New Solution Techniques for the Traffic Matrix Estimation Problem*

Zhou Xiang, Susana Scheimberg de Makler, Edmundo de Souza e Silva

[1]COPPE/PESC- Federal University of Rio de Janeiro
P.O Box 68511. Zip 21941-972–Rio de Janeiro – RJ – Brazil

{zhouxiang,edmundo}@land.ufrj.br, susana@cos.ufrj.br

Abstract. *We propose an approach which formulates the traffic matrix estimation problem as a non-negativity constrained optimization problem, and then a projection method is used to solve it. We conduct experiments both on synthetic and real measurement data obtained from the Abilene network. The results indicate that the estimated traffic matrices are more accurately estimated when our approach is used than when the tomogravity method is employed. We also develop a novel approach for estimating traffic matrices when optimal multi-path routing is employed. We formally formulate the problem as a bilevel programming problem. Then a genetic algorithm is used to solve it.*

1. Introduction

A traffic matrix (TM) reflects the volume of traffic that flows between any source-destination pair of nodes in a communication network. The nodes could be links, routers, Points-of-Presence (PoPs) in the Internet. The choice of node type affects the granularity and type of the traffic matrix. For example, in a router-router traffic matrix, the traffic that flows in and out of a given router includes all of the clients and peers attached to that router. Obtaining traffic matrices is a very important problem to the network operators because many traffic engineering and network management tasks need the information provided by traffic matrices, in order to improve the performance and efficiency of the network. These tasks include logical topology design, capacity planning, routing protocol configuration, load balancing and network reliability analysis.

The problem of traffic matrix estimation depends on the routing mechanism in the network. The most commonly used intra-domain Internet routing protocols today are the shortest path first (SPF) protocols such as Open Shortest Path First (OSPF) and Intermediate System-Intermediate System (IS-IS). In these protocols, each link is associated with a weight and the length of a path is defined as the sum of the weights of all the links on that path. The shortest path is that with minimum length. Traffic is routed along the shortest path to the destination. However, these traditional routing protocols that rely on a single path between a source-destination pair, may not efficiently utilize the network resources. To make optimal use of network resources and minimize delays, traffic between source-destination pairs may often have to be split and routed along multiple paths each carrying a fraction of the total flow. This routing mechanism is commonly called, optimal multi-path routing [Bertsekas and Gallager 1992].

In this work we tackle two traffic matrix estimation problems. The first (**Problem A**) proposes a solution for the case in which the network employs *shortest path routing*;

*Este trabalho é parcialmente financiado pelo CNPq e Faperj.

In the second (**Problem B**), a new technique is presented when the *optimal multi-path routing* mechanism is assumed.

All of the previous works that attempt to solve problem A can be roughly classified into two categories: direct and indirect methods. The direct methods, as the name indicates, directly measure traffic volumes between source and destination nodes. However, those approaches have not been fully explored in large IP networks because they still face challenging engineering obstacles such as the lack of proper measurement infrastructure, high communication and computational costs of transferring large amounts of information. Indeed, most of the previous approaches attempt to estimate traffic matrices from other available data, typically link load measurements and routing configurations. Roughly, the estimation problem is based on solving a system of linear equations $\mathbf{Ax} = \mathbf{y}$, where $\mathbf{x}$ is the traffic matrix vectorized as a column vector, $\mathbf{y}$ is a vector of link counts, and $\mathbf{A}$ is a matrix reflecting the routing, where element A_{ij} is equal to 1 if the OD pair j traverses link i, or zero otherwise. Moreover, the elements A_{ij} may be positive numbers between 0 and 1 if traffic splitting is supported. The link counts $\mathbf{y}$ can be readily obtained through standard Simple Network Management Protocol (SNMP) measurements. The routing matrix $\mathbf{A}$ can be obtained by gathering IGP link weights and networks topology information and then computing the shortest paths between all OD pairs, in case shortest path routing is used. The problem at hand is to compute the traffic matrix $\mathbf{x}$ given the link counts $\mathbf{y}$ and the routing matrix $\mathbf{A}$. However, this is not a straightforward task because the linear system is highly under-determined or ill-posed since the number of OD pairs is much larger than the number of links in almost all networks. This means that there is and infinite number of feasible solutions for $\mathbf{x}$.

In order to deal with the ill-posed problem, some previous works [Vardi 1996, Cao et al. 2000] are able to estimate the TM after assuming that the traffic flows follow a Poisson or Gaussian distribution. Typically, after assuming Poisson or Gaussian models for OD traffic pairs, additional constraints are introduced to obtain a solution. TM estimation is performed with both constraints on the first and second order moments. Vardi [Vardi 1996] proposes a Poisson distribution model for traffic demands and use the maximum likelihood estimation technique to estimate the traffic demands. Cao et al [Cao et al. 2000] followed Vardi's approach, but use a Gaussian model with the assumption of a power-law relationship between the mean and variance of the traffic demands. A comparative study of these methods in [Medina et al. 2003] shows that they are highly dependent on an initial starting point, called *prior*.

