq}}(\mathbf{g}^{l+1}) = \sum_{s \in S_{pq}} \delta_{as} \pi_s^l, \qquad a \in \bar{A}, \quad pq \in PQ. \tag{16}$$

Therefore,

$$\frac{\partial Z(\mathbf{g}^{l+1})}{\partial g_{pq}} = (g_{pq}^{l+1} - G_{pq}) + k \sum_{s \in S_{pq}} \pi_s^l \sum_{a \in \bar{A}} \delta_{as} \left(v_a(\mathbf{g}^{l+1}) - V_a\right), \quad pq \in PQ. \tag{17}$$

Remark. The assumption $\pi_s^{l+1} \approx \pi_s^l$ is very reasonable, specially when the sequence $\{\mathbf{g}^l\}$ is close to the optimum. It not only simplifies the computation of the gradient $\nabla Z(\mathbf{g})$, but also gives a "linear behavior" to $v_a(\mathbf{g})$, since in this case $v_a(\mathbf{g} + \lambda \mathbf{d}) \approx v_a(\mathbf{g}) + \lambda v(\mathbf{d})$ when $||\mathbf{d}||$ is small.
Concerning the optimal step length λ_l in (11), it can be found as the minimum of the one–dimensional subproblem $\phi(\lambda) = Z\left(\mathbf{g}^l + \lambda\, \mathbf{d}^l\right)$. More precisely

$$\min_{\lambda} \phi(\lambda) \approx \frac{1}{2} \sum_{pq \in PQ} \left(g_{pq}^l + \lambda\, d_{pq}^l - G_{pq}\right)^2 + \frac{k}{2} \sum_{a \in \bar{A}} \left(v_a(\mathbf{g}^l) + \lambda\, v_a(\mathbf{d}^l) - V_a\right)^2 \tag{18}$$

$$\text{subject to } \lambda \mathbf{d}^l \leq 1, \text{ and } g_{pq}^l \geq 0 \ \forall\ pq \in PQ. \tag{19}$$

This optimization problem has the solution

$$\lambda_l \approx \frac{\displaystyle\sum_{pq \in PQ} d_{pq}^l (G_{pq} - g_{pq}^l) + k \sum_{a \in \bar{A}} v_a(\mathbf{d}^l)\,(V_a - v_a(\mathbf{g}^l))}{\displaystyle\sum_{pq \in PQ} (d_{pq}^l)^2 + k \sum_{a \in \bar{A}} v_a(\mathbf{d}^l)^2}. \tag{20}$$

After obtaining $\mathbf{g}^{l+1}$, the new conjugate direction is computed using formula (12), where the value of β_l is calculated by an adaptation of the Hestenes–Stiefel formula (see ref. [12]). We obtain

$$\beta_l = \frac{\displaystyle\sum_{pq \in PQ} g_{pq}^{l+1} \frac{\partial Z(\mathbf{g}^{l+1})}{\partial g_{pq}} \left(\frac{\partial Z(\mathbf{g}^{l+1})}{\partial g_{pq}} - \frac{\partial Z(\mathbf{g}^l)}{\partial g_{pq}}\right)}{\displaystyle\sum_{pq \in PQ} d_{pq}^l \left(\frac{\partial Z(\mathbf{g}^{l+1})}{\partial g_{pq}} - \frac{\partial Z(\mathbf{g}^l)}{\partial g_{pq}}\right)}. \tag{21}$$

Notice that in this formula we have also multiplied each term in the numerator by g_{pq}^{l+1} to keep the multiplicative structure of the iterative algorithm.
The statement of the multiplicative conjugate algorithm is given next.

Multiplicative conjugate gradient algorithm for matrix adjustment

Initialization. Given the initial demand $\mathbf{g}^0 = \mathbf{G}$, do the following.

1. Assign the demand to get the volumes: $\mathbf{v}(\mathbf{g}^0) = assign(\mathbf{g}^0)$.
2. Compute the initial direction: $d_{pq}^0 = -\frac{\partial Z(\mathbf{g}^0)}{\partial g_{pq}}$, $pq \in PQ$, using formula (17) with $l = -1$.

Descent. For $l \geq 0$, assuming that we know $\mathbf{g}^l$, $\mathbf{d}^l$, do the following steps.

