Network-Design Sensitivity Analysis

Paul Tune and Matthew Roughan
School of Mathematical Sciences
The University of Adelaide, Australia
{paul.tune,matthew.roughan}@adelaide.edu.au

ABSTRACT

Traffic matrices are used in many network engineering tasks, for instance optimal network design. Unfortunately, measurements of these matrices are error-prone, a problem that is exacerbated when they are extrapolated to provide the predictions used in planning. Practical network design and management should consider sensitivity to such errors, but although robust optimisation techniques exist, it seems they are rarely used, at least in part because of the difficulty in generating an ensemble of admissible traffic matrices with a controllable error level. We address this problem in our paper by presenting a fast and flexible technique of generating synthetic traffic matrices. We demonstrate the utility of the method by presenting a methodology for robust network design based on adaptation of the *mean-risk analysis* concept from finance.

Categories and Subject Descriptors

C.2.1 [**Computer Communications**]: Network architecture and design; C.4 [**Performance of Systems**]: Modeling Techniques

Keywords

Mean-risk analysis; network design; network planning; sensitivity analysis; traffic matrix synthesis

1. INTRODUCTION

Network planning is a crucial task for maintaining the operational status quo of an existing network, as well as integration of future link expansions and technology into the network. Examples of some planning tasks include optimising the capacity of links and assigning an optimal routing plan based on the network topology and link capacities. All these tasks require a key input: the *traffic matrix*.

Unfortunately, the traffic matrix is not easy to measure accurately [26]. Inferring the traffic matrix from indirect measurements is an ill-conditioned, underconstrained problem, resulting in errors. Direct measurement methods are rarely economically viable, and sampling introduces errors. Moreover, planning must be performed using predicted traffic matrices, and *planning horizons* may be as long as a year in advance. Prediction errors may therefore be substantial, even when good data is available. In green-fields planning [19], predictions may be highly inaccurate.

As a result, an alternative set of ideas have been proposed: oblivious routing [1], and Valiant network design [31], which seek to design a network and its routing so as to work well with any traffic matrix. These network designs are *oblivious* to the traffic matrix, *i.e.*, the network performance is guaranteed for any matrix. The penalty is a loss of efficiency, as the network must be over-engineered by as much as a factor of two.

In reality, there is some information about the network beforehand, and this is useful in constraining the set of likely traffic matrices. The information allows operators to make informed guesses even in green-fields planning. The information may be coarser than needed and certainly contains errors, so we need an approach that finds the middle ground between complete ignorance and omniscience.

The other issue for oblivious approaches is that different operators may have different degrees of *risk aversion*. There is a continuous tradeoff between over-engineering to allow for potential unknowns, and the risk of congestion and its resultant performance degradation. Some operators may be sanguine about congestion, allowing it on occasion in order to provide a cheaper service, whereas others may wish to provide a "gold-class" service, with guaranteed SLAs, and are prepared to pay the extra cost required.

We start with the simple idea that operators should at least understand the sensitivity of their designs to prediction errors, and take these into account as part of planning. Arguably, the simplest approach requires a method to generate an ensemble of traffic matrices that model the errors in the predicted matrix. Such matrices must be:

- *admissible*, *i.e.*, that satisfy some set of constraints (*e.g.*, non-negativity);
- *centred*, *i.e.*, their average centres on the predicted matrix; and
- *controlled*, *i.e.*, their variance around the predicted matrix can be controlled, ideally through a simple, linear parameter.

The canonical example of such an ensemble in many practical situations is IID (Independent Identically Distributed)

SIGMETRICS'14, June 16–20, 2014, Austin, Texas, USA.

http://dx.doi.org/10.1145/2591971.2591979.

Gaussian additive noise. However, in our problem, simple additive noise fails the first two criteria. We present here a method satisfying these criteria.

In addition, we wish the ensemble to be as realistic as possible, though this is intrinsically hard to test given the fact that we can't (by the nature of the problem) obtain a complete ensemble of such data against which to compare the synthetic matrices. Instead, here we test the *usefulness* of our model[1] in the network capacity planning problem.

We do not aim to provide novel optimisation algorithms, but simply to demonstrate the importance and utility of an ensemble of "error" matrices. We do so through a proposed framework to optimise traffic engineering metrics of a network in a systematic way. Our framework is based on *mean-risk analysis* ideas, commonly used in finance for portfolio management. The key idea here is to develop both a *utility function* and simple model of *risk averseness* of a network operator with regard to chosen traffic engineering metrics against which we can optimise.

To summarise, our main contributions are:

- a fast, flexible and simple algorithm to generate an ensemble of synthetic traffic matrices around a predicted traffic matrix possessing desirable criteria such as admissibility, centredness and controlled by a single linear parameter,
- theoretical results on the performance of the algorithm, which are empirically validated on real traffic,
- evaluation of the sensitivity of important network designs with respect to edge capacity prediction errors via the use of our synthetic traffic matrices, and
- a simple framework for network design optimisation incorporating robustness into the design using our synthetic traffic matrices.

We validated our results using real traffic from the Abilene network [15].

2. BACKGROUND

2.1 Traffic Matrices

A traffic matrix describes the volume of traffic (typically in bytes) from one point in a network to another during some time interval. Its natural representation is a three dimensional array $\mathbf{T}(\tau)$ with elements $t_{i,j}(\tau)$ that represent the traffic volume from source i to destination j during the time interval $[\tau, \tau+\Delta\tau)$. Time intervals depend on the time resolution of measurements, and the aim is to be able to observe a stationary sequence of measurements. Typical intervals are of the order of minutes to hours (5, 15, and 60 minutes are very common). In our applications, the locations i and j will either be routers or, more typically Points-of-Presence (PoPs), and we only consider Ingress/Egress (IE) traffic matrices here (not underlying Origin/Destination (OD) matrices as these are usually unobservable). We will denote the number of ingress and egress points by N. Throughout the paper, we concentrate on a single snapshot of the traffic matrix at time τ, so we will omit the time index for convenience. For a much more complete description of all of the issues involved in such a matrix please see [26].

[1] We remind the reader of the quote "Essentially, all models are wrong, but some are useful." [2, p. 424]

The best current practice for measuring a network's traffic matrix is to use sampled flow-level measurements (*e.g.*, [7]). These measurements provide traffic matrices on a time granularity on the order of minutes, but these matrices will contain errors. Such errors arise from sampling and from lack of synchronisation in time intervals used in measurements. Control and calibration of the size of these errors, as far as we are aware, is not currently implemented. However, one might hope they were fairly small, say at the level of a few percent, in a well-engineered measurement system.

Alternative approaches to direct (sampled) collection include a body of research [13, 28–30] devoted to developing traffic matrix inference methods from more easily collected link-load measurements. These methods, however, are limited by the underconstrained nature of the problem, so average errors on the order of 10–20% are possible. Despite these errors, [22] showed that traffic engineering tasks, such as routing, can certainly benefit from an inferred traffic matrix as compared to having no knowledge whatsoever.

More importantly, one of the underlying themes of that research is that it is much easier to measure the traffic volumes on links than it is to determine where that traffic is going (and hence the traffic matrix). As a result, it is quite easy to measure values such as the row sums of the traffic matrices $r_i = \sum_j t_{i,j}$ (which tell us the total traffic coming into a PoP) and the column sums $c_j = \sum_i t_{i,j}$ (which tell us the total traffic leaving through a PoP).

Network planning requires forecasting future traffic matrices to the date relevant for the plans, the time in advance often called the *planning horizon*. The prediction process may vary, but it will inevitably introduce prediction errors.

Common sources of error are

- *Statistical inference errors:* due to stochastic variability within the model, *i.e.*, the model for the data is a good approximation, but the particular realisation we observed varies from that predicted.
- *Large, short-term fluctuations from the prediction:* *e.g.*, routing changes (either caused by internal link failures, or external routing policy changes), can alter the egress points of traffic, altering the IE traffic matrix (even if the underlying OD matrix remains unchanged) [25]; or flash-crowds can cause large, short-term changes to traffic. Some of these changes might be considered simple stochastic variation (as above), but some are larger, and more sudden than can be accounted for by normal variation, and their causes can often be identified as singular events.
- *Modelling errors:* resulting from use of a prediction technique whose underlying model is inaccurate over the prediction period. For instance, growth in traffic may appear to be exponential over some period, but actually be logistic (appearing exponential early on, but slowing as demand becomes saturated).

Modelling errors should be preventable by careful analysis of sufficient periods of historical data. The other two fluctuations are intrinsically hard to prevent, and it is these that we primarily focus on here.

The level of aggregation of traffic is often a large factor in the size of these stochastic prediction errors. Large aggregates of traffic are more predictable because they statistically aggregate the behaviour of many more sources. It

is natural then to expect errors to have some dependence on the size of traffic matrix element. Traffic matrix elements have a somewhat skewed distribution (with many small values, and few large) though not formally heavy-tailed [17, 18, 24], implying that we have a large group of smaller, more inaccurate predictions, and a smaller group of large, but comparatively accurate predictions. The Norros model [16], empirically tested in [20] and somewhat based on equivalent assumptions to the independent flow model [6] proposed for traffic matrices, proposes that variance should be proportional to the mean of the traffic. Moreover, the model has consistent variances for aggregated traffic. We need an error model satisfying this property, which we call *proportional error variance aggregation*, or PEVA.

We might also reasonably assume that prediction errors for aggregates such as the $2N$ row and column sums of a traffic matrix will be smaller than for the N^2 elements. Moreover, prediction of PoP (in/out) volumes can take advantage of demographic, marketing, business and financial predictions much more easily than can the $t_{i,j}$.

The point we are making is that traffic matrix (and related data) contains considerable value, but it must be assumed that it contains errors. However, these errors are not completely arbitrary. Larger matrix elements have smaller relative errors, and aggregates (such as row and column sums) often have much smaller errors than those in the traffic matrix elements themselves.

2.2 Mean-Risk Analysis

The literature on network planning (and related optimisation problems) is vast (for instance see [3]), and we do not aim to survey it here. However, relatively few network operators seem to use formal optimisation as part of their design process, and none that we are aware of use robust optimisation (where robust is used here to mean that the solutions are not sensitive to uncertainty in the inputs).

The approach we adopt here extends the ideas of mean-risk analysis – one approach for the analysis of the potential volatility in financial portfolio returns.

Generally, operators are interested in maximising the utilisation of the network, but at the same time, providing enough capacity to meet network policies and quality of service (QoS) constraints for the customers' applications. A common approach to allow for uncertainty (say, because of prediction errors) is to include some engineering *overhead*, or *over-dimensioning*. That is, instead of designing a network to carry the predicted traffic, the network is designed to carry this traffic, multiplied by some *safety* factor.