Unfortunately the statistical methods cannot guarantee that all of the estimated TM elements A_{ij} will have positive values. To cope with the problem, an iterative proportional fitting algorithm (IPF) is applied to perform final adjustments on the estimated traffic matrix. The IPF algorithm has been used extensively in the context of contingency tables [Deming and Stephan 1940]. The idea is to express the linear constraints given by Equation $\mathbf{Ax} = \mathbf{y}$, using a contingency table composed of the estimated traffic matrix and an extra value for each row and each column, corresponding to the row and column sums, respectively. The IPF algorithm proceeds to adjust the values of the estimated traffic matrix such that the row and column sum errors are minimized. The convergence of the IPF algorithm is proven [Deming and Stephan 1940].

These statistical inference methods do not work well in estimating the TM because they are very sensitive to the assumptions (Poisson, Gaussian distributions), which do not hold in real network traffic data [Medina et al. 2003]. This limitation has led to other models that employ extra information and different assumptions. The most prominent approach is the tomogravity model [Zhang et al. 2003]. The key idea behind the tomogravity method is to find the best among all of feasible traffic matrix solutions based on some criterion. The criterion can be described by an objective function, and the linear equations $\mathbf{Ax} = \mathbf{y}$ are the constraints in an optimization problem. A gravity model is used to obtain the prior traffic matrix, then the least-square solution which minimize the Euclidean distance to the prior traffic matrix is obtained.

The least-square solutions may result in negative values which, of course, have no physical meaning. In order to achieve a fast and effective solution, the Iterative Proportional Fitting is also used to ensure non-negativity instead of formulating the problem as a constrained optimization problem. As described in [Zhang et al. 2003], the initial starting point is not as complex as that in [Cao et al. 2000]. They just use the least square solution for the traffic matrix, with zero replacing the negative elements of the matrix. It is also reported in [Zhang et al. 2003] that it only takes a few iterations to reduce errors in the constraint equations to the point at which they are negligible in practice.

Since both the least square optimization and IPF techniques are based on the system $\mathbf{Ax} = \mathbf{y}$ and the measured link traffic always contains errors, the following question may be asked:

> *Instead of using the tomogravity method (least square + IPF), can we formulate the estimation problem as a non-negative constrained optimization problem? How?*

Contributions for Problem A

We propose an approach which formulates the traffic matrix estimation problem as a non-negative constrained optimization problem and a projection method is then used to solve it. We conduct experiments both on synthetic and real measurement data obtained from the Abilene network. We show that our approach is capable of estimating traffic matrices more accurately than the tomogravity method.

Contributions for Problem B

Existing traffic matrix estimation techniques are not applicable to the optimal multi-path routing networks. This is true since the relationship between the original traffic matrix $\mathbf{x}$ and the link data $\mathbf{y}$ changes from linear to nonlinear. This means that routing is no longer static, but dynamic and depends on the original traffic demands. In other words, the routing matrix $\mathbf{A}$ and the link data $\mathbf{y}$ are nonlinear functions of the original traffic matrix $\mathbf{x}$.

To the best of our knowledge, this paper is the first work also concerned with the traffic matrix estimation problem in optimal multi-path routing networks. We formally formulate the traffic matrix estimation problem in an optimal multi-path routing network as a bilevel programming problem. In the bilevel programming problem, the upper level problem is responsible for the traffic matrix estimation and the the lower level problem represents the optimal routing problem. A genetic algorithm is used to solve the bilevel programming problem.

The remaining of this paper is organized as follows. In Section 2, we present the background material. Section 3 presents our methods for dealing with problems A and B. The non-negativity constrained traffic estimation problem is proposed (problem A) and a projection method used to obtain the final solution. For problem B, a bilevel programming model is proposed and a genetic algorithm used to solve it. In Section 4, we validate the results for problem A, based both on synthetic and real network traffic matrices. For Problem B, we show numerical results based on synthetic network traffic. Our conclusions are presented in Section 5.

2. Background

2.1. Tomogravity Method

In [Zhang et al. 2003], the authors developed a method for estimating traffic matrices that starts by building a prior TM using a gravity model. The key assumption underlying the gravity model is that the traffic entering the network at any given node exits the network at a particular egress node proportionally to the total traffic exiting at that egress. Let $x(i,*)$ denote the total traffic entering an ingress node i. Let $x(*,j)$ denote the total traffic departing the network from node j. The gravity model postulates that,

$$x(i,j) = x(i,*)\frac{x(*.k)}{\sum_k x(*,k)} \tag{1}$$

This implies that the total amount of data node i sends to node j is proportional to the amount of traffic departing the network at j relative to the total amount of traffic departing the entire network. The authors call this assumption the *simple gravity model*.

This simple gravity model essentially assumes complete independence between sources and destinations. However, as pointed out in [Zhang et al. 2003], traffic transit between peering networks behaves very differently. This has led to the generalized gravity model, where traffic between peers is forced to zero.

