# An Evaluation of Traffic Matrix Estimation Techniques for Large-Scale IP Networks

by

Titus Olufemi Adelani

A Thesis Submitted to the Faculty of Graduate Studies

In Partial Fulfillment of the Requirements for the Degree

of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba, Canada

# Table of Contents

# Abstract

The information on the volume of traffic flowing between all possible origin and destination pairs in an IP network during a given period of time is generally referred to as traffic matrix (TM). This information, which is very important for various traffic engineering tasks, is very costly and difficult to obtain on large operational IP network, consequently it is often inferred from readily available link load measurements.

In this thesis, we evaluated 5 TM estimation techniques, namely Tomogravity (TG), Entropy Maximization (EM), Quadratic Programming (QP), Linear Programming (LP) and Neural Network (NN) with gravity and worst-case bound (WCB) initial estimates. We found that the EM technique performed best, consistently, in most of our simulations and that the gravity model yielded better initial estimates than the WCB model. A hybrid of these techniques did not result in considerable decrease in estimation errors. We, however, achieved most significant reduction in errors by combining iterative proportionally-fitted estimates with the EM technique. Therefore, we propose this technique as a viable approach for estimating the traffic matrix of large-scale IP networks.

# Acknowledgments

First of all, I would like to express my deep sense of gratitude to my supervisor Dr. A. S. Alfa for his guidance, encouragement and patience all through my study - he has been more of a father than an advisor to me. I am also grateful to Bell Canada for providing the opportunity and funding for this research.

Many thanks to all the academic professors in the department of Electrical and Computer Engineering, University of Manitoba (UofM) for their contribution to my success through teaching and motivation. I am grateful to my examiners, Professors B. Kordi and J. Montufar, for their invaluable reviews and comments.

A special thanks to my colleagues at Communication and Network Engineering Research group of Robert Alan Kennedy Communications Laboratory for their friendship and useful technical discussions. A token of appreciation goes to all my friends and their families, who made living in Winnipeg a wonderful experience for me.

Finally, I am greatly indebted to my beloved wife, 'Tomi, and my daughter, Blessing, for their support and sacrifice toward this academic achievement. I am grateful to my parents and parents-in-law for their prayers and encouragement too. Praise God!

# List of Acronyms

| | |
|---|---|
| **AS** | autonomous system |
| **BGP** | border gateway protocol |
| **BR** | backbone router |
| **CR** | core router |
| **EGP** | exterior gateway protocol |
| **EM** | entropy maximization |
| **ER** | edge router |
| **IGP** | interior gateway protocol |
| **IP** | internet protocol |
| **IPF** | iterative proportional fitting |
| **ISIS** | intermediate system-intermediate system |
| **ISP** | internet service provider |
| **LP** | linear programming |
| **MRE** | mean relative error |
| **NN** | neural network |
| **OD** | origin-destination |
| **OSPF** | open shortest path first |
| **POP** | point-of-presence |
| **QoS** | quality of service |
| **QP** | quadratic programming |
| **SD** | source-destination |
| **SNMP** | simple network management protocol |
| **TG** | tomogravity |
| **TM** | traffic matrix |
| **WCB** | worst-case bound |
| **WLSE** | weighted least-squares estimation |

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 General Background

An IP network typically consists of IP routers and interconnecting links between the routers, under a single administrative domain or autonomous system (AS). Internet Service Providers (ISPs) usually divide their IP network functionally into two parts - the edge and the backbone as shown in Figure 1.1. The network *edge* provides connectivity to customers, via customer access links, as well as to other ISPs via peering links. The backbone of the network performs high-speed routing and switching functionality from one edge of the network to another. The network backbone may be sub-divided logically into a core and distribution layer for ease of administration and enforcement of policies and security.

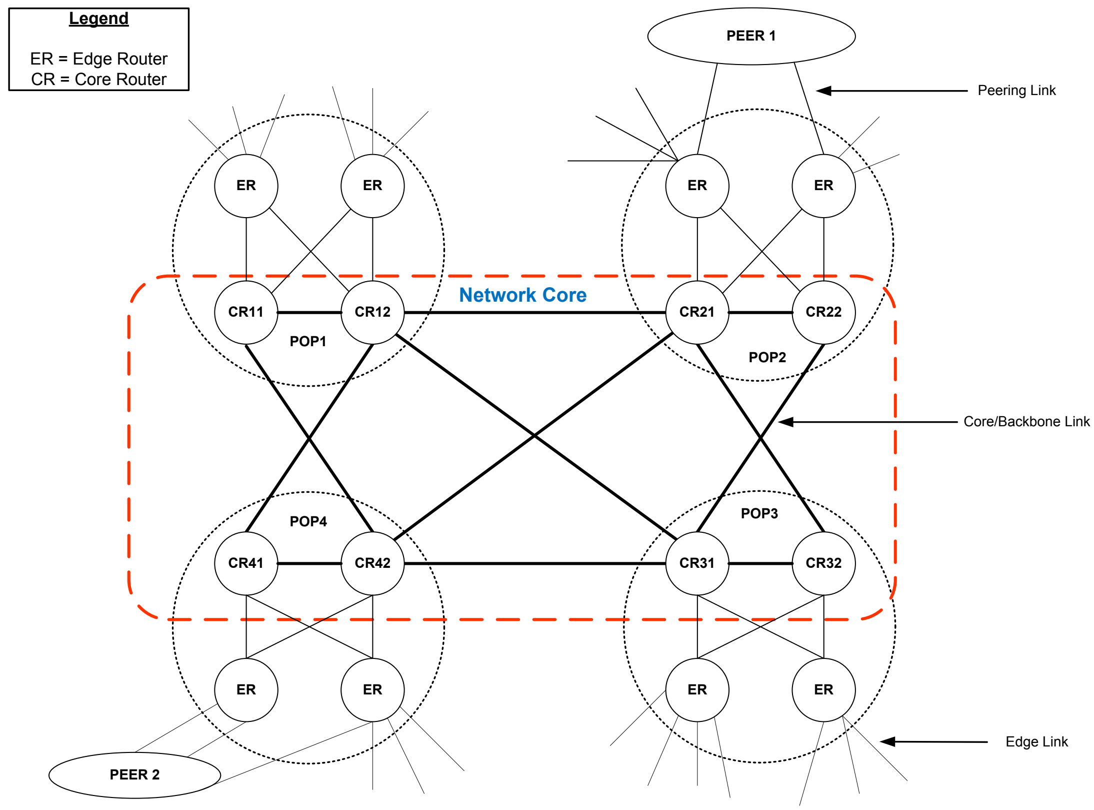


Figure 1.1: Simplified 4-POP ISP Topology

Geographically, ISPs often segment a large IP network into smaller units, each unit is referred to as a point-of-presence (POP). A POP typically provides connectivity to customers residing in an area or geographical location through *access* links. It also provides connectivity to other areas through high-speed *backbone* links. A POP may also have one or more *peering* links. In large IP networks such as those managed by ISPs, the flow of traffic is determined by forwarding/routing table on each router. Routers build routing tables based on configured parameters of the routing protocols and use these tables in making decisions on how to forward packets within the network or autonomous system or from the network to other autonomous systems. An interior gateway protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System - Intermediate System (ISIS) is an *intra*-domain routing protocol, while the Border Gateway Protocol (BGP) - an exterior gateway protocol (EGP) - is the typical routing protocol used for *inter*-domain routing.

IP traffic matrix (TM) measures the total amount of traffic that goes from any entry (*ingress*) node to any exit (*egress*) in an IP network during a given period of time. The information provided by the TM is an essential input for many network design and traffic engineering tasks such as load balancing, routing protocol configuration, capacity planning, link failure analysis, Quality of Service (QoS) provisioning and anomaly detection. The choices that IP network operators make in managing the network depend on the knowledge of how much traffic flows through the network, which is captured by the TM.

In spite of its importance, TM is difficult to obtain on large IP networks, and has to be inferred from readily available link counts obtained using simple network management protocol (SNMP). SNMP is part of the Internet Protocol suite and it is designed for management and monitoring of network devices. Most of the challenges associated with direct measurement of TM on IP networks is due to poor support in network equipment and high cost of extracting the information from large amount of data that flows through the network [13]. This cost, which consists primarily of the storage, computational and communication overheads associated with the collection and processing of traffic flow data across the network depends on the granularity and frequency at which the TM is being estimated.

IP traffic matrix can be estimated at various levels of granularity: POP, router, link, or prefix levels, in increasing order of complexity and size [1]. For example, in the simplified 4-POP ISP topology shown in Figure 1.1, estimating TM at POP level would involve determining the volume of traffic flowing from one POP to each of the other POPs (that is $POP1 \rightarrow POP2, POP1 \rightarrow POP3, POP1 \rightarrow POP4, POP2 \rightarrow POP1, POP2 \rightarrow POP3, ...., POP4 \rightarrow POP3$) resulting in a 12-element POP-to-POP traffic matrix. On the other hand, estimating the TM at backbone or core router (CR) level for the same network would involve determining the volume of traffic flowing from each of the 8 core routers to the others (that is, $CR11 \rightarrow CR12, CR11 \rightarrow CR21, CR11 \rightarrow CR22, ...., CR42 \rightarrow CR41$), resulting in a 56-element router-to-router traffic matrix. At link and prefix levels, the complexity and size of the matrix increases proportionally. Typical Tier-2 ISPs have POPs in the order of tens with core/backbone routers ranging from hundreds to a few thousands and TM for most traffic engineering applications are measured at POP and router levels [4, 7, 9].

In terms of frequency, TM can be estimated every 5-minutes, every 15-minutes, hourly, over the busy-hour (high traffic) period of the day, daily, weekly, etc. The time scale or frequency of estimation is usually dependent on the time-scale at which link load data is measured.

## 1.2 Purpose of Thesis

The goal of this thesis is to perform a thorough, independent evaluation of major IP traffic matrix estimation techniques proposed in the literature to date and provide recommendation and guidelines to ISPs on suitable approaches and techniques to adopt in performing TM estimation on their IP network. Another motivation is that, since most service providers do not have accurate TM information and they are not willing to measure it directly on their network, our evaluation of TM estimation techniques on a similar network using real Internet traffic data will provide them an idea of the expected accuracy of the techniques and enable them to account for these errors when using estimated TM for traffic engineering purposes. In addition, many estimation techniques often perform well or poorly depending on topology and traffic distribution within the network. By capturing these parameters that affect the results of TM estimation in our evaluation, we provide a way for ISPs to assess which technique or combination of techniques is more suitable for TM estimation of their networks, without having to experiment with each method on an operational network.

## 1.3 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, we reviewed various estimation techniques that have been proposed in the literature. In Chapter 3, we described our evaluation methodology as well as the data/parameters used. We evaluated 5 of the major estimation techniques using 3 different network topologies and real Internet traffic data from Abilene research network [33]. We also evaluated the performance of hybrid techniques, formed by combining any two of the well-known techniques in estimating the traffic matrix, and proposed a new method of TM estimation from previous measurements. Numerical results for each evaluation is presented in Chapter 4. Finally we concluded the thesis in Chapter 5 with recommendations to ISPs based on our comparative study. We also discussed the relevance of thesis to Engineering and provided some directions for future work in this area.

# Chapter 2

# Literature Review:

# IP Traffic Matrix Estimation

# Techniques

## 2.1 Introduction

The problem of estimating origin-destination (OD) traffic matrix has been well-studied in the literature for telephone networks and road transportation network dating as far back as the 1930s. It was not until 1996 that the problem was addressed for IP networks. Vardi [10] in 1996 was the first to study the problem of estimating traffic intensity between all OD pairs in an IP network from repeated measurement of traffic flow along the directed links connecting the nodes. He coined the term "network tomography" for the problem, perhaps due to its similarity with tomographic problems in medicine and other sciences.

## 2.2 Problem Definition and Notations

Consider a network with $n$ nodes and $r$ directed links. On an IP network, each node corresponds to a router or POP and each link corresponds to the physical communication media carrying the traffic. Each node generates traffic (data, voice or video) destined for other nodes in the network. For this network, there are typically $c = n^2$ or $n(n-1)$ OD traffic elements. The path through the network is defined by the routing matrix, $A$, whose elements, $A_{i,j}$, denote the fraction of traffic for the OD pair $j, j = 1, 2, ...., c$ that is carried by link $i, i = 1, 2, ..., r$. Figure 2.2 shows a 4-node (4-router) network with 3 bidirectional or 6 unidirectional links (solid lines) and 12 OD pairs (dashed lines).

The objective of IP traffic matrix estimation is to estimate $c$ OD traffic demands given $r$ link load measurements and the routing matrix $A$. $A$ is a $r \times c$ matrix and it is assumed to be constant and known. The relationship between the demands and the link counts is often represented by the following linear equation,

$$\mathbf{Y}^{(k)} = \mathbf{A}\mathbf{X}^{(k)} \tag{2.1}$$

where

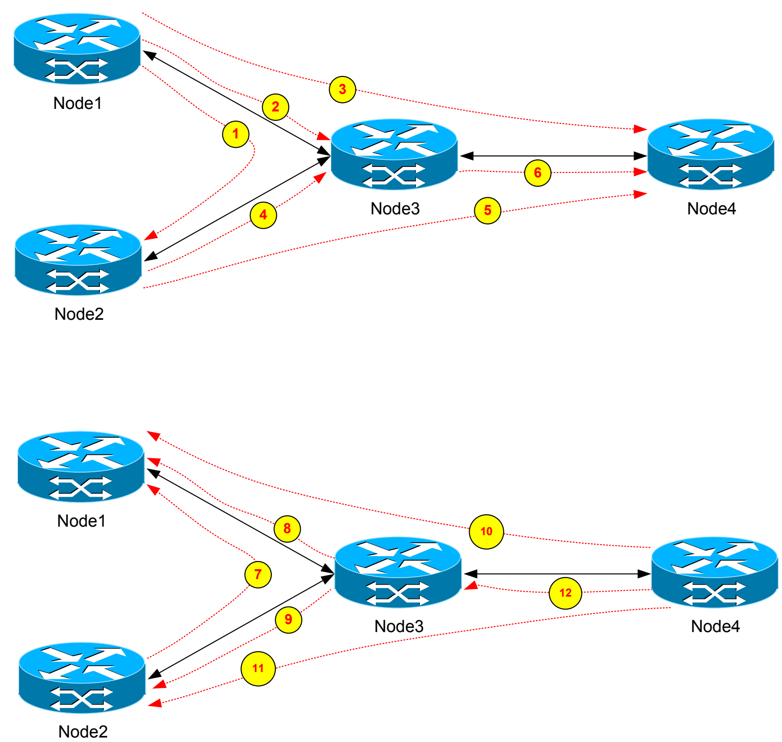


Figure 2.1: Relationship between Nodes and OD pairs

$Y_i^{(k)}$ is the measured traffic on link $i$, $1 \leq i \leq r$ during the time interval $k$, and $\mathbf{Y}^{(k)} = [Y_1^{(k)}, Y_2^{(k)}, ...., Y_r^{(k)}]^T$ is the measured traffic on all links of the network at time period $k$, written as a column vector.