3. Assign the demand on the network to get $\mathbf{v}(\mathbf{g}^l) = assign(\mathbf{g}^l)$. Compute $\mathbf{v}(\mathbf{d}^l)$ with (15).
4. Compute λ_l in formula (20) with the known values $\mathbf{g}^l$, $\mathbf{d}^l$, $\mathbf{v}(\mathbf{g}^l)$, $\mathbf{v}(\mathbf{d}^l)$.
5. Update the demand matrix: $g_{pq}^{l+1} = g_{pq}^l \left(1 + \lambda_l\, d_{pq}^l\right)$ for all $pq \in PQ$.
6. Compute the gradient of the objective function at $\mathbf{g}^{l+1}$: apply formula (17).

Stopping condition and new conjugate gradient direction. Given $0 < \epsilon \ll 1$ (the stopping parameter), do the following.

7. If $\|\nabla Z(\mathbf{g}^{l+1})\| \leq \epsilon \, \|\nabla(Z(\mathbf{g}^0)\|$, take $\mathbf{g} = \mathbf{g}^{l+1}$, stop and exit.
8. Otherwise, compute β_l with formula (21).
9. Compute the new conjugate gradient direction $\mathbf{d}^{l+1}$ with formula (12).
10. Update the index: $l = l + 1$ and go to 3.

4. NUMERICAL RESULTS

For the numerical experiments we employed the transit network from the city of Winnipeg, Canada, obtained from the standard EMME/4 Winnipeg demonstration database [9]. This is a network of 154 zones, 906 nodes, 3005 directional links, 4347 transit segments and 136 volume counts. The network is displayed in Figure 1, where the set of segments with available volume counts ($\bar{A}$) is shown in green.

Figure 1: Segment counts on the Winnipeg network.

We built up the following synthetic scenario: we first did a transit assignment to the Winnipeg network, with the exact O–D matrix $\mathbf{g}$ at the peak hour in the morning. From the result of this assignment we extracted the volumes, which play the role of measured volumes $\{V_a\}_{a \in \bar{A}}$ in the numerical experiments. Finally, we generated an '*obsolete*' O–D matrix $\mathbf{G}$ by doing a stochastic perturbation of 20% of the exact O–D matrix $\mathbf{g}$.

Then, with the generated data, we applied our multiplicative conjugate gradient method (CG) to see how much we can recover of the original demand matrix. This CG algorithm was programmed with the macro language of EMME and, consequently, the transit assignments in steps 1 and 3 of the CG algorithm were

done with a trial version of the software package EMME/4. To stop the iterations in the CG algorithm (step 7) we chose $\epsilon = 10^{-3}$. All the numerical calculations were done in a HP–Pavilion dm4 computer desk, which has an Intel(R) Core(TM) i5 processor and 3 GB RAM.
The numerical results show that this new algorithm improves the performance of the traditional multiplicative steepest descent method of Spiess (SD). Table 1 shows the values of a least squares fit of the deviation from the original O–D matrix obtained with both iterative algorithms. Similarly, Table 2 shows the correspondent values of the least squares fit of the adjusted volumes versus the real volumes. In both tables, A and B are the parameter values of the regression line, thus the adjustment is better for those points that are closer to the correspondent line. On the other hand, R^2 and $RMSE$ are the correlation coefficient and the square root of the mean squared error, respectively. A more detailed information about these regression parameters can be found in [4] and [6]. In both tables, Iters denotes the number of iterations to achieve convergence up to the given accuracy. Finally, in the last column we included the values of the quadratic sums $\sum_{pq\in PQ}(g_{pq}-G_{pq})^2$ and $\sum_{a\in\bar{A}}(v_a-V_a)^2$, respectively.

Table 1: Demand deviation regression coefficients and convergence w.r.t. k.

k	Method	A	B	R^2	$RMSE$	Iters.	$\sum_{pq}(g_{pq}-G_{pq})^2$
100	SD	-0.003	0.976	0.993	0.581	126	2.69
	CG	-0.003	0.976	0.993	0.580	37	2.70
1000	SD	-0.008	0.977	0.992	0.586	128	2.79
	CG	-0.008	0.977	0.992	0.586	27	2.79
10000	SD	-0.009	0.977	0.992	0.587	128	2.80
	CG	-0.009	0.978	0.992	0.586	29	2.79
∞	SD	-0.009	0.978	0.992	0.587	128	2.80
	CG	-0.009	0.977	0.992	0.588	27	2.80

Table 2: Volume deviation regression coefficients and convergence w.r.t. k.

