

# Asymptotically Optimal Importance Sampling for Jackson Networks with a Tree Topology

Ali Devin Sezer  
Institute of Applied Mathematics  
Middle East Technical University  
Ankara, Turkey

November 21, 2018

## Abstract

Importance sampling (IS) is a variance reduction method for simulating rare events. A recent paper by Dupuis, Wang and Sezer (Ann. App. Probab. 17(4):1306- 1346, 2007) exploits connections between IS and subsolutions to a limit HJB equation and its boundary conditions to show how to design and analyze simple and efficient IS algorithms for various overflow events for tandem Jackson networks. The present paper uses the same subsolution approach to build asymptotically optimal IS schemes for stable open Jackson networks with a tree topology. Customers arrive at the single root of the tree. The rare overflow event we consider is the following: given that initially the network is empty, the system experiences a buffer overflow before returning to the empty state. Two types of buffer structures are considered: 1) A single system-wide buffer of size  $n$  shared by all nodes, 2) each node  $i$  has its own buffer of size  $\beta_i n$ ,  $\beta_i \in (0, 1)$ .

## 1 Introduction

Importance sampling (IS) is a method for simulation of rare events. It is used in many applications including simulation of communication systems, computation of credit risk and pricing of financial derivatives. The idea in IS is to change the sampling distribution (and modify the Monte Carlo estimator accordingly) to reduce estimator variance. Queuing processes are basic stochastic models that are commonly used in a wide range of application areas. The simplest type of queuing processes are Jackson networks, in which the arrival and service times at the nodes of the network are assumed to be independent and exponentially distributed with constant rates.

In the present paper we build an IS algorithm, which is optimal in a certain asymptotic sense (see Section 3), to simulate buffer overflows of stable open Jackson networks with a tree topology. The system is stable in the sense that the average service rate at each node is faster than the average arrival rate to that node. Customers arrive at the single root of the tree. The rare overflow event we consider is the following: given that initially the network is empty the system experiences a buffer overflow before returning to the empty state. Two types of buffer structures are considered: 1) A single system-wide buffer of size  $n$  shared by all nodes 2) each node  $i$  has its own buffer of size  $\beta_i n$ ,  $\beta_i \in (0, 1)$ .

To construct our optimal IS algorithms we use an optimality result from [16] which was obtained using the optimal control/subsolution approach to IS of [12, 3, 4, 6, 5]. This result states that to construct optimal IS algorithms for the simulation of a wide range of buffer overflow events of any stable Jackson network it is sufficient to build appropriate smooth subsolutions to a Hamilton Jacobi Bellman (HJB) equation and its boundary conditions (these are given in (7) in the context we study in the current paper). This HJB equation and the boundary conditions are the main tools of the optimal control/subsolution approach and are derived from an optimal control representation of the IS distribution construction problem.

The main contribution of the present paper is a recursive algorithm which takes as input the parameters of an arbitrary Jackson network with a tree topology and constructs a smooth subsolution to the HJB equation and its boundary conditions given in (7). The constructed subsolution is of the form of a smoothed minimum of affine functions, as was the case in previous works using the subsolution approach, e.g. [12, 16]. The quantities that appear in the subsolution (and hence the algorithm) have simple heuristic interpretations as *effective* utilities and rates of nodes in the system. They are “effective” in the sense that they depend on whether a node is empty or nonempty. These concepts are explained in detail in subsection 4.1. The main results of the paper are Lemmas 4.2 and 6.1 which prove that the subsolutions arising from the effective rates and utilities satisfy all the conditions of the general optimality theorem in [16] for both type of buffer structures that we will be studying in this paper. Numerical results in Sections 5 and 6 demonstrate the practical usefulness of the resulting IS algorithms.

Since the initial writing [15] of the present paper a recent paper by Dupuis and Wang [7] appeared that treat the IS problem for any stable Jackson network using the subsolution approach. The relation between the results in the current paper and those in [7] is discussed in Section 7.

There is a tremendous amount of work on the IS of queueing networks, which include, [13, 18, 14, 2, 10, 9, 11, 8, 1]. The problem of constructing IS algorithms for buffer overflow of queueing networks was first posed for the simple two node tandem network in [11], which also proved that a static

large deviations based change of measure is asymptotically optimal for certain parameter values of the system. An asymptotically optimal IS algorithm with optimality proofs for buffer overflow of stable tandem Jackson networks was first developed in [12] using the optimal control/subsolution approach. The discontinuous dynamics of the queuing process near the boundaries of its state space (i.e., when few customers remain in some of the nodes) makes the IS construction problem for queuing networks difficult [12, 8]. This property rules out iid sampling distributions (such as those developed in [17] in the context of a random walk on the real line and in [11] in the context of two tandem Jackson nodes) as candidates for efficient IS samplers and forces one to search for a good IS distribution among dynamic distributions, where indeed the subsolution approach locates the optimal IS distributions. For a more in depth discussion of these issues we refer the reader to [12, 16, 8, 3].

## 2 Setup

We consider Jackson networks with a tree topology. Customers arrive only at the root of the tree. Our goal is to construct optimal IS algorithms to estimate the following probability:

$$P_0(\text{system experiences an overflow before it empties}). \quad (1)$$

This overflow event depends on the buffer structure of the network, which will be made precise in subsection 2.2. For the computation of  $p_0$  it is enough to consider the embedded discrete time random walk of the Jackson network. The normalized service and arrival rates and the routing probabilities of the Jackson network are the jump probabilities of the embedded random walk.

### 2.1 Notation and Definitions

The tree consists of  $d$  nodes.  $X(i)$  is the population of  $i^{\text{th}}$  node at the jump times in the network.  $i \rightarrow j$  denotes that node  $j$  is a child of node  $i$ . For  $i \rightarrow j$ ,  $\mu_{i,j} > 0$  is the rate at which customers are served in node  $i$  and are either [sent to node  $j$  if  $j > 0$ ] or [leave the system if  $j = 0$ ].