Similarly, $X_j^{(k)}$ is the demand for OD pair $j$, $1 \leq j \leq c$ during the time interval $k$ and $\mathbf{X}^{(k)} = [X_1^{(k)}, X_2^{(k)}, ...., X_c^{(k)}]^T$ is a column vector of OD traffic matrix elements that we aim to estimate. $[\mathbf{.}]^T$ denotes the transpose operator. The measurement time period $k$ can be 5-minute, 15-minute or an hour. We used hourly measurements in this work and assumed that a total of $K$ measurements are available, that is, $k = 1, 2, ...., K$. It has been shown that fanouts are generally more stable over the space of an hour and most traffic engineering applications are targeted toward long time scales [7, 13].

The total number of links, $r$, may range from $O(n)$ to $O(n^2)$, but generally of $O(n)$ implying that $c > r$. This implies that 2.1 is an under-determined system of equations in which, $c$, the number of unknowns is much greater than, $r$, the number of known, hence, there is no unique solution. The inability to obtain a unique solution stems from the fact that it is not possible to find the exact inverse of the rectangular routing matrix, $A$, whose rank is usually less than or equal to the number of links, $r$. The traffic matrix estimation problem therefore, finding the "best" solution to an under-constrained problem. This involves choosing one solution (out of many plausible solutions) that is consistent with observed link loads and is closest to the actual traffic that generates the link loads.

Several techniques have been proposed for estimation of traffic matrices from link loads. Experience in solving similar large-scale ill-posed inference problems described by equation 2.1 requires the incorporation of additional (side) information and assumptions about the nature of the problem, in order to make the problem less under-determined or to guide in the selection of the most probable solution out of all possible estimates [4]. The process of introducing additional information in order to solve an ill-posed problem is generally referred to in mathematics as *regularization*. Regularization in TM estimation takes the form of computing an initial estimate or prior distribution of the traffic matrix, which is then refined by some statistical or optimization algorithm.

## 2.3 Initial Traffic Estimate or Prior Distribution

We consider some of the various choices of possible initial estimates proposed in the literature namely - the gravity model, the worst case bound (WCB) and the fanout estimates.

### 2.3.1 Gravity Model

The gravity modelling is based on the Newton's law of gravitation and has been used by social scientists to predict the movement of people, goods, services and information between cities or geographical locations by taking in the consideration the population and distance factors. Gravity model has also been used in estimating telephone demands. An application of this model to IP traffic matrix estimation was first proposed by Roughan *et. al* [2] and is based on the total amount traffic entering and leaving each node in the network and the total traffic in the network. The model estimates $X_{i,j}$, the volume of traffic between ingress node $i$ and egress node $j$ as

$$X_{i,j} = \frac{N_{in}(i) * N_{out}(j)}{\sum_{k=1}^{n} N_{in}(k)} \tag{2.2}$$

or

$$X_{i,j} = \frac{N_{in}(i) * N_{out}(j)}{\sum_{k=1}^{n} N_{out}(k)} \tag{2.3}$$

where

$N_{in}(i)$ is the total amount of traffic originating from node $i$;

$N_{out}(j)$ is the total amount of traffic destined for node $j$.

In an ideal network, $\sum_{k=1}^{n} N_{in}(k)$ and $\sum_{k=1}^{n} N_{out}(k)$ should yield the same result based on flow conservation principle. However, in practice, due to packet losses, delay and other network errors, the total traffic into and out of the network do not match, hence, equations 2.2 and 2.3 do not yield the same result. Furthermore, the gravity model is rarely used in isolation, but in combination with (or as the starting point of) other techniques, because its estimates are often poor and generally inconsistent with link load constraint equation (2.1). We implemented this procedure using a simple MATLAB code.

### 2.3.2 Worst-Case Bound

An alternative choice of initial estimate of the TM is the worst-case bound (WCB) approach proposed by Gunnar *et. al* [7]. This technique is based on computing the mean of the lower and upper bound of each traffic demand using linear programming (LP). In particular, $X_p^-$, the lower bound of demand for each OD pair $p$ is found by solving the LP formulation:

$$\min \quad \{X_p\} \tag{2.4}$$

$$\text{subject to } \mathbf{AX} = \mathbf{Y}$$

$$\mathbf{X} \geq \mathbf{0}$$

Similarly, $X_p^+$, the upper of bound of demand for each OD pair $p$ is found by solving the LP formulation

$$\max \quad \{X_p\} \tag{2.5}$$

$$\text{subject to } \mathbf{AX} = \mathbf{Y}$$

$$\mathbf{X} \geq \mathbf{0}$$

$$X_p \leq \max\{Y_l\}, \forall l \in \mathcal{L}(p)$$

where

$\mathcal{L}(p)$ is the set of all links traversed by the traffic of OD pair, $p$.

One drawback of this technique is that it is computationally demanding, in terms of the number of computations to be performed and time required to obtain a TM estimate, because it involves solving two linear programming problems for each OD pair over a single set of link load measurement. Furthermore, the bounds (that is upper bound and lower bound of each demand) tend to be loose, especially for large demands. We implemented this procedure in MATLAB using the optimization toolbox [30].

### 2.3.3 Constant Fanout Model

Gunnar *et. al* [7] proposed the estimation of *fanout* from time-series of link load measurement. Fanout is the proportion of traffic flowing from a node to all other nodes. It is equivalent to the probability of a node sending traffic to all other nodes captured in the network (or included in the TM), hence the sum of fanout for a node is equal to 1. The fanout estimate is based on the assumption that the fanout of each node is relatively constant over a period of time and that link load fluctuations are caused by the changes in the total traffic generated by each node . The constant fanout model estimates the traffic matrix by solving the following equality-constrained quadratic optimization problem.

$$\min \sum_{k=1}^{K} ||\mathbf{A}\mathbf{S}^{(k)}\mathbf{P} - \mathbf{Y}^{(k)}||_2^2 \tag{2.6}$$

$$\textstyle\sum_{j=1}^{n} p_{ij} = 1$$

$$1 \leq i \leq n$$

$$0 \leq p_{ij} \leq 1$$

We implemented this procedure using MATLAB optimization toolbox. However, we found that, for most of our data sample, it was difficult to find a feasible solution satisfying the objective and constraints. Consequently, we did not include this technique in our evaluation.

### 2.3.4 Choice Model

Medina *et. al* [9] proposed the "choice" model which is similar to the constant fanout. This model employs multinomial logistic regression to estimate fanout based on the total incoming and outgoing traffic of each node. A knowledge of the fanout of each node is required in order to obtain the parameters that provide the best fit to the model, which is equivalent to the TM estimation problem itself. We did not include this model in our evaluation. Furthermore, since we use real data for our simulations, we can easily obtain fanouts directly from the data as explained in Section 3.2.4.

### 2.3.5 Iterative Proportional Fitting

Iterative proportional fitting (IPF) is one of the techniques that have long been used by researchers to adjust two-dimensional tables to known marginals [27,28]. The technique is a simple two-step arithmetic procedure. In the first step, each element of the matrix is multiplied by a factor that makes the sum of each row equal to the known marginal of each row. In the second step, each element is multiplied by a factor that makes the sum of each column equal to the known marginal of each column. The two steps are repeated until convergence is reached, either when the difference in value of each cell or marginal becomes less a predefined threshold, $\delta$ or a maximum number of iterations, $MaxIter$, has been performed.

In the context of IP TM estimation, the known marginals are the link loads. Given an initial estimate which is inconsistent with link load constraint equation 2.1, we iteratively adjust this estimate to the known link loads. The initial estimate is often the output of a traffic matrix estimation technique, such as tomogravity or artificial neural network, that contains negative values. However, since traffic matrix elements are non-negative, the negative values are set to zero and the IPF technique is used to adjust the resultant estimate to link loads. IPF has been used as a post-estimation technique in [3,19], but it can also be used to obtain an initial estimate of traffic distribution from sampled traffic matrix as proposed in Section 3.2.4. We implemented this procedure in MATLAB with $\delta = 0.01$ and $MaxIter = 20,000$.

## 2.4 Traffic Matrix Estimation Techniques

### 2.4.1 Tomogravity

The tomogravity technique proposed by Zhang *et al.* [3] is a combination of two techniques, the *tomography* estimation and the *gravity* modelling. This technique attempts to solve the traffic estimation problem by solving a quadratic programming problem formulated below.

$$\min ||\mathbf{X}^{(k)} - \mathbf{X}_g^{(k)}||_2^2 \tag{2.7}$$

$$\text{subject to } \mathbf{A}\mathbf{X}^{(k)} = \mathbf{Y}^{(k)}$$

where $\mathbf{X}_g^{(k)}$ is the vector of prior estimate obtained using gravity model discussed in Section (2.3.1) and $|| \bullet ||_2^2$ is the square of the $L_2$ norm of a vector.

Although the problem is formulated as an optimization, the solution is obtained using Singular Value Decomposition (SVD) of the routing matrix was employed to find a least square solution to the quadratic programming problem. In MATLAB, this is achieved by computing the pseudo-inverse or the Moore-Penrose inverse [22,23] of the routing matrix. The resulting solution sometimes contain negative values, hence, iterative proportional fitting (IPF) procedure, described in Section 2.3.5 is applied after setting the negative values to zero, to ensure a non-negative solution which satisfies the link load constraint is achieved. The authors also investigated weighted least square (WLSE) solution to the problem and found that the square-root weight provided the best estimates, although the difference in performance to other was not too significant.

### 2.4.2 Entropy Maximization

Zhang *et al.* [4] also applied regularization in solving the traffic estimation problem, drawing from experience in solving similar ill-posed problems in other scientific and engineering fields. On the assumption of conditional independence of source and destination on the network, they employed a regularization functional that minimizes the mutual information of each OD pair. We refer to this approach as Entropy Maximization technique. The formulation is

$$||\mathbf{Y}^{(k)} - \mathbf{A}\mathbf{X}^{(k)}||_2^2 + \lambda^2 I(S, D) \tag{2.8}$$

$$\text{subject to } X_i^{(k)} \geq 0; \text{ for } i = 1, 2, ..., c$$

where

$$I(S, D) = \sum_{j:g_j>0} X_j^{(k)} \log\left(\frac{X_j^{(k)}}{X_{g_j}^{(k)}}\right) \tag{2.9}$$

and $X_{g_j}^{(k)}$ is the gravity model estimate of the demand $X_j^{(k)}$.

### 2.4.3 Quadratic Programming

Tebaldi and West [11] proposed the use of Bayesian statistics for solving the TM estimation problem. Their approach entails finding the joint posterior distribution $p(X^{(k)}|Y^{(k)})$ for all OD pairs $X^{(k)}$ given the observed link loads $Y^{(k)}$. They assumed the prior distribution $p(X^{(k)})$ is Poisson. Gunnar *et al.* [7] however, have shown that, if one chooses a Gaussian prior distribution model instead of Poisson, and assumes that the link loads are subject to white noise with unit variance, the maximum a posteriori estimate of the traffic matrix can be found by solving the quadratic program below.

$$\min ||\mathbf{Y}^{(k)} - \mathbf{A}\mathbf{X}^{(k)}||_2^2 + \sigma^{-2}||\mathbf{X}^{(k)} - \mathbf{X}_p^{(k)})||_2^2 \tag{2.10}$$

$$\text{subject to } X_i^{(k)} \geq 0; \text{ for } i = 1, 2, ..., c$$

where $\mathbf{X}_p^{(k)}$ is a prior estimate of the traffic matrix $\mathbf{X}^{(k)}$

### 2.4.4 Linear Programming

We evaluated the Linear Programming(LP) proposed by Conway and Li [5], which is a form of fanout estimation. This technique estimates the traffic matrix by finding the fanout factors, $p_{ij}$, which is the probability of a random packet leaving the network through node $j$ given that it enters through node $i$. Using the notation in sec 2.2, the problem is stated using the following set of equations.

$$\mathbf{Y}^{(k)} = \mathbf{A}\mathbf{S}^{(k)}\mathbf{P}^{(k)} \tag{2.11}$$

$$\mathbf{X}^{(k)} = \mathbf{S}^{(k)}\mathbf{P}^{(k)} \tag{2.12}$$

and

$$\sum_{j=1}^{n} p_{ij}^{k} = 1; 1 \leq i \leq n; 0 \leq p_{ij} \leq 1, \tag{2.13}$$

where $\mathbf{S}^{(k)}$ is a $c \times c$ diagonal scaling matrix, whose elements are total traffic entering the network at each node, replicated $n$ times and $\mathbf{P}^{(k)}$ is the column vector of fanouts $p_{ij}^{k}$ ordered according to $\mathbf{X}^{(k)}$. We solved the LP problem using linear *goal programming* approach proposed in [8]. The final estimated is computed using equation 2.12.

### 2.4.5 Artificial Neural Network Approach

A new class of methods was recently introduced into traffic estimation techniques repertoire from the field of artificial intelligence. These techniques, which are based on various artificial neural network (ANN) model, attempt to provide a functional approximation to the inverse relationship and/or recognize any pattern between link counts and actual OD traffic matrix. Typically, an ANN model is developed and representative data from the IP network is used to "train" the ANN model. The resultant trained artificial neural network can be used for future estimation and prediction.