The gravity model is used inside a least square optimization problem. They formulate the optimization problem as

$$\begin{aligned} \min_{\mathbf{x}} \quad & (\mathbf{x}-\mathbf{x}_g)^T(\mathbf{x}-\mathbf{x}_g) \\ \text{subject to:} \quad & (\mathbf{A}\mathbf{x}-\mathbf{y})^T(\mathbf{A}\mathbf{x}-\mathbf{y}) \text{ is minimized} \end{aligned} \tag{2}$$

The idea is that among all the traffic matrices that satisfy the link constraints, the method chooses one closest to the gravity model $\mathbf{x}_g$. The overall method works as a two step process within each time interval t: first an initial estimate $\mathbf{x}_g$ is calculated using (1) and then, the optimization problem in (2) is solved.

2.2. The Formulation of Optimal Routing Problem

The objective of optimal multi-path routing is to distribute traffic demand on its all possible paths such that the total network "cost" is minimized. The cost may be average network delay, etc. The network is modeled by a graph $\mathcal{G} = (\mathcal{L}, \mathcal{M})$ which $\mathcal{L}$ is the set of L nodes and $\mathcal{M}$ is the set of M directed links. Each link $a \in \mathcal{M}$ has a known capacity C_a

and is assigned a cost function l_a based on its capacity and the amount of traffic passing through it. The most popular link cost function is the link delay function which is derived from the M/M/1 queueing model [Kleinrock 1964].

$$l_a(C_a, y_a) = \frac{y_a}{C_a - y_a} \quad (3)$$

where y_a is the traffic at link a.

The formulation of the optimal multi-path routing problem is

$$\begin{aligned} \min_{\mathbf{y}, x_{ij}} \quad & \textstyle\sum_{a \in \mathbf{M}} \frac{y_a}{C_a - y_a} \\ \text{subject to} \quad & \textstyle\sum_j x_{ij} = x_i, i \in \mathbf{N} \\ & \textstyle\sum_{ij} x_{ij} \delta_a^{ij} = y_a \\ & x_{ij} \geq 0 \\ & 0 \leq y_a < C_a \end{aligned} \quad (4)$$

where x_i is the traffic of OD pair i (recall that we reduce the TM to a vector) and x_{ij} is the traffic of the j path of OD pair i. The indicator δ_a^{ij} is 1 if the j path of OD pair i pass through link a, and zero otherwise.

3. Solutions

3.1. Solutions for Problem A: Non-negativity Constrained Traffic Matrix Estimation Problem

3.1.1. Uncertainty in Gravity Model and SNMP Data

Since the initial estimate in gravity model, $\mathbf{x}_g$, is calculated from (1), it is very unlikely that the initial estimate satisfies the equation $\mathbf{Ax} = \mathbf{y}$. As pointed out in [Roughan et al. 2003], SNMP link data $\mathbf{y}$ has many limitations. Missing data (SNMP uses unreliable UDP transport), incorrect data (through poor router vendor implementations), and a coarse sampling interval (five minutes is typical) are examples of these limitation. Therefore, the SNMP link data collected is not error free, and its reliability may vary.

In the gravity model presented in the last section, errors in SNMP data are not considered. The maximum belief is placed on it. But one possible consequence of errors in SNMP data is that the equation $\mathbf{Ax} = \mathbf{y}$ becomes inconsistent. Even if those errors are small, it could bring large errors to the estimated traffic matrix.

One way to handle this inconsistency is formulate the following weighted least square problem

$$\min \quad f(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_g)^T(\mathbf{x} - \mathbf{x}_g) + (\mathbf{Ax} - \hat{\mathbf{y}})^T \mathbf{W}^{-1}(\mathbf{Ax} - \hat{\mathbf{y}}) \quad (5)$$

where $\hat{\mathbf{y}}$ is measured SNMP link data, $\mathbf{W}^{-1}$ is a weight matrix that is positive semi-definite. In this present paper, we consider $\mathbf{W}^{-1}$ as an identity matrix.

Since the objective function (5) is convex, we have that a point $\mathbf{x}$ is a solution if and only if $\nabla f(\mathbf{x}) = 0$ holds. By the Matrix Inversion Lemma [Tylavsky and Sohie 1986], the solution can be expressed as,

$$\mathbf{x} = \mathbf{x}_g + \mathbf{A}^T(\mathbf{AA}^T + \mathbf{W})^{-1}(\hat{\mathbf{y}} - \mathbf{Ax}_g) \quad (6)$$

3.1.2. Handling the non-negativity

The solution of (6) doesn't ensure the non-negativity of the elements of the traffic matrix. In [Zhang et al. 2003], the Iterative Proportional Fitting (IPF), as suggested in [Cao et al. 2000], is used to ensure non-negativity. The estimated traffic matrix, solution of (2), with zero replacing the negative elements of the matrix is used as an initial starting point of the iterative fitting process. It is reported in [Zhang et al. 2003] that the IPF proceeds by successively refining the estimate and it only takes a few iterations to reduce errors in the constraint equations to the point at which they are negligible in practice. Note that the IPF is also based on the linear system $\mathbf{Ax} = \mathbf{y}$. Thus, the errors in y also have great influence on the accuracy of estimated traffic matrix, when the IPF procedure is used.