A simple tradeoff arises here: if the network is over–engineered with a high factor, there is much lower risk in the sense that it is able to cope with unexpected traffic, but at a higher cost. Different operators use different factors, based on experience and their degree of risk aversion. Yet, few if any operators formally calculate risk, and specify precisely what their degree of risk aversion is. Our approach – mean-risk analysis – makes this tradeoff explicit through the calculation of an appropriate measure of risk for networks.

To the best of our knowledge, a direct application of mean-risk analysis to network design has not been developed. One paper introducing a measure of risk in the context of routing, called the *bit risk mile*, is [5]. Risk here is directly related to the geographic distance and the expected outage between a source and a destination PoP on a particular path. Although a concept of *risk* was used, their definition is not the same as ours, as they study routing, whereas here we study the network design, which requires the traffic matrix.

The closest work to introduce some of these ideas, in the context of maximising bandwidth revenue of a network, is [14]. The paper discussed how a network operator can balance between wholesale and retail bandwidth revenue against his risk averseness by formulating an optimisation problem and studying the properties of the solution. However, it was assumed the underlying distribution of the revenue (a truncated Gaussian distribution) is known, and the network operator is thus able to compute the mean and standard deviation. Such an assumption is common in finance problems.

However, the multi-dimensional nature of traffic matrices, and the constraints on these matrices prevent us from providing a closed-form for the distribution of these matrices. Instead, we use our synthetic model to calculate risk, and it is this we describe next.

3. TRAFFIC MATRIX SYNTHESIS

Our main goal is to generate an ensemble of traffic matrices to test a network design. Not every matrix, however, can be a traffic matrix. As noted in the introduction such a traffic matrix must be *admissible*, which means that it satisfies some set of constraints. There are multiple possible constraints, but for illustration purposes in this paper we will focus on 4 simple, intuitive, and yet reasonably powerful constraints. For a matrix $\mathbf{T} = [t_{i,j}]$, we have

(i) *Non-negativity*: $t_{i,j} \geq 0$, $\forall i, j = 1, 2, \cdots, N$,

(ii) *Row sums*: $\sum_j t_{i,j} = r_i$, $\forall i$,

(iii) *Column sums*: $\sum_i t_{i,j} = c_j$, $\forall j$, and

(iv) *Total traffic*: $\sum_{i,j} t_{i,j} = \sum_i r_i = \sum_j c_j = T$,

where row and column sums are summarised in vectors $\mathbf{r}$ and $\mathbf{c}$ respectively. The last constraint is actually redundant given the previous constraints, but we include it for its explanatory value.

The constraints above are motivated by the fact that the row and column sums represent the total ingress and egress traffic of a network, which are relatively easy to obtain from SNMP information [26]. We aim to preserve the row, column and total traffic constraints because these will allow us to make fair comparisons between different network designs (see case studies in Sections 4 and 5). In general we might replace the row, column and total traffic constraints with any set of linear (or more generally, convex) constraints on the matrix, but we use the single constraints above for illustration because of their clear meanings in context.

Previous work [17, 18] focussed on generating such matrices, with the additional aim of matching some statistical properties of empirically measured matrices. The main difference between those papers, and ours is that we also aim to generate matrices which are *centred* [2] around a predicted matrix, with *controlled* variation around that matrix.

[2] It is perhaps more common to refer to "centring" as lack of bias, but we prefer the term here as we shall use it to mean lack of bias with respect to particular functions of expectation, not simple expectation.

For instance, the most commonly used noise models are additive, or multiplicative white noise, *i.e.*, we generate a matrix $\mathbf{Y}$ from prediction $\mathbf{T}$ by either

$$\begin{aligned} \textbf{Additive}: &\quad y_{i,j} = t_{i,j} + \sigma z_{i,j}, \\ \textbf{Multiplicative}: &\quad y_{i,j} = t_{i,j}\big(1 + \sigma z_{i,j}\big), \end{aligned}$$

where the $z_{i,j}$ are IID random variables forming the *error* or *noise* matrix $\mathbf{Z}$. In many cases, the $z_{i,j}$ would be Gaussian distributed with mean zero, and variance 1, *i.e.*, $z_{i,j} \sim \mathcal{N}(0,1)$, thus the parameter σ *controls* the variance of the errors, and both are centred (with respect to the average, or the average of the log, respectively).

However, simple generation methods are subtly difficult. We immediately see that with $z_{i,j} \sim \mathcal{N}(0,1)$ the two models above may generate inadmissible matrices, because they may violate non-negativity. Naturally, we truncate the distribution to prevent this, but that results in a significant number of "zero" elements of the resulting matrix, and this *uncentres* the matrix. Moreover, the resulting matrices no longer satisfy the row, column or total traffic constraints.

So the question remains: how can we generate a randomised traffic matrix satisfying the set of constraints? There are many possible noise models, and procedures for fitting constraints, however, we shall aim here to provide an approach which is intuitively well-matched to traffic matrix generation, as well as simple and fast (speed is important, as traffic matrices inhabit a high-dimensional space, so sensitivity analysis may require generation of a large number of these matrices to explore this space adequately).

3.1 Error Model and Non-Negativity

The goal of our approach is to guarantee the admissibility constraints are (at least in part) automatically satisfied by any matrix we generate. To do so we divide the constraints into *particular* constraints (the summation constraints, which are dependent on the particular network), and *universal* constraints (non-negativity), which all traffic matrices should satisfy.

Given the non-negativity, it is possible to write any traffic matrix $\mathbf{T} = [t_{i,j}]$ in the form

$$\mathbf{T} = \left[a_{i,j}^2\right],$$

where $a_{i,j} = \sqrt{t_{i,j}}$. The modification seems trivial, but given real values $a_{i,j}$, the matrix is now inherently non-negative. Moreover, the total traffic matrix constraint becomes

$$\sum_{i,j} a_{i,j}^2 = T,$$

so we can see the $a_{i,j}$s as lying on the N^2-dimensional hypersphere with radius $\sqrt{T}$.

Our approach, therefore, is to perturb the matrix by finding a new point on this hypersphere via an approximately additive model. This allows us to add noise in a controlled and centred manner, while preserving the universal constraint.

Our *Spherically Additive Noise Model* (SANM) is

$$y_{i,j} = (a_{i,j} + \beta z_{i,j})^2, \text{ for all } i,j, \tag{1}$$

where $z_{i,j} \sim \mathcal{N}(0,1)$, and $\beta \in [0,\infty)$ is a parameter we can use to tune the strength of the noise. We chose a simple IID noise process, as we have no *a priori* reason to assume correlations in the noise.

Although we allow the possibility of large β, as we shall see, there is a restricted range of reasonable values of β in practice. However, large β is not a detriment, as the model saturates to the gravity model as $\beta \to \infty$, see [27]. We will discuss this in-depth below.

The beauty of the simplicity of this model is that it guarantees non-negativity without truncation, and so not only does it not introduce zeros into the matrix, it is also simple to analyse. Since $z_{i,j} \sim \mathcal{N}(0,1)$, independent of $a_{i,j}$, we get

$$\begin{aligned} \mathbb{E}[y_{i,j}] &= \mathbb{E}\left[(a_{i,j} + \beta z_{i,j})^2\right] \\ &= \mathbb{E}\left[a_{i,j}^2 + 2\beta a_{i,j} z_{i,j} + \beta^2 z_{i,j}^2\right] \\ &= t_{i,j} + 2\beta \mathbb{E}[a_{i,j} z_{i,j}] + \beta^2 \mathbb{E}\left[z_{i,j}^2\right] \\ &= t_{i,j} + 2\beta a_{i,j} \mathbb{E}[z_{i,j}] + \beta^2 \\ &= t_{i,j} + \beta^2. \end{aligned}$$

This is not ideal (yet), because the resulting matrix clearly doesn't satisfy the centring condition, so a procedure is needed to force the matrix to satisfy the additional constraints.

3.2 Iterative Proportional Fitting (IPF)

The matrix $\mathbf{Y}$ no longer sits on the hypersphere defined by the total traffic constraints (or the manifolds defined by the other constraints). Intuitively, what we need is a procedure to *project* the perturbed solution onto the space defined by the set of constraints. We use the Iterative Proportional Fitting (IPF). Algorithm 1 and 2 outline our method. The algorithm is very simple indeed, with the only complication occurring in the implementation of IPF, which we detail below.

Input: $\mathbf{T}$, the predicted traffic matrix
Input: β, noise variance
Input: $\mathbf{r}$, $\mathbf{c}$, row and column sum constraints
Input: ϵ, tolerance for IPF
Output: $\mathbf{S}$, the synthetic traffic matrix
1 Generate $\mathbf{Z}$, with $z_{i,j} \sim \mathcal{N}(0,1)$
2 Generate $\mathbf{Y}$, where $y_{i,j} = (t_{i,j}^{1/2} + \beta z_{i,j})^2$
3 $\mathbf{S} = \text{IPF}(\mathbf{Y}, \mathbf{r}, \mathbf{c}, \epsilon)$ `/* See Algorithm 2 */`

Algorithm 1: Traffic Matrix Perturbation

IPF was originally developed to adjust contingency tables in statistics such that their marginals, given by the row and column sums, satisfy known constraints [4]. This is almost exactly what we aim to do, the only difference being terminology, where we apply IPF to traffic matrices instead. More recently, IPF has been adapted to work with an arbitrary set of measurement constraints (provided the constraints lie in a convex set) [11]. This implies that many types of constraints in network design can be incorporated into IPF. Here, we restrict our attention to only row and column sum constraints which naturally map to our current problem.

IPF consists of iteratively scaling the rows and columns of the matrix until the scaled row and column sums match the objective row and column sums (see Algorithm 2). Here, $*$ denotes a wildcard to specify the rows or columns of the matrix at once. The typical test for convergence is to compare the row and column sums of the iterate $\mathbf{S}^{(k)}$ to $\mathbf{r}$ and $\mathbf{c}$ at the end of each iteration k. Once the total difference falls below some required tolerance ϵ, the algorithm has converged.