The task of using ANN model to estimate IP traffic matrix involves identifying which model to use, selecting appropriate training/learning algorithm and generating representative data for supervised training of the model. The last of these tasks is the most challenging, since actual traffic matrix of the network are not available. To date, all published research that employed this technique have used data from Abilene research network [33] on which actual TM measurement has been carried, hence, it is doubtful if anyone has applied this technique on a real ISP network. Some of the ANN models that have been proposed include the feedforward backpropagation neural network (BPNN) [19], radial basis function neural network (RBF) [20] and multilayer recurrent neural network (RNN) [21]. These techniques often incorporate an iterative proportional fitting (IPF) procedure at the end to handle negative results generated in the estimation process. Figure 2.2 shows the block diagram of the artificial neural network used in this work.

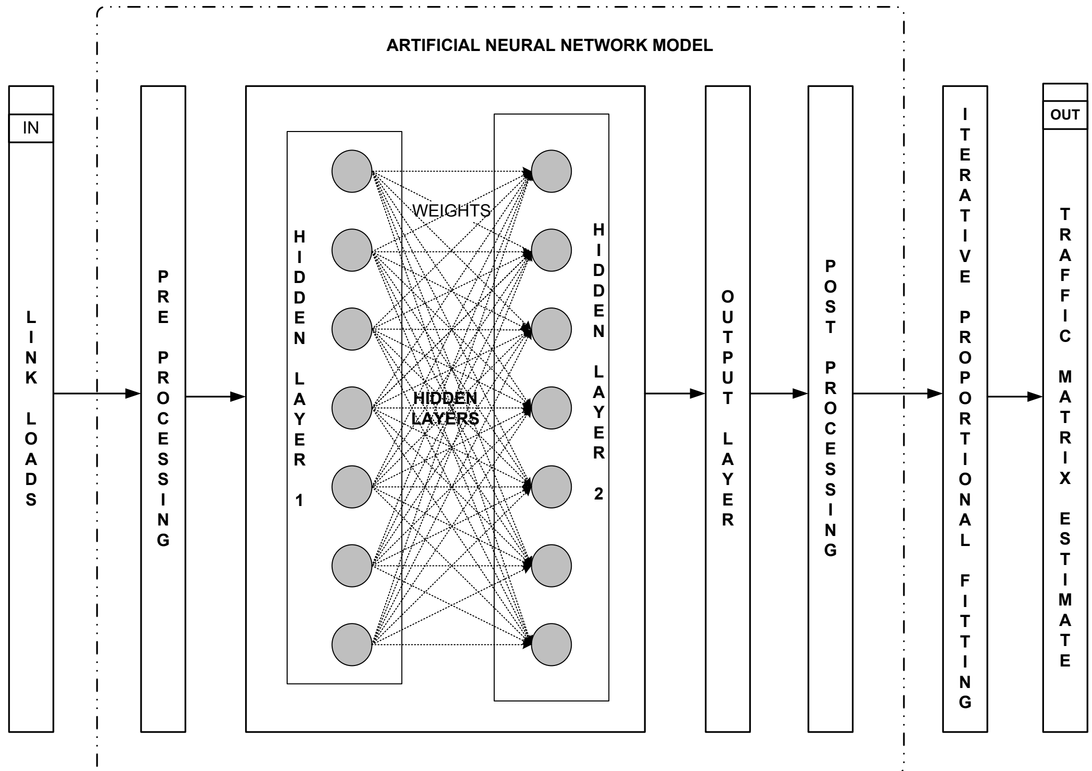


Figure 2.2: Block diagram of the Artificial Neural Network Model

### 2.4.6 Other Techniques

A few other techniques have been proposed in the literature, which were not evaluated in this thesis. Most of these techniques exploit the statistical properties of the readily available SNMP link counts in the estimation of OD traffic matrix. These are often augmented with an assumption about the prior distribution of each OD traffic. We refer the interested reader to the original references cited for each technique for details.

We only provide a summary of these other techniques here.

Vardi [10], on the assumption that OD traffic matrix elements follow a Poisson distribution, proposed the use of covariance of link counts to generate additional constraint equations, which can be used to augment equation (2.1). He argued that under this assumption, the mean rates of OD flows are identifiable. Medina *et. al* however showed to that Poisson assumption is not generally valid [9]. Gunnar *et. al* also showed, using measured traffic from an ISP network, that it is difficult to find mean rates satisfying this set of equations even when the system of equations is no longer under-determined [7]. Tebaldi and West [11] improved on Vardi's work, following the same assumption, but using Markov chain monte-carlo (MCMC) to find the mean rates (as well as traffic estimates for a subset of the traffic demand and them computing the remaining demands by matrix inversion. Their method is based on a combination of QR decomposition of the routing matrix $A$ and Bayesian inference using "Metropolis-within-Gibbs" algorithm. Vaton and Bedo [12] improved on the works of Tebaldi and West by assuming that the traffic matrix is a mixture of Gaussian distributions. Their approach known as iterative bayesian estimation technique in which the initial distribution is based on either the gravity model or the method of moments. These techniques require extensive simulation and often yield inaccurate results when the underlying assumptions are violated. [8].

The authors in [17] combined non-linear programming with the pseudo-inverse of the routing matrix. Using two sets of synthetic data drawn from Poisson and Gaussian distributions, and a 4-node network, they showed that this method is better than the bayesian approach. Authors in [18] however, combined linear programming with the pseudo inverse approach and validated their method with Abilene data. [33]. Nucci *et. al* [25] proposed a method of changing the IGP link weights in order to obtain a set of measurements that makes equation 2.1 full rank so that the the routing matrix becomes invertible and a direct solution can be obtained. The authors in [14] also proposed a method for estimating variance of OD flows as well OD flow estimate using the pseudo-inverse approach. This method assumes a full-rank routing matrix obtained using the IGP link weight change technique proposed in [25]. Considering the potential impact of routing changes on large IP network, it is doubtful if service providers would be willing to implement both techniques.

Some authors have assumed that a general power-law or generalized-scaling relationship exists between the mean and variance OD traffic flows. This relationship states that, if the mean of traffic rate for an OD pair is $\lambda$ and the variance is $\Sigma$, then a relationship of the form $\Sigma = \Phi\lambda^c$ exists, where $\Phi$ and $c$ are parameters to be determined. In [16], the authors proposed a technique for estimating the mean and variance of OD flows from the covariance of link loads based on the generalized scaling law. The final TM final TM estimate is then calculated using a projection method to ensure consistency with link loads. The authors in [7] and [9] proved, using both real and synthetic data, that the power-law relationship is generally not valid.

# Chapter 3

# Evaluation of Traffic Matrix Estimation Techniques

We compared the performance of 5 traffic matrix estimation techniques namely Tomogravity (TG) [3], Entropy Maximization (EM) [4], Quadratic Programming (QP) [4], Linear Programming (LP) [5] and Artificial Neural Network (NN) [19]. These techniques were chosen as representative of most of the techniques proposed in the literature today based on their reported performance. We evaluated the tomogravity technique using the WLSE code published by the authors in [3] with square-root weight and also applied the IPF procedure to ensure non-negativity of results. We implemented the EM and QP technique using the same PDSCO code [32] used by the authors [4] and a regularization parameter $\lambda = 0.01$. LP and NN techniques were implemented using MATLAB's optimization toolbox [30] and neural network toolbox [29] respectively. Table 3.1 contains important parameters used for the neural network simulation.

Table 3.1: Neural Network Simulation Parameters

| Parameter | Value |
|---|---|
| Model | Feedforwarded BackPropagation (newff) |
| Number of Layers | 2 |
| Number of Neurons in Hidden Layers | Layer 1 = size of link loads |
| | Layer 2 = size of OD pairs |
| Training Algorithm | 4 Node = Levenberg-Marquardt (trainlm) |
| | 12/14 Node = Scaled Conjugate Gradient (transcg) |
| Learning Algorithm | Gradient Descent with Momentum (learngdm) |
| Number of Epochs | 4 Node = 500 |
| | 12/14 Node = 1000 |
| Pre-Processing Function | Zero-mean normalization (mapstd) |
| Post-Processing Function | Reverse zero-mean normalization (mapstd) |
| Goal | 1.00E-03 |

## 3.1 Network Topology, Data Set and Performance Measures

### 3.1.1 Network Topology

We performed our evaluation using three networks of different sizes and topologies - a 4-node network, a 12-node Abilene network and a 14-node network. Figures 3.1, 3.2 and 3.3 show the 4-node, 12-node and 14-node network topologies respectively. We used the 4-node network to gain an insight into the performance of the estimation techniques. The small size of the network also allowed us to observe the details of each estimation technique.

The 12-node network is a typical POP network whose IGP weights have been tuned to ensure an "all-or-nothing" routing. Typically, routing protocols, such as OSPF would distribute the traffic for an OD pair over multiple paths, if those paths have equal cost. However, in all-or-nothing routing assignment, all traffic for an OD pair is forced to flow through only a single path either by tweaking the parameters that the routing protocol uses in computing the cost and consequently determine the best path or by manually defining the path for that OD pair's traffic.

The 14-node network [8] is a variant of the ISP POP topology used by other researchers [9] and implements a pure OSPF routing, which allows traffic for an OD pair to travel over multiple paths if the paths have equal cost. This combination of topology enables us to capture the effect of topology and routing dynamics in our evaluation.

### 3.1.2 Routing Matrix

The routing matrix for the 4-node and 14-node network were computed using shortest path first (SPF) algorithm based on the topology in Figures 3.1 and 3.3. In computing the SPF routing matrix, we assume that all links have equal capacity and consequently assign them a weight of 1 unit. The routing matrix for the 12-node network was obtained as part of the evaluation data set.

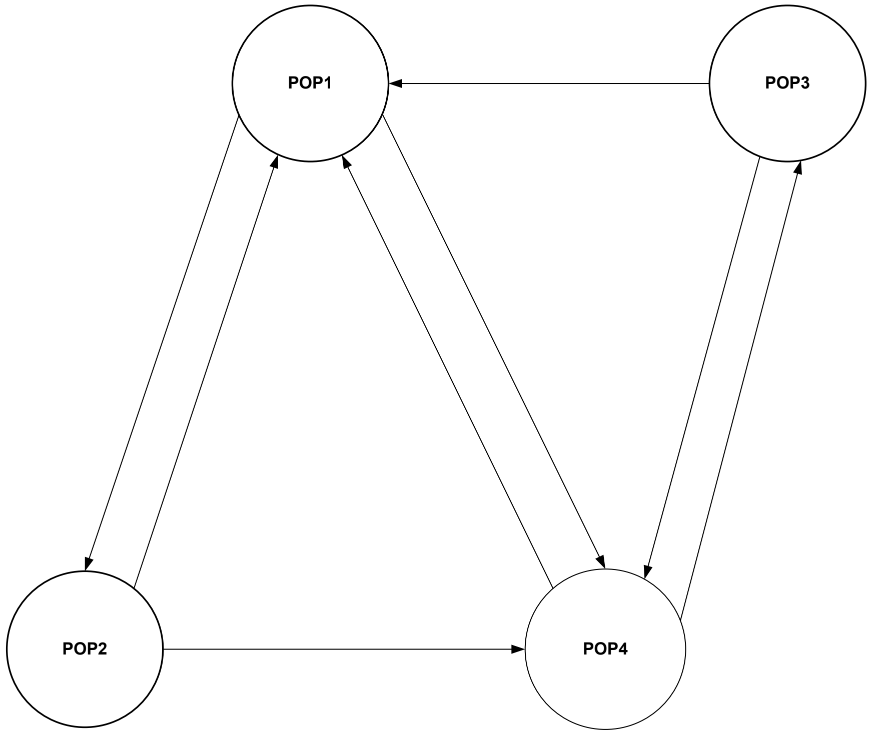


Figure 3.1: Topology of the 4-node Network

### 3.1.3 Evaluation Data Set

Many of the previous works [6, 8, 9] have used synthetic data, generated using a predetermined probability distributions and parameters, to evaluate the accuracy of traffic matrix estimation techniques, because real matrices were not available. These authors showed that most of the traffic estimation techniques are generally biased toward the distribution used and that any assumption of a particular probability distribution concerning OD flows in real matrix is generally not valid.

Nucci *et. al* [24] addressed this problem by attempting to fit some measured Inter-

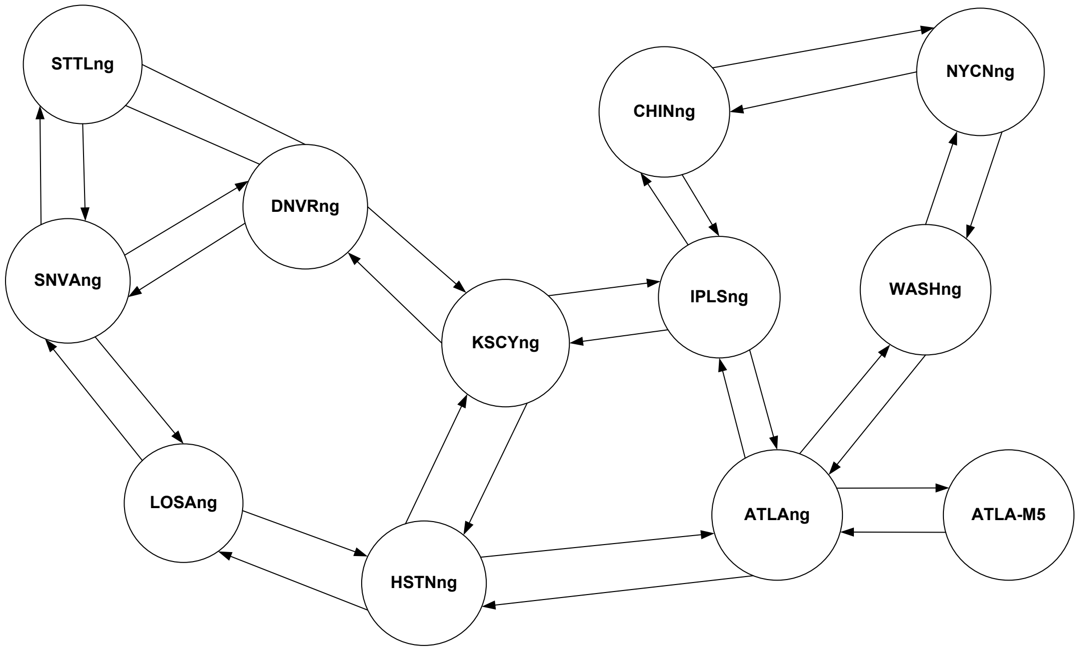


Figure 3.2: Topology of the 12-node Network

net traffic data to about 12 well-known distributions, including the uniform, Gamma, Weibull,LogLogistic, Lognormal and Inverse-Gaussian distributions, and concluded that none of these distributions provided a perfect fit. However, the lognormal distribution was found to provide the best fit to the aggregated data set (which is intuitive considering the aggregated nature of Internet traffic) and was therefore recommended. In [26], the author proposed a synthesis or real traffic matrix using the gravity model. He also showed that this method is simpler and provided similar traffic matrix to those generated using the LogNormal distribution.

