RC24807 (W0906-051) June 12, 2009
Mathematics

IBM Research Report

Improved Approximations for the Erlang Loss Model

J. Anselmi
INRIA and LIG Laboratory
MontBonnot Saint-Martin, 38330
France

Y. Lu, M. Sharma, M. S. Squillante
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Improved Approximations for the Erlang Loss Model

J. Anselmi*

INRIA and LIG Laboratory
MontBonnot Saint-Martin, 38330, FR
jonatha.anselmi@imag.fr

Y. Lu, M. Sharma, M.S. Squillante
Mathematical Sciences Department
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598, USA
{yingdong,mxsharma,mss}@us.ibm.com

Abstract

Stochastic loss networks are often a very effective model for studying the random dynamics of systems requiring simultaneous resource possession. Given a stochastic network and a multi-class customer workload, the classical Erlang model renders the stationary probability that a customer will be lost due to insufficient capacity for at least one required resource type. Recently a novel family of *slice* methods has been proposed by Jung *et al* to approximate the stationary loss probabilities in the Erlang model, and has been shown to provide better performance than the classical Erlang fixed point approximation in many regimes of interest. In this paper, we propose some new methods for loss probability calculation. We propose a refinement of the 3-point slice method of Jung *et al* which exhibits improved accuracy especially when heavily loaded networks are considered, at comparable computational cost. Next we exploit the structure of the stationary distribution to propose randomized algorithms to approximate both the stationary distribution and the loss probabilities. Whereas our refined slice method is exact in a certain scaling regime and is therefore ideally suited to the asymptotic analysis of large networks, the latter algorithms borrow from volume computation methods for convex polytopes to provide approximations for the unscaled network with error bounds as a function of the computational costs.

1 Introduction

Starting with the seminal work of Erlang [4], stochastic loss networks (SLNs) have been applied to the study of many diverse communication and computer systems. In general, a SLN can be used as a model for any system which allocates non-idling resource capacities to fulfill various classes of requests if possible, each class requiring simultaneous possession of resources for a random duration from a pool of different types of resources. And quite often a SLN is able to effectively capture the dynamics and uncertainty of the computer/communication system being modeled. Examples include telephone networks, communication networks, distributed computing systems, database systems, data centers, wireless networks, and multi-item inventory systems; see, e.g., [16, 7, 21, 8, 9, 10, 18, 15, 22, 17].

An important performance measure arising in the analysis of such loss networks is the stationary loss probability for each customer class. Given a stochastic network and a multi-class customer workload, the classical Erlang model renders the stationary probability that a customer will be lost due to insufficient capacity for at least one required resource type. While the initial results of Erlang [4] were for the particular case of Poisson arrivals and exponential service times, Sevastyanov [19] demonstrates that the Erlang formula holds under general finite-mean distributions for

*This work is supported by the Conseil Régional Rhône-Alpes, Global competitiveness cluster Minalogic contract SCEPTRE.

the customer service times. Subsequently the formula has been shown to hold under more general conditions [3, 2]. A multi-period version of the loss network has also been recently studied [1].

Unfortunately, the problem of evaluating the exact (multi-dimensional) Erlang formula is known to be $\sharp P$-complete[1] in the size of the network [12], thus rendering the exact formula of limited use for many large networks in practice. The well-known Erlang fixed-point approximation (EFPA) has been developed and extensively used and studied as a tractable approach for calculating the stationary loss probabilities. The method assumes that customer losses are caused by independent blocking events on each of the resources used by that customer. Estimates of the blocking probabilities of the individual resources are in turn calculated using the one-dimensional Erlang formula with arguments that are functions of the blocking probabilities of other resources. Surprisingly, the EFPA has been shown to be asymptotically exact in the limiting regime where arrival rates grow in proportion to the resource capacities [8]. Roughly speaking, this result holds true because in that limiting regime the mean number of calls in the system can be well approximated by the mode of the stationary distribution of the network states.

The popularity of the EFPA can be attributed to the fact that, in addition to its favorable theoretical properties, the estimates provided by the method have been found to be remarkably accurate in traditional application areas such as large communication networks. Recently, SLNs have been used for resource planning within the context of workforce management in the information technology (IT) services industry, where a collection of IT service products are offered each requiring a set of resources with certain capabilities [1, 14]. In such applications, one is frequently confronted with critically loaded workload processes for which the EFPA performs poorly [6]. A novel family of *slice* methods has been proposed in [6] to compute the stationary loss probabilities using Little's law and a decomposition of the computation along hyperplane sections that are parallel to the route axes (called slices), i.e., subsets of the network states where the number of calls on a given route is kept fixed. It is shown that the idea of approximating the probability content of each slice by its mode indeed yields more accurate results, especially in critically loaded situations [5]. The general slice method solves a strictly convex program along each slice of each route; thus its computational complexity depends on the resource capacities. A 3-point slice method is also proposed in [6] which uses a suitable interpolation scheme to reduce computational complexity at a small loss of accuracy.

In this paper, we propose some new methods for loss probability calculation. First we present a refinement of the 3-point slice method which provides improved accuracy especially when heavily loaded networks are considered. This new approach has nearly the same computational complexity of the 3-point slice method. Next we exploit the structure of the stationary distribution to propose randomized *contour* algorithms to approximate both the stationary distribution and the loss probabilities. The contour algorithms borrow from volume computation methods for convex polytopes to approximate the probability content of carefully chosen contours. The two sets of proposed algorithms differ in their applicability: whereas the refined slice method is asymptotically exact in a certain scaling regime and is therefore ideally suited to the asymptotic analysis of large networks, the contour algorithms provide approximations corresponding to the unscaled network with error bounds as a function of the computational cost. Hence, there is a tradeoff between computational complexity and (probabilistic) accuracy guarantees in choosing among our refined slice method and our contour methods.

[1] Refer to [20] for details on the $\sharp P$-complete complexity class.

2 Preliminaries

2.1 Erlang Loss Model

Consider a SLN with J links, labeled $1, 2, \ldots, J$, and a set of K fixed routes, denoted by $\mathcal{R} = \{1, \ldots, K\}$, where each link j has C_j units of capacity and $\underline{C} = (C_1, \ldots, C_J)$ denotes the vector of link capacities. Calls on route r arrive according to an independent Poisson process of rate ν_r, with $\underline{\nu} = (\nu_1, \ldots, \nu_K)$ denoting the vector of these rates, and require A_{jr} units of capacity on link j, $A_{jr} \geq 0$. An arriving call on route r is admitted to the network if sufficient capacity is available on all links used by route r; otherwise, the call is lost. The call service times are i.i.d. and follow a general distribution with unit mean. Note, however, that our results are not limited to this unit mean assumption since the quantities of interest remain unchanged in the stationary regime for general traffic intensities.

Let $\underline{n}(t) = (n_1(t), \ldots, n_K(t)) \in \mathbb{Z}_+^K$ be the vector of the number of active calls in the network at time t. By definition, $\underline{n}(t) \in \mathcal{S}(\underline{C})$ where

$$\mathcal{S}(\underline{C}) = \left\{\underline{n} \in \mathbb{Z}_+^K : A\underline{n} \leq \underline{C}\right\}.$$

As Erlang originally established, followed by extensions of various researchers, it is well known that there is a unique stationary distribution π on the state space $\mathcal{S}(\underline{C})$ such that for $\underline{n} \in \mathcal{S}(\underline{C})$

$$\pi(\underline{n}) = G(\underline{C})^{-1} \prod_{r \in \mathcal{R}} \frac{\nu_r^{n_r}}{n_r!},$$

where $G(\underline{C})$ is the normalizing constant

$$G(\underline{C}) = \sum_{\underline{n} \in \mathcal{S}(\underline{C})} \prod_{r \in \mathcal{R}} \frac{\nu_r^{n_r}}{n_r!}. \tag{1}$$

The stationary probability that a call on route r is lost can be expressed as

$$L_r \;=\; 1 \;-\; G(\underline{C})^{-1} G(\underline{C} - A\underline{e}_r),$$

where $\underline{e}_r$ is the unit vector corresponding to a single active call on route r.

2.2 Erlang Fixed-Point Approximation

Due to the computational complexity of (1), which is known to be $\sharp P$-complete in the size of the network [12], the EFPA has been long used as a more efficient alternative to the exact Erlang loss formula. The EFPA is based on approximating the stationary blocking probabilities of the individual links, denoted by E_j, by the following set of equations:

$$E_j = E(\rho_j, C_j) = \frac{\rho_j^{C_j}}{C_j!} \left[\sum_{i=0}^{C_j} \frac{\rho_j^i}{i!}\right]^{-1} \qquad \text{where} \qquad \rho_j = \frac{1}{1 - E_j} \left[\sum_r \nu_r A_{jr} \prod_i (1 - E_i)^{A_{ir}}\right].$$

Then the stationary loss probability for route r can be approximated in terms of the per-link blocking probabilities as

$$L_r \;\approx\; 1 - \prod_j (1 - E_j)^{A_{jr}}.$$

The approximation assumes that lost calls are caused by independent blocking events on each of the links used by the routes, and uses the one-dimensional Erlang loss formula for each link with an appropriately thinned arrival rate.