Input: $\mathbf{Y}$, the raw perturbed traffic matrix
Input: $\mathbf{r}$, $\mathbf{c}$, row and column sum constraints
Input: ϵ, tolerance for IPF
Output: $\mathbf{S}$, the synthetic traffic matrix

$\mathbf{S}^{(0)} \leftarrow \mathbf{Y}$
$k \leftarrow 1$
while *(not converged)* **do**
  /* Scale the rows */
  for $i \leftarrow 1$ ***to*** N **do**
    $\mathbf{S}^{(k-1/2)}_{i,*} \leftarrow \mathbf{S}^{(k-1)}_{i,*} r_i / \sum_i \mathbf{S}^{(k-1)}_{i,*}$
  end
  /* Scale the columns */
  for $j \leftarrow 1$ ***to*** N **do**
    $\mathbf{S}^{(k)}_{*,j} \leftarrow \mathbf{S}^{(k-1/2)}_{*,j} c_j / \sum_i \mathbf{S}^{(k-1/2)}_{*,j}$
  end
  $k \leftarrow k + 1$
end

Algorithm 2: Iterative Proportional Fitting (IPF)

IPF has a strong connection to the generalised Kullback-Leibler (GKL) divergence used in non-negative matrix decomposition [10]. IPF is an iterative, fixed-point solution to the problem of minimising the GKL divergence between the input to the final solution, subject to constraints (see proof in [27]). More complex algorithms can be used to solve this problem, but IPF's simplicity and fast convergence are ideal here.

An additional property of IPF worth mentioning is that it is guaranteed to converge to a solution satisfying the row and column constraints only if the initial input matrix is non-negative [4]. Our error model is guaranteed to be non-negative, so convergence is assured.

Moreover, IPF preserves zeros [4, 23]. Thus problematic zeros introduced by truncation (if a simplistic error model was used) would be retained by this step. This is undesirable, as real traffic matrices generally do not have many zero elements. Our model is unlikely to have zero elements for $\beta > 0$. Such an event occurs only if $t^{1/2}_{i,j} = -\beta z_{i,j}$, but since $z_{i,j}$ is a continuous random variable, this would occur with probability 0. So our noise model creates elements that are almost-surely positive (simulations confirm this, the results of which we omit here in the interest of space). Therefore, IPF's outputs are also positive.

IPF isn't novel, even in the context of traffic matrices, where it has been used to enforce row and column constraints before [29]. However, in the cases we are aware of, the method was being used as part of an inference technique, whereas here it is being used for synthesis.

3.3 Discussion

In this synthesis problem, we are essentially finding a set of random matrices $\mathbf{S} = \mathbf{T} + \mathbf{W}$ that satisfy a set of constraints

$$\mathcal{A}(\mathbf{S}) = \mathbf{b}.$$

The operator $\mathcal{A}$ might represent a set of measurement operations, where $\mathbf{b}$ are observed measurements, or in the preceding section we considered $\mathcal{A}$ to take row and column sums and $\mathbf{b}$ to be the specified values of those sums.

In general, however, we consider linear operators $\mathcal{A}$, and hence we are requiring that $\mathcal{A}(\mathbf{T}) = \mathcal{A}(\mathbf{S}) = \mathcal{A}(\mathbf{T} + \mathbf{W}) = \mathcal{A}(\mathbf{T}) + \mathcal{A}(\mathbf{W}) = \mathbf{b}$, or

$$\mathcal{A}(\mathbf{W}) = \mathbf{0},$$

i.e., the noise $\mathbf{W}$ must lie in the null space of $\mathcal{A}$, though with the additional complication of requiring non-negativity.

One method of finding such matrices is by defining a manifold on which all valid traffic matrices reside and forming the above additive model on this manifold. However, that requires (i) defining invertible maps between the manifold and the generated traffic matrices and (ii) search algorithms on manifolds (which are much slower than IPF). For instance, see [8] for an example of a manifold and its relationship to IPF. So the approach we propose is solving a more general problem, but in a fast, easy to implement manner.

Although some measurements are more accurate, they are not perfect. For example, the row and column sums may contain errors, albeit small. The row and column sums themselves may be subject to variations in traffic over time. Either way, regularisation methods may be used to handle variations of these measurements, but a simpler way we adopt in the paper is to add a small amount of noise to the constraint values $\mathbf{b}$ prior to traffic matrix generation. For a valid traffic matrix, the constraints must be self-consistent, for instance, the total of the row and column sums must be equal to the total traffic traversing the network, but this is easy to ensure. So the SANM allows controllable errors in the constraints as well as the generated traffic matrices themselves. We test SANM on variations of the row and column sums in a case study in Section 4.2.

It is worth noting that our method does not constraint the individual elements of the traffic matrix directly. One motivation for direct constraints occurs between the different customers of an ISP, as each may have different SLAs, and are therefore each sensitive to different variations in traffic. To capture these variations, ideally, one would have to define a distribution to specify the traffic variations. It is possible to do so in our model, however, this would (i) require additional work in specifying the type of noise used in SANM, as the noise may no longer be IID, and (ii) define new constraints in IPF.

3.4 Properties

Two criteria for this model were that it be centred and controllable. The SANM is nonlinear, and hence somewhat hard to analyse, but intuitively, IPF is finding the *closest* point on the appropriate manifold to project the perturbed solution to, so we might hope for these properties.

Analytically, we consider the simplest case where the row and column constraints are ignored, and only the total traffic constraint is included (for more complex row and column problems we quantify the effect in the following section). In the simple case, IPF will scale the values so that the total is correct, *i.e.*,

$$\mathbb{E}[s_{i,j}] = c\mathbb{E}[y_{i,j}] = c\left(t_{i,j} + \beta^2\right).$$

The scaling constant c will be chosen so that $\sum_{i,j} s_{i,j} = T$, so we know that

$$\mathbb{E}\left[\sum_{i,j} s_{i,j}\right] = c\mathbb{E}\left[\sum_{i,j} t_{i,j} + \beta^2\right] = c\left(T + N^2\beta^2\right) = T,$$

and so

$$c = \frac{1}{1 + N^2\beta^2/T}.$$

Take the average matrix element to be $\bar{t} = T/N^2$ and we get

$$\mathbb{E}[s_{i,j}] = \frac{t_{i,j} + \beta^2}{1 + \beta^2/\bar{t}} = t_{i,j} + \beta^2 \left(1 - \frac{t_{i,j}}{\bar{t}}\right) + O(\beta^4).$$

So, in expectation, larger elements of the matrix become smaller, and smaller elements become larger, but the effect is $O(\beta^2)$, and so for small values of β the non-linearity is negligible. We therefore have approximate centring of the synthetic matrices.

In addition, we can consider the variance of the synthetic matrix about the predicted matrix by looking at the variance of $s_{i,j}$ and noting that, at least approximately this is the same as the variation around $t_{i,j}$:

$$\mathrm{Var}[s_{i,j}] = \mathbb{E}\left[s_{i,j}^2\right] - \mathbb{E}[s_{i,j}]^2 \simeq \mathbb{E}\left[s_{i,j}^2\right] - t_{i,j}^2 = \mathbb{E}\left[(s_{i,j} - t_{i,j})^2\right],$$

and

$$\begin{aligned} \mathbb{E}\left[s_{i,j}^2\right] &= c^2 \mathbb{E}\left[y_{i,j}^2\right] = \frac{\mathbb{E}\left[\left(a_{i,j} + \beta z_{i,j}\right)^4\right]}{\left(1 + \beta^2/\bar{t}\right)^2} \\ &= t_{i,j}^2 + \beta^2 t_{i,j} \left(6 - 2\frac{t_{i,j}}{\bar{t}}\right) + O(\beta^4), \end{aligned} \quad (2)$$

so the standard deviation of the variability around the predicted matrix is approximately linear in β, for small β. Averaging over the whole matrix to obtain a relative measure of standard deviation around matrix elements we get

$$\sqrt{\frac{1}{N^2} \sum_{i,j} \frac{\mathbb{E}\left[s_{i,j}^2\right] - t_{i,j}^2}{t_{i,j}^2}} \simeq \beta \sqrt{\frac{1}{N^2} \sum_{i,j} \frac{6}{t_{i,j}} - \frac{2}{\bar{t}}}. \quad (3)$$

We will examine the range of linearity of this approximation empirically for the more general case including row and column constraints in the following section. We find the approximation to be very good in the range $0 \leq \beta \leq 0.2$, and reasonable for β up to 0.4.

Finally, as the noise level increases, *i.e.*, $\beta \to \infty$, and the errors are very large, $\mathbf{Y}$ becomes less informative as a prior, as useful information from $\mathbf{T}$ is overwhelmed by noise, resulting in a saturated response of each $s_{i,j}$. We prove that this response approaches the response of the gravity model $\mathbf{r}\mathbf{c}^{\mathrm{T}}/T$ in [27].

This well-behaved asymptotic behaviour is another advantage of our model over the additive and multiplicative models. As the noise strength increases for the latter models, the generated traffic matrices depart significantly from realistic traffic matrices, due to the presence of a large number of zeros (the effect of truncation). In contrast, with our model, even with high noise, the generated traffic matrices can be used to study network designs under high uncertainty, *i.e.*, when predictions are way off the mark.

3.5 Validation

In the following results, the predicted traffic matrix $\mathbf{T}$ is a single snapshot from the Abilene [15] data, the average traffic over time 0200 to 0205 hours, 1st March 2004.

We first test the approximation (2). In Figure 1, we set $\beta = 0.2$ and plot approximation (2) against empirical measurements of $\mathbb{E}\left[s_{i,j}^2\right]$ (which also includes the row and column sum constraints), averaged over 1000 trials. The middle solid line denotes the equality between elements, *i.e.*, if the approximation equals $\mathbb{E}\left[s_{i,j}^2\right]$. We find that the fit is very good indeed, as the data points lie close to the solid line, despite the fact the approximation did not try to account

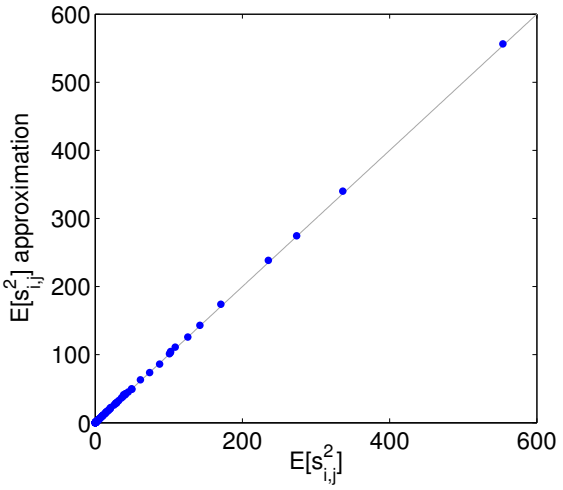


Figure 1: Testing the approximation (2) against IPF's output $\mathbb{E}\left[s_{i,j}^2\right]$ on a single snapshot from the Abilene [15] data, the average traffic over time 0200 to 0205 hours, 1st March 2004, with $\beta = 0.2$.

for the row and column sum constraints. We have tested this with various predicted traffic matrices with similar results. The approximation is fair up to $\beta = 0.4$, beyond which the approximation quality degrades. This is not surprising since with small noise, there is less distortion to the row and column sums of $\mathbf{Y}$, so the total traffic and non-negativity constraints have a bigger effect on the final output of IPF.