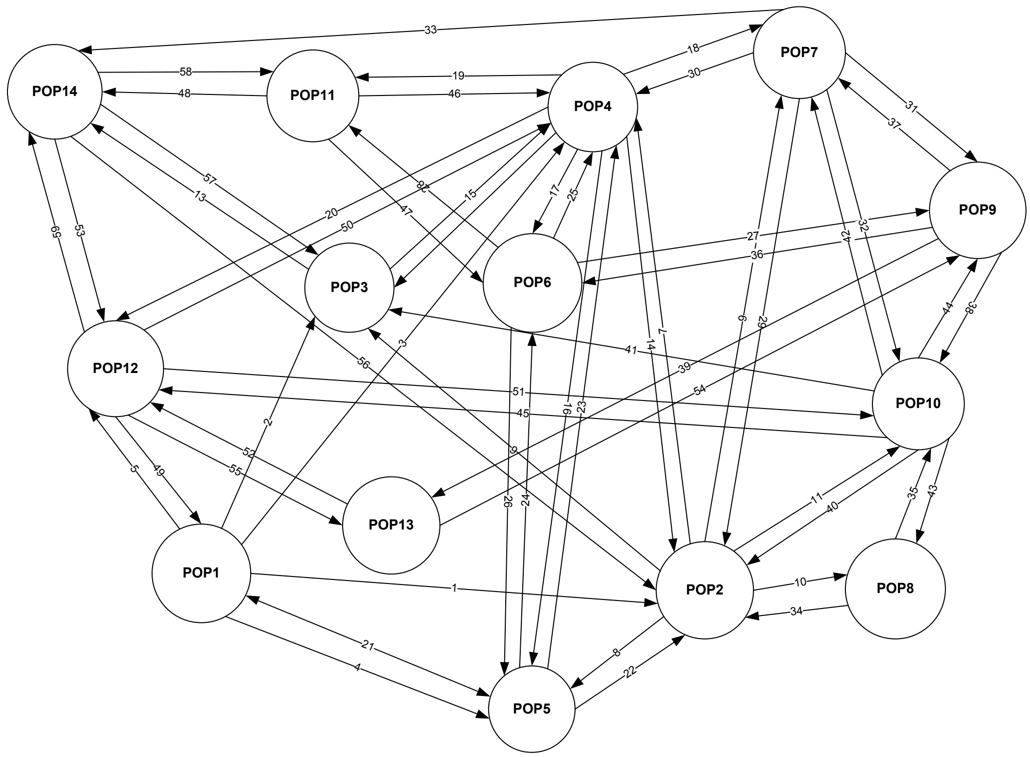


Figure 3.3: Topology of the 14-node Network

In this work, we avoided the problems discussed in [24] by using real network traffic data obtained from Abilene network [33]. Abilene network, which was created by the Internet2 community and is currently part of the Internet2 network, is a large-scale, high-speed IP backbone network, connecting several universities and affiliated institutions in the United States [34]. The network provides U.S. research and academic community with scalable, cost-effective and innovative hybrid optical and packet network. The data set used in our evaluation consists of real OD traffic matrices for 144 OD pairs, were collected at fixed intervals of 5-minutes, on the 12-node Abilene network shown in 3.2, over a period of 24 weeks starting from March 1, 2004 on 12 routers by Professor Y. Zhang of the University of Texas at Austin [33]. Table 3.2 lists the names of the routers and their locations. This data has been made available to the Internet community for research purposes and has been used by several authors [19, 21, 24, 26].

Table 3.2: List of Abilene Network Routers and Locations as of March 1, 2004 [33]

| **R**outer Name | **C**ity |
|---|---|
| ATLA-M5 | Atlanta_GA |
| ATLAng | Atlanta_GA |
| CHINng | Chicago_IL |
| DNVRng | Denver_CO |
| HSTNng | Houston_TX |
| IPLSng | Indianapolis_IN |
| KSCYng | Kansas_City_MO |
| LOSAng | Los_Angeles_CA |
| NYCMng | New_York_NY |
| SNVAng | Sunnyvale_CA |
| STTLng | Seattle_WA |
| WASHng | Washington_DC |

The files containing this data are labelled “Xuv”, where “uv” is a two-digit number representing the week. For example, the file X01, contains measurement for week 1 starting March 1, 2005 while the file X22 contains measurement for week 22 starting August 21, 2004. Each row of the file contains a single traffic matrix, hence there are $12 \times 24 \times 7 = 2016$ (corresponding to 12 measurements by hour at 5-minutes interval, 24 hours per day, 7 days per week) measurement of OD pair traffic matrices in each file. Each row contains 720 columns, however, only the first column and subsequent 5th columns contain the real traffic matrix. Others contain results of estimation using some estimation techniques. A "readme.txt" file that provides additional details about this data can be found at [33]. First, we extracted only the required data from each file. Thereafter, we converted this 5-minute traffic matrix to hourly average traffic matrix by taking an average of the 12 sample measurements for each hour.

In evaluating the 12 node network, we used the hourly traffic matrix as it is. For the 4-node network simulations, we select by random permutation, traffic matrix for only 12 of the 144 OD pairs. Random permutation is done using MATLAB function *randperm*, which yields a random ordering of the 144 OD pairs from which we select the first 12 indices. In the case of 14-node network, where we require 182 OD pair, we duplicated the traffic matrix thus yielding a matrix of 288 OD pairs. We use the same random permutation as in the 4-node network to select the first 182 OD pairs.

### 3.1.4 Performance Measure

Several measures have been adopted by various authors in evaluating the accuracy of traffic matrix estimation techniques. One popular measure is the Root Mean Square Error (RMSE) used in [3, 6, 8]. This measure is often combined with the root mean square relative error (RMSRE) in order to obtain a proper assessment of nature and distribution of the error. In this work, we measure error in terms of the absolute relative error. We prefer this method because of its simplicity and physical meaning compared to other error measures. The value defined by the relative absolute error defines how much the estimate differs from the actual data in terms of ratio or percentage of the original matrix. For example, a MRE of 0.0 implies perfect estimation of all traffic matrix elements, that is without any error, while a MRE of 0.1 implies that the estimate deviates from the actual demand by 10%. In general, the closer the MRE value is to 0, the more accurate the estimated traffic matrix is. However, when the mean of the errors are considered, the result may not give a proper perspective of the errors when a few large errors dominate the mean. Thus, an additional measure, such as the standard deviation, the coefficient of variation or a probability distribution plot may be required to effectively characterize the errors in such cases.

Given $K$ sample TMs of a network, each containing $c$ OD pairs, we define the error $\epsilon_j^{(k)}$ in estimating $j^{th}$ OD pair of traffic matrix sample $k$ as

$$\epsilon_j^{(k)} = \left| \frac{X_j^{(k)} - \hat{X}_j^{(k)}}{X_j^{(k)}} \right| \tag{3.1}$$

where $X_j^{(k)}$ is the actual value and $\hat{X}_j^{(k)}$ is the estimated value of OD pair $j$. $\epsilon_j^{(k)}$ represents the absolute value of the relative error. In evaluating each technique, we primarily use the mean of the relative error,MRE, denoted as $\epsilon_\mu$, calculated over the entire sample. We define

$$MRE = \epsilon_\mu = \frac{1}{cK} \sum_{k=1}^{K} \sum_{j=1}^{c} \left| \frac{X_j^{(k)} - \hat{X}_j^{(k)}}{X_j^{(k)}} \right| \tag{3.2}$$

## 3.2 Evaluation Methodology

For each simulation, we compute the link loads from the real traffic matrix and the routing matrix using equation 2.1. The link load and routing matrix formed the input to the traffic matrix estimation process. The output of the estimation process is then evaluated against the original traffic matrix using the performance measure discussed in section 3.1.4.

We conducted our evaluation of the traffic matrix estimation techniques through four main comparative analysis as follows:

- Comparison of Gravity and WCB initial estimates
- Comparison of 5 TM estimation techniques using both gravity and WCB initial estimates
- Comparison of hybrid techniques of two each of the 5 TM estimation techniques
- Comparison of 5 TM estimation techniques using a previously measured TM sample

### 3.2.1 Comparison of Gravity and WCB Initial Estimates

We compared two choices of initial starting point or prior distribution of the traffic matrix - the gravity and the worst-case bound. We evaluated each model on the 3 topologies using all the 24 samples in our data set.

### 3.2.2 Comparison of 5 Estimation Techniques using Gravity and WCB Initial Estimates

We evaluated the performance of 5 techniques - TG, EM, QP, LP and NN - using the gravity prior distribution. In evaluating the TG, EM and QP techniques, the gravity and WCB estimates served as initial solutions, whereas the LP technique utilized them as starting points. NN technique employs these estimates as training data sets for the artificial neural network model.

### 3.2.3 Comparison of 5 Hybrid Techniques

We evaluated a hybrid approach to traffic matrix estimation in which the estimate of one technique serves as a prior estimate for another technique, with the goal of further driving down the estimation errors. Soule *et. al* [14] have noted that some ISPs have indicated that they would not use traffic matrices whose errors are above 10% mark (corresponding to an MRE of 0.1) for traffic engineering purposes. Although, it is not clear what performance measures are desired, we believe the MRE used in this work provides a reasonable measure.

EM hybrid techniques leverage on the estimates of the TG, QP, LP and NN techniques as initial estimate in the original Entropy Maximization technique. We expect an improvement in the overall estimation since these new initial estimates are better than the gravity model estimates. In the QP hybrid techniques, the gravity model estimates is replaced with the estimates obtained using the TG, EM, LP and NN techniques.

We re-evaluated the linear programming technique using the estimates from TG, EM, QP and NN as starting points to obtain what we called the LP hybrid techniques. In NN hybrid techniques, we investigated the benefit, if any, of training the artificial neural network model with TG, EM and QP and LP estimates and then using the trained network to estimate the traffic demands from link loads.

### 3.2.4 Comparison of Traffic Matrix Estimation Techniques using Previous Demand Measurements

Finally, we explored the effect of using a known traffic matrix to estimate current and future traffic matrices from link load data. This is a technique that has been widely adopted in road transport traffic forecasting where a sample of today's traffic is adjusted to estimate the full matrix for the day. This estimate is then used to predict future traffic demands. This is different from the technique investigated by other authors such as [7–9] where some demands are measured and incorporated into the traffic matrix estimation process to make the system of equations less under-constrained and thus obtain better estimates. Whereas, in their own case, known demands is combined with link load measurements from the same period to estimate traffic matrix, here, we use complete measured demands or its estimates from a different period to predict future estimates given the link loads. We considered three possible uses of the previously measured demands - raw data, fanout from raw data and iterative proportionally-fitted data.

In the raw-data approach, we use the first sample in our 24-sample data set as an initial estimate in estimating the other 23 samples (weeks) of traffic matrix. We know that this initial estimate is not consistent with current link loads, in the same way the gravity model estimate is inconsistent with network link loads. Our goal is to assess how a knowledge of past demands affects current demands or its estimate.

The fanout approach evaluated here computes fanout estimate from the hourly data of sample 1. This is different from the constant fanout model described in Section 2.3.3. The goal is to ascertain if fanouts are constant over a long range of time, in which case, it should translate into more accurate estimation if known. In our simulation, we first compute the fanout estimate using the fanout factor computed from sample 1. The fanout factor is then combined with current edge link load (production) of each node to determine a fanout-based estimate of current demand. The estimated demand served as initial estimate to the 5 techniques being evaluated.

In the iterative proportionally-fitted estimation approach, first, we adjusted the previously measured traffic matrix (Sample 1) to current link loads (Samples 2 - 23) using iterative proportional fitting. Note that this procedure produces estimates that are consistent with current link loads. The samples can be fitted to either the core/backbone link loads or to both core and edge link loads. Then, the adjusted is used as initial estimates in each of the 5 techniques evaluated.

Most of the results of our comparisons are presented in Chapter 4. The remaining results can be found in Appendix B.

# Chapter 4

# Numerical Results

## 4.1 Comparison of Gravity and WCB Initial Estimates

Table 4.1 compares the mean relative error for the gravity and WCB estimate for the 4-node network. The WCB model was more accurate in estimating the TM for the 4-node network, yielding an average MRE of 0.48 over the 24 samples (weeks) of hourly data used, than the gravity model with an MRE of 7.99.