Total service rate at node  $i$  is defined as  $\mu_i \doteq \sum_k \mu_{i,k}$ . Arrival rate to node  $\Lambda_j$  at node  $j$  equals  $\lambda$  if  $j$  is the root node. Otherwise it equals  $\Lambda_j \doteq \Lambda_i \frac{\mu_{i,j}}{\mu_i}$  where node  $i$  is the parent of node  $j$ . It is no loss of generality to assume that  $\lambda + \sum_{i=1}^d \mu_i$  equals 1 ; otherwise one can change the time unit so that the equality holds. The utility of node  $i$  is defined as:  $\rho_i \doteq \Lambda_i / \mu_i$ . The Jackson network is called stable if  $\rho_i < 1$  for all  $i \in \{1, 2, \dots, d\}$ . Therefore we assume that  $\prod_{i=1}^d \rho_i < 1$ . This stability assumption implies that the buffer overflow events of interest we study in the present paper decay exponentially in  $n$  (see (10) and (19)). Asymptotic optimality of an IS algorithm is stated in terms of this exponential decay (see Section 3).

The evolution of the random walk  $X$  takes place in the state space  $\mathbb{Z}_+^d$ . This set has  $2^d - 2$  different boundaries:  $\partial_i \doteq \{x = (x_1, x_2, \dots, x_d) \in \mathbb{Z}_+^d : x_i = 0\}$ ,  $i \in \{1, 2, \dots, d\}$ ,  $\partial_{\{i_1, i_2, \dots, i_k\}} \doteq \bigcap_{l=1}^k \partial_{i_l}$ ,  $\{i_1, i_2, \dots, i_k\} \subset \{1, 2, \dots, d\}$ . As we have remarked earlier the dynamics of  $X$  depends on whether  $X$  is on one of these boundaries and if so it further depends on which one. We will find it convenient to identify these boundaries with bitmaps  $b \in \{0, 1\}^d$ .  $b$  describes the following state of the network:  $b(i) = 0$  signifies that node  $i$  is empty,  $b(i) = 1$  signifies that it is non-empty. Define  $v_{0,1} = (1, 0, \dots, 0)$  and

$$\begin{aligned} \mathcal{V}_2 &\doteq \{v_{i,j}, i, j \in \{1, 2, \dots, d\} : i \rightarrow j, \\ &\quad v_{i,j}(i) = -1, v_{i,j}(j) = 1, v_{i,j}(k) = 0, k \in \{1, 2, \dots, d\} - \{i, j\}\} \\ \mathcal{V}_3 &\doteq \{v_{i,0}, i \in \{1, 2, \dots, d\} : v_{i,0}(i) = -1, v_{i,0}(k) = 0, k \in \{1, 2, \dots, d\} - \{i\}\} \end{aligned}$$

Let  $\mathcal{V} \doteq \{v_{0,1}\} \cup \mathcal{V}_2 \cup \mathcal{V}_3$ .  $\mathcal{V}$  are the set of all possible jumps the process  $X$  can make.  $v_{0,1}$  corresponds to a new customer arriving at the root node,  $v_{i,j} \in \mathcal{V}_2$  corresponds to server  $i$  serving a customer in queue  $i$  and sending it to queue  $j$  with  $i \rightarrow j$ , and finally  $v_{i,0} \in \mathcal{V}_3$  corresponds to a customer leaving the system after being served by server  $i$ .

Let  $Y = \{Y_k : k = 0, 1, 2, \dots\}$  be an iid sequence such that  $P_x(Y_k = v_{0,1}) = p(v_{0,1}) \doteq \lambda$ ,  $P_x(Y_k = v_{i,j}) = p(v_{i,j}) \doteq \mu_{i,j}$  for  $v_{i,j} \in \mathcal{V}_2$ ,  $P_x(Y_k = v_{i,0}) = p(v_{i,0}) \doteq \mu_{i,0}$  for  $v_{i,0} \in \mathcal{V}_3$ , for all  $x \in \mathbb{Z}_+^d$ .  $Y_k$  are the unconstrained increments of the process  $X$ . We assume the existence of a probability space  $(\Omega, \mathcal{F})$  equipped with the probability distributions  $P_x$ . The subscript  $x$  denotes the initial position of the queuing system  $X_0$ : under  $P_x$ ,  $X_0 = x$  almost surely.

$X \in \partial_{\{i_1, i_2, \dots, i_k\}}$  if the Jackson network has no customers in queues  $i_1, i_2, \dots, \text{and } i_k$ . Therefore  $v_{l,j}$ ,  $j \in \{0, 1, 2, \dots, d\}$ ,  $l \in \{i_1, i_2, \dots, i_k\}$ , cannot be an increment of  $X$  when  $X \in \partial_{\{i_1, i_2, \dots, i_k\}}$ . The constraining map  $\pi : \mathbb{R}_+^d \times \mathcal{V} \rightarrow \mathcal{V} \cup \{0\}$  will make sure that this does not happen:

$$\pi(x, v) = \begin{cases} 0, & \text{if } x \in \partial_i \text{ for some } i \in \{1, 2, \dots, d\} \text{ and } \langle v, n_i \rangle < 0, \\ v, & \text{otherwise,} \end{cases}$$

where  $n_i$  is normal to the boundary  $\partial_i$ :  $n_i(i) = 1$  and  $n_i(j) = 0$  for  $j \neq i$ .  $X$  can now be written as

$$X_{k+1} \doteq X_k + \pi(X_k, Y_k). \quad (2)$$

$X_0$  is the initial state of the system and under  $P_x$  it equals  $x \in \mathbb{Z}_+^d$  almost surely.