Our model also coincides with the intuition about the modelling of aggregated flows per the Norros model [21]. In real traffic, as flows are combined, the aggregated flow has a higher variance, and this variation is modelled to be proportional to the size of the aggregated traffic flow in the Norros model. These larger flows would naturally have larger measurement error. From approximation (2), we find the errors scale in proportion to $t_{i,j}$, aligning with the intuition of the Norros model. In the SANM, the error would therefore scale proportional to the size of the predicted traffic matrix's elements. We call this the *proportional error variance aggregation* (PEVA) property.

Another consideration is how far the generated traffic matrices are from the original input. Since the SANM is non-linear, will there be significant distortions of the matrices?

We measure the distortion using the Mean Relative Square Error (MRSE) of the generated traffic matrices to the input $\mathbf{T}$, defined by

$$\mathrm{MRSE}(\beta) = \sqrt{\frac{1}{N^2} \sum_{i,j} \frac{(s_{i,j}(\beta) - t_{i,j})^2}{t_{i,j}^2}}.$$

The metric quantifies the effect of the perturbation of IPF's solution to the predicted traffic matrix. Ideally, we would like a linear relationship between β and the MRSE, implying that $s_{i,j}$ is proportional to $t_{i,j}$, and this is predicted for small β by (3), but without row and column sum constraints. Here, we include those constraints and test the MRSE response.

We examine the MRSE response of IPF in Figure 2. The solid curve is the average response of the generated traffic matrices. The curve was generated from 200 data points,

Model	Parameter	Mean % zeros	Admissible	Centred	Controlled	PEVA
Additive	$\sigma = \bar{t}$	≈ 50.0	✓	✗	✓	✗
Multiplicative	$\sigma = 1$	13.6	✓	✗	✓	✗
SANM	$\beta = 0.2$	0.0	✓	✓	✓	✓

Table 1: Comparing the (truncated) additive, multiplicative and SANM over several important criteria. PEVA implies that the error of the generated matrix elements scale in proportion to the size of the predicted traffic matrix element as noise increases. Model parameters were chosen so that their MRSEs are roughly 1.

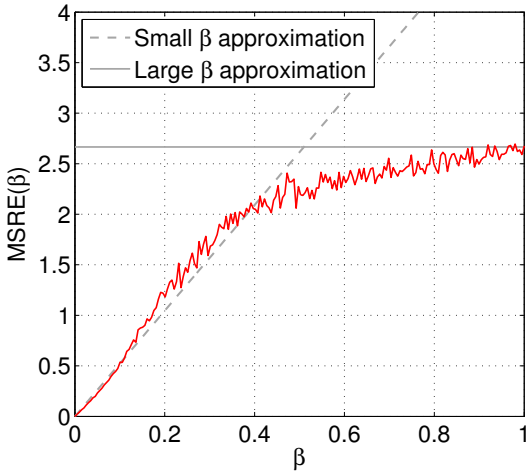


Figure 2: MRSE as a function of β. The solid curve is the average response of IPF's output over 1000 trials per data point. The predicted traffic matrix T is from Abilene data, the average traffic over time 0200 to 0205 hours, 1st March 2004. The dashed and flat lines show the approximation for small β (see approximation (3)) and large β (see details in [27])
.

each data point an average of 1000 trials. The sloped dashed line is the response of the approximation (3) for the small β regime, while the flat line is the response of the gravity model for the large β regime.

In Figure 2, the response of the MRSE to β is almost linear up to about $\beta = 0.4$ before starting to saturate to the response of the gravity model (see [27]).

Table 1 compares the SANM against the (truncated) additive and multiplicative models across several important criteria. Their parameters were chosen so that their MRSEs are roughly 1. The SANM satisfies all the required criteria. Also, because the other models produce a high number of zeros, convergence of IPF based on their inputs is not guaranteed. In contrast, our model has both theoretical and empirical convergence guarantees.

In practice, the saturation is not a handicap in any way. Since we are mostly interested in variations around $\mathbf{T}$ in the MRSE range $(0, 1]$, so β does not have to exceed 0.2 in most cases. If there are large errors present in traffic measurements, then the measurements essentially convey no information in the first place. Our model handles large errors gracefully, since it produces a positive matrix almost surely, so IPF is guaranteed to converge to a solution, and that solution is the gravity model which is a reasonable choice in the absence of other information.

4. CASE STUDY 1: DESIGN COMPARISON

It is intrinsically difficult to determine whether our test matrices are "accurate". We are generating an ensemble of synthetic traffic matrices to model variability WRT a single real traffic matrix, so we don't have anything to compare against. Instead, we opt to demonstrate that they are *useful* via two simple case studies.

In our first study, we consider the question of whether oblivious design is wasteful given some, potentially flawed, information about the traffic matrix. Our approach is to test five design strategies:

Abilene: We will perform our test using the Abilene network data (from 2004), and so our baseline design will be the actual Abilene network, but rather than use the actual link capacities, we choose link capacities that are the minimum required to carry $\mathbf{T}$, the predicted TM. Note that although these are not the real network capacities, this choice is made so that we can perform a fair comparison across all designs.

Robust Abilene: The previous network design assumed that we knew the traffic matrix $\mathbf{T}$ accurately, and chose capacities accordingly. Now we use the same set of links, but choose capacities to allow for errors in the traffic predictions. This is common practice in network design: first optimise the network according to an objective, then over-engineer to allow a margin of safety. In practice, the over-engineering may appear in the form of a constant factor determined empirically. Here, we shall explicitly use our synthesis methodology to determine how much over-engineered capacity is required. We generate a set of 100 synthetic matrices and set the link capacities so that we can carry any of these. This *robust* design then allows for errors in the traffic matrix inputs, but with some extra "overhead" capacity, which we will assess below as a function of β. Note that when $\beta = 0$, this design will be the same as the Abilene design.

Star: Also known as a "hub-and-spoke" design, it consists of a single central node with all other nodes connected to it. The central node has the highest access capacity as it needs to handle all incoming traffic from the other nodes. All traffic flows on this network travel at most two hops. The star design, with clever link capacity choices and load balancing, is the design with the minimum capacity required to serve *any* admissible traffic matrix with prescribed edge capacities, *i.e.*, row and column sums. It is therefore the optimal oblivious network. Although a common design amongst enterprise level networks, backbone networks do not use the star design as failure of the central node leads to catastrophic failure of the entire network.

Valiant (oblivious) design: The key idea here is a design of a network that can carry, without congestion, any admissible traffic matrix [31] with some failure resilience. This

may sound impossible, but it does so by using the clever idea of indirect routing from switching. The network design is a clique (with links between all pairs of routers), and each packet is routed along two hops, rather than traversing a direct route. The first hop is spread amongst all of the available routers and the second is to the destination router. In [31], the authors show that this type of network design and load balancing combination can guarantee that any possible traffic matrix (given access link capacities) can be carried as long as sufficient capacity is allocated to the links. Moreover, they presented an algorithm for selecting the proportions of traffic which go to each intermediate node (oblivious of the actual traffic matrix) and for designing the correct network capacities. We apply this technique to derive a set of link capacities in a network, using the row and column sums of the traffic matrix to derive lower bounds on the access link capacity. We use the row and column sums of the predicted traffic matrix in our capacity planning rather than an arbitrary access link capacity because this will produce a tighter network design, and allow for a fairer comparison against other capacity planning techniques. The capacity of link connecting i to j is calculated according to [31]

$$x_{i,j} = \left(\sum_{k=1}^{N} \frac{w_k}{T - 2w_k} \right)^{-1} \left(\frac{w_i w_j}{T - 2w_j} + \frac{w_i w_j}{T - 2w_i} \right), \quad (4)$$

where $w_i := \max(r_i, c_i)$. The total capacity of the Valiant design is at most twice that of the star design.

Robust clique: If one seeks to minimise the bandwidth-distance product on a network, the optimal network design is a clique, where traffic only ever has to traverse a single hop. Although this is an obvious simplification of real costs, it provides a simple comparison against the Valiant design, which also uses a clique. The minimal link capacities are given by the corresponding traffic matrix element. However, as noted, we will not use the capacities determined by one traffic matrix, but rather, choose capacities that can carry any of our synthetic traffic matrices.

4.1 Experiment 1: Fixed c and r

We test the five designs with traffic matrices from Abilene [15]. The 2004 network had $N = 12$ PoPs. We draw one week of Abilene traffic matrices, starting from the first week of April 2004, aggregated into 168 one-hour traffic matrices, and we use these matrices as if they were the predicted traffic matrices $\mathbf{T}$. For each predicted matrix, we generate $M = 100$ synthetic matrices (using the procedure described above, assuming the $\mathbf{c}$ and $\mathbf{r}$ given by the matrix are correct), and calculate the total bandwidth $\times$ distance product for the network.

According to the simple optimisation used here, an efficient network would have a low bandwidth $\times$ distance product. When the errors in the prediction are zero (*i.e.*, $\beta = 0$) the clique will produce the optimal design, and so we use the bandwidth-capacity product of this network as a baseline, and measure all other networks as a factor of this capacity, hence the minimum possible value is 1. Thus we see the degree of "over-build" required by each strategy, as a function of the size of the prediction errors.

Figure 3(a) plots the total bandwidth miles (bandwidth $\times$ distance) for the different network designs.

Observe that the Valiant, star and Abilene network designs are constant WRT to β. The overbuild of the Valiant network is slightly over two because our choice of the baseline is not the star design. However, it is within a factor of two compared to the star design (about 1.3), confirming the results in [31]. These designs are two extremes: the first two assume no knowledge, and the second takes no account of errors, so neither changes WRT to β. Obviously, the Abilene strategy needs a much smaller overbuild, at the cost of being sensitive to errors in the inputs.

The second curve to note is the robust clique, whose performance stretches from optimal for $\beta = 0$ (as expected, though it might be surprising that the Abilene design does almost as well), to being quite poor for large β. This type of result is the key motivation for approaches such as Valiant design, which achieves much better performance when there is no information about traffic.