The EFPA can provide relatively poor estimates for the per-route loss probabilities L_r in various model instances. In [6], it is shown that this is particularly the case when the network is operating in the critically loaded regime [5]. This motivated the authors to investigate alternative approximation algorithms for the exact loss probabilities and to develop the family of slice methods.

2.3 Slice Methods

The Erlang loss model can be thought of as a stable system where admitted calls experience an average delay of 1 and lost calls experience a delay of 0. Hence, the average delay experienced by calls on route r is given by

$$D_r = (1 - L_r) \times 1 + L_r \times 0 \;=\; (1 - L_r),$$

which together with Little's law [11] yields

$$1 - L_r = \frac{\mathbb{E}[n_r]}{\nu_r}, \tag{2}$$

and therefore L_r can be obtained through $\mathbb{E}[n_r]$. By definition,

$$\mathbb{E}[n_r] = \sum_{k=0}^{\infty} k \Pr[n_r = k]$$

and thus $\mathbb{E}[n_r]$ can be obtained through approximations of $\Pr[n_r = k]$. Note that $\Pr[n_r = k]$ corresponds to the probability mass along the "slice" of the polytope defined by $n_r = k$. An exact solution for $\mathbb{E}[n_r]$ can be obtained by using the exact values of $\Pr[n_r = k]$, but obtaining the probability mass along a "slice" can be as computationally hard as the original problem. The family of slice methods introduced in [6] is therefore based on approximations for $\Pr[n_r = k]$ which assume that the mass along each slice is concentrated around the mode of the distribution restricted to the slice.

From the definition of the stationary distribution $\pi(\cdot)$, the mode $\underline{n}^*$ corresponds to a solution of the optimization problem

$$\begin{aligned} \max \quad & \sum_r n_r \log \nu_r - \log n_r! \\ \text{s.t.} \quad & \underline{n} \in \mathcal{S}(\underline{C}). \end{aligned}$$

A natural continuous relaxation of the state space $\underline{n} \in \mathcal{S}(\underline{C})$ is

$$\bar{\mathcal{S}}(\underline{C}) = \left\{ \underline{x} \in \mathbb{R}_+^K : \; A\underline{x} \leq \underline{C} \right\}, \tag{3}$$

for which we obtain the corresponding optimization problem (P1):

$$\begin{aligned} \max \quad & \sum_r x_r \log \nu_r - \log \Gamma(x_r + 1) \\ \text{s.t.} \quad & \underline{x} \in \bar{\mathcal{S}}(\underline{C}), \end{aligned}$$

where the Gamma function is an extension of the factorial function to real numbers,

$$\Gamma(x+1) = \int_0^\infty e^{-t} t^x dt, \qquad \Gamma(n+1) = n!.$$

For each value of $k \in \{n_r : \underline{n} \in \mathcal{S}(\underline{C})\}$ that is along each *slice*, we define $\underline{x}^*(k, r)$ to be the solution of the optimization problem (P2):

$$\begin{aligned} \max \quad & \sum_r x_r \log \nu_r - \log \Gamma(x_r + 1) \\ \text{s.t.} \quad & \underline{x} \in \bar{\mathcal{S}}_{k,r}(\underline{C}) \equiv \bar{\mathcal{S}}(\underline{C}) \cap \{\underline{x} : x_r = k\}. \end{aligned} \tag{4}$$

A computationally implementable version of the primal problems (P1) and (P2) can be obtained by using Stirling's approximation, $\log \Gamma(x_r + 1) = x_r \log x_r - x_r + O(\log x_r)$, and ignoring the $O(\log n_r)$ term. In particular, this respectively yields the following convex relaxations for (P1) and (P2):

$$\begin{aligned} \max \quad & \sum_r x_r \log \nu_r + x_r - x_r \log x_r \\ \text{s.t.} \quad & \underline{x} \in \bar{\mathcal{S}}(\underline{C}) \end{aligned} \tag{5}$$

and

$$\begin{aligned} \max \quad & \sum_r x_r \log \nu_r + x_r - x_r \log x_r \\ \text{s.t.} \quad & \underline{x} \in \bar{\mathcal{S}}_{k,r}(\underline{C}). \end{aligned} \tag{6}$$

In the general slice method [6], for each route r, the optimization problem (5) is solved for the mode of the distribution $\underline{x}^*$ and the optimization problem (6) is solved for each slice defined by $n_r = k$, $k \in \{n_r : \underline{n} \in \mathcal{S}(\underline{C})\}$. To obtain a variation of the general slice method that reduces this computational complexity and provides computational complexity similar to that of the EFPA, a 3-point slice method is presented in [6]. Instead of computing $\underline{x}^*(k, r)$ for all $k \in \{n_r : \underline{n} \in \mathcal{S}(\underline{C})\}$, the 3-point slice algorithm consists of solving the optimization problem (6) for $k = 0$ and the maximum value of k, and also obtaining the mode of the distribution $\underline{x}^*$ by solving (5). Then $\underline{x}^*(k, r)$ is approximated for all other values of k by linear interpolation between pairs of the 3 computed modes. Formally, the interpolation scheme to calculate $\underline{x}^*(k, r)$ is as follows:

(a) If $k \leq x_r^*$, then

$$\underline{x}^*(k, r) = \underline{x}^* \cdot \frac{k}{x_r^*} + \underline{x}^*(0, r) \cdot \frac{x_r^* - k}{x_r^*}.$$

That is, $x^*(k,r)$ is the point of intersection (in the space $\mathbb{R}_+^K$) of the slice $x_r = k$ with the line passing through the two points $\underline{x}^*$ and $\underline{x}^*(0,r)$.

(b) For $x_r^* < k \leq n^{\max}(r)$, where $n^{\max}(r) = \max\{n_r : \underline{n} \in \mathcal{S}(\underline{C})\}$, set

$$\underline{x}^*(k,r) = \underline{x}^*(n^{\max}(r),r) \cdot \frac{k - x_r^*}{n^{\max}(r) - x_r^*} + \underline{x}^* \cdot \frac{n^{\max}(r) - k}{n^{\max}(r) - x_r^*}.$$

Finally, let us denote the objective function from (5) of the Stirling approximated primal problem (P1) (also (P2)) by

$$q(\underline{x}) = \sum_r x_r \log \nu_r + x_r - x_r \log x_r.$$

The estimate of $\mathbb{E}[n_r]$ is then obtained as

$$\mathbb{E}[n_r] = \frac{\sum_k k \exp(q(\underline{x}^*(k,r)))}{\sum_k \exp(q(\underline{x}^*(k,r)))},$$

from which, using (2), we calculate $L_r = 1 - \mathbb{E}[n_r]/\nu_r$.

2.4 Asymptotic Optimality

Kelly [8, 9] established that the EFPA is asymptotically exact in a large network limiting regime where the arrival rates and resource capacities are increased in a proportional manner. More formally, given a SLN with parameters $\underline{C}$ and $\underline{\nu}$, a scaled version of the system is defined by the scaled capacities

$$\underline{C}_N = N\underline{C} = (NC_1, \ldots, NC_K)$$

and the scaled arrival rates

$$\underline{\nu}_N = N\underline{\nu} = (N\nu_1, \ldots, N\nu_K),$$

where $N \in \mathbb{N}$ is the system scaling parameter. The corresponding feasible region of calls is given by $\mathcal{S}(N\underline{C})$. Notice that a normalized version of this region, defined as

$$\mathcal{S}_N(\underline{C}) = \left\{ \frac{1}{N}\underline{n} : \underline{n} \in \mathcal{S}(N\underline{C}) \right\},$$

tends in the large N limit to the *continuous relaxation* (3), that is, $\bar{\mathcal{S}}(\underline{C}) = \{\underline{x} \in \mathbb{R}_+^K : A\underline{x} \leq \underline{C}\}$.

In [6], it is shown that the slice method is also asymptotically exact in the above large network limiting regime. The slice method is further established to provide improved accuracy over the EFPA in the critically loaded regime [5]. Numerical results presented in [6] provide additional evidence that under arbitrary load, the slice methods produce better results than the EFPA.

3 Refined Slice Method

In this section, we present a refinement to the 3-point slice method which computes approximations for the stationary loss probabilities $\underline{L}$ using points $\underline{x}^*(k,r)$ that have higher probability content than the points used by the 3-point slice method. Before introducing the method, we present some notation and results that will be used below.