Since (5) is a weighted least square optimization problem, the direct way to ensure non-negativity is to add the non-negativity constraint into the problem. We have,

$$\begin{array}{rc} \min & f(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_g)^T(\mathbf{x}-\mathbf{x}_g) + (\mathbf{Ax}-\hat{\mathbf{y}})^T\mathbf{W}^{-1}(\mathbf{Ax}-\hat{\mathbf{y}}) \\ \textbf{subject to:} & \mathbf{x} \geq 0 \end{array} \tag{7}$$

Note that the solution of (6) is not the solution of the above problem, since the non-negativity isn't ensured in it. Therefore, the gradient projection method is used to solve this issue.

3.1.3. Gradient Projection Method

Projection onto non-negativity sets

Let $\mathrm{C} \subset \mathrm{R}^n$ be a closed, convex set. Given $\mathbf{x} \in \mathrm{R}^n$, the Euclidean projection of $\mathbf{x}$ onto C is $P_{\mathrm{C}}(\mathbf{x}) = \arg\min_{\mathbf{v}\in\mathbf{C}} \|\mathbf{x}-\mathbf{v}\|$. In other words, $P_{\mathrm{C}}(\mathbf{x})$ is the closest point in **C** to **x** related to the euclidean norm. In our case, **C** is given by, $\mathbf{C} = \{x \in \mathbf{R}^n | x_i \geq 0, i = 1, 2, \ldots, n\}$ Hence, the i-component of $P_{\mathrm{C}}(\mathbf{x})$ is, $P_{\mathrm{C}}(\mathbf{x})_i = \max(x_i, 0) \quad i = 1, 2, \ldots, n$.

Gradient Projection Algorithm

The idea of the method is that, at a current point, $\mathbf{x} = \mathbf{x}_k$, a steepest descent direction for the unconstrained problem, $\mathbf{p} = -\nabla f(\mathbf{x})$ is considered. A search along the line through $\mathbf{x}_k$ in the direction $-\mathbf{p}$, a new point $\mathbf{x} = \mathbf{x} + \tau_p\mathbf{p}$ is found. Then this new point is projected onto the nonnegative sets to ensure its feasibility. The algorithm continues until the a final point has been found. This can be viewed as a generalization of the steepest descent method for unconstrained optimization.

In Algorithm 1 below, instead of a one-dimension search that many optimization algorithms use, the stepsize τ_k in (9), can be expressed as

$$\tau_k = -\frac{\mathbf{p}_k^T(\mathbf{x}_k - \mathbf{x}_g) + (\mathbf{Ap}_k)^T\mathbf{W}^{-1}(\mathbf{Ax}_k - \hat{\mathbf{y}})}{\mathbf{p}_k^T\mathbf{p}_k + (\mathbf{Ap}_k)^T\mathbf{W}^{-1}(\mathbf{Ap}_k)} \tag{12}$$

since $\mathbf{p}_k$ is a descent direction and $f(\mathbf{x}_k + \tau\mathbf{p}_k)$ is a strictly convex function.

Algorithm 1 Gradient Projection Method

Step 1 Initialization.$k = 0; x_0 \geq 0$(nonnegative initial solution);

Step 2 Gradient Projection Iterations

$$\mathbf{p}_k = -\nabla f(\mathbf{x}_k); \tag{8}$$

$$\tau_k = \arg\min_{\tau>0} f(\mathbf{x}_k + \tau \mathbf{p}_k); \tag{9}$$

$$\mathbf{x}_{k+1} = P(\mathbf{x}_k + \tau_k \mathbf{p}_k); \tag{10}$$

$$k = k + 1; \tag{11}$$

Step 3 If $\mathbf{p}_k = 0$ or $\max_i \mid x^i_{k+1} - x^i_k \mid \leq \epsilon$, the algorithm is terminated. Otherwise, return to step 2.

3.2. Solutions for Problem B

We begin the section by presenting the proposed estimation model which is illustrated in Figure 1. The model is divided in two levels. In the upper level, we use the least square method to perform the traffic matrix estimation. In the lower level, we model the optimal routing by a nonlinear programming. Both problems are unified by formulating a bilevel programming model. Then, we present a genetic algorithm to solve it. The goal

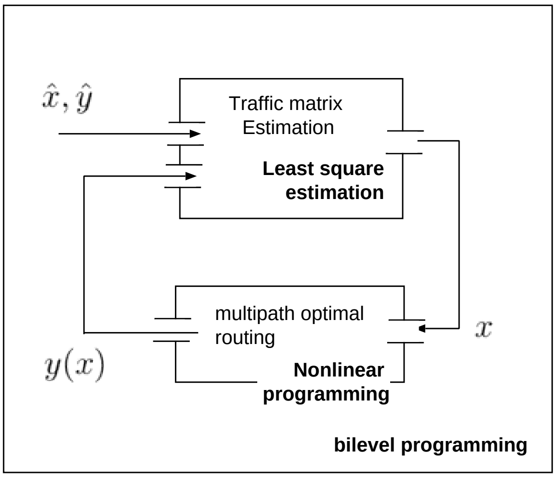


Figure 1. The proposed traffic matrix estimation model

is to estimate the traffic matrix in the optimal multi-path routing based network given the measured link traffic $\hat{y}$.