The big surprise is that the robust Abilene design is even more efficient than both of the alternative robust designs when there are significant prediction errors. What is happening here is that the shared nature of many of the main paths through the Abilene network allows errors in the traffic matrix to balance out (a fact that we shall make further use of below). In any case, we can see that for this relatively simple network, the designers did a pretty good job. More to the point, the completely oblivious approach is wasteful, given even moderately accurate data about traffic.

4.2 Experiment 2: Row/column variations

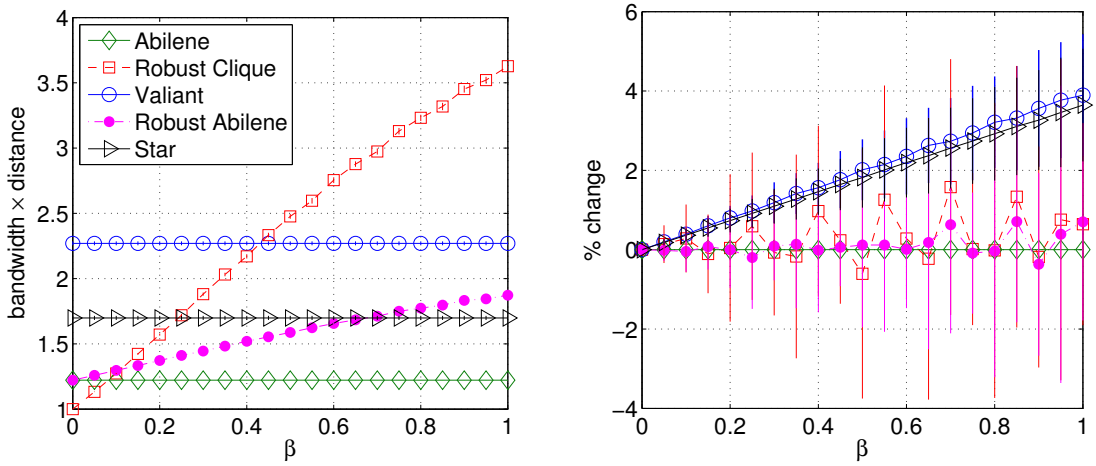
In the Valiant and star designs, the total access capacity at each PoP is assumed known. However, networks do evolve over time, say, as customers upgrade links or depart the network, so the total access capacity will change. Thus, there can be errors in predictions of the edge capacities. As with other designs, both the Valiant and star designs could accommodate such errors by over-engineering, and so in this section we compare robustness of the different designs allowing both variations in the traffic matrix, and variations in the row and column sums.

We chose a simple model for the errors in the predictions of the edge capacities by simply adding IID Gaussian noise to the row and column sums in our constraints. We scale these errors by $\beta/\sqrt{N}$, intentionally simulating that predictions for total incoming and outgoing traffic of a PoP are likely to be more accurate in comparison to predictions of individual traffic matrix elements.

Figure 3(b) plots the %–change WRT to the results in Figure 3(a). Most designs behave almost as before. The prediction errors in rows and columns are small compared to those in the traffic matrices, and so have little effect.

The robust clique has the highest variation, which is expected, because there is greater variation in generated traffic matrices due to the row and column sum variations.

The Valiant and star designs must account for these explicitly to remain oblivious, and hence their overall capacity requirement increase as the errors in the edge capacities increase. Although the star design maintains a total capacity less than twice that of the Valiant, its confidence intervals are larger. Due to prediction errors, the "optimal" central node chosen according to predictions may not be the optimal central node based on the true traffic matrix. One mistake would disrupt the carefully planned traffic load balance.



(a) Fixed row and column sums

(b) Variable row and column sums

Figure 3: Network design tests based on an average of performance over 168 Abilene traffic matrices relative to the optimal design. 95th percentile confidence intervals are given for the data. In (a), in most cases are so small as to be almost unobservable. In (b), the %-change over the fixed case is plotted, with little statistical variation except for the Valiant and star designs.

Both designs demonstrate a near linear relationship with β. These designs allocate capacities proportional to the total T. The assigned capacities would scale linearly with β, since we assign capacities by choosing the design with the maximum total capacity, which is a Gaussian distributed random variable with a standard deviation of $O(\beta)$, due to the prediction errors.

The example illustrates an interesting principle of robustness. Both designs attempt to carefully optimise the network, not against a particular traffic matrix, but all matrices given the edge capacities. They have strong protection against errors in individual elements of the traffic matrix, provided the measurement of edge capacities are accurate. When this assumption is violated, the performance of these designs degrade.

We point out that there are a number of simplifications here in our cost model and system constraints (typical link capacities are symmetric and constrained to take a fixed set of values). However, we have shown how the choice of network design algorithm can be influenced by the nature of the data we have available, and that illustrates very simply the value of being able to generate an ensemble of potential matrices to use in planning.

However, for all these approaches, there is no ability to tradeoff risk and value. Network operators would have to design the network to allow for any possible traffic generated by the model. In reality, operators have different degrees of risk aversion, and we consider network design in that context in the following section.

Moreover, our design approaches are only robustified versions of existing designs, *i.e.*, we choose the link capacities, but don't assess where the links should be. In the next case study we will allow for a full design process.

5. CASE STUDY 2: RISK TRADEOFFS

5.1 Mean-Risk Analysis and Network Design

The fundamental ideas of portfolio analysis trace back to Markowitz's famous work [12]. The objective was to construct a portfolio of financial assets yielding a better overall return compared to the individual assets at lower overall risk by selecting assets whose risk and return profiles are uncorrelated to balanced the total risk of the portfolio. In network design, we might consider multiplexing multiple traffic matrix elements on the same links to reduce the relative variance of the link flows to achieve a similar effect.

Mean-risk analysis is a simple, but useful approach within the (much larger) field of portfolio design. The goal (as suggested by the name) is to optimise the portfolio with respect to the average risk and utility. The assumption is that risk is bad, and so portfolio managers are *risk averse*, but to different degrees, so we aim solve the problem

$$\max_{\mathbf{x}} \quad \mathbb{E}_{\mathbf{y}}[U(\mathbf{x},\mathbf{y})] - \delta\,\mathbb{E}_{\mathbf{y}}[R(\mathbf{x},\mathbf{y})]\,,$$

where $\mathbf{x}$ is a set of decision variables, and $\mathbf{y}$ are the potential outcomes (expectations are across the space of outcomes), and the optimisation may also contain some constraints. The parameter $\delta > 0$ allows the portfolio manager to tune their tradeoff between utility U and risk R. In the context of portfolios, utility is a given function of return (sometimes simply just the return itself), and there are various notions of risk, depending on context, often based on the variance or the standard deviation of the utility function.

It is worth noting, though, that choice of risk metric is non-trivial, and there is a considerable body of work on this factor alone in finance. Here, we propose a rudimentary mean-risk analysis for network design and management. A network operator wants to maximise a certain objective, say, the utilisation of the entire network, given a predicted traffic demand. In this case, the decision variables are the link capacities (which we will denote by $\mathbf{x}$, noting that $x_\ell > 0$ indicates that a link ℓ is present), the outcomes are the L link traffics $\mathbf{y}$, and the utility function is the *total utilisation*

of the network

$$U(\mathbf{x}, \mathbf{y}) = \sum_{x_\ell > 0} y_\ell / x_\ell.$$

Note that high utilisation of the network is important because high utilisation lowers the network cost per unit bit, and increases profits for given transit charges.

However, total utilisation is not the only metric that can be incorporated into the framework. Depending on the operator's objective, other metrics such as total link delay can certainly be included.

Risk is slightly harder to define. Simplistically, the risk here is that the bandwidth will be insufficient to accommodate traffic, resulting in congestion. The cost of congestion is that performance will suffer, and so SLAs may not be satisfied, or disappointed customers will leave the network, but quantification of these risks is difficult, so we aim to simply provide a measure of congestion. In finance, several decades of research has not yielded a perfect risk metric and this suggests there is potential for much further research on network risk metrics. We chose two simple alternatives for this case study, without claiming that either is the perfect choice. The two are:

- *Hard threshold*: $R(\mathbf{x}, \mathbf{y}) = \sum_{x_\ell > 0} [y_\ell / x_\ell - \gamma]^+$, where $[z]^+ = \max(z, 0)$. This measure enforces a hard threshold, with a tuneable *translation* parameter $\gamma > 0$ determining the placement of the threshold.
- *Soft threshold*: $R(\mathbf{x}, \mathbf{y}) = \sum_{x_\ell > 0} y_\ell / (x_\ell - y_\ell)$, though we improve performance by using the piece-wise linear approximation of $R(\mathbf{x}, \mathbf{y})$ developed for optimising OSPF weights [9], a relaxed version of the former. This measure provides more incentive to avoid heavily over-utilised links, and thus balance overall traffic more evenly across the network.

Note that these risk measures deal with correlation in traffic variations in very different ways, and the resulting network design would be reflective of the chosen risk measure. One could imagine all admissible traffic matrices as *stocks* in a portfolio, with link capacities as the *allocation proportion* for the entire network to form a joint portfolio. The hard threshold risk measure caps variations at each link to achieve overall lower utlisation, but the soft threshold allows some links with lower capacity to be sacrificed, *i.e.*, allowing more variations for a small number of links, while ensuring a low global utlisation. The former risk measure is similar to uniformly limiting the volatility of aggregates of stocks, while the latter is akin to proportionally allocating more to the best stocks in a portfolio. This will be clearer in our case study below.

Our optimisation problem is

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbb{E}_{\mathbf{y}}[U(\mathbf{x}, \mathbf{y})] - \delta\, \mathbb{E}_{\mathbf{y}}[R(\mathbf{x}, \mathbf{y})] \\ \text{s.t.} \quad & \mathbf{x} \geq \mathbf{0}, \text{ and } \mathbf{y} = \mathcal{A}_{\mathbf{x}}(\mathbf{S}),\ \mathbf{S} \in \mathcal{S}_{TM}, \end{aligned} \tag{5}$$

where $\mathcal{S}_{TM}$ is the set of admissible traffic matrices, and $\mathcal{A}_{\mathbf{x}}(\mathbf{S})$ represents the link loads resulting from routing the traffic matrix $\mathbf{S}$ across the network with capacities $\mathbf{x}$ ($\mathbf{x} \geq \mathbf{0}$ denotes element-wise non-negativity).

The optimisation is, in theory, performed on the expectation of our metrics over all admissible traffic matrices. Unfortunately, this is impractical as we don't know their true distribution, and even if we did the optimisation would likely be computationally intractable. Instead, our approach is to use our synthesis algorithm to generate an ensemble of traffic matrices, over which the solution to both problems can be approximated. This has a two-fold effect: first, performance of the most likely variations around the predicted traffic matrix can be assessed, and second, the time required to obtain a solution can be controlled by the number of generated traffic matrices.