Consider the unnormalized stationary probability distribution

$$\hat{\pi}(\underline{n}) = \pi(\underline{n})G(\underline{C}), \tag{7}$$

and let

$$\underline{x}^*(k,r) = \text{argmax}_{\underline{x}\in\bar{\mathcal{S}}_{k,r}(\underline{C})}\hat{\pi}(\underline{x}) \tag{8}$$

be the mode at slice $n_r = k$ in the continuous case, which is the solution of (4). Given that the elements of the matrix A are non-negative, the projection of $\bar{\mathcal{S}}_{k,r}(\underline{C})$ must be a subset of the projection of $\bar{\mathcal{S}}_{k-1,r}(\underline{C})$, $k \geq 1$, from which, together with the structural properties of $\hat{\pi}$, the next proposition follows.

Proposition 1. *Assume $k \geq 1$ and $(k, x_2, \ldots, x_K) \in \bar{\mathcal{S}}_{k,r}(\underline{C})$. Then the following identity holds true*

$$\text{argmax}_{x_2,\ldots,x_K}\hat{\pi}(k, x_2, \ldots, x_K) = \text{argmax}_{x_2,\ldots,x_K}\hat{\pi}(k-1, x_2, \ldots, x_K). \tag{9}$$

From Proposition 1, the modes at slices $n_r = k$ and $n_r = k-1$ are related by the equation

$$\underline{x}^*(k,r) = \underline{x}^*(k-1,r) + \underline{e}_r \tag{10}$$

if $\underline{x}^*(k-1,r) + \underline{e}_r \in \bar{\mathcal{S}}_{k,r}(\underline{C})$, $k \geq 1$. This can be rewritten as

$$\underline{x}^*(k,r) = \underline{x}^*(0,r) + k\underline{e}_r \tag{11}$$

if $\underline{x}^*(0,r) + k\underline{e}_r \in \bar{\mathcal{S}}_{k,r}(\underline{C})$, $k \geq 1$. Hence, once the value of $\underline{x}^*(0,r)$ is known, then the modes along a number of route-r slices are immediately given by (11).

The next proposition covers the cases in which $\underline{x}^*(0,r) + k\underline{e}_r \notin \bar{\mathcal{S}}_{k,r}(\underline{C})$.

Proposition 2. *Assume $\underline{x}^*(0,r) + k\underline{e}_r \notin \bar{\mathcal{S}}_{k,r}(\underline{C})$ for $k \in \{n_r : \underline{n} \in \mathcal{S}(\underline{C})\}$. Then $\nabla\hat{\pi}(\underline{x}) > \underline{0}$ for all $\underline{x} \in \bar{\mathcal{S}}_{k,r}(\underline{C})$ where the inequality holds component-wise. Consequently, along each slice $n_r = k$, $\underline{x}^*(k,r)$ must belong to the boundary of $\bar{\mathcal{S}}_{k,r}(\underline{C})$ and, thus, of $\bar{\mathcal{S}}(\underline{C})$.*

Proof. Follows simply from the fact that $\hat{\pi}$ (the unnormalized stationary distribution), within the polytope $\bar{\mathcal{S}}_{k,r}(\underline{C})$, is a monotonically increasing function since the univariate Poisson distribution is increasing prior to the mode, as assumed in the proposition. □

3.1 Description of the Method

For the slice $n_r = k$ of route r, define

$$proj(\underline{x}) = \underline{x} + d\underline{\alpha}^r, \;\; \underline{x} \in \bar{\mathcal{S}}(\underline{C}), \tag{12}$$

where $\underline{\alpha}^r$ is a given vector with the r^{th} element fixed to 0 and d is the maximum positive real number such that $\underline{x} + d\underline{\alpha}^r \in \bar{\mathcal{S}}_{k,r}(\underline{C})$. Namely, $proj(\underline{x})$ represents the projection of $\underline{x}$ to the boundary of $\bar{\mathcal{S}}_{k,r}(\underline{C})$ along the direction of $\underline{\alpha}^r$. The value of d can be efficiently computed by a suitable binary search.

Our method is based on refining the interpolation of the mode of each slice from that of the 3-point slice method. We first compute the fundamental points characterizing the 3-point slice method, i.e., $\underline{x}^*$, $\underline{x}^*(0,r)$ and $\underline{x}^*(n^{\max}(r),r)$, as described in [6]. Next, denote by k^* the positive integer such that $\underline{x}^*(k^*,r) \in \bar{\mathcal{S}}(\underline{C})$ and $\underline{x}^*(k^*,r) + \underline{e}_r \notin \bar{\mathcal{S}}(\underline{C})$. Then, for $k = 1, \ldots, k^*$, compute the modes at slice $n_r = k$ as

$$\underline{x}^*(k,r) = \underline{x}^*(k-1,r) + \underline{e}_r. \tag{13}$$

Proposition 1 ensures that the modes computed through (13) are exact.

Now, turning to the case where $k > k^*$, the mode at slice $n_r = k$ must belong to the boundary of $\bar{\mathcal{S}}_{k,r}(\underline{C})$ by Proposition 2. There are two cases to consider:

1. If $k^* > x_r^*$, then for $k = k^* + 1, \ldots, n^{\max}(r)$, approximate $\underline{x}^*(k,r)$ by interpolation using the formula

$$\underline{x}^*(k,r) = proj\left(\underline{x}^*(n^{\max}(r),r)\frac{k-k^*}{n^{\max}(r)-k^*} + \underline{x}^*(k^*,r)\frac{n^{\max}(r)-k}{n^{\max}(r)-k^*}\right), \tag{14}$$

   which represents the projection to the boundary of $\bar{\mathcal{S}}(\underline{C})$ of the line segment connecting points $\underline{x}^*(k^*,r)$ and $\underline{x}^*(n^{\max}(r),r)$, i.e., $\underline{\alpha}^r$ in (12) is given by $\underline{x}^*(k^*,r) - \underline{x}^*(n^{\max}(r),r)$ and setting the r^{th} component to 0.

2. If $k^* \leq x_r^*$, then

   (a) For $k = k^* + 1, \ldots, \lfloor x_r^* \rfloor$, approximate $\underline{x}^*(k,r)$ by interpolation as follows

$$\underline{x}^*(k,r) = proj\left(\underline{x}^*\frac{k-k^*}{x_r^*-k^*} + \underline{x}^*(k^*,r)\frac{x_r^*-k}{x_r^*-k^*}\right), \tag{15}$$

   which represents the projection to the boundary of $\bar{\mathcal{S}}(\underline{C})$ of the line segment connecting points $\underline{x}^*(k^*,r)$ and $\underline{x}^*$, i.e., $\underline{\alpha}^r$ in (12) is given by $\underline{x}^*(k^*,r) - \underline{x}^*$ and setting the r^{th} component to 0.

   (b) For $k = \lfloor x_r^* \rfloor + 1, \ldots, n^{\max}(r)$, interpolate $\underline{x}^*(k,r)$ using the formula

$$\underline{x}^*(k,r) = proj\left(\underline{x}^*(n^{\max}(r),r)\frac{k-x_r^*}{n^{\max}(r)-x_r^*} + \underline{x}^*\frac{n^{\max}(r)-k}{n^{\max}(r)-x_r^*}\right), \tag{16}$$

   which represents the projection to the boundary of $\bar{\mathcal{S}}(\underline{C})$ of the line segment connecting points $\underline{x}^*$ and $\underline{x}^*(n^{\max}(r),r)$, i.e., $\underline{\alpha}^r$ in (12) is given by $\underline{x}^* - \underline{x}^*(n^{\max}(r),r)$ and setting the r^{th} component to 0.

Thus, when $k^* \leq x_r^*$, the refined method interpolates the modes at different slices among the *four* points $\underline{x}^*(0,r)$, $\underline{x}^*(k^*,r)$, $\underline{x}^*$ and $\underline{x}^*(n^{\max}(r),r)$. On the other hand, when $k^* > x_r^*$ the refined method interpolates among the *three* points $\underline{x}^*(0,r)$, $\underline{x}^*(k^*,r)$ and $\underline{x}^*(n^{\max}(r),r)$. Notice that in this case by definition, the mode $\underline{x}^*$ belongs to the interpolation line (13). We further remark that (14), (15) and (16) provide approximations of the mode for each slice.

Once the points $\underline{x}^*(k,r)$ are computed, the mean number of route-r calls is given by

$$\mathbb{E}[n_r] = \frac{\sum_{k=0}^{n^{\max}(r)} k\ \hat{\pi}(\underline{x}^*(k,r))}{\sum_{k=0}^{n^{\max}(r)} \hat{\pi}(\underline{x}^*(k,r))} \tag{17}$$

and the stationary loss probability of route-r calls is obtained from (2).