k	Method	A	B	R^2	$RMSE$	Iters.	$\sum_a(v_a-V_a)^2$
100	SD	0.060	1.000	1.000	0.278	126	0.0105
	CG	0.059	1.000	1.000	0.306	37	0.0126
1000	SD	0.051	1.000	1.000	0.267	128	0.0096
	CG	0.028	1.000	1.000	0.285	27	0.0109
10000	SD	0.050	1.000	1.000	0.266	128	0.0096
	CG	0.038	1.000	1.000	0.288	29	0.0112
∞	SD	0.050	1.000	1.000	0.267	128	0.0096
	CG	0.028	1.000	1.000	0.262	27	0.0092

The previous tables show that both iterative methods yield almost the same results, specially for large values of k. Also, those tables show a convergent behavior of the method when $k\to\infty$. To further illustrate this behavior, we show in Figures 2 and 3, the demand deviations and a comparison of volumes for $k = 1000$ and $k = \infty$, respectively. In the demand matrix scatter plots (left) each red point has coordinates (g_{pq}, g^L_{pq}), with L the last iteration, *i.e.* $L = 128$ for SD and $L = 27$ in the case of CG. Similarly, for the volume scatter plots (right) each red point has coordinates $(V_a, v(\mathbf{g}^L)_a)$. These figures also corroborate that the results obtained with the SD algorithm are quite similar to those obtained with the CG algorithm. The

only significant difference is that the steepest descent algorithm requires about 4.5 more iterations than the conjugate gradient algorithm to achieve the same accuracy.
Finally, in Figure 4 we show how the objective function, for the model of Spiess (1), decreases at each iteration with both algorithms. We only show the first 27 iterations, since this is the number of iterations needed by the CG algorithm to converge to the minimum with the given accuracy. This figure clearly shows the better ability of the CG algorithm to solve the quadratic programming model.

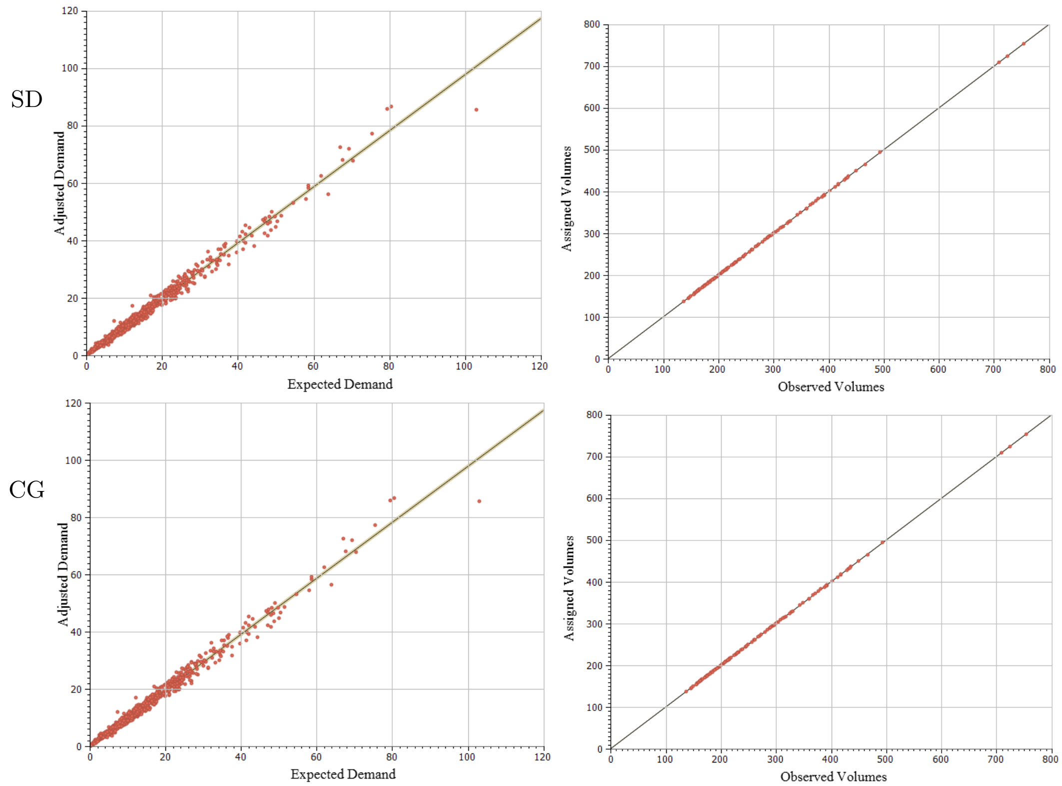

Figure 2: Demand deviations (left) and flow comparison (right) with CG and $k = 1000$.