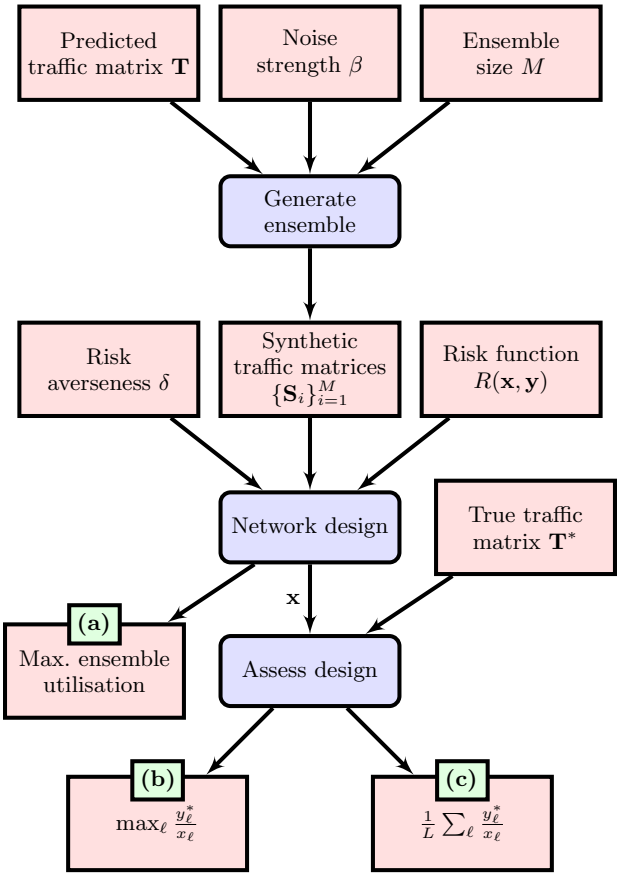


Figure 4: Flow chart of the design methodology with mean and risk tradeoffs. The rectangles denote the inputs and outputs at each stage, while the rounded rectangles are the stages in the methodology. Here, y^*_ℓ denotes the load on link ℓ with the true matrix $\mathbf{T}^*$. Labels on outputs match to Figures 5 and 6.

5.2 Experiments

We test the idea presented above as shown in Figure 4, though the reader should note that this is not intended to be a full exposition on network optimisation, but rather an example of how to use the synthetic matrices.

We start by using a predicted traffic matrix $\mathbf{T}$ to generate an ensemble of traffic matrices $\{\mathbf{S}_i\}_{i=1}^M$ with noise strength β. In order to make the scenario somewhat realistic, we will once again use the traffic matrix data from Abilene [15] (from 0000 to 0005 hours on the 1st March 2004), but this time the network designers do not have access to the true traffic matrix. They are only given a forecast on the incoming and outgoing traffic of the network, and they form a predicted matrix $\mathbf{T}^*$ using the gravity model [29]. This is the

natural estimate given no other information [30], though it certainly does contain errors [29]. The tests below are over a range of 40 values of $\beta \in (0, 0.4]$, each with $M = 100$ matrices.

The real errors between $\mathbf{T}$ and $\mathbf{T}^*$ are known to us and we can calibrate the meaning of the results. The real MRSE is 1.33, which corresponds to $\beta_{err} \simeq 0.17$, well within the approximate linearity range, *i.e.*, $\beta \leq 0.2$. Note $\mathbf{T}^*$ here is not the same as that in Figure 2 and β_{err} was computed using approximation (3).

Once we have an ensemble of possible matrices, we perform the network design. We do so for the two risk functions described above (with $\gamma = 1$ in the hard threshold), and a range of risk averse parameters δ. Optimisation was performed by using the Matlab function `fmincon`. Despite the non-convex nature of the optimisation problem, the time taken to generate the 100 synthetic matrices and compute the optimal capacity design was approximately 2 minutes on a machine with a 3.06 GHz processor and 4 GB of RAM.

As the optimisation problem is non-convex, there is the possibility that the optimisation performs badly, so our first set of figures tests the quality of the optimisation. Although we optimise against total utilisation and risk, we assess performance here against maximum utilisation. We do so for two reasons: (i) the maximum provides a worst case measure of how badly the overall design can fail, and (ii) we wish to avoid the circularity of testing against exactly the same criteria against which we optimise. Figures 5(a) and 6(a) show the maximum utilisation across the whole ensemble. One immediately notes that for all δ, this increases almost linearly with β. This is entirely reasonable – as the variety in the ensemble increases (approximately linearly with β), the optimisation finds it harder to both ensure high average utilisation, and low maximum utilisation, and the maximum utilisation increases.

Importantly, however, the maximum utilisation decreases as we become more risk averse, and it is precisely this effect we are trying to achieve by tuning the parameter δ.

The performance of the network designs on the true traffic matrix is more important than the performance across the artificial ensemble. Figures 5 and 6 plots (b) and (c) show two performance metrics: the maximum and average utilisations for the network design, given the true Abilene traffic matrix $\mathbf{T}^*$. The maximum is a worst case measure of the risk (lower is better), and the average utilisation is a measure of the utility achieved (higher is better).

The maximum utilisation (for the true matrix) generally decreases with increasing risk aversion, though surprisingly there is little difference for the hard threshold between $\delta = 5$ and 10. As δ increases, we see that the average utilisations decrease. This is simply because as we become more risk averse, we provide more excess capacity to provide for inaccuracies, and so overall utilisation decreases.

Moreover, the maximum utilisation roughly decreases as β increases. Increasing β indicates that we have less confidence in our estimate, and so again we need to allow more *headroom* to allow for errors in predictions. Consequently we see that as β increases the average utilisation decreases.

However, note that we aren't just providing extra capacity to account for errors. As the parameters δ and β change, the actual network designs change. When we are more risk averse, or less sure about the accuracy of predictions, it makes sense to multiplex larger streams of traffic onto fewer links to reduce the relative variance of the link loads. We can see this in the non-smooth, non-linear nature of these functions. We can also see this in the fact that for the hard threshold, and moderate degrees of risk aversion, *e.g.*, $\delta = 5$, the average network utilisation is relatively insensitive to β, even as the maximum utilisation decreases. This is highly desirable, as we might only be able to estimate the prediction errors approximately.

The choice of risk metric obviously has a large effect on the results, though neither is universally *better*.

The Fortz-Thorup risk penalises heavy over-utilisation, but this depends on the load of the link. For links with small loads, a compromise is accepted where these links are allowed to be congested, as long as the overall network has a favourable average utilisation. Links with large loads matter more under this risk measure, so their utilisation is kept below 1. This can be seen in Figure 6(b) when β is small with $\delta = 1$, where the maximum utilisation with $\mathbf{T}^*$ only stabilises after $\beta = 0.1$, as the traffic matrix has a proportion of small elements. Once β increases, elements of the synthetic ensemble get larger, so the subsequent designs are over-provisioned as there are less links with small loads that can be "sacrificed". Thus, designs using this risk metric maintain average utilisation below 1, but may sacrifice some links in an effort to do so.

In contrast, the hard threshold penalises all overloads linearly, without normalising according to load size. Designs using the hard threshold would attempt to keep maximum utilisation near 1 for all links. So we see a more uniform response of the maximum and average utilisation across β when tested on $\mathbf{T}^*$.

Overall, for most given network designs the Fortz-Thorup risk is therefore larger, forcing both the maximum and average utilisations lower for a given δ. It also shows more sensitivity to the parameter β for maximum utilisation, and less for average, due to its sensitivity to the more extreme utilisations resulting from the traffic matrix ensemble.

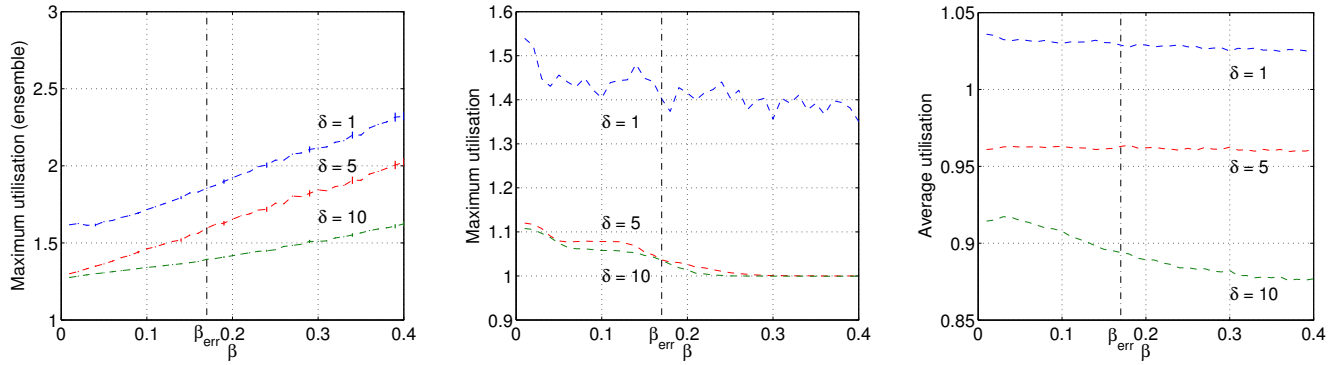
Although increasing β and δ have somewhat similar effects (to decrease the risk, and to compensate, the overall utilisation also decreases), the parameters have different meanings. The parameter β is a technical measure of the accuracy of our predictions, while the parameter δ is a business parameter expressing the degree of our concern about congestion.

Finally, the main point of this given the actual prediction errors for Abilene, $\beta_{err} \simeq 0.17$, is that we could use the above to start to choose between designs. Given δ we can examine for the given risk metrics the designs based on a prediction, and use these to build a network robust against prediction errors, while still as efficient as possible.