3.2 Comparison with the Original 3-Point Slice Method

Denoting the unconstrained mode of $\hat{\pi}(\cdot)$ by

$$\underline{\hat{\nu}} = \text{argmax}_{\underline{x}\in\mathbb{R}_+^K}\ \hat{\pi}(\underline{x}), \tag{18}$$

we have the following comparison between the original 3-point slice method and our refinement of this method.

Theorem 1. *The points identified by the refined slice method have higher probability than those identified by the 3-point slice method when*

1. $\underline{\hat{\nu}} \in \bar{\mathcal{S}}(\underline{C})$ *and* $x_r^* \neq x_r^*(n^{\max}(r), r)$,
2. $\underline{\hat{\nu}} \notin \bar{\mathcal{S}}(\underline{C})$ *and* $x_r^* \neq x_r^*(0, r)$.

In all other cases, both methods are equivalent.

Proof. In the first case, we note that the interpolation line segment connecting points $\underline{x}^*$ and $\underline{x}^*(n^{\max}(r), r)$ is in general not on the boundary of $\bar{\mathcal{S}}(\underline{C})$. However, the refined method chooses the line segment along the r^{th} dimension according to (13), and then chooses a line segment belonging to the boundary of $\bar{\mathcal{S}}(\underline{C})$. Within a given slice, the fact that the points (13) have higher probability is implied by Proposition 1 and the fact that the boundary points have higher probability is implied by the monotonically increasing behavior of $\hat{\pi}$ for states $\underline{n}$ such that $n_s \leq x_s^*, \forall s \neq r$.

In the second case, we note that the interpolation line segment connecting points $\underline{x}^*(0,r)$ and $\underline{x}^*$ cannot be on the boundary of $\bar{\mathcal{S}}(\underline{C})$. However, the refined method always chooses a line segment belonging to the boundary of $\bar{\mathcal{S}}(\underline{C})$. The fact that these boundary points have higher probability is implied by the monotonically increasing behavior of $\hat{\pi}$ for states $\underline{n}$ such that $n_s \leq x_s^*, \forall s \neq r$.

Finally, the equivalence of the two methods in all other cases follows by definition. □

As will be demonstrated in Section 5, instances belonging to either one of cases *1* or *2* of Theorem 1 occur quite frequently in practice (recall that the refined slice method and the 3-point slice method are identical when these cases do not occur). Moreover, the refined slice method is computationally comparable to the original 3-point slice method. In particular, we observe that once the value of k^* is known, the first k^* terms appearing in the numerator and denominator of $\mathbb{E}[n_r]$ in (17) respectively sum to $\frac{1}{2}k^*(k^*+1)\hat{\pi}(\underline{x}(0,r))$ and $(k^*+1)\hat{\pi}(\underline{x}(0,r))$. The efficient computation of $\mathbb{E}[n_r]$ is facilitated by obtaining the value of k^* using binary search between 0 and $n^{\max}(r)$. Finally,

if $T(\underline{x}^*)$ and $T(\underline{x}^*(k,r))$ are the (polynomial-time) costs of computing the modes $\underline{x}^*$ and $\underline{x}^*(k,r)$ using a standard convex programming solver, then the computational complexity of the refined slice method is given by

$$O\left(T(\underline{x}^*) + \sum_{r=1}^{K}[T(\underline{x}^*(0,r)) + T(\underline{x}^*(n^{\max}(r),r)) + n^{\max}(r)]\right). \tag{19}$$

Remark. An immediate extension of the refined slice method is to define a neighborhood around the mode of each slice, and include the probabilities of a fixed, small number of points (possibly of maximal probability) from within that neighborhood. This (potential) improvement in accuracy can be achieved at the additional computational cost on the order of the neighborhood size. We do not consider this extension further since it only complicates the exposition without providing any significant new insight.

4 Contour Method

Our refined slice method provides an efficient means to approximate the stationary loss probabilities $\underline{L}$. However, it does not adequately exploit geometric properties of the multidimensional Poisson distribution with respect to the underlying polytope. To address this, we propose randomized approximation algorithms that are essentially based on a piece-wise approximation of $\hat{\pi}$ and on the volume computation of convex bodies. Approximations are provided for both the normalizing constant $G(\underline{C})$ and the loss probabilities L_r for each route $r = 1, 2, \ldots, K$. While these contour-based methods have a polynomial computational complexity and thus require a greater computational effort than our refined slice method, they generate sample estimation of the desired quantities with probabilistic guarantees on estimation accuracy. In addition, we establish explicit error bounds on the piece-wise approximation of $\hat{\pi}$. Hence, there is a tradeoff between computational complexity and (probabilistic) accuracy guarantees in choosing among our refined slice method and our contour methods.

4.1 Normalizing Constant Algorithm

By definition, estimating the normalizing constant $G(\underline{C})$ is equivalent to estimating a summation over a subset of the high dimensional lattice $\mathbb{Z}_+^K$. To be more precise,

$$G(\underline{C}) = \sum_{\underline{n} \in S(\underline{C})} \hat{\pi}(\underline{n}).$$

For any $\epsilon > 0$, select an integer $M > 0$ and $\varpi_i, i = 0, 1, \ldots, M+1$ such that $\Delta\varpi_i = \varpi_{i+1} - \varpi_i$ are sufficiently small so that

$$\left|G(\underline{C}) - \sum_i (\varpi_{i+1} - \varpi_i)|\chi_i|\right| < \frac{\epsilon}{2},$$

where

$$\chi_i = \{\underline{n} \in S(\underline{C}) : \pi(\underline{n}) \geq \varpi_i\}, i = 0, 1, \ldots, M,$$

partitions $S(\underline{C})$ into M disjoint subsets $\chi_1 - \chi_0, \ldots, \chi_M - \chi_{M-1}$. Meanwhile, define

$$H_i = \{\underline{x} \in \bar{\mathcal{S}}(C) : \sum_r x_r \log \nu_r + x_r - x_r \log x_r \geq \log \varpi_i\},$$

i.e., the polytope that roughly contains all the points in χ_i. We know that the only difference between $|\chi_i|$ and $Vol(H_i)$ occurs for some integer points around the boundary, so

$$\big||\chi_i| - Vol(H_i)\big| = o(Vol(H_i)).$$

Hence, we can refine the selection of M and ϖ_i to make $\Delta\varpi_i$ sufficiently small such that

$$\left|G(\underline{C}) - \sum_i (\varpi_{i+1} - \varpi_i) Vol(H_i)\right| < \frac{\epsilon}{2}.$$

Therefore, the key task is to calculate the volumes of H_i for $i = 1, 2, \ldots, M$. For this purpose, we adapt a multi-phase Monte-Carlo based volume calculation algorithm of Lovász and Vempala [13]. The basic idea is to calculate a sequence of multivariate integrations through Monte-Carlo methods.

First, we need to apply an affine transformation to make sure that the image of each H_i contains the unit ball $B(0,1)$ in $\mathbb{R}^K$ and is contained in $B(0,D)$ where $D = O(\sqrt{K}\log(1/\epsilon))$. Second, for each H_i, define

$$H_i' = ([0, 2D] \times H_i) \cap F,$$

where

$$F = \left\{ x = (x_0, x_1, \ldots, x_K) \in \mathbb{R}^{K+1} : 2\sqrt{\sum_{i=1}^{K} x_i^2} \leq x_0 \right\}.$$

Lovász and Vempala [13] note that, for any $a \leq \epsilon/D$, the quantity

$$Z_i(a) = \int_{H_i'} e^{-a x_0} d\underline{x}$$

serves as a good approximation for $Vol(H_i)$. Now, consider a sequence of positive numbers $a_0 > a_1 > \ldots > a_m$ for $m = 2\lceil \sqrt{K} \log \frac{K}{\epsilon} \rceil$, satisfying $a_0 > 2K$ and $a_m \leq \epsilon^2/D$. Then the volume of H_i can be approximated by

$$Z_i = Z_i(a_0) \prod_{\ell=0}^{m-1} \frac{Z(a_{\ell+1})}{Z(a_\ell)}.$$

At each step $i = 1, 2, \ldots, m$ of the algorithm, $k = \frac{8}{\epsilon^2}\sqrt{K}\log(\frac{K}{\epsilon})$ samples $X_i^1, X_i^2, \ldots, X_i^k$ will be drawn from the distribution of $e^{-a_i x_0}/Z_M(a_i)$. Then for each $j = 1, 2, \ldots, M$, those samples that belong to H_j can be treated as samples drawn from the distribution $e^{-a_i x_0}/Z_j(a_i)$. This feature of the algorithm enables us to adapt the algorithm to estimate the volumes of M convex bodies together, instead of running the original algorithm M times. Therefore, our

contour algorithm has the same complexity as the volume calculation algorithm in [13]. (We note that this is the only part of our algorithm which is different from that of Lovász and Vempala [13].) As for the output, the quantity

$$W_i^j = \frac{1}{k}\sum_{\ell=1}^{k} e^{(a_i - a_{i+1})(X_i^\ell)_0} \cdot I\{X_i^\ell \in H_j\}$$

can serve as an unbiased estimator of $Z_j(a_{i+1})/Z_j(a_i)$, where $I\{\cdot\}$ is the indicator function. The output of the algorithm then will be a vector $(Z_1, Z_2, \ldots, Z_M)$, where

$$Z_j = K!\varpi_K a_m^{-(K+1)} W_1^j \ldots W_m^j, j = 1, 2, \ldots, M.$$

Note that the vector $(Z_1, Z_2, \ldots, Z_M)$ is a probabilistic estimation of the volumes of $H_1, H_2, \ldots, H_M$ that satisfy, with probability at least $1 - \delta$,

$$\frac{|Z_i - Vol(H_i)|}{Vol(H_i)} < \frac{\epsilon}{2}.$$

Therefore,

$$\left| G(\underline{C}) - \sum_i (\varpi_{i+1} - \varpi_i) Z_i \right| < \epsilon.$$