## 2.2 Overflow event of interest

We would like to develop IS algorithms to estimate (1). We now define what we mean by an overflow. Let  $\partial_+^d \doteq \{x \in \mathbb{R}_+^d : \forall_i x(i) = 1\}$ .

**Assumption 1.** *The system has a buffer whose structure is determined by a normalized exit set  $\mathcal{S} \subset [0, 1]^d$  with the following properties: 1)  $\mathcal{S}$  is closed and connected, 2)  $0 \notin \mathcal{S}$ , 3) Any continuous curve in  $[0, 1]^d$  that contains 0 and a point from  $\partial_+^d$  must also contain a point from  $\mathcal{S}$ . 4) For  $S_n \doteq \{x \in \mathbb{Z}_+^d : x/n \in \mathcal{S}\}$ ,*

$$\gamma \doteq \lim_{n \rightarrow \infty} -\frac{1}{n} \log P_{\mathbf{s}}(X \text{ hits } S_n \text{ before } 0) \quad (3)$$

*exists and is nonzero.*

In this article we are interested in two types of buffer structures: 1)  $\mathcal{S}_1 \doteq \{x \in \mathbb{R}_+^d : x(1) + x(2) + \dots + x(d) = 1\}$ .  $S_n \doteq \{x \in \mathbb{Z}_+^d : x/n \in \mathcal{S}_1\}$  corresponds to a single buffer of size  $n$  shared by all queues. For  $\beta \in \mathbb{R}_+^d$   $\mathcal{S}_2 = \{x \in \mathbb{R}_+^d : x(i) = \beta(i) \text{ for some } i \text{ and } x(j) \leq \beta(j) \text{ for all } j\}$ . Then  $S_n \doteq \{x \in \mathbb{Z}_+^d : x/n \in \mathcal{S}_2\}$  corresponds to  $d$  independent buffers, one for each node. The size of the buffer for node  $i$  is given by  $n\beta(i)$ . Without loss of generality we will assume that  $\forall_i \beta(i) = 1$ .

Define the initial point  $\mathbf{s} \doteq (1, 0, 0, 0, \dots, 0)$ . Fix a buffer structure  $\mathcal{S}$  and define the exit boundaries  $S_n$  as above. We now rewrite the exit probability of interest precisely as:  $p_n \doteq P_{\mathbf{s}}(X \text{ hits } S_n \text{ before it hits } 0)$ . We consider the case  $\mathcal{S} = \mathcal{S}_1$  (all nodes share a single buffer) in Section 4 and the case  $\mathcal{S} = \mathcal{S}_2$  (one buffer for each node) in Section 6.

### 3 Importance Sampling

In order to simulate  $X$  using importance sampling one specifies a sampling distribution  $\bar{p}(v|x)$ ,  $v \in \mathcal{V}$  and  $x \in \mathbb{Z}^+$  and simulates  $X$  from this distribution. Note that we allow  $\bar{p}$  to depend on  $x$ , the current position of  $X$ . Define  $A_n$  to be the set of sample paths that hit the exit set  $S_n$  before 0 and let  $T_n$  denote the first time  $X$  hits  $S_n$  or 0. The IS estimator of  $p_n$  using  $K$  sample paths is then:

$$\frac{1}{K} \sum_{k=1}^K \hat{p}_n^k, \quad \hat{p}_n^k \doteq 1_{A_n}(X^k) \cdot \prod_{i=1}^{T_n-1} \frac{p(Y_i^k)}{\bar{p}(Y_i^k|X_i^k)}, \quad (4)$$

where  $X^k$  denotes the  $k^{\text{th}}$  independent sample path used in the simulation. The increments  $\{Y^k\}$  are iid copies of the increment process  $Y$  sampled from  $\bar{p}$ .  $X^k$  is built along with  $Y^k$  using the dynamics (2). The product is the likelihood ratio of  $P_{\mathbf{s}}$  and  $\bar{P}$ , which appears in the estimator to cancel off the effect of changing the sampling distribution from  $p$  to  $\bar{p}$ .

$\hat{p}_n \doteq \hat{p}_n^1$  is an unbiased estimator of  $p_n$  and therefore the variance of  $\hat{p}_n$  depends on the sampling distribution only through the second moment of  $\hat{p}_n$ . Because  $p_n$  decays exponentially, one would like the second moment of  $\hat{p}_n$  to decay exponentially as well. However, Jensen's inequality implies that

$$\limsup_n -\frac{1}{n} \log \hat{\mathbb{E}}[\hat{p}_n^2] \leq \limsup_n -\frac{2}{n} \log \hat{\mathbb{E}}[\hat{p}_n] \equiv 2\gamma.$$

In other words, the exponential decay rate of the second moment can be at most twice that of the probability. The IS estimator is said to be *asymptotically optimal* if the upper bound is achieved, i.e., if  $\liminf_n -\frac{1}{n} \log \mathbb{E}[\hat{p}_n^2] \geq 2\gamma$ .

### 3.1 Definitions from the subsolution approach

In this subsection we will give only the definitions from the subsolution approach that we need to present the results and the algorithm for the tree Jackson networks. A full development of the subsolution approach ideas can be found in [6, 5, 12].

#### Hamiltonians, the limit HJB equation and the boundary conditions.

For a bitmap  $b \in \{0, 1\}^d$  and  $q \in \mathbb{R}^d$  define

$$N_b(q) \doteq \lambda e^{-q(1)/2} + \sum_{i:b(i)=1} \sum_{i \rightarrow j} \mu_{i,j} e^{\frac{q(i)-q(j)}{2}} + \sum_{i:b(i)=1} \mu_{i,0} e^{\frac{q(i)}{2}} + \sum_{i:b(i)=0} \mu_i, \quad (5)$$

$$H_b(q) = -2 \log N_b(q).$$

$H_b$  is the Hamiltonian associated with boundary  $b$ . We denote  $H_b$  by  $H$  if  $b = (1, 1, 1, \dots, 1, 1)$ .