6. CONCLUSION AND FUTURE WORK

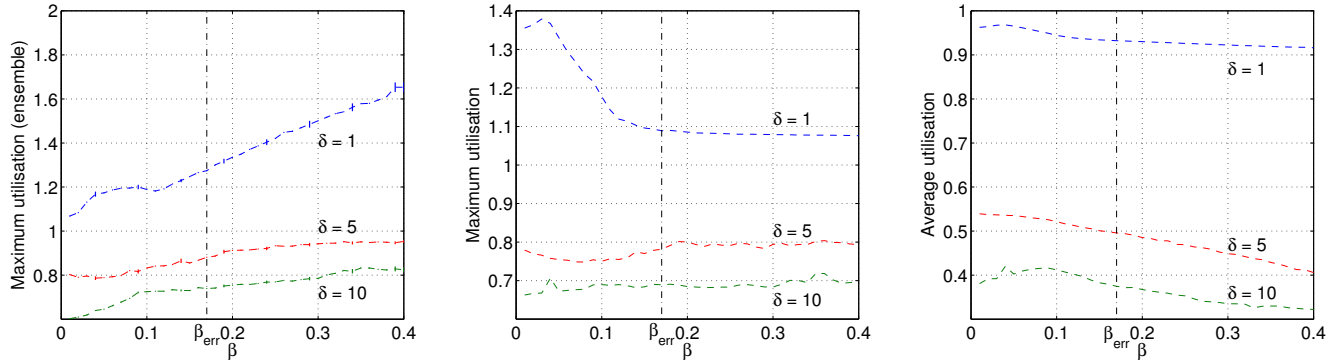
In this paper, we examined the generation of synthetic traffic matrices for sensitivity analysis of network design algorithms. The main point we aim to get across is that all measurements contain errors and likewise with predictions. To further compound the problem, there are no "valid" traffic matrices and obtaining a ground truth to evaluate a network design with is nearly impossible. Thus, we advocate using fast, simple traffic matrix synthesis models so as to allow testing of network designs under variations around the predicted traffic matrix.

The results in our paper demonstrate the usefulness of our approach. We can use our synthetic traffic matrices to



(a) Maximum utilisation over ensemble (b) Maximum utilisation (c) Average utilisation

Figure 5: Evaluating the network design with risk averseness $\delta = \{1, 5, 10\}$ using the hard threshold risk, with $\gamma = 1$, in the network optimisation for (a) the maximum utilisation over an ensemble of $M = 100$ synthetic matrices per data point, (b) the maximum utilisation and (c) average utilisation, when the true traffic matrix, a single snapshot taken from Abilene in 2004, is used in the network. $\beta_{err} = 0.17$ denotes the required noise strength to match the MRSE with respect to the true matrix $\mathbf{T}^*$.



(a) Maximum utilisation over ensemble (b) Maximum utilisation (c) Average utilisation

Figure 6: Evaluating the network design with risk averseness $\delta = \{1, 5, 10\}$ using the Fortz–Thorup risk in the network optimisation for (a) the maximum utilisation over an ensemble of $M = 100$ synthetic matrices per data point, (b) the maximum utilisation and (c) average utilisation, when the true traffic matrix, a single snapshot taken from Abilene in 2004, is used in the network. $\beta_{err} = 0.17$ denotes the required noise strength to match the MRSE with respect to the true matrix $\mathbf{T}^*$.

understand network designs better, such as Valiant's lack of sensitivity to variations in incoming and outgoing traffic. Moreover, we can design networks with a degree of robustness to input errors. We have shown how robust versions of existing network designs can often be more resilient when faced with significant changes in the traffic matrix.

We further advanced a mean-risk analysis framework for network design. The rationale is that different network operators have different risk averseness. We used simple utility and risk functions, coupled with our traffic matrix synthesis algorithm to assign capacities to links of a network. These ideas were tested on real data and simulations. Future work will further extend and validate the framework with more traffic snapshots, and study other risk measures.

Sensitivity of network design algorithms to errors is important but is often ignored. If sensitivity and robustness to error receive more care, network optimisations might receive more attention from the operator community.

7. ACKNOWLEDGEMENTS

The authors wish to thank the anonymous reviewers for their useful feedback. This work was supported by the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS).

APPENDIX

A. IPF MINIMISES KL DIVERGENCE

Throughout the discussion, we consider the application of IPF to a general matrix $\mathbf{Y} \in \mathbb{R}^{M\times N}$. It has been claimed that IPF minimises the KL divergence [11], but we are not aware of a proof of this claim, so we include our own proof for reference.

Let us define the matrix generalisation of the KL divergence [10] as

$$D_{GKL}(\mathbf{X}\,||\,\mathbf{Y}) = \sum_{i,j} x_{i,j} \log \frac{x_{i,j}}{y_{i,j}} - \sum_{i,j} x_{i,j} + \sum_{i,j} y_{i,j}, \quad (6)$$

between the matrices $\mathbf{X}$ and $\mathbf{Y}$, reducing to the canonical KL divergence when $x_{i,j},\ y_{i,j} \in [0,1]$ for all i,j and $\sum_{i,j} x_{i,j} = \sum_{i,j} y_{i,j} = 1$, *i.e.*, probabilities. Here, we use the standard convention $\log(0/0) = 0$.

Given an initial matrix $\mathbf{Y}$, IPF solves the following optimisation problem successively for each iteration k, $k = 1, 2, \cdots, K$ (assume convergence to the solution at K, according to an appropriate criterion):

$$\mathbf{X}^\star = \underset{\mathbf{X}\in\mathbb{R}^{M\times N}}{\operatorname{argmin}} \quad D_{GKL}(\mathbf{X}\,||\,\mathbf{Y}), \quad (7)$$

subject to the constraint set (recall the set in Section 3)

$$x_{i,j} \geq 0,\ \forall i,j = 1,2,\cdots,N, \quad (8)$$

$$\sum_j x_{i,j} = r_i,\ \forall i, \quad (9)$$

$$\sum_i x_{i,j} = c_j,\ \forall j, \quad (10)$$

$$\sum_{i,j} x_{i,j} = \sum_i r_i = \sum_j c_j = T. \quad (11)$$

The objective function is convex in $\mathbf{X}$ and the constraints form a convex set, hence, the optimisation problem is convex. Since the objective function is strictly convex, there is one global minimum with a single optimal point. Assume $\mathbf{Y}$ is non-negative, so the non-negativity constraint is inactive.

To prove the claim, we define the Lagrangian with multipliers $\boldsymbol{\lambda} \in \mathbb{R}^N$, $\boldsymbol{\mu} \in \mathbb{R}^M$, $\psi \in \mathbb{R}$,

$$\begin{aligned} L(\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu},\psi;\mathbf{Y}) := &\sum_{i,j} x_{i,j} \log \frac{x_{i,j}}{y_{i,j}} - \sum_{i,j} x_{i,j} + \sum_{i,j} y_{i,j} \\ &+ \sum_j \lambda_j \Big(\sum_i x_{i,j} - c_j \Big) \\ &+ \sum_i \mu_i \Big(\sum_j x_{i,j} - r_i \Big) + \psi \Big(\sum_{i,j} x_{i,j} - T \Big). \end{aligned} \quad (12)$$

Differentiating with respect to $x_{i,j}$, $\forall i,j$, λ_j, $\forall j$ and μ_i, $\forall i$, and ψ,

$$\frac{\partial L(\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu},\psi;\mathbf{Y})}{\partial x_{i,j}} = \log x_{i,j} - \log y_{i,j} + \lambda_j + \mu_i + \psi \quad (13)$$

$$\frac{\partial L(\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu},\psi;\mathbf{Y})}{\partial \lambda_j} = \sum_i x_{i,j} - c_j \quad (14)$$

$$\frac{\partial L(\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu},\psi;\mathbf{Y})}{\partial \mu_i} = \sum_j x_{i,j} - r_i \quad (15)$$

$$\frac{\partial L(\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\mu},\psi;\mathbf{Y})}{\partial \psi} = \sum_{i,j} x_{i,j} - T. \quad (16)$$

Setting the derivatives to 0, we find the last three return the column, row and total sum constraints. Solving equation (13) yields the fixed point equation

$$x_{i,j} = \eta_i \nu_j y_{i,j},\ \forall i,j \quad (17)$$

where $\eta_i = \exp(-\mu_i - \psi_i)$, $\nu_j = \exp(-\lambda_j - \psi_j)$ and $\psi_i + \psi_j = \psi$, $\forall i,j$.

One way to find the optimal point is to iteratively alternate between solving the Langrangian multipliers, first holding ν_j fixed, then holding η_i fixed. One does this by taking constraints (9) and (10) into account to solve for the multipliers. Beginning from an initialisation $\mathbf{X}^{(0)} = \mathbf{Y}$, the steps are

$$\begin{aligned} \eta_i^{(k-1/2)} &= \frac{r_i}{\sum_j x_{i,j}^{(k-1)} \nu_j^{(k-1)}}, \forall i, \\ \nu_j^{(k)} &= \frac{c_j}{\sum_i x_{i,j}^{(k-1)} \eta_i^{(k-1/2)}}, \forall j. \end{aligned}$$

Substituting the above into the fixed point equation (17), this is equivalent to iteratively solving the equations

$$x_{i,j}^{(k-1/2)} = \frac{r_i}{\sum_j x_{i,j}^{(k-1)}}\ x_{i,j}^{(k-1)}, \forall i, \quad (18)$$

$$x_{i,j}^{(k)} = \frac{c_j}{\sum_i x_{i,j}^{(k-1)}}\ x_{i,j}^{(k-1/2)}, \forall j. \quad (19)$$

This is precisely what IPF does. Speedup is possible by replacing the denominator in (19) with $\sum_i x_{i,j}^{(k-1/2)}$ instead, so as to propagate new updates as soon as possible.

B. ASYMPTOTICS OF THE SANM

Here we obtain intuition on the average behaviour of the Spherically Additive Noise Model (SANM) in the limit of large β. The limit corresponds to the case where we have very inaccurate traffic matrix predictions. The matrices must still satisfy admissibility constraints: non-negativity, row and columns sums, and total, but otherwise we have essentially no information about these matrices. In this limit, it would make sense to use a maximum entropy model, that is, to use a model that postulates the minimum additional assumptions about the traffic matrix, in the absence of any prior beliefs beyond the explicit constraints. Prior work [**?**,30] has shown that the maximum entropy model for a traffic matrix under these constraints is the gravity model. We indeed found that this was a reasonable approximation for the limiting distribution of our matrices, and so we seek to explain it intuitively in this appendix.

We are interested in $\mathbb{E}[\mathbf{S}]$ when $\beta \to \infty$. The major complication is the nonlinear action of IPF on $\mathbf{Y}$, making it difficult to analyse the output of IPF. Moreover, since the input to the generalised KL divergence are random matrices, $D_{GKL}(\mathbf{X}\,||\,\mathbf{Y})$ is itself a random quantity (in this sense, it is not the same as the canonical KL divergence), the solution of (7) is only deterministic conditioned on knowing $\mathbf{Y}$.