In our model, we want to find the vector **x** satisfying the link traffic by minimizing the distance between measured link traffic and real link traffic and the distance between the prior traffic matrix and the real traffic matrix. The objective function of the upper level is

$$(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x}) + (\hat{\mathbf{y}} - \mathbf{y})^T \mathbf{W}^{-1}(\hat{\mathbf{y}} - \mathbf{y}) \tag{13}$$

where $\mathbf{W}^{-1}$ is a given positive semidefinite matrix.

By combining the upper and the lower level problems, the formulation of problem B is,

$$\begin{aligned} \min_{\mathbf{x}} \quad & (\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x}) + (\hat{\mathbf{y}} - \mathbf{y})^T\mathbf{W}^{-1}(\hat{\mathbf{y}} - \mathbf{y}) \\ \text{subject to} \quad & \mathbf{x} \geq 0 \end{aligned}$$

where $\mathbf{y}$ is the solution of,

$$\begin{aligned} \min_{\mathbf{y}, x_{ij}} \quad & \sum_{a \in \mathbf{M}} l_a(C_a, y_a) \\ \text{subject to} \quad & \sum_j x_{ij} = x_i, i \in \mathbf{N} \\ & \sum_{ij} x_{ij}\delta_a^{ij} = y_a \\ & x_{ij} \geq 0 \\ & 0 \leq y_a < C_a \end{aligned} \tag{14}$$

where $l_a(C_a, y_a) = \frac{y_a}{C_a - y_a}$

Genetic algorithm

It is important to observe that the lower problem has only one solution because the objective function is strictly convex on the feasible set. So, problem B is well defined. A bilevel problem is nonconvex since its feasible set, given by the solutions of the lower problem, is nonconvex. Nonconvexity implies the existence of local solutions and so it can be difficult to find the global optimal solution.

Some algorithms have been developed based on classical optimization approaches such as the branch-and-bound technique, the variable elimination method based on Kuhn Tucker approach and algorithms based on the penalty function approach [Vicente and Calamai 1994]. It is reported in [R. Mathieu and Anandalingam 1994] that most of the traditional approaches are problem-dependent relying on knowledge of the search space and are not sufficiently robust. Hall et al. [Hall et al. 1980] propose a heuristic iterative algorithm to find a solution of the bilevel problem applied in the evolution of traffic management schemes. However, the iterative algorithm may not converge if the upper-level and lower-level problems are decoupled.

In this paper, a genetic algorithm is proposed to solve the bilevel programming problem. Genetic algorithms (GAs) are inspired by the theory of evolution. Initially, a population of chromosomes, each of which are potential solutions, are generated. These chromosomes are altered or modified using the genetic operators (crossover and mutation) in order to create a new generation. This evolutionary process is repeated a predetermined number of times or until the solution is satisfied.

The motivation of using genetic algorithms (GAs) is due to its simplicity, less problem restrictions, globality and implicit parallelism. In addition, it is quite robust in dealing with non-convex as well as nondifferential problems. So, it can handle the problem of nonconvexity resulting from the bilevel programming problem.

The algorithm starts with a population $\{x_1^{(0)}, \ldots, x_N^{(0)}\}$. In the iterative procedure, given $x_i^{(k)}(k = 0, 1, \ldots)$, $y_i^{(k)}$ is calculated by solving the lower optimal routing problem.

The upper objective function is evaluated at each pair $(x_i^{(k)}, y_i^{(k)})$ obtaining a selection of the population which is used to generate a new population $x_i^{(k+1)}$. The proposed genetic algorithm is given as follows,

Algorithm 2 The Genetic Algorithm

Step 1. Randomly generate an initial N population of $\mathbf{x}_i^{(k)}$. $i \in N$. Set k = 1.
Step 2. Solve the lower-level optimal multi-path routing optimization problem problem with each fixed $\mathbf{x}_i^{(k)}$ and find its corresponding lower-level variable $\mathbf{y}_i^{(k)}$.
Step 3. Return to upper-level problem and calculate the fitness function for each individual $\mathbf{x}_i^{(k)}$ with $\mathbf{y}_i^{(k)}$.
Step 4. Tournament selection is used to select fit individuals $\mathbf{x}_i^{(k)}$. Perform the crossover operation with probability P_c. Perform the mutation operation with probability P_m. Use elitism strategy: the best solution is copied to the population in the next generation. This is done to prevent losing the best solution found so far. Then a new population $\mathbf{x}^{(k+1)}$ is generated.
Step 5. If k = maximum number of generations, the algorithm terminates. Else, set $k = k + 1$ and return to Step 2.

4. Evaluation Methodology

In what follows we evaluate our traffic matrix estimation methods described in the previous section.

4.1. Evaluation Methodology for Problem A

4.1.1. Synthetic Data Experiments

We consider a small 4 node topology, shown in Figure 2. This topology is also used in [Medina et al. 2003]. All node pairs and their traffic can be enumerated, which is useful for illustrating how the methods behaves.