For  $x \in \mathbb{R}_+^d$ , define  $b_x \in \{0, 1\}^d$  as follows:

$$b_x(i) \doteq \begin{cases} 0, & \text{if } x(i) = 0, \\ 1, & \text{otherwise.} \end{cases} \quad (6)$$

$b_x$  indicates which boundary  $x$  is on (if  $b_x = (1, 1, \dots, 1, 1)$  then  $x$  is in the interior of  $\mathbb{R}_+^d$ ).

**Definition of a subsolution.** The limit HJB equation and its boundary conditions that are in the center of the subsolution approach are as follows:

$$H(DV(x)) = 0, \quad H_{b_x}(DV(x)) = 0, \quad (7)$$

where  $DV$  denotes the gradient of  $V$ . A subsolution to (7) is defined as follows:

**Definition 3.1.**  $\bar{V}$  is an  $\epsilon$ -subsolution to (7) if it is  $C^1(\mathbb{R}^d, \mathbb{R})$  and

- (a)  $H_{b_x}(D\bar{V}(x)) \geq -\epsilon$  for all  $x \in \mathbb{R}_+^d$ ,
- (b)  $\bar{V}(0) \geq 2\gamma - \epsilon$ ,
- (c)  $\bar{V}(x) \leq \epsilon, x \in \mathcal{S}$ ,

where  $\gamma$  is the decay rate associated with the buffer structure  $\mathcal{S}$ .

For  $q \in \mathbb{R}^d$  and bitmap  $b$  define the jump probabilities:

$$\bar{p}_b^*(q)(v_{i,j}) = \begin{cases} \lambda \frac{\exp(-q(j)/2)}{N_b(q)}, & i = 0, j = 1 \\ \mu_{i,j} \frac{\exp((q(i)-q(j))/2)}{N_b(q)}, & i \neq 0, b(i) = 1, i \rightarrow j \\ \mu_{i,0} \frac{\exp(q(i)/2)}{N_b(q)}, & i \neq 0, b(i) = 1 \\ \mu_{i,j} \frac{1}{N_b(q)}, & i \neq 0, b(i) = 0, i \rightarrow j \text{ or } j = 0. \end{cases} \quad (8)$$

Any smooth function  $W : \mathbb{R}^d \rightarrow \mathbb{R}$  can be used to define a stochastic kernel  $\bar{p}$  as follows:

$$\bar{p}_W(v|x) = \bar{p}_{b_x}^*(v|DW(x/n)), \quad (9)$$

where  $DW$  is the gradient of  $W$ .

Theorem 4.1.1 of [16] asserts that the IS transition kernel defined by smooth subsolutions to (7) satisfying growth conditions on their Hessians are asymptotically optimal. For completeness we quote this theorem below.

**Theorem 3.1** (Theorem 4.1.1 of [16]). *Let  $\{\bar{V}_n\}$  be a sequence of  $C^2([0, 1]^d, \mathbb{R})$  functions that satisfy 1)  $\bar{V}_n$  is a  $\epsilon_n$ -subsolution 2)  $\left| \frac{\partial^2 \bar{V}_n}{\partial x_i \partial x_j} \right| \leq \frac{C}{\delta_n}$  for  $i, j \in \{1, 2, \dots, d\}$ , for some fixed constant  $C < \infty$  and a pair of non negative sequences  $\{\delta_n\}$  and  $\{\epsilon_n\}$  that converge to 0 and satisfy  $n\delta_n \rightarrow \infty$ . Then the IS scheme defined by the subsolutions  $\bar{V}_n$  is asymptotically optimal.*

In the next section we will construct a sequence of smooth subsolutions to (7) that satisfy the conditions of this theorem by piecing together at most  $2^d$  affine functions for the buffer structure  $\mathcal{S}_1$ . We will find out in Section 6 that the same sequence also works for  $\mathcal{S}_2$  (one individual buffer for each node).

## 4 Single shared buffer

In this section we will be working with  $\mathcal{S} = \mathcal{S}_1 = \{x \in \mathbb{R}_+^d : x(1) + x(2) + \dots + x(d) = 1\}$ . As noted before,  $\mathcal{S}_1$  corresponds to a single buffer shared by all queues in the system. To remind the reader, we are interested in the overflow probability:  $p_n \doteq P_{\mathbf{s}}(X \text{ hits } S_n \text{ before it hits } 0)$ , where  $S_n \doteq \{x \in \mathbb{Z}_+^d : x/n \in \mathcal{S}_1\}$ . It is proved in [8] that

$$\lim_{n \rightarrow \infty} -\frac{1}{n} \log p_n = \gamma_1 = \min_i -\log \rho_i. \quad (10)$$

In particular, this implies that  $\mathcal{S}_1$  satisfies the conditions of Assumption 1.

### 4.1 The smooth subsolution

We define the following quantities to write down the subsolution to (7) that we have in mind.

**The effective rate  $M_i(b)$  of node  $i$  at boundary  $b$ .**

$$M_i(b) \doteq \begin{cases} \mu_i, & \text{if } b(i) = 1, \\ \min\left(\mu_i, \sum_{k:i \rightarrow k} M_k(b) + \mu'_{i,0}\right), & \text{if } b(i) = 0, \end{cases} \quad (11)$$

where  $\mu'_{i,0} \doteq \Lambda_i \frac{\mu_{i,0}}{\mu_i}$  is the traffic that leaves the system through node  $i$ . The recursive formula (11) is the main ingredient of our construction and is suggested by the definition of the Hamiltonians (5) and the HJB equation (7) to which we are constructing a subsolution. The form of (11) and the role  $M_i(b)$  plays in the solution to the problem suggests the following interpretation of (11). (11) seems to compute an “effective” service rate for each node taking into account whether the node is empty or nonempty. If a node is nonempty its effective service rate is simply its service rate. If the node is empty, (11) seems to consider it as a system whose components are the nodes it directly feeds and computes the effective rate as the total effective rates of the components. There is also an upper bound on the effective rate, namely the service rate and if the aforementioned total exceeds this bound then again the effective rate is set to be the service rate. In this interpretation  $\mu'_{i,0}$  can be thought of as the effective rate of outside of the network for the empty node  $i$ .