This may not be unexpected, due to the fact that the technique is based on linear programming, which is well known to estimate traffic matrix of small networks with relatively high accuracy and the fact that it utilizes the link load constraint information. Furthermore, by computing the MRE over only OD pairs that account for 95% and 90% of total network traffic, the error drops significantly to an average of 0.08 and 0.05

Table 4.1: MRE of Gravity and WCB Estimate for 4-Node Network

| Sample | Gravity | WCB | Gravity-95 | WCB-95 | Gravity-90 | WCB-90 |
|---|---|---|---|---|---|---|
| 1 | 0.92 | 0.18 | 0.59 | 0.09 | 0.57 | 0.05 |
| 2 | 4.76 | 0.23 | 0.47 | 0.09 | 0.44 | 0.05 |
| 3 | 60.33 | 2.38 | 0.56 | 0.07 | 0.54 | 0.03 |
| 4 | 34.18 | 0.67 | 0.44 | 0.07 | 0.42 | 0.01 |
| 5 | 47.50 | 0.72 | 0.58 | 0.06 | 0.55 | 0.01 |
| 6 | 0.87 | 0.22 | 0.34 | 0.05 | 0.26 | 0.04 |
| 7 | 1.13 | 0.41 | 0.35 | 0.09 | 0.32 | 0.06 |
| 8 | 1.98 | 0.47 | 0.35 | 0.06 | 0.30 | 0.04 |
| 9 | 3.24 | 0.35 | 0.36 | 0.06 | 0.34 | 0.04 |
| 10 | 4.44 | 0.30 | 0.36 | 0.05 | 0.33 | 0.02 |
| 11 | 3.52 | 0.48 | 0.35 | 0.07 | 0.35 | 0.04 |
| 12 | 7.08 | 0.32 | 0.35 | 0.06 | 0.35 | 0.03 |
| 13 | 2.31 | 0.30 | 0.44 | 0.08 | 0.40 | 0.05 |
| 14 | 1.56 | 0.43 | 0.40 | 0.09 | 0.36 | 0.05 |
| 15 | 1.71 | 0.31 | 0.41 | 0.07 | 0.35 | 0.04 |
| 16 | 1.42 | 0.26 | 0.50 | 0.06 | 0.44 | 0.03 |
| 17 | 1.88 | 0.35 | 0.46 | 0.05 | 0.40 | 0.03 |
| 18 | 2.18 | 0.80 | 0.42 | 0.08 | 0.38 | 0.05 |
| 19 | 2.03 | 0.35 | 0.45 | 0.09 | 0.38 | 0.05 |
| 20 | 2.32 | 0.27 | 0.42 | 0.06 | 0.39 | 0.04 |
| 21 | 3.17 | 0.48 | 0.46 | 0.08 | 0.41 | 0.05 |
| 22 | 1.28 | 0.56 | 0.43 | 0.10 | 0.38 | 0.07 |
| 23 | 1.02 | 0.29 | 0.38 | 0.11 | 0.35 | 0.08 |
| 24 | 0.92 | 0.43 | 0.38 | 0.14 | 0.33 | 0.12 |
| **Mean** | **7.99** | **0.48** | **0.43** | **0.08** | **0.39** | **0.05** |

respectively. However, the gravity model outperformed the WCB in estimating the TM for 12-node and 14-node network as shown in Table 4.2 and 4.3. Although, both errors are large, on the average, the MRE computed using top 95% and 90% of OD flows shows a significant reduction in errors, indicating that small OD flows were the most poorly estimated by both techniques.

Table 4.2: MRE of Gravity and WCB Estimate for 12-Node Network

| Sample | Gravity | WCB | Gravity-95 | WCB-95 | Gravity-90 | WCB-90 |
|---|---|---|---|---|---|---|
| 1 | 1.28 | 6.40 | 0.45 | 1.04 | 0.40 | 0.73 |
| 2 | 7.70 | 12.40 | 0.47 | 0.95 | 0.43 | 0.70 |
| 3 | 91.55 | 46.43 | 0.54 | 0.34 | 0.49 | 0.30 |
| 4 | 77.03 | 33.32 | 0.66 | 0.36 | 0.60 | 0.31 |
| 5 | 51.61 | 39.63 | 0.45 | 0.37 | 0.41 | 0.32 |
| 6 | 1.67 | 8.32 | 0.63 | 0.82 | 0.56 | 0.59 |
| 7 | 5.14 | 23.76 | 0.53 | 0.84 | 0.49 | 0.62 |
| 8 | 3.64 | 22.45 | 0.58 | 0.82 | 0.53 | 0.62 |
| 9 | 4.45 | 23.11 | 0.55 | 0.88 | 0.49 | 0.66 |
| 10 | 4.74 | 24.52 | 0.55 | 0.87 | 0.48 | 0.66 |
| 11 | 4.49 | 29.89 | 0.54 | 0.88 | 0.48 | 0.67 |
| 12 | 4.57 | 25.52 | 0.58 | 0.91 | 0.53 | 0.69 |
| 13 | 5.00 | 27.30 | 0.58 | 0.94 | 0.53 | 0.70 |
| 14 | 8.33 | 35.46 | 0.53 | 1.00 | 0.49 | 0.77 |
| 15 | 8.49 | 37.93 | 0.55 | 1.04 | 0.52 | 0.79 |
| 16 | 8.04 | 33.03 | 0.52 | 1.07 | 0.48 | 0.80 |
| 17 | 10.36 | 39.28 | 0.51 | 1.09 | 0.47 | 0.80 |
| 18 | 9.50 | 36.71 | 0.50 | 1.09 | 0.45 | 0.79 |
| 19 | 9.47 | 36.54 | 0.57 | 1.05 | 0.52 | 0.78 |
| 20 | 15.35 | 47.39 | 0.55 | 1.09 | 0.49 | 0.82 |
| 21 | 18.51 | 65.34 | 0.56 | 1.14 | 0.49 | 0.84 |
| 22 | 12.30 | 51.12 | 0.55 | 1.14 | 0.48 | 0.82 |
| 23 | 5.86 | 32.83 | 0.61 | 1.08 | 0.57 | 0.81 |
| 24 | 6.48 | 111.00 | 0.50 | 0.98 | 0.47 | 0.75 |
| Mean | 15.65 | 35.40 | 0.54 | 0.91 | 0.49 | 0.68 |

## 4.2 Comparison of 5 Techniques with Gravity and WCB Initial Estimates

Tables 4.4, 4.5 and 4.6 show the MRE for the five techniques evaluated using the 4-node, 12-node and 14-node network topology respectively.

The EM technique performed best of all the 5 techniques compared, regardless of topol-

Table 4.3: MRE of Gravity and WCB Estimate for 14-Node Network

| **S**ample | **G**ravity | **W**CB | **G**ravity-95 | **W**CB-95 | **G**ravity-90 | **W**CB-90 |
|---|---|---|---|---|---|---|
| 1 | 16.76 | 42.14 | 0.62 | 1.77 | 0.51 | 1.14 |
| 2 | 77.45 | 171.67 | 0.60 | 1.57 | 0.51 | 0.96 |
| 3 | 374.92 | 730.29 | 0.62 | 0.81 | 0.58 | 0.61 |
| 4 | 355.14 | 588.15 | 0.74 | 1.09 | 0.66 | 0.80 |
| 5 | 370.36 | 765.65 | 0.55 | 0.90 | 0.53 | 0.59 |
| 6 | 28.44 | 72.72 | 0.87 | 1.72 | 0.61 | 1.01 |
| 7 | 98.86 | 232.25 | 0.70 | 1.54 | 0.56 | 0.93 |
| 8 | 102.40 | 254.44 | 0.81 | 1.68 | 0.60 | 1.03 |
| 9 | 105.92 | 278.36 | 0.78 | 1.52 | 0.61 | 1.01 |
| 10 | 112.00 | 279.56 | 0.67 | 1.39 | 0.57 | 0.94 |
| 11 | 98.46 | 247.25 | 0.64 | 1.46 | 0.54 | 0.94 |
| 12 | 90.43 | 244.98 | 0.66 | 1.46 | 0.57 | 0.95 |
| 13 | 90.17 | 233.33 | 0.66 | 1.44 | 0.57 | 1.00 |
| 14 | 81.73 | 250.04 | 0.65 | 1.47 | 0.55 | 0.98 |
| 15 | 98.59 | 274.27 | 0.65 | 1.48 | 0.57 | 0.99 |
| 16 | 70.46 | 207.26 | 0.64 | 1.48 | 0.54 | 1.00 |
| 17 | 84.11 | 270.39 | 0.61 | 1.43 | 0.53 | 1.01 |
| 18 | 82.96 | 230.85 | 0.62 | 1.45 | 0.53 | 0.98 |
| 19 | 117.37 | 287.88 | 0.69 | 1.41 | 0.59 | 0.93 |
| 20 | 102.33 | 295.42 | 0.64 | 1.41 | 0.53 | 0.93 |
| 21 | 106.16 | 326.42 | 0.66 | 1.45 | 0.55 | 0.94 |
| 22 | 71.69 | 213.53 | 0.68 | 1.55 | 0.55 | 1.04 |
| 23 | 71.67 | 141.26 | 0.65 | 1.55 | 0.56 | 1.12 |
| 24 | 62.66 | 239.83 | 0.59 | 1.53 | 0.50 | 1.05 |
| **M**ean | **1**19.63 | **2**86.58 | **0**.67 | **1**.44 | **0**.56 | **0**.95 |

ogy. It is however interesting to know that, the errors for the 12-node and 14-node were much higher than those of the 4-node network. While the errors for the 4-node network were typically between 0.28 and 0.49, the best performing technique, EM, had an average error of 8.02 and 69.10 for the 12-node and 14-node networks respectively.

We therefore plot the empirical cumulative distribution (ECDF) of one of the samples

Table 4.4: MRE of 5 Techniques using Gravity Prior Distribution for 4-Node Network

| **S**ample | **T**G | **E**M | **Q**P | **L**P2 | **L**P1 | **N**N |
|---|---|---|---|---|---|---|
| 1 | 0.32 | 0.08 | 0.11 | 0.67 | 0.76 | 0.28 |
| 2 | 0.33 | 0.11 | 0.13 | 1.65 | 2.31 | 0.34 |
| 3 | 1.37 | 1.01 | 0.81 | 27.04 | 38.14 | 2.72 |
| 4 | 1.27 | 2.09 | 2.13 | 3.40 | 1.69 | 1.29 |
| 5 | 1.52 | 0.62 | 0.50 | 1.96 | 2.17 | 1.43 |
| 6 | 0.24 | 0.10 | 0.11 | 0.93 | 0.94 | 0.40 |
| 7 | 0.30 | 0.16 | 0.24 | 0.97 | 1.05 | 0.22 |
| 8 | 0.24 | 0.14 | 0.21 | 1.44 | 1.07 | 0.30 |
| 9 | 0.32 | 0.13 | 0.15 | 1.42 | 0.90 | 0.39 |
| 10 | 0.26 | 0.14 | 0.15 | 0.74 | 0.92 | 0.28 |
| 11 | 0.23 | 0.14 | 0.15 | 1.02 | 1.17 | 0.29 |
| 12 | 0.31 | 0.19 | 0.22 | 0.74 | 0.89 | 0.30 |
| 13 | 0.32 | 0.14 | 0.18 | 1.58 | 1.97 | 0.32 |
| 14 | 0.34 | 0.13 | 0.15 | 1.30 | 1.48 | 0.26 |
| 15 | 0.34 | 0.17 | 0.19 | 0.80 | 0.78 | 0.29 |
| 16 | 0.35 | 0.13 | 0.13 | 0.79 | 0.78 | 0.32 |
| 17 | 0.34 | 0.15 | 0.16 | 0.98 | 0.98 | 0.27 |
| 18 | 0.26 | 0.18 | 0.13 | 1.86 | 1.86 | 1.84 |
| 19 | 0.40 | 0.16 | 0.19 | 0.83 | 0.80 | 0.22 |
| 20 | 0.30 | 0.15 | 0.18 | 0.79 | 0.76 | 0.40 |
| 21 | 0.32 | 0.21 | 0.20 | 1.45 | 1.72 | 1.07 |
| 22 | 0.28 | 0.20 | 0.19 | 1.37 | 1.44 | 0.33 |
| 23 | 0.20 | 0.11 | 0.12 | 0.85 | 1.09 | 0.32 |
| 24 | 0.20 | 0.11 | 0.11 | 0.97 | 1.23 | 0.20 |
| **M**ean | **0**.43 | **0**.28 | **0**.29 | **2**.31 | **2**.79 | **0**.59 |

(sample 1) for the EM technique to understand the distribution of the errors. Figure 4.1 shows the plot of ECDF of MRE for EM estimation of sample1 using the three network topologies.

The error distribution appeared to have a heavy tail, especially for the 4-node and 12-node network, where more than 80% of the errors were well below an MRE value of 1,

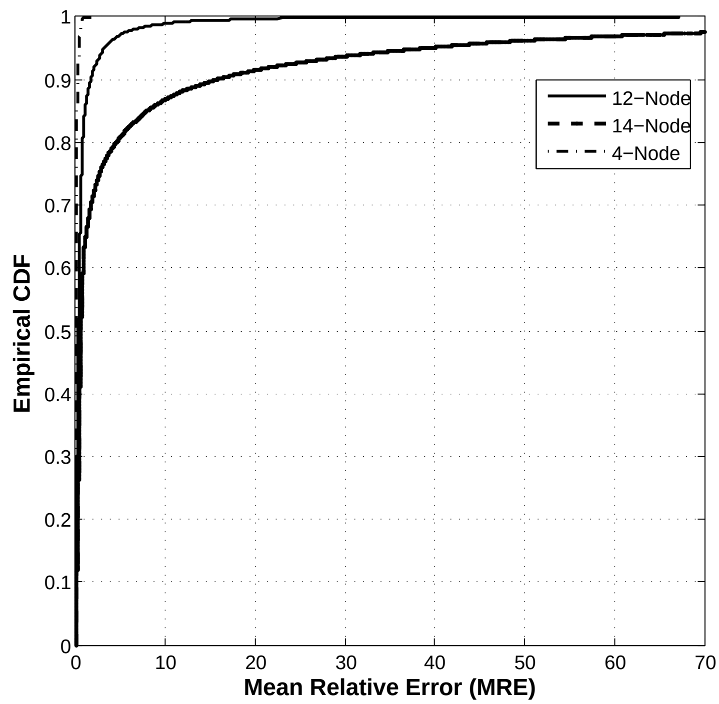


Figure 4.1: Empirical CDF of MRE for Entropy Technique with Data Sample 1

Table 4.5: MRE of 5 Techniques using Gravity Prior Distribution for 12-Node Network