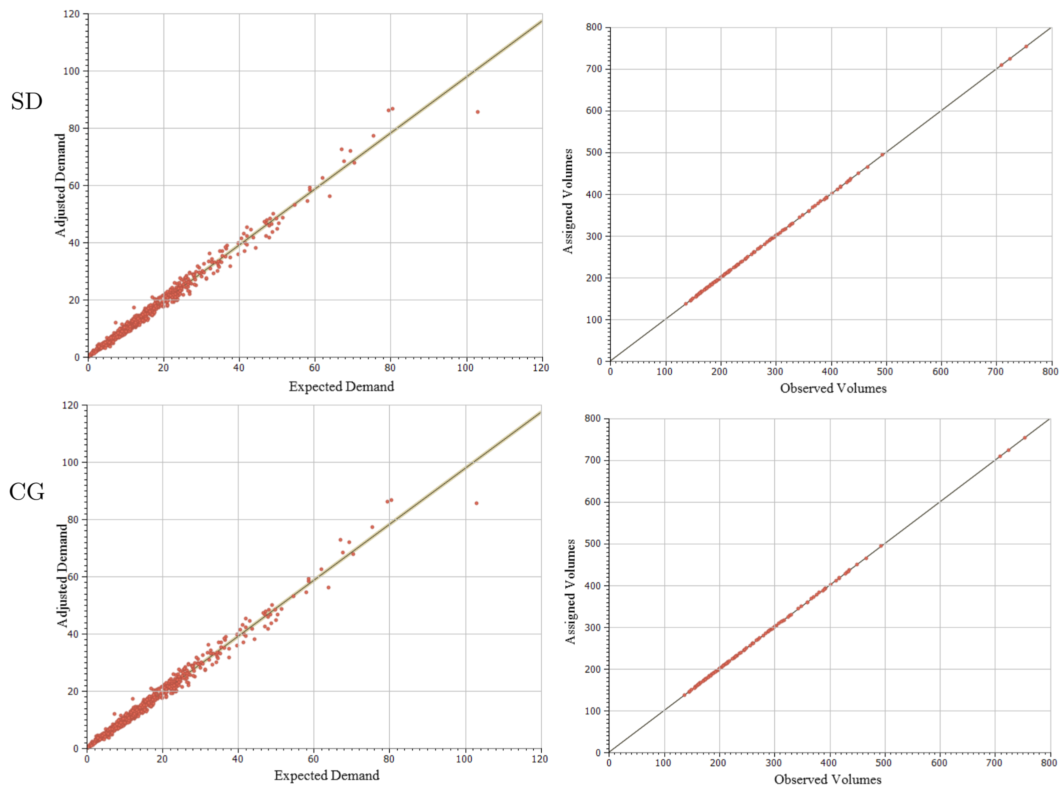

Figure 3: Demand deviations (left) and flow comparison (right), obtained with CG. Model of Spiess.

5. CONCLUSIONS

In this work we studied an O–D matrix demand adjustment model within the context of public transportation systems. This model is based on available volume data on some known links in the network; it considers the difference between observed and assigned volumes as a constraint, and incorporates those differences to the objective function as penalized quadratic terms, see (7)–(8). To solve the resultant optimization problem we introduced a multiplicative conjugate gradient method. The performance of this algorithm was compared with the method of steepest descent of Spiess [15]. Both methods yielded very similar solutions, but with the advantage that the conjugate gradient algorithm does much less iterations to get the same accuracy.
It can be observed that the CG algorithm does more operations at each iteration than the SD algorithm, mainly due to the calculation of β_l in (21). However, this is compensated by the fact that the CG algorithm does much less iterations than the SD algorithm to reach the same accuracy. Furthermore, it may be possible to find a good preconditioner for the CG algorithm to reduce the number of iterations even more. This a topic of future research.
There are additional issues that deserve further research, like the convergence properties of the CG algorithm. Also, it will be interesting to investigate the possibility of generalize the penalized model (4)–(6), allowing

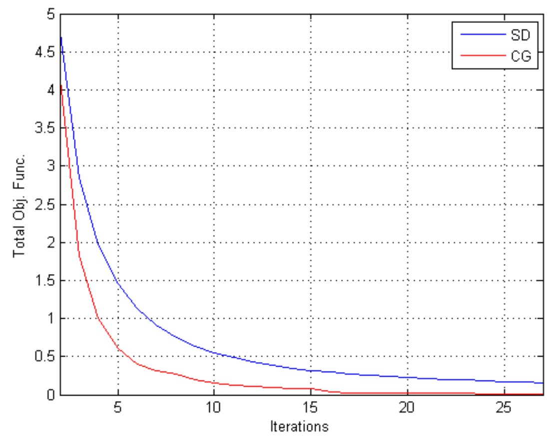

Figure 4: Objective function vs iteration for SD and GC algorithms. Model of Spiess (1).

different penalization parameters for different links. An important pending task, is to test the reliability of our approach for large scale problems, and how it compares with more recent models and techniques, like bi level programming techniques, but in the context of transit assignment. Preliminary numerical experiments with the transit network that represents the metropolitan area of Mexico City and surroundings are promising. This is an ongoing research.