**The effective utility  $\rho_i(b) \doteq \frac{\Lambda_i}{M_i(b)}$ .** The effective utility of a node is the ratio of its arrival rate to its effective service rate. If node  $i$  is nonempty then it coincides with the ordinary utility  $\rho_i$ .

**The effective gradient  $q \in \mathbb{R}^d$  associated with boundary  $b$ .**

$$q(i) \doteq 2 \log \rho_i(b) = 2 \log \frac{\Lambda_i}{M_i(b)}, \quad (12)$$

where  $q(i)$  denotes the  $i^{\text{th}}$  component of the vector  $q$ . We will use the affine functions defined by the effective gradients to construct our subsolution of (7). The effective gradient  $q$  of the boundary  $b$  will be the gradient of the smooth subsolution around that boundary.

For each boundary  $b$  there is an effective gradient  $q$ . It may happen that two boundaries  $b_1$  and  $b_2$  have the same effective gradients. Let  $EG \doteq \{q_1, q_2, \dots, q_L\}$ ,  $L \leq 2^d$ , be the set of unique effective gradients. We identify two extreme elements of the set  $EG$ : firstly, the effective gradient corresponding to the boundary  $0 = (0, 0, 0, \dots, 0, 0)$  (all nodes empty) is  $0 = (0, 0, 0, \dots, 0, 0)$  (this follows from (11) and the definition of  $\mu'_{i,0}$ ). Secondly, the effective gradient corresponding to the boundary  $1 = (1, 1, 1, \dots, 1, 1)$  (all nodes non-empty) is the vector whose  $i^{\text{th}}$  component is  $\log \Lambda_i / \mu_i$ .

Now define

$$m_i(b) \doteq \begin{cases} \mu_i, & \text{if } b(i) = 1, \\ \sum_{k:i \rightarrow k} m_k(b) + \mu'_{i,0}, & \text{if } b(i) = 0. \end{cases} \quad (13)$$



The simple gradient  $q = (q_1, q_2, \dots, q_d)$  associated with boundary  $b$  is defined as  $q(i) \doteq 2 \log \frac{\Lambda_i}{m_i(b)}$  where as before  $\Lambda_i$  is the arrival rate to node  $i$ . The following lemma relates simple and effective gradients. Bitmaps  $b'$  and  $b$  satisfy  $b' \geq b$  if  $b'(i) \geq b(i)$  for all  $i \in \{1, 2, 3, \dots, d\}$ .

**Lemma 4.1.** *Let  $q$  be the effective gradient associated with boundary  $b$ . Then there exists a boundary  $\bar{b} \geq b$  such that  $q$  is the simple gradient associated with  $\bar{b}$ .*

*Proof.* If  $b = (1, 1, 1, \dots, 1, 1)$  then there is nothing to prove because for this boundary the effective gradient and the simple gradient are the same. Then we assume that there are some empty nodes indicated by  $b$ .  $\bar{b} \geq b$  is constructed as follows. Initially set  $\bar{b} = b$ . For each empty node  $i$  in  $b$  set  $\bar{b}_i$  to 1 if  $M_i(b) = \mu_i$ . (see (11)). It is clear that 1)  $\bar{b} \geq b$  and 2) the effective and simple gradients of  $\bar{b}$  are the same vector which is the effective gradient of  $b$ .  $\square$

**Definition 4.1.** *For an effective gradient  $q_l \in EG$  let  $\bar{b}$  be the boundary whose simple gradient equals  $q_l$ . Define  $\alpha_l$  to be the number of 0's in  $\bar{b}$  plus 1.*

The  $\alpha_l$ 's will determine the size of the regions where the change of measure defined by  $q_l$  is used for IS. Now define the piecewise affine subsolution

$$W_l^\epsilon(x) = 2\gamma_1 - \alpha_l \epsilon + \langle q_l, x \rangle, \quad W^\epsilon(x) = \bigwedge_{l=1}^L W_l^\epsilon(x), \quad (14)$$

where  $L$  is the number of effective gradients and  $q_l$  are the effective gradients.  $W^\epsilon$  is piecewise affine and not smooth in general. To obtain the sequence of smooth subsolutions satisfying the assumptions of Theorem 4.1.1 of [16] one has to let  $\epsilon$  depend on  $n$  and then smooth  $W^\epsilon$ . One smoothing method that is simple and easy to implement on a computer is the following [6]. Define

$$W^{\epsilon, \delta}(x) \doteq -\delta \log \sum_{l=1}^L \exp \left\{ -\frac{1}{\delta} W_l^\epsilon(x) \right\}. \quad (15)$$

This smoothing algorithm is based on the following fact: For  $d$  real numbers  $a_1, a_2, \dots, a_d$ :  $-\lim_{\delta \rightarrow 0} \delta \log \left( \sum_{i=1}^d e^{-a_i/\delta} \right) = \bigwedge_{i=1}^d a_i$ . By Lemma 3.12 of [12],  $W^{\epsilon, \delta} \rightarrow W^\epsilon$  uniformly as  $\delta \rightarrow 0$ . In addition,  $W^{\epsilon, \delta}$  is continuously differentiable and a simple direct calculation gives

$$DW^{\epsilon, \delta}(x) = \sum_{l=1}^L w_l^{\epsilon, \delta}(x) q_l, \quad w_l^{\epsilon, \delta}(x) \doteq \frac{\exp \{-W_l^\epsilon(x)/\delta\}}{\sum_{k=1}^L \exp \{-W_k^\epsilon(x)/\delta\}}. \quad (16)$$