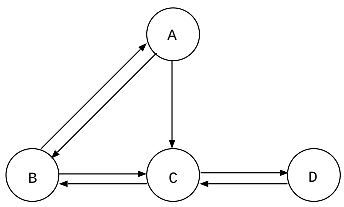


Figure 2. Topology of a four-node network

Synthetic data is very useful in order to evaluate the performance of traffic matrix estimation techniques. This is true since we can study the behavior of the technique under evaluation with respect to complete traffic matrices rather than only partial matrices obtained from measured traffic. By performing synthetic-data experiments we can better assess the errors yielded by the evaluated techniques. In general, studies and comparative evaluations in the literature rely on synthetically generated traffic matrices based on strong assumptions regarding the underlying distributions of the traffic demands between origin-destination (OD) pair [Medina et al. 2003, Vardi 1996, Cao et al. 2000]. For example, a common approach assumes that OD demands are distributed according to Gaussian or Poisson distributions. In our synthetic-data experiment, we consider the synthetically generated Poisson TMs which is used in [Medina et al. 2003].

The synthetic TM is obtained by generating the Poisson parameters λ_i for each OD pair i uniformly from an interval $[100, 500]$. Then, Poisson traffic demands are generated according to $x_i \sim Poisson(\lambda_i)$. We route this TM onto the 4-node topology to obtain synthetic link traffic. In order to evaluate the impact of SNMP measurement errors, we introduce an error vector ϵ. The error vector is formed by multiplying **y** with a noise vector, $\varepsilon = \mathbf{y} * N(0, \phi)$, where $*$ denotes the element by element vector multiplication, $N(0, \phi)$ is a vector with random entries drawn form a normal distribution with mean 0 and standard deviation ϕ. In summary, the synthetic measured link traffic are generated by adding white noise to **y**, that is $\hat{y} = y + \varepsilon$. Finally the prior traffic matrix is generated from the simple gravity model (1). Its mean relative error (MRE) is about 33%, and so it is **not** a good prior. Note that if the prior is good (close to the final solution) the evaluation process would not be appropriate.

In order to evaluate our methods, we need to know the accuracy of the estimated traffic matrices. First we consider the MRE metric, which is defined as,

$$MRE = \frac{1}{N_T} \sum_{i:x_i>T} \left| \frac{\hat{x}_i - x_i}{x_i} \right| . \tag{15}$$

Here, x_i is the true traffic matrix element and $\hat{x}_i$ is the corresponding estimate. N_T is the number of matrix elements that are greater than a threshold value T. In this experiment, since the TM is generated with Poisson distribution and the difference between its largest and smallest value is not big, we consider all traffic matrix elements in MRE evaluation.

Table 1 compares the estimated TM obtained by the tomogravity method with our approach. We observer that the tomogravity method estimate for the traffic between OD pair CB is a negative value. After IPF, this value is adjusted to approximately zero. Note that this leads to a very large error. With our approach, the estimate value is 53.51 which is considerably better than the tomogravity estimate. It can also be observed that, when the tomogravity method is used, the Mean Relative Error slightly decreases after IPF. In this example, our method just needs a single step, and does not need to engage an iterative process. Its MRE is smaller than that obtained by the tomogravity method.

4.1.2. Real-data Experiments

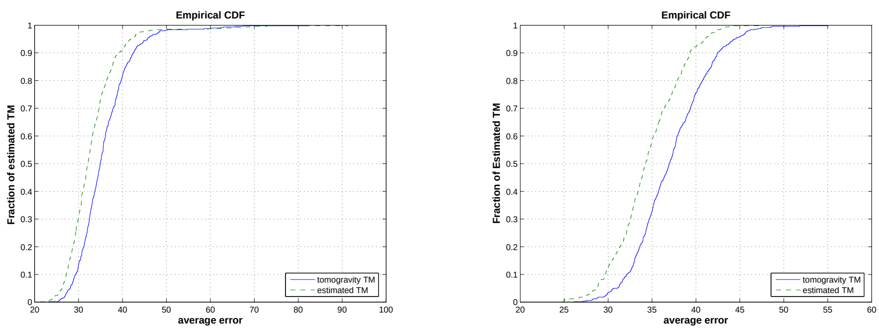
We use the 6 months of Abilene traffic matrices collected by Zhang [Zhang]. The routing matrix and the gravity model solution is also provided to the research community. We perform two experiments on five hundred 5-minutes traffic matrices time series. The beginning time of the first time series is March 1, 2004 and the second is July 31, 2004. The corresponding link traffic can be derived by calculating the product of the traffic matrix and the provided routing matrix A. In order to validate our approach, we simulate the measurement noise described in Section 5.1.

TM			Tomogravity		Our Method
OD Pair	Original TM	prior TM (gravity)	TM before IPF	TM after IPF	TM
AB	318	241.26	260.28	248.77	165.81
AC	289	344.40	221.17	228.91	192.42
AD	312	142.48	111.44	383.52	312.57
BA	294	256.79	538.94	292.91	223.33
BC	292	574.77	378.	258.95	238.51
BD	267	237.79	376.12	376.12	322.22
CA	305	350.82	165.18	147.95	224.70
CB	289	550.09	-48.50	0.81	53.51
CD	324	324.86	322.17	322.17	288.71
DA	283	157.52	308.52	298.68	246.45
DB	277	247.01	183.04	160.81	163.47
DC	291	352.60	396.69	428.76	275.10
	MRE	33.33%	29.84%	29.42%	26.88%

Table 1. Comparison with the tomogravity model

The most important traffic to be estimated is the largest OD pair traffic since its accuracy has great impact on traffic engineering tasks such as load balancing or failure analysis. In this experiment, we choose the value of T (defined in (15)) as 5.000.000 so that the considered OD flows comprise approximately 85% of total traffic.