The variance and covariance of the sequence of random samplings have been calculated, or estimated, by Lovász and Vempala [13]. All of those results can be directly applied here to guarantee the accuracy and complexity of our algorithm. In summary, we have

Theorem 2. *For any $\epsilon > 0$ and $\delta > 0$, then, with probability at least $1 - \delta$, $G(\underline{C})$ can be approximated with error no more than ϵ by an algorithm whose complexity is*

$$O\left(\frac{K^4 M}{\epsilon^2} \log^7 \frac{K}{\epsilon\delta}\right). \tag{20}$$

.

4.2 Loss Probability Algorithm

4.2.1 Algorithm Description

Our algorithm computes the loss probability L_r through the following macro steps:

1. For each slice $n_r = k$, choose M points on the segment connecting $\underline{e}_r k$ and $\underline{x}^*(k, r)$. Let $\underline{p}_m = \underline{p}_m(k, r)$, $1 \leq m \leq M$, denote these points such that $\hat{\pi}(\underline{p}_m) \leq \hat{\pi}(\underline{p}_{m+1})$, $1 \leq m < M$. Also let $\underline{p}_0 = \text{argmin}_{\underline{n} \in S_{k,r}(\underline{C})} \hat{\pi}(\underline{n})$ and $\underline{p}_{M+1} = \underline{x}^*(k, r)$.

2. Compute $\mathbb{E}[n_r]$ as

$$\mathbb{E}[n_r] = \frac{\sum_{k=0}^{n^{\max}(r)} k \sum_{m=0}^{M} |\Phi(m,k,r)| \, (\hat{\pi}(\underline{p}_m) + \hat{\pi}(\underline{p}_{m+1}))}{\sum_{k=0}^{n^{\max}(r)} \sum_{m=0}^{M} |\Phi(m,k,r)| \, (\hat{\pi}(\underline{p}_m) + \hat{\pi}(\underline{p}_{m+1}))} \tag{21}$$

where $|\Phi(m,k,r)|$ is the cardinality of the set

$$\Phi(m,k,r) = \left\{ \underline{n} \in S(\underline{C}) : n_r = k, \ \hat{\pi}(\underline{p}_m) < \hat{\pi}(\underline{n}) \leq \hat{\pi}(\underline{p}_{m+1}) \right\}. \tag{22}$$

The stationary loss probability L_r is obtained from (2).

Along each slice, in Step 1 the algorithm initially selects a number of points on the segments connecting the r-axis and the modes $\underline{x}^*(k,r)$. The details of such selection will be discussed below. Then, in Step 2, our algorithm computes the average per-route number of calls through a piece-wise approximation of $\hat{\pi}$ defined on each slice and among pairs of the points identified in Step 1. In particular, this piece-wise approximation ensures that the unnormalized stationary probability of the network states along slice $n_r = k$ is given by

$$\frac{\hat{\pi}(\underline{p}_{m-1}) + \hat{\pi}(\underline{p}_m)}{2} \tag{23}$$

for all $\underline{n} \in \Phi(m,k,r)$, $0 < m \leq M$. The rationale behind approximation (23) is to take the average value between two piece-wise functions representing an upper and a lower bound of $\hat{\pi}$. These bounds are respectively given by

$$\hat{\pi}_{\text{upper}}(\underline{n}) = \hat{\pi}(\underline{p}_m) \ \text{ iff } \ \underline{n} \in \Phi(m,k,r) \tag{24}$$

and

$$\hat{\pi}_{\text{lower}}(\underline{n}) = \hat{\pi}(\underline{p}_{m-1}) \ \text{ iff } \ \underline{n} \in \Phi(m,k,r). \tag{25}$$

For a two-route network and a given slice, Figure 1 depicts an example of such piece-wise functions with respect to the continuous representation of $\hat{\pi}$ and assuming $M = 2$. With approximation (23), several points $\underline{n} \in S(\underline{C})$ are characterized by the same value of (23) (see, e.g., Figure 1) and these are captured by the cardinalities of the sets $\Phi(m,k,r)$. In fact, these sets can be interpreted as the inverse images of $\hat{\pi}$ associated with the subsets $[\hat{\pi}(\underline{p}_m), \hat{\pi}(\underline{p}_{m+1})]$. Formula (21) is then obtained by the exact definition of $\mathbb{E}[n_r]$, which can be expressed as

$$\mathbb{E}[n_r] = \frac{\sum_{k=0}^{n^{\max}(r)} k \sum_{\underline{n} \in S_{k,r}(\underline{C})} \hat{\pi}(\underline{n})}{\sum_{k=0}^{n^{\max}(r)} \sum_{\underline{n} \in S_{k,r}(\underline{C})} \hat{\pi}(\underline{n})}, \tag{26}$$

and grouping together all states $\underline{n}$ having the same value of (23). This allows us to significantly speed up the computation of (26).

We now focus on two remaining issues concerning our algorithm, with the first being the selection of the points $\underline{p}_1, \ldots, \underline{p}_M$. The simplest approach one can choose is to distribute the points on such segments in a balanced manner, i.e., choosing the points

$$m \cdot \frac{\underline{x}^*(k,r)}{M+1}, \tag{27}$$

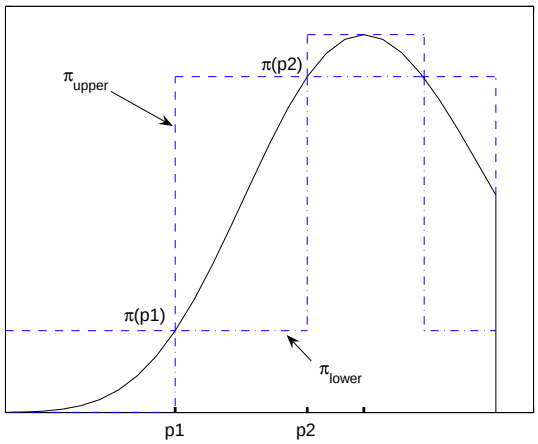


Figure 1: Graphical representation of $\hat{\pi}_{\text{lower}}$ and $\hat{\pi}_{\text{upper}}$ in the single-route case.

$1 \leq m \leq M$. Given that the structure of the polytope $\mathcal{S}_{k,r}(\underline{C})$ is in general different from $\mathcal{S}_{k+1,r}(\underline{C})$, even the arrangement of points at different slices are different. Such arrangements are efficient and in the experimental results section we show that they yield very accurate results.

The second issue is the computation of the number of elements in $\Phi(m,k,r)$. By the same arguments as those in Section 4.1, such a task can be achieved with high accuracy by computing the volume of the body defined by $\Phi(m,k,r)$. To determine the volume of $\Phi(m,k,r)$, we again exploit the randomized algorithm of Lovász and Vempala [13] which computes the volume of convex bodies in polynomial time and within a given precision threshold. However, $\Phi(m,k,r)$ is in general not convex and, thus, we solve this issue with auxiliary sets.

For route r, slice $n_r = k$ and point $\underline{p}_m$, consider the set

$$\Phi'(m,k,r) = \left\{ \underline{n} \in S(\underline{C}) : n_r = k,\ \hat{\pi}(\underline{p}_m) < \hat{\pi}(\underline{n}) \right\}. \tag{28}$$

The following relations are straightforward:

$$\Phi(m,k,r) = \Phi'(m,k,r) \setminus \Phi'(m+1,k,r),\ 0 \leq m < M; \tag{29}$$

$$\Phi(M,k,r) = \Phi'(M,k,r). \tag{30}$$

We further note that all sets $\Phi'(m,k,r)$ are convex because they represent the inner points of contours of $\hat{\pi}$, and $\Phi'(m+1,k,r) \subseteq \Phi'(m,k,r)$ for each $m = 0,\ldots,M-1$. Hence, to compute the volume of $\Phi(m,k,r)$, $0 \leq m \leq M$, we first compute the volume of $\Phi'(m,k,r)$ and then we exploit relations (29) and (30).