**Lemma 4.2.**  *$W^{\epsilon, \delta}$  defined in (15) satisfies:*

1.  $H_{b_x}(DW^{\epsilon, \delta}(x)) \geq -C_1 \exp(-\frac{\epsilon}{\delta})$ ,

2.  $W^{\epsilon, \delta}(0) \geq 2\gamma_1 - \epsilon \left( \frac{\delta}{\epsilon} \log \sum_{l=1}^L \exp \left\{ \frac{\rho_l}{\delta/\epsilon} \right\} \right),$
3.  $W^{\epsilon, \delta}(x) \leq 0$  for  $x \in \mathcal{S}_1,$
4.  $\left| \frac{\partial^2 W^{\epsilon, \delta}}{\partial x_i \partial x_j} \right| \leq \frac{C_2}{\delta},$

where  $C_1$  and  $C_2$  are constants that only depend on the parameters of the network (arrival and service rates and the routing probabilities).

The proof of Lemma 4.2 is in Appendix A. This lemma directly implies that, for  $\epsilon_n = -\delta_n \log \delta_n$  and  $\delta_n$  chosen such that  $\delta_n \rightarrow 0$  and  $n\delta_n \rightarrow \infty,$  the sequence of smooth subsolutions  $W^{\epsilon_n, \delta_n}$  (where  $W^{\epsilon, \delta}$  is defined as in (15)) satisfy the conditions of the optimality Theorem 4.1.1 [16]. This means that the IS scheme defined by these subsolutions through (9) is asymptotically optimal.

Here we repeat an idea from [6, 12]. The formula (9) can be used to translate any smooth function into an IS transition kernel. However, for the smooth subsolutions there is a slightly different way of defining IS transition kernels which turn out to be very convenient in computer simulations.

For  $x \in \mathbb{Z}_+^d$  define

$$\bar{p}^*(v_{i,j}|x) = \sum_{l=1}^L w_l^{\epsilon, \delta}(x/n) \bar{p}_{b_x}^*(q_l)(v_{i,j}), \quad (17)$$

i.e., we switch the order of taking the average against the weights  $w_l^{\epsilon, \delta}$  and applying the map  $\bar{p}_{b_x}^*(\cdot)$  of (8). The advantage of  $\bar{p}^*$  of (17) is that it requires the computation of  $\bar{p}_b^*(q_l)$  only once at the beginning of the estimation procedure. During the simulation only the weights are computed dynamically and averages of the precomputed  $\bar{p}_b^*(q_l)$  will be the IS rates. Theorem 4.1.1 of [16] doesn't cover this way of computing the IS rates. However, the modification of this theorem to accommodate direct averaging entails no significant changes. In the next section we report on the numerical performance of these algorithms.

## 4.2 Interpretation of the IS algorithm defined by the subsolution

Let  $b$  a boundary and  $q$  its effective gradient. (17) essentially uses  $\bar{p}_b(q)$  as the IS change of measure when the queueing process is on the boundary  $b$  and away from the lower dimensional boundaries contained in  $b$ . Looking at (12) and (8) one sees that  $\bar{p}_b(q)$  is simply the following change of measure:

$$\bar{\mu}_{i,j} = \begin{cases} \mu_{i,j}, & \text{if node } i \text{ is empty,} \\ \mu_{i,j} \frac{\rho_i(b)}{\rho_j(b)}, & \text{if node } i \text{ is nonempty,} \end{cases} \quad (18)$$

where  $\rho_i(b)$  and  $\rho_j(b)$  are the effective utilities of nodes  $i$  and  $j$ . These new rates are renormalized so that they sum to 1. By convention  $\rho_0(b) = 1$ , i.e., the outside of the system is thought of as a node with utility 1. The IS scheme given by (17) uses a convex combination of (18) when the simulated queuing process transitions from one boundary to another.

(18) illustrates well how the IS change of measure given by the subsolution approach works. In the course of a simulation, the IS change of measure depends on which nodes are currently empty and nonempty. The service probabilities of empty nodes are not modified. The service probability  $\mu_{i,j}$  of a nonempty node  $i$  is modified through a comparison of the traffic at the source  $i$  and the target  $j$ ; the service rate is increased if the source is busier, decreased otherwise. The goal seems to be to direct traffic to the less strained node. The traffic is measured by the effective utilities. For an empty node the effective utility is a value that takes into account the traffic in the nodes that follow it immediately. We also note that the arrival rate  $\lambda$  is replaced by  $\bar{\lambda} = \lambda \frac{1}{\rho_1(b)}$  which is always larger than  $\lambda$ . Therefore the rate of traffic from outside is always increased. Similarly, the rate of traffic to outside is always decreased.

We would like to also note that the standard state independent heuristic IS algorithms based on large deviations results can be thought of as variants of (18) in which the standard utilities are used instead of the effective utilities.

## 5 Numerical Results

**Choice of  $\epsilon$  and  $\delta$ .** The IS algorithm defined by  $W^{\epsilon,\delta}$  of (15) has two parameters  $\epsilon$  and  $\delta$ . The optimality Theorem 3.1 suggest  $\delta \approx C/n$  and  $\epsilon \approx -\delta \log \delta$ . Asymptotic optimality criterion is not precise enough to impose a value for  $C$ . For the choice of this constant we used experimental evidence.

Once  $\epsilon$  and  $\delta$  are fixed,  $\bar{p}^*(v|x)$  of (17) is used as the IS change of measure. The effective gradients  $q_1, q_2, \dots, q_L$  and their  $\alpha_l$ 's are computed by iterating over all boundaries  $b$  and computing the effective gradient of each of them using the formulas (11) and (12) and the Definition 4.1.