Figures 3(a) and 3(b) show the empirical cumulative distribution function of the estimated traffic matrix for the two methods we compare. For instance, from Figure 3(a), we can see that only approximately 10% of the traffic matrix estimated by the Tomogravity approach has errors which are smaller that 30%, while 30% of the TM estimated by our approach has errors smaller than 30%. Figure 3(b) shows similar behavior. If we are concerned with errors smaller than 35%, only 30% of the estimated TM by the Tomogravity technique are under this threshold in comparison with 60% of the TM estimated by our method. From the figures we can conclude that our method has an improved performance in comparison with the tomogravity method. This shows the advantages of incorporating $\mathbf{Ax} - \hat{\mathbf{y}}$ in the objective function.



(a) March 1,2004 (b) July 31, 2004

Figure 3. The empirical cumulative distribution function of estimate TM

In what follows we evaluate the impact of different levels of noise in SNMP measurement on the traffic matrix estimation. In particular, the noise level ϕ is that used in $\varepsilon = \mathbf{y} * N(0, \phi)$. Figure 4 shows the comparison of the Mean Relative Error (MRE) with the tomogravity method under different noise levels for one specific traffic matrix. Since the SNMP measurement error is synthetically generated by random normal distribution with mean 0 and variance ϕ, it is reasonable that the MRE increases with increasing ϕ, as shown in the figure.

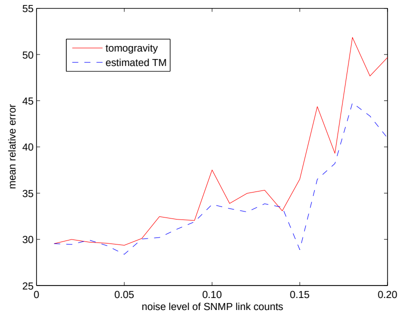
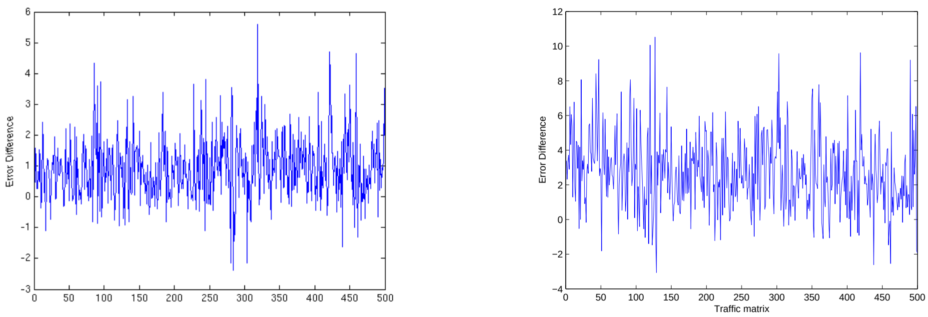


Figure 4. Comparison with tomogravity method with different noise levels

Figure 5(a) shows the difference between the MRE of the tomogravity method and our technique for five hundred traffic matrices, when the noise level is 0.05. We see that 411 (82.2%) values are positive. This means that, in most cases, our method has a smaller MRE. The mean value of this difference is 0.93. Therefore, when the noise level is 0.05, our method has a MRE which is about 0.93% smaller than that of the tomogravity method. Figure 5(b) shows the results when the noise level is 0.1. In this case 453 (90.6%) values are positive, and consequently, our method has a smaller MRE. The the mean value of this difference is 2.69. This means that, when the noise level is 0.1, our method has a MRE which is about 2.69% smaller than that of the tomogravity technique.



(a) March 1,2004 and noise level = 0.05 (b) July 31, 2004 and noise level = 0.1

Figure 5. Comparison with the tomogravity method with different noise levels

Table 2 shows the mean value of MRE differences with different noise levels for five hundred traffic matrices. The table shows that the accuracy of our method increases with the level of noise, as expected.