Acknowledgements. We want to thank professor Michael Florian and his assistant, Yolanda Noriega, for their hospitality at CIRRELT while the second author was doing an internship at the University of Montreal to get training in the use of EMME/4 software. Also, we would like to acknowledge the financial support provided by CONACYT and, of course, the support and help from the graduate program of the Department o Mathematics at UAM-Iztapalapa to make possible this academic project.

RECEIVED: SEPTEMBER, 2014.
REVISED: MARCH, 2015.

REFERENCES

[1] BERA, S. and RAO, K. V. (2011): Estimation of origin–destination matrix from traffic counts: the state of the art. **Trasporti Europei**, . 49, 2–23.

[2] BIERLAIRE, M. (1995): Mathematical Models for Transportation Demand Analysis. **Ph.D. Thesis**, Facultés Universitaires Notre–Dame de la Paix de Namur, Faculte des Sciences, Départment de Mathématique.

[3] CASCETTA, E. and NGUYEN, S. (1988): A unified framework for estimating or updating origin/destination matrices from traffic counts. **Transpn. Res B**, 22–6,. 437–455.

[4] CHÁVEZ, M. V. (2013): Modelos matemáticos para análisis de demanda en transporte. **Master's thesis**, Department of Mathematics, Universidad Autónoma Metropolitana-Iataplapa, Mexico, D.F.

[5] CODINA, E., GARCIA, R. and MARIN, A. (2006): New algorithmic alternatives for the O–D matrix adjustment problem on traffic networks. **European Joournal of Operation Research**, 175, 1484–1500.

[6] DRAPER, N. R. and SMITH, H. (1998): **Applied Regression Analysis**. John Wiley and Sons, 3rd edition, N. York.

[7] FERNÁNDEZ, A. G. (2013): Modelos Matemáticos de Asignación de Tránsito. Aplicación a la Red Metropolitana del Valle de México y sus Efectos en el STC−Metro. **Master's thesis**, Departamento de Matemáticas, Universidad Autónoma Metropolitana−Iztapalapa.

[8] HESTENES, M. R. and STIEFEL, E. (1952): Methods of Conjugate Gradients for Solving Linear Systems. **Journal of Research of the National Bureau of Standards**, 49, 6.

[9] INRO, The evolutionof transport planning. Available at http://www.inrosoftware.com/en/products/emme/index.php. **Consulted** 15-03, 2015.

[10] JUÁREZ L. H., FERNÁNDEZ, A. G., DELGADO, CHÁVEZ, M. V. and OMAÑA, E. (2013): Asignación de Tránsito en la Red Metropolitana del Valle de México y su Impacto en el STC–Metro. **Contactos, Revista de Educación en Ciencias e Ingeniería**, 3, 84–95.

[11] LUNDGREN, J. and PETERSON A. (2008): A heuristic for the bilevel origin–destination–matrix estimation problem. **Transportation Research Part B: Methodological**, vol. 42, issue 4, pp. 339–354.

[12] NOCEDAL, J. and WRIGHT S. (2006): **Numerical optimization**. Springer Series in Operations Research and Financial Engineering, 2nd ed. Berlin

[13] NORIEGA, Y. and FLORIAN, M. (2009): Some Enhancements of the Gradient Method for O–D Matrix Adjustment. **CIRRELT–2009–04**, 1–23.

[14] SPIESS, H. (1987): A maximum–likelihood model for estimating origen–destination matrices, **Transpn. Res. B** 21 5, 395–412.

[15] SPIESS, H. (1990): A Gradient Approach for the O-D Matrix Adjustment Problem. Publication 693, **Centre for Reasearch on Transportation**, University of Montreal, Canada.

[16] SPIESS, H. and FLORIAN, M. (1989): Optimal strategies: A new assignment model for transit networks **Transpn. Res. B** 23, pp. 83–102.

[17] VAN UYLEN, H. J. and WILLUMSEN, L. G. (1980): The Most Likely Trip Matrix Estimated from Traffic Counts. **Transpn. Res. B** 14, 281–293.