Once again, adapting the algorithm in [13], the volume of $\Phi'(m,k,r)$ can be approximated within a relative error of ϵ with probability at least $1-\delta$ and with (polynomial) computational complexity

$$O\left(\frac{K^4 M}{\epsilon^2} \log^7 \frac{K}{\epsilon\delta} \right). \tag{31}$$

4.2.2 Analysis of the Algorithm

Computational Complexity

In summary, for any $\epsilon > 0$ and $\delta > 0$, to obtain the stationary loss probabilities of all routes within a relative error of ϵ with probability at least $1 - \delta$, the computational complexity of our contour method is

$$O\left(\sum_{r=1}^{K}\sum_{k=0}^{n^{\max}(r)}\left[T(\underline{x}^*(k,r)) + T(\Phi(0,k,r),...,\Phi(M,k,r))\right]\right) \tag{32}$$

where $T(\underline{x}^*(k,r))$ is the cost of computing the mode $\underline{x}^*(k,r)$ and $T(\Phi(0,k,r),...,\Phi(M,k,r))$ is the cost of computing the cardinalities for the vector $(\Phi(0,k,r),...,\Phi(M,k,r))$ with the randomized algorithm. Thus, the computational requirement is polynomial with respect to the input parameters.

Error Analysis

Another source of error of the algorithm comes from the selection of M and $p_1, \ldots, p_M$. It is evident that the approximation (23) converges to $\hat{\pi}$ (in the continuous case) when $M \to \infty$. Hence, the choice of which M must be used is related to the computational costs that we are willing to incur to solve SLN models. We now explicitly compute such costs. In what follows, we establish error bounds for the contour method as a function of the number of points M to distribute on each slice. From this error bound, we can conclude that the error decays exponentially as M grows.

Let us consider the unnormalized distribution $\hat{\pi}$ in the continuous case. From its definition, we know that there are positive constants c_1 and c_2 determined only by $\bar{S}(\underline{C})$ and ν_r such that

$$c_1 \leq \frac{d\hat{\pi}}{dx} \leq c_2, \tag{33}$$

i.e., the likelihood ratio between $\hat{\pi}$ and the Lebesgue measure is bounded from below and above. Since the approximation (23) converges to $\hat{\pi}$ as $M \to \infty$, the error of the contour method approaches zero as $M \to \infty$ (in the continuous case). We now evaluate the rate of such convergence.

Within slice $n_r = k$, let

$$\text{Err}_{k,r}(M) = \left| \int_{\mathcal{S}_{k,r}(\underline{C})} \frac{1}{2}(\hat{\pi}_{\text{upper}}(\underline{x}) + \hat{\pi}_{\text{lower}}(\underline{x})) - \hat{\pi}(\underline{x})\, \mathrm{d}\underline{x} \right| \tag{34}$$

be the error of the proposed piece-wise approximation where $\hat{\pi}_{\text{upper}}$ and $\hat{\pi}_{\text{lower}}$ are given by (24) and (25). Given that the average distance between $\hat{\pi}(\underline{x})$ and (23) is never greater than the distance between $\hat{\pi}_{\text{upper}}(\underline{x})$ and $\hat{\pi}_{\text{lower}}(\underline{x})$ (see, for instance, Figure 1), we must have

$$\text{Err}_{k,r}(M) \leq G^+_{k,r}(M) - G^-_{k,r}(M) = \text{Err}^*_{k,r}(M), \tag{35}$$

where

$$G^+_{k,r}(M) = \int_{\mathcal{S}_{k,r}(\underline{C})} \hat{\pi}_{\text{upper}}(\underline{x})\, \mathrm{d}\underline{x} = \sum_{m=1}^{M} |\Phi(m,k,r)| \hat{\pi}(\underline{p}_m), \tag{36}$$

$$G^{-}_{k,r}(M) = \int_{\mathcal{S}_{k,r}(\underline{C})} \hat{\pi}_{\text{lower}}(\underline{x})\, \mathrm{d}\underline{x} = \sum_{m=1}^{M} |\Phi(m,k,r)| \hat{\pi}(\underline{p}_{m-1}). \tag{37}$$

We now focus on $\text{Err}^{*}_{k,r}(M)$ to determine its rate of convergence as M increases.

It is easy to see from (33) that

$$|\Phi(m,k,r)| \approx \alpha_m \frac{1}{M+1}. \tag{38}$$

Hence, we have

$$\begin{aligned}
\text{Err}^{*}_{k,r}(M) \leq & \max_{m \leq M^*} \frac{\alpha_m}{M} \sum_{m=0}^{M^*} \hat{\pi}(\underline{p}_{m+1}) - \hat{\pi}(\underline{p}_m) + \min_{m > M^*} \frac{\alpha_m}{M} \sum_{m=M^*+1}^{M-1} \hat{\pi}(\underline{p}_{m+1}) - \hat{\pi}(\underline{p}_m) \\
= & \; O\left(\max_m \hat{\pi}(\underline{p}_{m+1}) - \hat{\pi}(\underline{p}_m) \right) \\
= & \; O\left(\max_m \prod_r \frac{\nu_r^{p_{mr}+\beta_r/M}}{(p_{mr}+\beta_r/M)!} - \frac{\nu_r^{p_{mr}}}{p_{mr}!} \right) \\
= & \; O\left(\max_m \prod_r \frac{\nu_r^{p_{mr}}}{p_{mr}!} \left(\nu_r^{\beta_r/M} \frac{p_{mr}!}{(p_{mr}-\beta_r/M)!} - 1 \right) \right) \\
\approx & \; O\left(\prod_r (\nu_r^{\beta_r/M} - 1) \right)
\end{aligned} \tag{39}$$

where $\beta_r = x^*_r(k,r)/(M+1)$ is obtained from (27) and M^* is such that $\hat{\pi}(\underline{p}_{M^*+1}) - \hat{\pi}(\underline{p}_{M^*}) \geq 0$ and $\hat{\pi}(\underline{p}_{M^*+2}) - \hat{\pi}(\underline{p}_{M^*+1}) < 0$ if it exists, otherwise it is M. This means that the error on each slice approaches zero exponentially.

5 Experimental Results

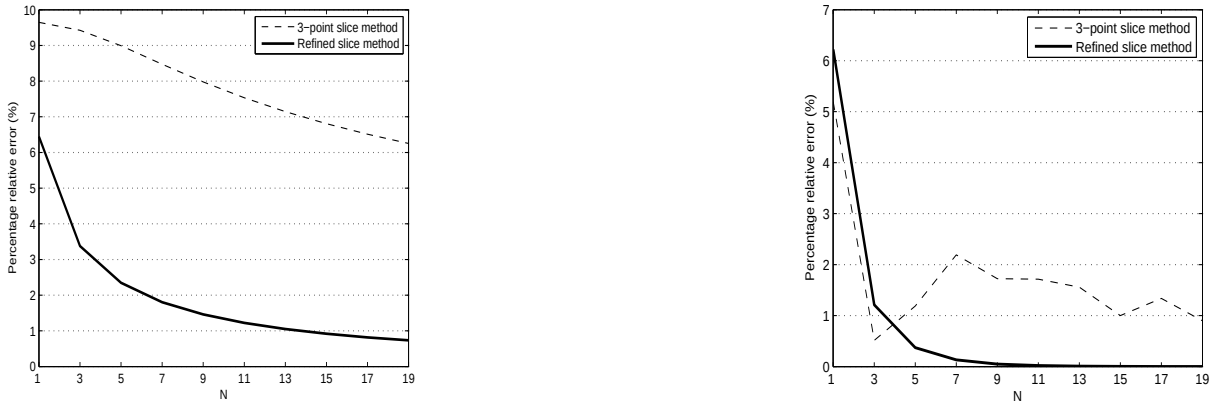
In this section, we numerically assess the accuracy of the refined slice method and of both contour methods. Our approximations are validated against exact solutions, and the refined slice method is also compared with the 3-point slice method [6]. Numerical results are presented relative to two different sets of experiments: first, we perform a validation against a small canonical network to emphasize the fundamental properties of the proposed solutions, and then we focus on a wide test-bed of randomly generated models to assess the general quality of the relative error percentages.