4.2. Evaluation Methodology for Problem B

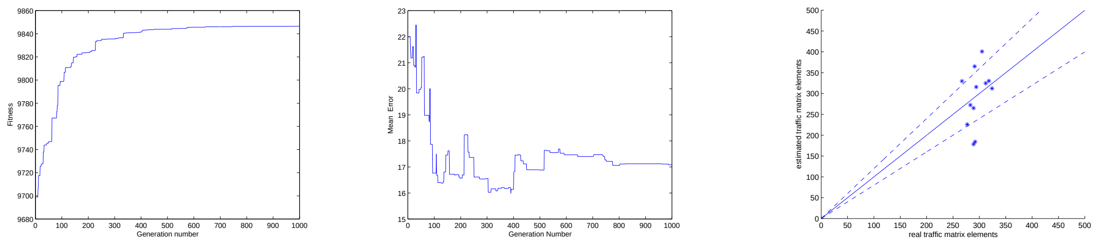
The topology we consider is the 4-node topology for Problem A, and is depicted in Figure 2. The routing mechanism is assumed to be the optimal multi-path routing in this case.

noise level	0.01	0.03	0.05	0.09	0.11	0.13	0.15
MRE difference	0.0592	0.3431	0.8665	1.5372	2,2174	2.9455	3.7584

Table 2. MRE difference with different noise levels

We generate synthetic traffic matrices from Poisson distribution and it is assumed that the capacity of each link is $2,500$ units. As described before, the prior TM is also generated using the gravity model. The link traffic is obtained by solving the optimal multi-path routing problem. We also simulate the noise in the link traffic, to obtain the final *observed* link traffic.

For this problem, the relationship of the objective function and the fitness function is defined as follows: $f(x) = C_{max} - O(x)$, where $f(x)$ is the fitness function, and $O(x)$ is the objective value of the upper-level problem. C_{max} is taken as $10,000$ in this paper.



(a) Fitness (b) Mean Error (c) Comparison to the 'Real' TM

Figure 6. Genetic Algorithm for Traffic Matrix Estimation

The convergence of the algorithm is shown in Figure 6(a). It can be seen that the algorithm converges quickly in the first 200 generations. Then it begins to converges slowly. After 1,000 generations, the fitness of the current optimal solution is $94,83$ which is approximate the fitness of the ideal optimal solution which is $10,000$. Therefore, we can take this near optimal solution as the final solution of the traffic matrix estimation problem.

Figure 6(b) shows the mean error of the solution in each generation. We can observe that the smaller upper-level objective function value achieved by the genetic algorithm does not necessarily imply a better estimate of the traffic matrix. This can be explained by the accuracy of the estimated TM that depends not only on the genetic algorithm itself but also on the upper-level objective function and the prior traffic matrix. Most importantly, the upper-level objective function (13), which minimizes the distance between the estimated TM and the prior TM, is an approximation to reach the real TM. Obviously, a smaller value of objective function (13) means that the estimated TM is closer to the prior TM, but may not be closer to the real TM.

The comparison of the "real" traffic matrix elements to the estimated traffic matrix elements is shown in Figure 6(c). The solid diagonal line shows equality, while the dashed lines show $\pm 20\%$ errors. We can see that, among 12 elements, the errors of 7 elements are smaller than 20%. The errors of the other 3 elements are close to 20%. We conclude then that the algorithm works well for TM Estimation.

5. Conclusions

In this paper we propose a new traffic matrix estimation method which formulates the traffic matrix estimation as a non-negativity constrained optimization problem. We conducted extensive experiments both on synthetic traffic data and real traffic data obtained from the Abilene network. We found that the traffic matrix is more accurately estimated with our method in comparison with the tomogravity method. Moreover, the relative accuracy increases with increasing noise levels.

We also develop a novel approach to estimate traffic matrices when the optimal multi-path routing mechanism is used. We formulate this problem as a bilevel programming problem. The upper level model represents the traffic matrix estimation problem and the lower level model represents the optimal multi-path routing problem. A genetic algorithm is used for the solution.

References

Bertsekas, D. and Gallager, R. (1992). *Data Networks*. Prentice Hall.

Cao, J., Davis, D., VanderWeil, S., and Yu, B. (2000). Time-varying network tomography: Router link data. *Journal of the American Statistical Association*, 95:1063–1075.

Deming, W. and Stephan, F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics*, pages 427–444.

Hall, M., Vliet, V. D., and Willumsen, L. (1980). Saturn: A simulation assignment model for the evaluation of traffic management schemes. *Traffic Eng. Control*, 21:168–176.

Kleinrock, L. (1964). *Communication Nets: Stochastic Message Flow and Delay*. McGraw-Hill, New York.

Medina, A., Taft, N., Salamatian, K., Bhattacharyya, S., and Diot, C. (2003). Traffic matrix estimation: Existing techniques and new directions. In *Sigcomm'02*, USA.

R. Mathieu, L. P. and Anandalingam, G. (1994). Genetic algorithm based approach to bi-level linear programming. *Operations Research*, 28(1):1–21.

Roughan, M., Thourp, M., and Zhang, Y. (2003). Traffic engineering with estimated traffic matrices. In *Proceedings of the USENIX/ACM Internet Measurement Conference*, Miami, USA.

Tylavsky, D. J. and Sohie, G. R. L. (1986). Generalization of matrix inversion lemma. *Proceeding of the IEEE*, 74(7).

Vardi, Y. (1996). Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*.

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

Zhang, Y. 6 months of abilene traffic matrices. `http://www.cs.utexas.edu/users/yzhang/research/AbileneTM/`.

Zhang, Y., Roughan, M., Duffield, N., and Greenberg, A. (2003). Fast accurate computation of lagre-scale IP traffic matrices from link data. In *ACM Sigmetrics*, CA.