In the following subsections we present simulation results for various Jackson networks with a tree topology. In all the estimations  $K = 10000$  sample paths were used.

**Example 1.** We first consider the network in Figure 1. Let us consider the case when  $\lambda = 0.04, \mu_{1,2} = \mu_{1,0} = 0.12, \mu_{2,0} = \mu_{2,3} = \mu_{2,4} = 0.08, \mu_{3,0} = \mu_{3,1} = \mu_{4,0} = \mu_{4,1} = 0.12$ . The node utilities in this case are:  $\rho_1 = 1/6, \rho_2 = 1/12, \rho_3 = \rho_4 = 1/36$ . In this example, the utilities are unevenly distributed and node 1 is the most strained node. We take  $n = 30$ . For  $n = 30$ , and with this four dimensional system, it is possible to compute

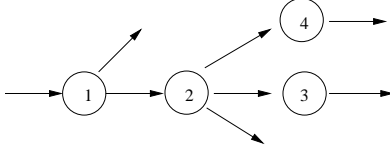


Figure 1: Example 1

Exact probability  $p_{30} = 3.269 \times 10^{-23}$

	Estimate $\hat{p}_n$	Standard Error	95 % CI
Est. 1	$3.50 \times 10^{-23}$	$0.19 \times 10^{-23}$	$[3.12, 3.88] \times 10^{-23}$
Est. 2	$3.22 \times 10^{-23}$	$0.16 \times 10^{-23}$	$[2.89, 3.54] \times 10^{-23}$
Est. 3	$3.28 \times 10^{-23}$	$0.17 \times 10^{-23}$	$[2.94, 3.61] \times 10^{-23}$
Est. 4	$3.32 \times 10^{-23}$	$0.17 \times 10^{-23}$	$[2.98, 3.66] \times 10^{-23}$
Est. 5	$3.16 \times 10^{-23}$	$0.16 \times 10^{-23}$	$[2.84, 3.48] \times 10^{-23}$

Table 1: Simulation Results for Example 1

$p_{30}$  without any simulation using the Markov property and straight-forward iteration. Such a computation yields  $p_{30} = 3.269 \times 10^{-23}$ . For the subsolution based IS algorithm we take  $\epsilon = 0.25$  and  $\delta = 0.08$ . There turns out to be only five effective gradients for the given rate values above. The results of five consecutive estimations using the subsolution based IS algorithm are displayed in Table 1. The ‘standard error’ column is the standard error of each estimation. The 95% confidence intervals are  $\hat{p}^n + [-2SE, 2SE]$ , where  $SE$  is the standard error displayed under the standard error column. These intervals are only formal, i.e., we make no assertion about the normality of these errors. Note that the estimation results are very close to the exact value and the “95% confidence intervals” are accurate: in all these estimations the exact value happened to be in the computed confidence interval. In total all five estimations took around 20 seconds on an ordinary laptop manufactured in 2004.

**Example 2.** Now we look at the 8-node network depicted in Figure 2. We

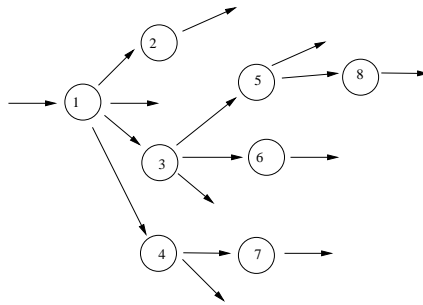


Figure 2: Example 3

	Estimate $\hat{p}_n$	Standard Error	95 % CI
Est. 1	$1.11 \times 10^{-6}$	$0.17 \times 10^{-6}$	$[0.78, 1.44] \times 10^{-6}$
Est. 2	$1.69 \times 10^{-6}$	$0.32 \times 10^{-6}$	$[1.04, 2.34] \times 10^{-6}$
Est. 3	$1.25 \times 10^{-6}$	$0.18 \times 10^{-6}$	$[0.89, 1.61] \times 10^{-6}$
Est. 4	$1.94 \times 10^{-6}$	$0.51 \times 10^{-6}$	$[0.92, 2.97] \times 10^{-6}$
Est. 5	$1.23 \times 10^{-6}$	$0.17 \times 10^{-6}$	$[0.89, 1.56] \times 10^{-6}$

Table 2: Simulation results for the network with eight nodes

take the arrival rate  $\lambda = 0.1248$ , The service rates are taken to be:  $\mu_{1,2} = 0.062442$ ,  $\mu_{1,3} = 0.1874$ ,  $\mu_{1,4} = 0.062442$ ,  $\mu_{1,0} = 0.062517$ ,  $\mu_{2,0} = 0.06$ ,  $\mu_{3,0} = 0.036$ ,  $\mu_{3,5} = 0.072$ ,  $\mu_{3,6} = 0.072$ ,  $\mu_{4,0} = 0.03$ ,  $\mu_{4,7} = 0.03$ ,  $\mu_{5,0} = 0.0365$ ,  $\mu_{5,8} = 0.0365$ ,  $\mu_{6,0} = 0.073$ ,  $\mu_{7,0} = 0.025$ ,  $\mu_{8,0} = 0.028$ . For this choice of the network parameters, the utility of each node turns out to be approximately:  $\rho_1 = 0.331738$ ,  $\rho_2 = 0.3465$ ,  $\rho_3 = 0.3466$ ,  $\rho_4 = 0.3465$ ,  $\rho_5 = 0.3419$ ,  $\rho_6 = 0.3466$ ,  $\rho_7 = 0.3465$ ,  $\rho_8 = 0.4158$ . All nodes are similarly utilized, although the load on node 8 is slightly heavier than the rest. A straightforward simulation with  $10^8$  samples estimate  $p_{30}$  to be  $1.2 \times 10^{-6}$  with a standard error of  $1.1 \times 10^{-6}$ . The subsolution based IS simulation results are given in Table 2. The parameters of the algorithm are taken to be  $\epsilon = 0.4$  and  $\delta = 0.1$ . Each estimation uses 10000 samples. For this network there are 256 effective gradients. Total run time for all these five estimations was about 20 minutes.