| **S**ample | **T**G | **E**M | **Q**P | **L**P2 | **L**P1 | **N**N |
|---|---|---|---|---|---|---|
| 1 | 0.81 | 0.86 | 0.94 | 1.56 | 1.39 | 1.20 |
| 2 | 1.89 | 2.32 | 2.91 | 16.10 | 18.04 | 1.72 |
| 3 | 23.40 | 25.07 | 24.41 | 114.11 | 101.06 | 16.44 |
| 4 | 27.48 | 27.54 | 44.53 | 91.33 | 94.67 | 20.20 |
| 5 | 16.84 | 16.26 | 10.59 | 102.43 | 87.53 | 10.26 |
| 6 | 1.16 | 1.21 | 1.57 | 1.91 | 1.64 | 1.02 |
| 7 | 3.58 | 3.63 | 4.12 | 7.12 | 8.72 | 4.79 |
| 8 | 3.19 | 3.26 | 4.48 | 5.36 | 7.38 | 3.84 |
| 9 | 3.31 | 3.49 | 4.49 | 5.80 | 5.79 | 3.14 |
| 10 | 4.05 | 4.24 | 5.83 | 7.32 | 6.57 | 2.45 |
| 11 | 3.85 | 3.98 | 6.08 | 7.65 | 7.08 | 4.20 |
| 12 | 3.40 | 3.50 | 4.38 | 6.60 | 5.88 | 2.80 |
| 13 | 3.84 | 3.96 | 4.37 | 6.82 | 4.73 | 3.14 |
| 14 | 6.89 | 6.80 | 8.07 | 8.80 | 7.33 | 12.13 |
| 15 | 7.87 | 7.67 | 9.41 | 10.73 | 10.12 | 6.23 |
| 16 | 6.66 | 6.25 | 6.48 | 7.42 | 5.45 | 5.89 |
| 17 | 11.08 | 9.06 | 10.27 | 8.33 | 4.79 | 7.73 |
| 18 | 10.06 | 8.53 | 9.74 | 7.72 | 5.76 | 8.26 |
| 19 | 8.22 | 7.75 | 8.54 | 8.56 | 7.33 | 27.91 |
| 20 | 12.81 | 12.13 | 12.97 | 12.01 | 10.41 | 14.16 |
| 21 | 13.99 | 13.63 | 13.32 | 15.49 | 10.24 | 11.64 |
| 22 | 10.14 | 9.65 | 8.94 | 11.24 | 6.61 | 12.76 |
| 23 | 5.46 | 5.28 | 5.48 | 9.06 | 3.66 | 4.89 |
| 24 | 6.28 | 6.39 | 22.18 | 201.17 | 4.08 | 33.58 |
| **M**ean | **8**.18 | **8**.02 | **9**.75 | **2**8.11 | **1**7.76 | **9**.18 |

which is far less than the average error shown earlier in tables 4.5 and 4.6. Consequently, we would endeavour to report the errors for the top 90% and 95% of OD flows in subsequent analysis, as a means of checking the distribution of errors in the estimate and evaluating the errors of large OD flows in the traffic matrix.

Table 4.7 gives a summary of the errors for the 5 techniques over the entire data set

Table 4.6: MRE of 5 Techniques using Gravity Prior Distribution for 14-Node Network

| **S**ample | **T**G | **E**M | **Q**P | **L**P2 | **L**P1 | **N**N |
|---|---|---|---|---|---|---|
| 1 | 9.22 | 8.17 | 7.56 | 7350.95 | 16.30 | 10.74 |
| 2 | 45.59 | 41.53 | 42.32 | 65.25 | 55.13 | 50.72 |
| 3 | 278.20 | 249.83 | 300.04 | 299.68 | 312.39 | 287.64 |
| 4 | 222.35 | 209.98 | 253.60 | 273.74 | 233.25 | 278.44 |
| 5 | 277.82 | 260.24 | 295.20 | 1955.82 | 266.85 | 254.91 |
| 6 | 19.51 | 15.40 | 15.49 | 18.83 | 18.14 | 19.81 |
| 7 | 58.70 | 49.70 | 57.14 | 47.96 | 44.46 | 74.97 |
| 8 | 63.56 | 48.93 | 50.27 | 59.31 | 55.93 | 66.16 |
| 9 | 71.82 | 56.07 | 56.78 | 3436.72 | 50.47 | 62.16 |
| 10 | 75.38 | 59.60 | 64.35 | 61.74 | 54.85 | 63.23 |
| 11 | 62.06 | 47.73 | 51.39 | 51.41 | 40.42 | 109.96 |
| 12 | 57.75 | 45.64 | 49.34 | 46.63 | 53.12 | 67.51 |
| 13 | 54.93 | 43.12 | 46.18 | 52.41 | 64.77 | 37.92 |
| 14 | 57.14 | 45.31 | 44.87 | 41.10 | 47.11 | 59.82 |
| 15 | 64.75 | 51.64 | 58.00 | 45.35 | 46.46 | 70.57 |
| 16 | 46.86 | 38.52 | 40.99 | 2693.57 | 39.43 | 72.22 |
| 17 | 62.42 | 50.26 | 50.26 | 1558.53 | 45.53 | 78.61 |
| 18 | 55.80 | 45.23 | 45.96 | 39.31 | 35.17 | 41.91 |
| 19 | 70.47 | 53.61 | 58.87 | 58.83 | 53.87 | 64.11 |
| 20 | 77.11 | 56.65 | 61.56 | 60.64 | 56.38 | 66.14 |
| 21 | 86.51 | 65.69 | 68.76 | 65.96 | 57.50 | 97.79 |
| 22 | 50.89 | 41.77 | 40.05 | 39.85 | 34.16 | 42.36 |
| 23 | 35.05 | 29.21 | 29.89 | 32.53 | 26.57 | 31.98 |
| 24 | 50.61 | 44.66 | 50.55 | 908.68 | 435.12 | 49.88 |
| **M**ean | **8**1.44 | **6**9.10 | **7**6.64 | **8**02.70 | **8**9.31 | **8**5.82 |

of 24 samples using gravity prior distribution. The EM technique produced the best estimate resulting in MRE values 0.01, 0.32 and 0.49 for the 4-node, 12-node and 14-node network topologies respectively using the top 90% OD flows. TG and QP have errors that were slightly higher but much better than the rest of the techniques.

Table B-1 provides a summary of the result for each of the network topology. The

Table 4.7: MRE of 5 Techniques using Gravity Prior Distribution

| | **T**G | **E**M | **Q**P | **L**P1 | **L**P2 | **N**N |
|---|---|---|---|---|---|---|
| **4**-Node | 0.43 | 0.28 | 0.29 | 2.31 | 2.79 | 0.59 |
| **1**2-Node | 8.18 | 8.02 | 9.75 | 28.11 | 17.76 | 9.18 |
| **1**4-Node | 81.44 | 69.10 | 76.64 | 802.70 | 89.31 | 85.82 |
| | | | | | | |
| **T**op 95% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P1 | **L**P2 | **N**N |
| **4**-Node | 0.07 | 0.02 | 0.03 | 0.48 | 0.50 | 0.11 |
| **1**2-Node | 0.39 | 0.37 | 0.46 | 0.53 | 0.60 | 2.61 |
| **1**4-Node | 0.63 | 0.57 | 0.72 | 163.36 | 1.16 | 0.85 |
| | | | | | | |
| **T**op 90% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P1 | **L**P2 | **N**N |
| **4**-Node | 0.04 | 0.01 | 0.02 | 0.45 | 0.44 | 0.07 |
| **1**2-Node | 0.34 | 0.32 | 0.39 | 0.49 | 0.51 | 3.03 |
| **1**4-Node | 0.52 | 0.49 | 0.59 | 139.11 | 0.98 | 0.71 |

summarized MRE is based on all the 24 samples. Generally, the WCB model results in slightly higher MRE than the gravity model for most of the techniques and the different topologies. The only exception being the NN technique which estimates the 4-node network better using the WCB initial estimate than using gravity estimate. The other techniques performed equally well or worse using the WCB prior than the gravity prior. LP with gravity prior also performed much better than the original LP. This may be due to the fact that the approach used here is not the classical linear programming approach but goal programming and the results may not be exact solutions but approximate.

## 4.3 Comparison of Hybrid Techniques

Table 4.8 presents the result of comparing the MRE of TG hybrid techniques. In comparison with the original tomogravity technique, use of estimates from other techniques as starting point yielded only marginal decrease in error, with the exception of neural network. Computing the error over top 90% demands reveals an interesting result - the marginal improvements of using these techniques as prior had been lost, except in the case of the EM technique. This implies that the improvement in performance was due to better estimation of small OD flows, which originally had high errors, at the expense of producing worst estimates of some large OD flows.

Table 4.8: MRE of Tomogravity Hybrid Technique using EM, QP, LP and NN as Prior Estimates

| | TG | TG-EM | TG-QP | TG-LP2 | TG-LP1 | TG-NN |
|---|---|---|---|---|---|---|
| 4-Node | 0.43 | 0.19 | 0.19 | 0.64 | 1.25 | 0.62 |
| 12-Node | 8.18 | 7.50 | 7.97 | 10.44 | 8.19 | 837.80 |
| 14-Node | 81.44 | 68.86 | 74.88 | 65.48 | 63.47 | 85.03 |
| | | | | | | |
| Top 95% of Demands | | | | | | |
| | TG | TG-EM | TG-QP | TG-LP2 | TG-LP1 | TG-NN |
| 4-Node | 0.07 | 0.02 | 0.03 | 0.11 | 0.16 | 0.11 |
| 12-Node | 0.39 | 0.37 | 0.46 | 0.49 | 0.55 | 86.58 |
| 14-Node | 0.63 | 0.57 | 0.71 | 0.85 | 1.11 | 0.86 |
| | | | | | | |
| Top 90% of Demands | | | | | | |
| | TG | TG-EM | TG-QP | TG-LP2 | TG-LP1 | TG-NN |
| 4-Node | 0.04 | 0.01 | 0.02 | 0.07 | 0.10 | 0.07 |
| 12-Node | 0.34 | 0.32 | 0.39 | 0.49 | 0.51 | 3.03 |
| 14-Node | 0.52 | 0.49 | 0.59 | 0.73 | 0.95 | 0.72 |

Table B-2 shows the result of the hybrid EM technique. None of the techniques produced a better estimate than the original EM. This may be due to the regularization parameter, $\lambda$. There may be need to adjust this parameter to put more weight on the prior estimate in the computation.

The result of the QP hybrid techniques shown in Table B-3 shows that the EM technique consistently resulted in better estimates, especially for the top 90-95% OD flows as well the 12- and 14-node network topologies. TG only produced better estimate consistently for the 14-node network. All other techniques resulted in worse estimates of the TM when combined with the QP technique.

Table B-4 summarizes the result of the LP hybrid techniques. None of the techniques could improve the result of estimation of the 4-node network using the LP technique, confirming that LP is best at estimating TM for small networks. There were significant reduction in errors for the 12 and 14 node networks by all the other technique, however, the overall error is still much higher than those achieved by those techniques individually, especially the EM, TG and QP techniques.

Table B-5 shows the average error of the final estimate obtained using the 24 samples of data. Only the EM technique consistently produced better overall estimates when used to train the neural network model. The TG resulted in better estimation of large OD flows and networks only, while all other techniques produced worse estimate than the gravity model used earlier.

## 4.4 Comparison of Traffic Matrix Estimation Techniques using Previous Demand Measurements

Table 4.9 shows that all the techniques, except LP improved their estimates using a previously measured matrix. MRE for both EM and TG decreased by approximately 81%, while NN and QP experienced a reduction in overall error of 75% and 43% respectively. Contrariwise, the LP produced worse estimates with the known demands. Furthermore, these gains appeared to have greater impact on smaller OD flows as the top 90% of OD flows only witnessed a maximum of 43% reduction in average error using the EM technique - which appeared to benefit most from the previous measurement.

Table 4.10 shows that the fanout-estimate itself performed poorly in the estimation of the 4-node network demands, consequently, all the techniques performed worse by using it as prior instead of the gravity prior estimates. The converse is true with the 12-node and 14-node network, where similar reduction in average errors as in the case of using the raw estimates were obtained.

Table 4.11 compares the performance of the 5 techniques using the proportionally-fitted data as prior estimate. Both the EM and NN consistently produced better estimates across the three topologies and over all demands; the estimates for the 4-node network

Table 4.9: MRE of Estimation using Past Measurement as Initial Estimate

| | **T**G | **E**M | **Q**P | **L**P2 | **N**N |
|---|---|---|---|---|---|
| **4**-Node | 2.46 | 0.26 | 0.42 | 2.54 | 0.92 |
| **1**2-Node | 6.19 | 4.12 | 8.50 | 12.72 | 6.39 |
| **1**4-Node | 15.31 | 13.13 | 43.98 | 305.41 | 21.41 |
| | | | | | |
| **T**op 95% of Demands | | | | | |
| | **T**G | **E**M | **Q**P | **L**P2 | **N**N |
| **4**-Node | 0.08 | 0.02 | 0.03 | 0.50 | 0.14 |
| **1**2-Node | 0.49 | 0.32 | 0.49 | 0.62 | 0.75 |
| **1**4-Node | 0.45 | 0.33 | 0.53 | 1.19 | 0.65 |
| | | | | | |
| **T**op 90% of Demands | | | | | |
| | **T**G | **E**M | **Q**P | **L**P2 | **N**N |
| **4**-Node | 0.05 | 0.01 | 0.02 | 0.43 | 0.10 |
| **1**2-Node | 0.43 | 0.28 | 0.42 | 0.54 | 0.66 |
| **1**4-Node | 0.38 | 0.28 | 0.44 | 1.02 | 0.55 |

were worse using the TG, QP and LP techniques. Over the 12 and 14-node network, all the techniques seemed to produce better estimates comparable to those obtained using the raw data or fanout of previous measurement. Although the errors of the NN technique are higher than those of EM, TG and QP techniques, the proportionally-fitted estimate provided the best means of training the network, compared to using the raw data or fanout estimates. Note that, using the iterative proportionally-fitted estimate with tomogravity did not result in any improvement, which shows that the initial estimate is very good.