5.1 A Canonical Network

Consider first a small network characterized by

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \underline{C} = \begin{bmatrix} 5 \\ 7 \\ 5 \end{bmatrix}, \tag{40}$$

i.e., link 2 is shared by both routes while links 1 and 3 are dedicated to route 1 and 2, respectively. Without loss of generality, we focus on the average number of calls of route 1, i.e., $\mathbb{E}[n_1]$. To quantify the benefits of the refined slice method, we consider the following cases for the value of $\underline{\nu}$:



(a) (b)

Figure 2: Comparison of the refined and the 3-point slice methods for (a) Stress case 1 and (2) Stress case 2.

- *Stress case 1*: $\underline{\nu} \notin \mathcal{S}(\underline{C})$ and $x_1^* \neq x_1^*(0,1)$, e.g., $\underline{\nu} = (6,8)$;
- *Stress case 2*: $\underline{\nu} \in \mathcal{S}(\underline{C})$ and $x_1^* \neq x_1^*(n^{\max}(1),1)$, e.g., $\underline{\nu} = (1,5)$.

Note that for an evaluation of two-route networks, these classes of values of $\underline{\nu}$ suffice since outside this load region the 3-point slice method and the refined slice method are identical.

With respect to the above stress cases and by varying the scaling parameter N, Figure 2 illustrates the relative error percentages of $\mathbb{E}[n_1]$, i.e.,

$$\frac{|\mathbb{E}[n_1]_{approximate} - \mathbb{E}[n_1]_{exact}|}{\mathbb{E}[n_1]_{exact}} \cdot 100\%, \tag{41}$$

when the 3-point slice method and the refined slice method are used. In Figure 2(a), we note that the refined slice method provides a remarkable improvement in accuracy for each N and the error (41) converges to zero much faster when compared to the 3-point slice method. In Figure 2(b), we observe that in stress case 2, the proposed method provides better results only when $N > 3$. This can be explained by the fact that when N is small, states close to the mode significantly affect the behavior of the slice methods. On the other hand, when N is large the shape of the distribution π concentrates around its mode making negligible the effect of the states in its neighborhood. Therefore the mode closely approximates the mean number of calls in the system only for N relatively large. In fact, we note that our results are very close to exact values for $N > 7$.

To provide a geometric explanation for the improvements in Figure 2, we depict in Figures 3(a) and 3(b) the polytope $\mathcal{S}(\underline{C})$ and the interpolation points calculated by the 3-point and the refined slice methods, respectively[2]. Along each slice, i.e., $n_1 = 0, \ldots, 5$, we notice that the points identified by our method are always the ones with highest probability. From Figure 3(a) this is true because these points belong to the boundary of the polytope and the distribution $\hat{\pi}$ is monotonically increasing in $\mathcal{S}(\underline{C})$. On the other hand, the 3-point slice method initially chooses the

[2]It suffices to consider the unscaled case, i.e., when $N = 1$, since the structure of the polytope and the interpolation lines chosen by both methods do not change with N.

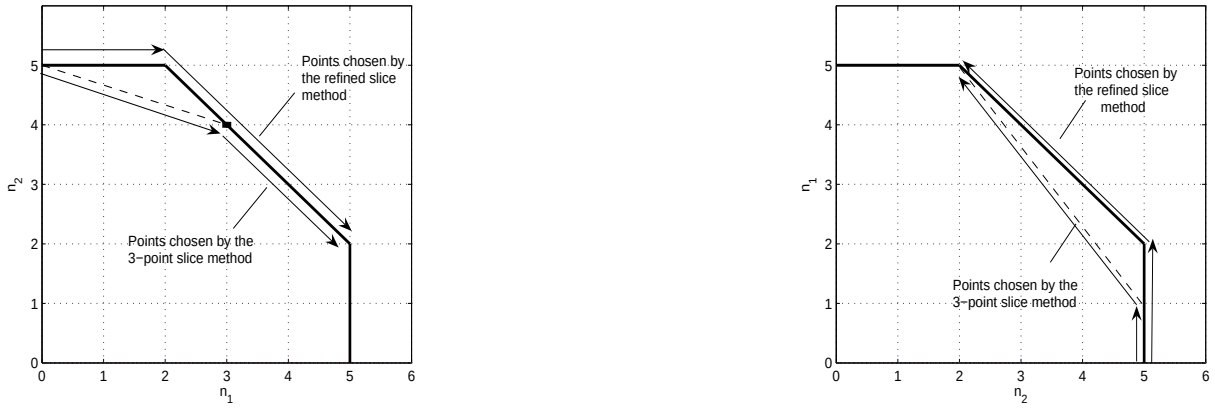


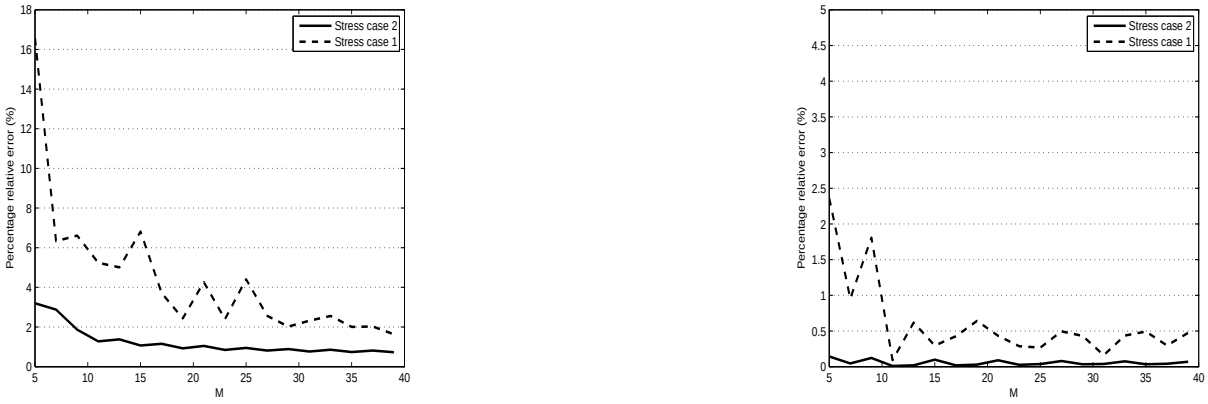
Figure 3: Interpolation points of both methods for (a) Stress case 1 and (2) Stress case 2.

points which follow the dashed segment connecting $\underline{x}^*(0,1) = (0,5)$ and the (global) mode $\underline{x}^* = (3,4)$. Clearly, these points have less probability than the respective ones belonging to the boundary of $\mathcal{S}(\underline{C})$. Likewise from Figure 3(b) we observe that the points connecting the (global) mode $\underline{x}^* = (1,5)$ and $\underline{x}^*(5,1) = (5,2)$ chosen by the 3-point slice method have less probability than the corresponding ones chosen by the refined slice method. Therefore, the results shown in Figure 2 agree with our intuition that better results are obtained if we interpolate over states with higher probability.

We now compare the accuracy of both contour methods when applied to the small network under stress cases 1 and 2 while varying the number of contours M from 5 to 40 in steps of size 2. Figure 4(a) shows the relative error percentage of the normalizing constant from the contour method. Stress case 2 yields better results than stress case 1 because the unnormalized probability distribution $\hat{\pi}$ in the former case is easier to approximate through contours. Analogously, Figure 4(b) depicts the error in computing $\mathbb{E}[n_1]$ by the contour method. Once again the proposed method yields very good results and the errors tend to 0.1% (stress case 2) and 0.5% (stress case 1) for M large. As in Figure 4(a), better results are obtained for stress case 2. Additionally, in both Figures 4(a) and (b) we observe small oscillations. These are caused simply because of the discrete nature of the approximating distribution used in the contour method, which while convergent to the continuous version of $\hat{\pi}$ for large M has an oscillating error for smaller M.

5.2 Random Models

To assess the accuracy of the proposed methods in a more general setting, we consider a test-bed of thousands of randomly-generated models larger than the canonical example discussed in the previous section. The input parameters used to validate our methods are shown in Table 1. We compare the quality of the refined slice method and of both contour methods with the exact results which we obtain by brute-force enumeration over the set of feasible states, i.e., $\mathcal{S}(\underline{C})$, for each random model. We do not consider networks with larger capacities or a larger number of routes because



(a) (b)

Figure 4: Percentage relative error vs. number of contours M for (a) the normalizing constant and (b) $\mathbb{E}[n_1]$.

	Interval
Number of routes	$\{2, 3, 4\}$
Number of links	$\{3, \ldots, 10\}$
Links capacities	$\{20, \ldots, 50\}$
Arrival rates	$[1, 50]$

Table 1: Input parameters used in the validation.

of the prohibitively expensive computational effort required to obtain the exact solution and the consequent cost of calculating robust accuracy results for the large number of test cases. Further, we consider only random instances such that no single link simultaneously limits the maximum number of calls on *all* routes. In fact, in this particular case the normalizing constant admits a very efficient expression by means of Newton's multinomial theorem.