Recall that IPF is performing the optimisation specified in (7)-(11), so

$$\mathbb{E}[\mathbf{S}] = \mathbb{E}\left[\underset{\mathbf{X}\in\mathbb{R}^{N\times N}}{\operatorname{argmin}}\ D_{GKL}(\mathbf{X}\,||\,\mathbf{Y}) \right], \quad (20)$$

subject to the usual admissibility constraints. On the other hand, we will find it easier to work with

$$\tilde{\mathbf{S}} = \underset{\mathbf{X}\in\mathbb{R}^{N\times N}}{\operatorname{argmin}} \; D_{GKL}\big(\mathbf{X} \,\big\|\, \mathbb{E}[\mathbf{Y}]\big), \tag{21}$$

subject to the admissibility constraints, which is the solution of the optimisation problem for the average matrix $\mathbb{E}[\mathbf{Y}]$.

The two are not equal, but we can argue that they are close in the limit. First, note that the $y_{i,j}$ are IID random variables, so $\mathbf{Y}$ is composed of an ensemble of N^2 IID random variables. The Asymptotic Equipartition Theorem [**?**, Ch. 3], guarantees that asymptotically the *typical* set of instances of $\mathbf{Y}$ will have probability near one, and elements of this set will have approximately constant probability. So we can loosely think of the ensemble $\mathbf{Y}$ as being largely composed of typical instances, each equiprobable. The constraints in the problem are linear, so the space onto which we are projecting is a linear subspace of $\mathbb{R}^{N\times N}$. Moreover, the optimisation is to minimise the "distance" to this subspace, and so we can think of it as projecting the ensemble $\mathbf{Y}$ onto the subspace. Finally expectation is a linear operator itself, so for N^2 large and $\beta \to \infty$, we have $\tilde{\mathbf{S}} \sim \mathbb{E}[\mathbf{S}]$.

As $\beta \to \infty$, the predicted traffic matrix has an insignificant effect on the matrix $\mathbf{Y}$, so we find that the $y_{i,j}$ are distributed as χ^2 random variables, with one degree of freedom, and that $\mathbb{E}[y_{i,j}] \simeq \beta^2$, *i.e.*, (1) leads to

$$y_{i,j} \simeq \beta^2 z_{i,j}^2, \tag{22}$$

where $z_{i,j} \sim \mathcal{N}(0,1)$, and hence $\mathbb{E}[\mathbf{Y}] = \beta^2 \mathbf{1}_N \mathbf{1}_N^{\mathrm{T}}$, where $\mathbf{1}_N$ is the vector of N ones.

Using the previous section, the effect of applying IPF to $\mathbb{E}[\mathbf{Y}]$ is to solve the optimisation problem specified in (7)-(11). That is, $\tilde{\mathbf{S}}$ is the $\mathbf{X}$ that minimises $D_{GKL}(\mathbf{X} \,||\, \beta^2 \mathbf{1}_N \mathbf{1}_N^{\mathrm{T}})$, or

$$\sum_{i,j} x_{i,j} \log x_{i,j} - \Big(\sum_{i,j} x_{i,j}\Big) \log \beta^2 - \sum_{i,j} x_{i,j} + N^2\beta^2.$$

When we include Lagrange multipliers for each constraint, the problem becomes that of minimising

$$\begin{aligned} L(\mathbf{X}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \psi) := & \sum_{i,j} x_{i,j} \log x_{i,j} - \Big(\sum_{i,j} x_{i,j}\Big) \log \beta^2 \\ & - \sum_{i,j} x_{i,j} + N^2\beta^2 + \sum_j \lambda_j \Big(\sum_i x_{i,j} - c_j\Big) \\ & + \sum_i \mu_i \Big(\sum_j x_{i,j} - r_i\Big) + \psi\Big(\sum_{i,j} x_{i,j} - T\Big). \end{aligned}$$

Differentiating with respect to $x_{i,j}$, $\forall i, j$, λ_j, $\forall j$ and μ_i, $\forall i$,

$$\frac{\partial L(\mathbf{X}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \psi)}{\partial x_{i,j}} = \log x_{i,j} - \log \beta^2 + \lambda_j + \mu_i + \psi, \tag{23}$$

where we omitted the last three equations (14)-(16). Setting the above equation to 0 and solving equation (23), yields the fixed point equation

$$s_{i,j} = \eta_i \nu_j \beta^2, \;\; \forall i, j \tag{24}$$

where $\eta_i = \exp(-\mu_i - \psi_i)$, $\nu_j = \exp(-\lambda_j - \psi_j)$ and $\psi_i + \psi_j = \psi$, $\forall i, j$.

In order to find the optimal solution, we use the constraint information. Substituting (24) into the constraints,

$$\beta^2 \eta_i \sum_j \nu_j = r_i, \forall i \;\; \textit{(row constraints)} \tag{25}$$

$$\beta^2 \nu_j \sum_i \eta_i = c_j, \forall j \;\; \textit{(column constraints)} \tag{26}$$

$$\beta^2 \sum_{i,j} \eta_i \nu_j = T, \;\; \textit{(total sum constraint)}. \tag{27}$$

By the total sum constraint (27), since (27) is equal to $\beta^2(\sum_i \eta_i)(\sum_j \nu_j)$,

$$\sum_i \eta_i = \frac{T}{\beta^2 \sum_j \nu_j}.$$

Substituting this into the row constraints (25), we obtain for all i

$$\frac{\eta_i}{\sum_i \eta_i} = \frac{r_i}{T}.$$

Similarly, using the total sum constraint (27), we can obtain from (26)

$$\frac{\nu_j}{\sum_j \nu_j} = \frac{c_j}{T}.$$

By rewriting (24), for all i, j, the optimal solution is

$$\begin{aligned} \tilde{s}_{i,j} &= \left(\Big(\sum_i \eta_i\Big)\Big(\sum_j \nu_j\Big)\beta^2\right)\left(\frac{\eta_i}{\sum_i \eta_i}\right)\left(\frac{\nu_j}{\sum_j \nu_j}\right) \\ &= T\left(\frac{r_i}{T}\right)\left(\frac{c_j}{T}\right) = \frac{r_i c_j}{T}. \end{aligned}$$

Thus, $\tilde{\mathbf{S}} = \mathbf{r}\mathbf{c}^{\mathrm{T}}/T$, *i.e.*, the *gravity model*, and as discussed earlier, will be close to $\mathbb{E}[\mathbf{S}]$ as $\beta \to \infty$. Intuitively, once the noise overwhelms any prior beliefs of the network's traffic obtained from the predicted matrix, the IPF defaults to a solution with maximum entropy, which matches the empirical results.

C. REFERENCES

[1] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *ACM SIGCOMM*, pages 313–324, August 2003.

[2] G. E. P. Box and N. R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley, 1987.

[3] R. S. Cahn. *Wide Area Network Design*. Morgan Kaufman, 1998.

[4] W. E. Deming and F. F. Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Ann. Math. Stat*, 11(4):427–444, 1940.

[5] B. Eriksson, R. Durairajan, and P. Barford. RiskRoute: A framework for mitigating network outage threats. In *CoNEXT '13*, pages 405–416, December 2013.

[6] V. Erramilli, M. Crovella, and N. Taft. An independent-connection model for traffic matrices. In *ACM IMC 2006*, pages 251–256, October 2006.

[7] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving demands for operational IP networks: Methodology and experience. *IEEE/ACM Trans. Networking*, 9:265–280, June 2001.

[8] S. E. Fienberg. An iterative procedure for estimation in contingency tables. *Ann. Math. Stat.*, 41(3):907–917, 1970.

[9] B. Fortz and M. Thorup. Optimizing OSPF/IS–IS weights in a changing world. *IEEE J. Selected Areas in Communications*, 20(4):756–767, May 2002.

[10] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[11] G. Liang, B. Yu, and N. Taft. A fast lightweight approach to origin-destination IP traffic estimation using partial measurements. *IEEE/ACM Trans. Networking*, 14:2634–2648, June 2006.

[12] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952.

[13] A. Medina, N. Taft, K. Salmatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *ACM SIGCOMM*, 2002.

[14] D. Mitra and Q. Wang. Stochastic traffic engineering for demand uncertainty and risk-aware network revenue management. *IEEE/ACM Trans. Networking*, 13(2):221–233, April 2005.

[15] NLANR. Abilene Trace Data. http://pma.nlanr.net/Special/ipls3.html.

[16] I. Norros. On the use of fractional Brownian motion in the theory of connectionless networks. *IEEE Journal on Selected Areas in Communications*, 13(6):953–961, 1995.

[17] A. Nucci, A. Sridharan, and N. Taft. The problem of synthetically generating IP traffic matrices: Initial recommendations. *SIGCOMM Comput. Commun. Rev.*, 35:19–32, July 2005.

[18] M. Roughan. Simplifying the synthesis of Internet traffic matrices. *SIGCOMM Comput. Commun. Rev.*, 35(5):93–96, 2005.

[19] M. Roughan. Robust network planning. In C. R. Kalmanek, S. Misra, and R. Yang, editors, *The Guide to Reliable Internet Services and Applications*, chapter 5, pages 137–177. Springer, 2010.

[20] M. Roughan and J. Gottlieb. Large-scale measurement and modeling of backbone Internet traffic. In *SPIE ITCom*, Boston, 2002.

[21] 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 *ACM IMW*, 2002.

[22] M. Roughan, M. Thorup, and Y. Zhang. Traffic engineering with estimated traffic matrices. In *ACM IMC 2003*, pages 248–258, October 2003.

[23] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2), 1967.

[24] 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. *SIGMETRICS Perform. Eval. Rev.*, 32(1):73–84, June 2004.

[25] R. Teixera, N. Duffield, J. Rexford, and M. Roughan. Traffic matrix reloaded: Impact of routing changes. In *Workshop on Passive and Active Measurements*, 2005.

[26] P. Tune and M. Roughan. Internet traffic matrices: A primer. In H. Haddadi and O. Bonaventure, editors, *Recent Advances in Networking, Vol. 1*. ACM SIGCOMM, August 2013.

[27] P. Tune and M. Roughan. Network-design sensitivity analysis. Technical report, School of Mathematical Sciences, University of Adelaide, January 2014.

[28] Y. Vardi. Network Tomography: Estimating source-destination traffic intensities from link data. *J. Am. Statist. Assoc.*, 91:365–377, 1996.

[29] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *ACM SIGMETRICS 2003*, pages 206–217, 2003.

[30] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. Estimating Point-to-Point and Point-to-Multipoint traffic matrices: An information-theoretic approach. *IEEE/ACM Trans. Networking*, 13(5):947–960, October 2005.

[31] R. Zhang-Shen and M. McKeown. Designing a predictable Internet backbone with Valiant load-balancing. In *Proceedings of the 13th International Conference on Quality of Service (IWQoS)*, pages 178–192, 2005.