Table 4.10: MRE of Estimation using Fanout Estimate of Previous Measurement

| | **T**G | **E**M | **Q**P | **L**P2 | **N**N | **F**O |
|---|---|---|---|---|---|---|
| **4**-Node | 2.46 | 0.26 | 0.42 | 2.54 | 0.79 | 9.39 |
| **1**2-Node | 3.94 | 4.15 | 9.07 | 15.18 | 5.33 | 24.96 |
| **1**4-Node | 16.42 | 13.13 | 44.57 | 134.83 | 23.32 | 16.95 |
| | | | | | | |
| **T**op 95% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P2 | **N**N | **F**O |
| **4**-Node | 0.08 | 0.02 | 0.03 | 0.49 | 0.17 | 0.89 |
| **1**2-Node | 0.37 | 0.32 | 0.43 | 0.58 | 1.21 | 0.45 |
| **1**4-Node | 0.44 | 0.33 | 0.54 | 1.20 | 0.70 | 0.66 |
| | | | | | | |
| **T**op 90% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P2 | **N**N | **F**O |
| **4**-Node | 0.05 | 0.01 | 0.02 | 0.43 | 0.11 | 0.84 |
| **1**2-Node | 0.33 | 0.28 | 0.37 | 0.50 | 1.29 | 0.40 |
| **1**4-Node | 0.37 | 0.28 | 0.45 | 1.02 | 0.59 | 0.60 |

Table 4.11: MRE of Estimation using IPF Estimate of Previous Measurement

| | **T**G | **E**M | **Q**P | **L**P2 | **N**N | **P**F |
|---|---|---|---|---|---|---|
| **4**-Node | 0.64 | 0.22 | 0.31 | 2.66 | 0.31 | 0.64 |
| **1**2-Node | 4.91 | 3.63 | 8.28 | 17.80 | 13.72 | 4.91 |
| **1**4-Node | 14.49 | 14.38 | 25.89 | 94.59 | 21.28 | 14.49 |
| | | | | | | |
| **T**op 95% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P2 | **N**N | **P**F |
| **4**-Node | 0.08 | 0.02 | 0.02 | 0.50 | 0.05 | 0.08 |
| **1**2-Node | 0.44 | 0.32 | 0.39 | 0.57 | 0.52 | 0.44 |
| **1**4-Node | 0.42 | 0.35 | 0.39 | 1.15 | 0.48 | 0.42 |
| | | | | | | |
| **T**op 90% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P2 | **N**N | **P**F |
| **4**-Node | 0.04 | 0.01 | 0.01 | 0.43 | 0.03 | 0.04 |
| **1**2-Node | 0.39 | 0.28 | 0.34 | 0.49 | 0.47 | 0.39 |
| **1**4-Node | 0.36 | 0.29 | 0.32 | 0.98 | 0.41 | 0.36 |

# Chapter 5

# Conclusion and Recommendations

## 5.1 Conclusion

Traffic Matrix of IP networks is a vital information required by network providers for various traffic engineering tasks. TM estimation from link loads is preferred on large-scale IP network because of the huge overhead of direct measurement. In this thesis, we have evaluated five important traffic matrix estimation techniques namely, tomogravity (TG), entropy maximization (EM), quadratic programming (QP), linear programming(LP) and artificial neural network (NN) using three topologies and real Internet traffic data. We conclude that EM technique is the best among these techniques as it performs consistently well on both small and large networks. We also found TG and QP techniques to produce good estimates, thought with slightly higher MRE values than the EM technique. The LP technique is only appropriate for small networks, because the MRE value of its estimates on the 12 and 14 node networks were so high that, sometimes, we had to round them off to only 4 significant values for the purpose of comparison with other techniques. We found out that a more accurate initial estimate than the gravity and WCB estimates, like those obtained from previous measurement or flow samples is required to train the artificial neural network model in order to generate estimates with reasonable accuracy on both small and large-scale IP networks.

We recommend that ISPs choose the EM over other techniques in performing large-scale IP traffic estimation. Our results on the use of past measured demands provides insight into the value of such measurements. We therefore recommend the use of available tools on routers to obtain a sampled traffic matrix or fanout, which could be adjusted to link load measurements using IPF in order to obtain an initial estimate of the TM. Sampling interval can be set in such a way that the processing and computational overhead is minimal. This would provide a better prior estimate for any TM estimation technique, compared to the gravity and WCB prior estimates, thus reducing the error in estimation.

## 5.2 Summary of Contributions

In this work, we have shown that, of all the techniques that we evaluated, the EM technique is the best and most robust technique for estimating traffic matrix of large-scale IP networks. We also showed that, if ISPs can invest a considerable effort and money into measuring of traffic matrix once, the overhead of continuous measurement can be avoided by using simple iterative proportional fitting procedure to estimate future demand from link loads with a guaranteed reduction of up to 80% in mean relative error, compared to just applying the techniques without any reasonable prior information about the OD flows. However, we observed that, achieving an upper bound of 10% on the estimated demand may be difficult, if not impossible to achieve using this technique or any other technique that we evaluated. We have also shown that, given a previous measurement or sampled flow, the best way to use this data for traffic estimation using artificial neural network is to first iteratively fit the data to link loads and use the resultant traffic matrix to train the network.

## 5.3 Proposal for Future Work

We have found the EM technique to be very accurate in estimating traffic matrix of large network. However, it would be interesting to investigate how to determine the optimal parameter of the regularization parameter based on the initial distribution. The authors have recommended a value of 0.01 which we utilized in this work, but when the initial estimate is more accurate than the gravity model provides, a slightly higher value may produce better estimate.

We would also like to evaluate other neural network models to see if they produce better TM estimates than the basic feedforward backpropagation model evaluated in this work. Currently, no author has compared these models in terms of their accuracy in measuring IP traffic matrices. Further research would also be needed to determine the optimal parameters as well as training and learning algorithms for neural network models employed in traffic matrix estimation large-scale IP networks.

Finally, it would be interesting to find better methods of utilizing past demands or fairly accurate estimates of past demands in predicting future demand other than the three approaches we have adopted here.

## 5.4 Relevance of Thesis to Engineering

One of the goals of engineering is to apply employ theoretical principles, mathematical techniques and scientific methods in the design, implementation and optimization of systems. This research is focused on a telecommunication system - a large-scale IP network operated by an Internet service provider. The objective of this thesis is to utilize readily available (SNMP link loads and network routing information) in providing a a non-existent information (the traffic matrix).

The information provided by the traffic matrix is critical for optimal design and management of IP networks, however, as it is many real life problems, there are constraints to acquiring this desired information. The constraints include the adverse effects of measurement on user traffic, such as network delay, packet losses and quality of service degradation. In addition, there are cost constraints to frequent upgrade and replacement of network equipments (both hardware and software) as well as links in order to overcome the difficulty of direct measurement. there are cost constraints to frequent upgrade and replacement of network equipments (both hardware and software) as well as links in order to overcome the difficulty of direct measurement.

Mathematically, there is no exact solution to an under-constrained system of linear equations. However, as engineers, we have proposed a technique in this research that significantly reduced the errors in estimating TM from link loads by combining existing optimization tools with sampling and extensive computer simulations. This approach has a minimal impact on network traffic and does not require costly network equipment upgrade or replacement.

# References

[1] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya and C. Diot. "Taxonomy of IP traffic matrices", In *Proceedings of Scalability and Traffic Control in IP Networks II*, SPIE vol. 4868, no. 1, pp. 200–213, Boston, MA, 2002.

[2] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates and Y. Zhang. "Experience in measuring backbone traffic variability: models, metrics, measurements and meaning". In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 91–92, Marseille, France, 2002.

[3] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. "Fast accurate computation of large-scale IP traffic matrices from link loads". In SIGMETRICS, pp. 206-213, San Diego, CA, June 2003.

[4] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. "An information theoretic approach to traffic matrix estimation. In *SIGCOMM 03:Proceedings of the 2003 conference on Applications, technologies,architectures, and protocols for computer communications*, pp. 301-312, Karlsruhe, Germany, August 2003.

[5] A. E. Conway and M. Li. "Estimating the traffic distribution matrix of a packet-switched network using aggregate traffic measurements." *Telecommunication Systems*, 24(1):9–28, 2003.

[6] Md. M. Rahman, S. Saha; U. Chengan, and A. S. Alfa. "IP Traffic Matrix Estimation Methods: Comparisons and Improvements", *IEEE International Conference on Communications*, 2006 (ICC '06), vol. 1, no., pp. 90–96, June 2006.

[7] A. Gunnar, M. Johansson, and T. Telkamp. "Traffic matrix estimation on a large ip backbone - a comparison on real data". In *SIGCOMM Internet Measurement Conference*, pages 149-160, Sicily, Italy, Oct. 2004.

[8] U. Chengan. "Traffic matrix estimation of an IP network", M.Sc Thesis, University of Manitoba, 2006.

[9] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. "Traffic matrix estimation: Existing techniques and new directions". In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer*, SIGCOMM 02, pp. 161–174, Pittsburg, USA, August 2002.

[10] Y. Vardi. "Network tomography: estimating source-destination traffic intensities from link data". *American Statistical Association*, 91:365-377, 1996.

[11] C. Tebaldi and M. West. "Bayesian inference on network traffic using link count data". *American Statistical Association*, 93:557-576, 1998.

[12] S. Vaton and J. S. Bedo. "Network Traffic Matrix: how can one learn the prior distributions from link counts only?" *IEEE Communication Society*, 93:2138-2141, 2004.

[13] K. Papagiannaki, N. Taft and A. Lakhina. "A distributed approach to measure IP Traffic Matrices. In *Proceedings of ACM Internet Measurement Conference*, IMC'04, October 25-27, 2004

[14] A. Soule, A. Nucci, R. Cruz, E. Leonardi and N. Taft. "How to identify and estimate the largest traffic matrix elements in a dynamic environment. In *Proceedings of the joint international conference on Measurement and modeling of computer systems*, , SIGMETRICS '04, pp. 73–84, New York, USA, 2004.

[15] J. Cao, D. Davis, S. V. Wiel, and B. Yu. "Time-varying network tomography". *American Statistical Association*, 95:1063-1075, 2000.

[16] I. Juva, S. Vaton, J. Virtamo, "Quick Traffic Matrix Estimation Based on Link Count Covariances", In *IEEE International Conference on Communications, 2006.* ICC '06. vol.2, no., pp.603-608, June 2006.

[17] L. Tan and X. Wang. "A novel method to estimate IP traffic matrix", In *IEEE Communication Letters*, vol. 11, no. 11, pp. 907-909, November 2007.

[18] D. Jiang, J. Chen, L. He and G. Hu, "A Fast Approach of Large-Scale IP Traffic Matrix Estimation". In *International Conference on Wireless Communications,*

*Networking and Mobile Computing, 2007*, WiCom 2007, pp. 1913-1916, 21-25 Sept. 2007.

[19] D. Jiang and G. Hu, “Large-Scale IP Traffic Matrix Estimation Based on Backpropagation Neural Network”, In *First International Conference on Intelligent Networks and Intelligent Systems, 2008*, ICINIS ’08. pp. 229-232, 1-3 Nov. 2008.

[20] D. Jiang and Guangmin Hu, “A Novel Approach to Large-Scale IP Traffic Matrix Estimation Based on RBF Neural Network”, In *4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008*, WiCOM ’08, pp. 1-4, 12-14 Oct. 2008.

[21] D. Jiang, G.Hu, “Large-Scale IP Traffic Matrix Estimation Based on the Recurrent Multilayer Perceptron Network”, In *IEEE International Conference on Communications, 2008*, ICC ’08, pp. 366-370, 19-23 May 2008.

[22] C. R. Rao and S. K. Mitra. *Generalized Inverse of Matrices and its Applications*, John Wiley & Sons Inc., New York, 1971.

[23] R.M. Pringle and A. A. Rayner. *Generalized Inverse Matrices with applications to statistics*, Charles Griffin & Company Limited, London, 1971.

[24] A. Nucci, A Sridharan and N. Taft. “The problem of synthetically generating IP traffic matrices: Initial Recommendations”. In *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, pp. 19 - 32, July, 2005.

[25] A. Nucci, R. Cruz, N. Taft and C. Diot, "Design of IGP Link Weight Changes for Estimation of Traffic Matrices, IEEE Infocom, Hong Kong, China, March 2004.

[26] M. Roughan. "Simplifying the synthesis of Internet traffic matrics", In *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 19 - 32, October, 2005.

[27] W. Edwards Deming, Frederick F. Stephan. "On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known". In *The Annals of Mathematical Statistics*, Vol. 11, No. 4, pp. 427-444, Dec., 1940.

[28] P. Norman. "Putting iterative proportional fitting on the researchers desk." Working Paper 99/3. School of Geography, University of Leeds, Leeds, 1999.

[29] H. Demuth, M. Beale and M. Hagan. *Neural Network Toolbox 6 User's Guide*, Version 6.0.1, Release 2008b, October 2008.

[30] MathWorks. *Optimization Toolbox 4 User's Guide*, Version 4.2, Release 2009a, March 2009.

[31] Cisco. *Netflow Services Solution Guide*, Cisco Systems, October 2004.

[32] M.A. Saunders and B. Kim. PDSCO: Primal-Dual Interior method for Separable Convex Objectives. `http://www.stanford.edu/group/SOL/software/pdsco`

[33] Abilene data: `http://www.cs.utexas.edu/~yzhang/research/AbileneTM/`.

[34] Internet2 Network: `http://www.internet2.edu/network/`.