In Figure 5 we compare the accuracies of the refined and 3-point slice methods for different values of the scaling parameter $N = 0.5, \ldots, 2.5$ in steps of size 0.5 and, to emphasize the trend of the slice methods, for $N = 5$. Each point in the graphs of Figure 5 represents the average of the errors (41) over 3,000 models. Even though the errors of both methods are small and converge to zero as N increases, we observe that the refined slice method provides more accurate results for each N. In the worst case, i.e., $N = 0.5$, our results have an average error of 2.2%.

In Figure 6(a), we show the quality of the results obtained by the contour method which computes the normalizing constant by varying the number of contours M from 5 to 40 in steps of size 5. In the figure, each point represents the relative error percentage in computing the normalizing constant, averaged over 3,000 models. The results show that the average relative error percentage of the normalizing constant tends to around 4%. As discussed above, we note that the piece-wise function characterizing our contour methods converges to the continuous version of distribution $\hat{\pi}$ when M increases. However, since the distribution $\hat{\pi}$ is discrete, the small limiting error in the graph of Figure 6(a) is essentially due to the error in assuming that $\hat{\pi}$ is continuous. In Figure 6(b), we analogously show the quality of the results obtained by the contour method in computing $\mathbb{E}[n_r]$. In this figure, we observe that our approach provides very

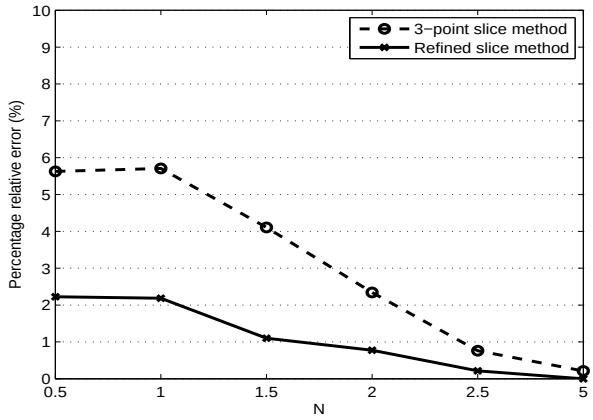
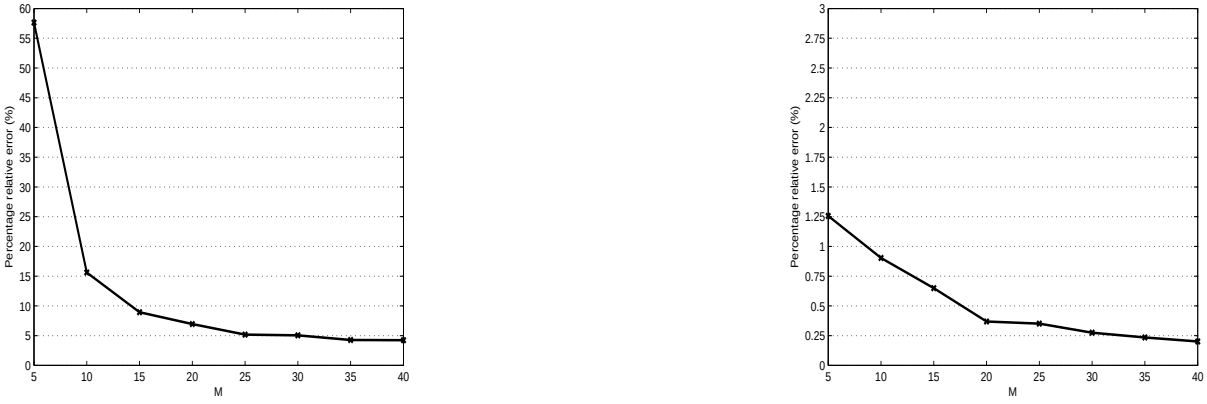


Figure 5: Comparison between the error in computing $\mathbb{E}[n_1]$ for the refined and the 3-point slice methods.



(a) (b)

Figure 6: Average errors in computing (a) the normalizing constants and (b) $\mathbb{E}[n_1]$ using the contour method.

accurate and robust results which tend to an average error of 0.2% as M increases.

Hence, at the cost of higher computational complexity than our refined slice algorithm, the contour method can be adapted to obtain nearly exact estimates of the average numbers of per-route calls and loss probabilities. It is worth recalling that the two sets of proposed algorithms differ in their applicability: the refined slice method is exact in the large network limiting regime ($N \to \infty$), while the contour algorithms give approximations corresponding to the unscaled network with error bounds that tend to zero as the computational costs increase ($M \to \infty$). Thus we cannot make a meaningful comparison between the accuracy of the refined slice and contour methods since it depends upon the respective parameters N and M. In practice, one can choose among the methods based upon which properties best suit their needs.

References

[1] BHADRA, S., LU, Y., AND SQUILLANTE, M. S. (2007). Optimal capacity planning in stochastic loss networks with time-varying workloads. In *Proceedings of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*. ACM, New York, 227–238.

[2] BONALD, T. (2006). The Erlang model with non-Poisson call arrivals. In *Proceedings of Joint SIGMETRICS/Performance Conference on Measurement and Modeling of Computer Systems*. ACM, New York, 276–286.

[3] BURMAN, D. Y., LEHOCZKY, J. P., AND LIM, Y. (1984). Insensitivity of blocking probabilities in a circuit-switching network. *Journal of Applied Probability 21*, 850–859.

[4] ERLANG, A. K. (1948). Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. In *The Life and Works of A. K. Erlang*, E. Brockmeyer, H. L. Halstrom, and A. Jensen, Eds. Academy of Technical Sciences, Copenhagen. Paper first published in 1917.

[5] HUNT, P. J. AND KELLY, F. P. (1989). On critically loaded loss networks. *Advances in Applied Probability* **21**, 4, 831–841.

[6] JUNG, K., LU, Y., SHAH, D., SHARMA, M., AND SQUILLANTE, M. S. (2008). Revisiting stochastic loss networks: Structures and algorithms. In *Proceedings of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*. ACM, New York, 407–418.

[7] KELLY, F. P. (1985). Stochastic models of computer communication systems. *Journal of the Royal Statistical Society, Series B 47*, 379–395.

[8] KELLY, F. P. (1986). Blocking probabilities in large circuit-switched networks. *Advances in Applied Probability* **18**, 2, 473–505.

[9] KELLY, F. P. (1988). Routing in circuit-switched networks: Optimization, shadow prices and decentralization. *Advances in Applied Probability* **20**, 1, 112–144.

[10] KELLY, F. P. (1991). Loss networks. *Annals of Applied Probability* **1**, 3, 319–378.

[11] LITTLE, J. D. C. (1961). A proof of the queuing formula $L = \lambda W$. *Operations Research 9*, 383–387.

[12] LOUTH, G., MITZENMACHER, M., AND KELLY, F. (1994). Computational complexity of loss networks. *Theoretical Computer Science* **125**, 1, 45–59.

[13] LOVÁSZ, L. AND VEMPALA, S. (2006). Simulated annealing in convex bodies and an O*(n^4) volume algorithm. *Journal of Computer and System Sciences* **72**, 2, 392–417.

[14] LU, Y., RADOVANOVIĆ, A., AND SQUILLANTE, M. S. (2007). Optimal capacity planning in stochastic loss networks. *Performance Evaluation Review* **35**, 2.

[15] MITRA, D., MORRISON, J. A., AND RAMAKRISHNAN, K. G. (1996). ATM network design and optimization: A multirate loss network framework. *IEEE/ACM Transactions on Networking* **4**, 4, 531–543.

[16] MITRA, D. AND WEINBERGER, P. J. (1984). Probabilistic models of database locking: Solutions, computational algorithms and asymptotics. *Journal of the ACM 31*, 855–878.

[17] MOMCILOVIC, P. AND SQUILLANTE, M. S. (2008). On throughput in linear wireless networks. In *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing*. ACM, New York.

[18] ROSS, K. W. (1995). *Multiservice Loss Models for Broadband Telecommunication Networks*. Springer-Verlag, New York.

[19] SEVASTYANOV, B. A. (1957). An ergodic theorem for Markov processes and its application to telephone systems with refusals. *Theoretical Probability Applications 2*, 104–112.

[20] VALIANT, L. (1979). The complexity of computing the permanent. *Theoretical Computer Science 8*, 189–201.

[21] WHITT, W. (1985). Blocking when service is required from several facilities simultaneously. *AT&T Bell Laboratories Technical Journal* **64**, 8, 1807–1856.

[22] XU, S., SONG, J. S., AND LIU, B. (1999). Order fulfillment performance measures in an assemble-to-order system with stochastic leadtime. *Operations Research* **47**, 1, 131–149.