# Appendix A:

# Generalized Inverse Approach

The problem of traffic matrix estimation can be viewed as an ill-posed linear inverse problem arising from inability to accurately compute the inverse of the rank-deficient routing matrix. A quick method of solving equation (2.1) would be to compute the generalized inverse of the routing matrix, $\mathbf{A}$, since the actual inverse does not exist. Given a real matrix $\mathbf{A}$ of arbitrary rank and order $m \times n$, the generalized inverse of $\mathbf{A}$ is an $n \times m$ matrix $\mathbf{G}$ such that $\mathbf{X} = \mathbf{GY}$ is a solution of the equation $\mathbf{AX} = \mathbf{Y}$ for any $\mathbf{Y}$ which makes the equation consistent [22]. Unfortunately, for the kind of routing matrix encountered in this problem, $rank(\mathbf{A}) \leq m < n$, consequently, there are so many matrices satisfying this property. However, all such generalized inverse matrix of $A$ must satisfy at least the first of the conditions.

$$\mathbf{AGA} = \mathbf{A} \tag{A-1}$$

$$\mathbf{GAG} = \mathbf{G} \tag{A-2}$$

$$(\mathbf{AG})^T = \mathbf{AG} \tag{A-3}$$

$$(\mathbf{GA})^T = \mathbf{GA} \tag{A-4}$$

A *one-condition* generalized inverse matrix, denoted as $G^1 - inverse$ satisfies only Equation A-1, while a *two-condition* generalized inverse matrices, denoted as $G^2 - inverse$ (also known as *reflexive inverse*) satisfies only Equations A-1 and A-2. A necessary and sufficient condition for $G$ to be a reflexive inverse of $A$ is that $rank(G) = rank(A)$. A *three-condition* generalized inverse matrices, denoted as $G^3 - inverse$ satisfies either Equations A-1, A-2 and A-3 or A-1, A-2 and A-4. The last class of generalized inverses, which is more widely used, is the Moore-Penrose inverse (also known as *the generalized inverse* or the *pseudo-inverse*) which satisfies all the four conditions. Unlike other generalized inverses, the pseudo-inverse can be uniquely determined by this property. Thus, for a given matrix $\mathbf{A}$, the pseudo-inverse, denoted as $\mathbf{G}^*$, satisfies all four conditions given by Equations (A-1, A-2, A-3 and A-4). Furthermore, $\mathbf{G}^*$ has the property that $\mathbf{G}^*\mathbf{Y}$ is the minimum norm least-squares solution of $\mathbf{AX} = \mathbf{Y}$ [22].

The general form of a generalized inverse is given by

$$\mathbf{A}^{-1} = \mathbf{G}^* + \mathbf{U} - \mathbf{G}^*\mathbf{AUAG}^* \tag{A-5}$$

where $\mathbf{G}$ represents the generalized inverse of $\mathbf{A}$ and $\mathbf{U}$ is any arbitrary $n \times m$ matrix. Given a generalized inverse, the general solution of the equation $\mathbf{AX} = \mathbf{Y}$ is given by

$$\mathbf{X} = \mathbf{GY} + \mathbf{(I - GA)Z} \tag{A-6}$$

where $\mathbf{Z}$ is an arbitrary vector [22], [23].

In addition to generalized inverses, one may also compute right and left inverses, denoted as $\mathbf{A}_R^{-1}$ and $\mathbf{A}_L^{-1}$ respectively, depending on the rank of the matrix. For a rectangular matrix $\mathbf{A}$ of dimension $m \times n$, if $rank(A) = m$, there exists a right inverse, $\mathbf{A}_R^{-1}$, of $\mathbf{A}$ which satisfies the property

$$\mathbf{A}\mathbf{A}_R^{-1} = \mathbf{I}_m \tag{A-7}$$

where $\mathbf{I}_m$ is the identity matrix of order $m$. Similarly, if $rank(A) = n$, then a left inverse $\mathbf{A}_L^{-1}$ of $\mathbf{A}$ exists satisfying the property

$$\mathbf{A}_L^{-1}\mathbf{A} = \mathbf{I}_n \tag{A-8}$$

where $\mathbf{I}_n$ is the identity matrix of order $n$. Clearly, right and left inverses exists only when the rank of the $m \times n$ matrix is either m or n and unless $m = n$, both inverses cannot exist. It is important to mention that most of the routing matrices encountered in IP traffic estimation problems are rectangular matrices with full row rank, hence the right inverse exists and can easily be computed.

We compared the estimates produced by the Moore-Penrose generalized inverse (MPInv) with those of other inverses namely the right inverse (RInv), two-condition generalized inverse (G2Inv) and three-condition generalized inverse (G3Inv). We also compared the result of using these inverses in the tomogravity (WLSE) technique with the original Moore-Penrose inverse. Our comparison is based on 10 different routing matrices generated from 10 topologies each of the 4-node and 14-node network used by the authors in [8].

Table A-1: MRE of Generalized Inverses for 10 topologies of 4-Node Network

| Top | RInv | G2Inv | G3Inv | MPInv | TG+RInv | TG+G2 | TG+G3 | TG+MPInv |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.88 | 0.88 | 0.88 | 0.88 | 1.37 | 1.00 | 44.92 | 1.37 |
| 2 | 112.23 | 112.23 | 48.93 | 112.23 | 23.16 | 1.00 | 126.20 | 23.16 |
| 3 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 |
| 4 | 74.17 | 74.17 | 11.36 | 74.17 | 25.40 | 1.00 | NaN | 25.40 |
| 5 | 34.45 | 34.45 | 59.32 | 34.45 | 21.85 | 1.00 | NaN | 21.85 |
| 6 | 12.74 | 12.74 | 12.74 | 12.74 | 1.94 | 1.00 | 12.35 | 1.94 |
| 7 | 79.94 | 79.94 | 14.77 | 79.94 | 55.52 | 1.00 | NaN | 55.52 |
| 8 | 13.63 | 13.63 | 54.57 | 13.63 | 10.47 | 1.00 | 18.46 | 10.47 |
| 9 | 43.14 | 43.14 | 189.35 | 43.14 | 41.88 | 1.00 | NaN | 41.88 |
| 10 | 59.12 | 59.12 | 59.74 | 59.12 | 46.02 | 1.00 | 119.72 | 46.02 |

Tables A-1 and A-2 show the MRE of the estimates obtained using sample 3 of our data set for the 4-node and 14-node respectively. In the case of the 4-node network, all the generalized inverses yielded the same estimate when applied solely. When combined with the tomogravity technique, RInv and MPInv produced the same result, G2Inv yields an MRE of 1 because all the estimates are 0s while the G3Inv yields estimates with higher or invalid (NaN) MRE values.

Table A-2: MRE of Generalized Inverses for 10 topologies of 14-Node Network

| Top | RInv | G2Inv | G3Inv | MPInv | TG+RInv | TG+G2 | TG+G3 | TG+MPInv |
|---|---|---|---|---|---|---|---|---|
| 1 | 264.04 | 264.04 | 197.05 | 264.04 | 278.20 | 1.00 | 310.71 | 278.20 |
| 2 | 271.62 | 271.62 | 526.48 | 271.62 | 291.93 | 1.00 | 313.66 | 291.93 |
| 3 | 266.83 | 266.83 | 285.76 | 266.83 | 278.28 | 1.00 | 283.03 | 278.28 |
| 4 | 249.47 | 249.47 | 160.39 | 249.47 | 284.47 | 1.00 | 289.23 | 284.47 |
| 5 | 337.68 | 337.68 | 307.07 | 337.68 | 333.01 | 1.00 | 312.38 | 333.01 |
| 6 | 289.15 | 289.15 | 491.85 | 289.15 | 277.57 | 1.00 | 243.25 | 277.57 |
| 7 | 228.53 | 228.53 | 207.22 | 228.53 | 248.10 | 1.00 | 258.47 | 248.10 |
| 8 | 293.68 | 293.68 | 267.63 | 293.68 | 304.21 | 1.00 | 295.08 | 304.21 |
| 9 | 388.98 | 388.98 | 818.61 | 388.98 | 375.00 | 1.00 | 378.53 | 375.00 |
| 10 | 360.25 | 360.25 | 421.19 | 360.25 | 318.64 | 1.00 | 430.23 | 318.64 |

Similar results were obtained in the case of the 14-node network, except that the G3Inv consistently produced estimates with higher MRE when applied solely or combined with the WLSE technique. We conclude that other generalized inverses are not better than the Moore-Penrose inverse in estimating IP traffic matrices.

# Appendix B:

# Additional Simulation Results

Table B-1: MRE of 5 Techniques using WCB Prior Distribution

| | **T**G | **E**M | **Q**P | **L**P1 | **L**P2 | **N**N |
|---|---|---|---|---|---|---|
| **4**-Node | 0.48 | 0.30 | 0.41 | 2.31 | 2.66 | 0.30 |
| **1**2-Node | 14.34 | 9.30 | 9.92 | 28.11 | 20.01 | 10.06 |
| **1**4-Node | 107.78 | 69.77 | 76.26 | 802.70 | 74.52 | 99.39 |
| | | | | | | |
| **T**op 95% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P1 | **L**P2 | **N**N |
| **4**-Node | 0.08 | 0.03 | 0.04 | 0.48 | 0.50 | 0.03 |
| **1**2-Node | 0.59 | 0.36 | 0.40 | 0.53 | 0.59 | 0.42 |
| **1**4-Node | 0.75 | 0.57 | 0.72 | 163.42 | 1.19 | 0.81 |
| | | | | | | |
| **T**op 90% of Demands | | | | | | |
| | **T**G | **E**M | **Q**P | **L**P1 | **L**P2 | **N**N |
| **4**-Node | 0.05 | 0.02 | 0.02 | 0.45 | 0.44 | 0.02 |
| **1**2-Node | 0.48 | 0.32 | 0.35 | 0.49 | 0.50 | 0.37 |
| **1**4-Node | 0.62 | 0.50 | 0.60 | 139.03 | 1.00 | 0.68 |

Table B-2: MRE of Entropy Maximization Hybrid Techniques

| | EM | EM-TG | EM-QP | EM-LP2 | EM-LP1 |
|---|---|---|---|---|---|
| 4-Node | 0.28 | 0.19 | 0.19 | 0.64 | 1.25 |
| 12-Node | 8.02 | 8.11 | 7.67 | 9.17 | 17.76 |
| 14-Node | 69.10 | 69.69 | 73.47 | 848.95 | 89.33 |
| | | | | | |
| Top 95% of Demands | | | | | |
| | EM | EM-TG | EM-QP | EM-LP2 | EM-LP1 |
| 4-Node | 0.02 | 0.04 | 0.02 | 0.23 | 0.43 |
| 12-Node | 0.37 | 0.39 | 0.45 | 0.41 | 0.60 |
| 14-Node | 0.57 | 0.59 | 0.70 | 688.45 | 1.16 |
| | | | | | |
| Top 90% of Demands | | | | | |
| | EM | EM-TG | EM-QP | EM-LP2 | EM-LP1 |
| 4-Node | 0.01 | 0.02 | 0.02 | 0.21 | 0.37 |
| 12-Node | 0.32 | 0.33 | 0.38 | 0.37 | 0.51 |
| 14-Node | 0.49 | 0.51 | 0.58 | 756.48 | 0.97 |

Table B-3: MRE of Quadratic Programming Hybrid Techniques

| | QP | QP-TG | QP-EM | QP-LP2 | QP-LP1 |
|---|---|---|---|---|---|
| 4-Node | 0.29 | 0.42 | 0.27 | 0.32 | 1.10 |
| 12-Node | 9.75 | 10.21 | 10.75 | 10.99 | 8.72 |
| 14-Node | 76.64 | 69.69 | 68.99 | 818.94 | 75.41 |
| | | | | | |
| Top 95% of Demands | | | | | |
| | QP | QP-TG | QP-EM | QP-LP2 | QP-LP1 |
| 4-Node | 0.02 | 0.04 | 0.02 | 0.03 | 0.14 |
| 12-Node | 0.46 | 0.39 | 0.37 | 0.50 | 0.48 |
| 14-Node | 0.72 | 0.63 | 0.57 | 749.49 | 0.95 |
| | | | | | |
| Top 90% of Demands | | | | | |
| | QP | QP-TG | QP-EM | QP-LP2 | QP-LP1 |
| 4-Node | 0.02 | 0.03 | 0.01 | 0.02 | 0.10 |
| 12-Node | 0.39 | 0.34 | 0.32 | 0.44 | 0.41 |
| 14-Node | 0.59 | 0.53 | 0.49 | 668.64 | 0.80 |

Table B-4: MRE of Linear Programming Hybrid Techniques

| | **L**P2 | **L**P2-TG | **L**P2-EM | **L**P2-QP | **L**P2-NN |
|---|---|---|---|---|---|
| **4**-Node | 2.31 | 2.65 | 2.64 | 2.64 | 2.65 |
| **1**2-Node | 28.11 | 19.40 | 19.02 | 18.14 | 18.77 |
| **1**4-Node | 802.70 | 101.02 | 97.24 | 92.93 | 76.07 |
| | | | | | |
| **T**op 95% of Demands | | | | | |
| | **L**P2 | **L**P2-TG | **L**P2-EM | **L**P2-QP | **L**P2-NN |
| **4**-Node | 0.48 | 0.50 | 0.50 | 0.50 | 0.50 |
| **1**2-Node | 0.53 | 0.56 | 0.55 | 0.60 | 0.68 |
| **1**4-Node | 163.36 | 1.15 | 1.14 | 1.16 | 1.20 |
| | | | | | |
| **T**op 90% of Demands | | | | | |
| | **L**P2 | **L**P2-TG | **L**P2-EM | **L**P2-QP | **L**P2-NN |
| **4**-Node | 0.45 | 0.44 | 0.44 | 0.44 | 0.44 |
| **1**2-Node | 0.49 | 0.48 | 0.47 | 0.51 | 0.59 |
| **1**4-Node | 139.11 | 0.97 | 0.96 | 0.98 | 1.02 |

Table B-5: MRE of Neural Network Hybrid Techniques

| | **N**N | **N**N-TG | **N**N-EM | **N**N-QP | **N**N-LP2 |
|---|---|---|---|---|---|
| **4**-Node | 0.59 | 1.83 | 0.16 | 0.72 | 0.72 |
| **1**2-Node | 9.18 | 8.54 | 7.99 | 12.64 | 12.64 |
| **1**4-Node | 85.82 | 78.01 | 68.77 | 104.07 | 104.07 |
| | | | | | |
| **T**op 95% of Demands | | | | | |
| | **N**N | **N**N-TG | **N**N-EM | **N**N-QP | **N**N-LP2 |
| **4**-Node | 0.11 | 0.06 | 0.02 | 0.19 | 0.19 |
| **1**2-Node | 2.61 | 0.96 | 1.50 | 3.86 | 3.86 |
| **1**4-Node | 0.85 | 0.67 | 0.58 | 1.23 | 1.23 |
| | | | | | |
| **T**op 90% of Demands | | | | | |
| | **N**N | **N**N-TG | **N**N-EM | **N**N-QP | **N**N-LP2 |
| **4**-Node | 0.07 | 0.05 | 0.01 | 0.13 | 0.13 |
| **1**2-Node | 3.03 | 1.04 | 1.70 | 4.43 | 4.43 |
| **1**4-Node | 0.71 | 0.57 | 0.50 | 1.07 | 1.07 |