As can be seen, the subsolution based IS algorithm performs very well for this high dimensional system too: the estimate is within the 95% confidence interval of the MC estimator and the formal 95% confidence intervals of the IS simulation do not wildly fluctuate.

## 6 Individual Buffers for each Node

In this section we look at the buffer structure  $\mathcal{S}_2$ : for  $\beta \in \mathbb{R}_+^d$

$$\mathcal{S}_2 = \{x \in \mathbb{R}_+^d : x(i) = \beta(i) \text{ for some } i \text{ and } x(j) \leq \beta(j) \text{ for all } j\}.$$

As we noted before,  $S_n \doteq \{x \in \mathbb{Z}_+^d : x/n \in \mathcal{S}_2\}$  corresponds to  $d$  independent buffers, one for each node. The size of the buffer for node  $i$  is given by  $n\beta(i)$ . Without loss of generality we will assume that  $\sum_i \beta(i) = 1$ . We are, as before, interested in:  $p_n \doteq P_{\mathbf{s}}(X \text{ hits } S_n \text{ before it hits } 0)$ , where  $\mathbf{s} = (1, 0, 0, \dots, 0)$ . One can prove, using arguments similar to those in [8] that

$$\lim_{n \rightarrow \infty} -\frac{1}{n} \log p_n = \gamma_2 = \min_i -\beta(i) \log \rho_i, \quad (19)$$

where  $\rho_i$  are the node utilities. In particular, this implies that  $\mathcal{S}_2$  satisfies the conditions of Assumption 1. Our goal now is to prove that the IS algorithm defined by  $W^{\epsilon_n, \delta_n}$  is asymptotically optimal for the buffer structure  $\mathcal{S}_2$  as well (when buffer structure is changed to  $\mathcal{S}_2$ ,  $\gamma_1$  in (14) needs to be replaced with  $\gamma_2$ ). To prove this, it is enough to prove a version of Lemma 4.2 for  $\mathcal{S}_2$ . Note

Exact probability  $p_{19} = 6.8601 \times 10^{-9}$

	Estimate $\hat{p}_n$	Standard Error	95 % CI
Est. 1	$7.33 \times 10^{-9}$	$0.42 \times 10^{-9}$	$[6.50, 8.17] \times 10^{-9}$
Est. 2	$6.81 \times 10^{-9}$	$0.34 \times 10^{-9}$	$[6.12, 7.50] \times 10^{-9}$
Est. 3	$7.30 \times 10^{-9}$	$0.38 \times 10^{-9}$	$[6.53, 8.06] \times 10^{-9}$
Est. 4	$7.05 \times 10^{-9}$	$0.39 \times 10^{-9}$	$[6.28, 7.83] \times 10^{-9}$
Est. 5	$7.01 \times 10^{-9}$	$0.37 \times 10^{-9}$	$[6.26, 7.76] \times 10^{-9}$

Table 3: Simulation results for the case when each node has a separate buffer

that only item 3 of this lemma depends on  $\mathcal{S}$  and therefore we only have to prove that the same item holds for  $\mathcal{S}_2$ , which is done in the next lemma.

**Lemma 6.1.** *Define  $W_l^{c,\epsilon}(x) \doteq 2\gamma_2 - \alpha_l\epsilon + \langle q_l, x \rangle$ , where  $\alpha_l$  and  $q_l$  are defined as in (12) and Definition 4.1 and  $\gamma_2$  is the large deviation rate associated with the boundary  $\mathcal{S}_2$  (19). Define  $W^{\epsilon,\delta}$  by the expression (15). Then:  $W^{\epsilon,\delta}(x) \leq 0$  for  $x \in \mathcal{S}_2$ .*

*Proof.* Take any  $x \in \mathcal{S}_2$ . Then, there is an  $i \leq d$  such that  $x(i) = \beta(i)$ . Let  $q_L$  be the effective gradient of the boundary  $1 = (1, 1, 1, \dots, 1, 1)$ .

$$W(x) = -\delta \log \sum_{l=1}^L \exp \left\{ -\frac{1}{\delta} (2\gamma_2 - \alpha_l\epsilon + \langle q_l, x \rangle) \right\} \leq 2\gamma_2 + \langle q_L, x \rangle - \alpha_L\epsilon.$$

By definition,  $q_L(i) = 2 \log \frac{\mu_i}{\Lambda_i}$  and the rest of the components of  $q_L$  are negative. These facts, (19),  $x \in \mathbb{R}_+^d$ , and  $x(i) = \beta(i)$  imply that the last display is less than  $-\alpha_L\epsilon$ . This finishes the proof of this lemma.  $\square$

**Numerical example** Consider a network with five nodes with the following service rates:  $\mu_{1,2} = 0.038$ ,  $\mu_{1,3} = 0.057$ ,  $\mu_{1,0} = 0.095$ ,  $\mu_{2,4} = 0.076$ ,  $\mu_{2,0} = 0.114$ ,  $\mu_{3,5} = 0.095$ ,  $\mu_{3,0} = 0.095$ ,  $\mu_{4,0} = 0.19$ ,  $\mu_{5,0} = 0.19$  and  $\lambda = 0.1$ . We will suppose that the buffer sizes for the nodes are respectively: 15, 15, 17, 18, 19. Then  $n = 19$  and  $\beta(1) = \beta(2) = 15/19$ ,  $\beta(3) = 17/19$ ,  $\beta(4) = 18/19$ ,  $\beta(5) = 1$ . The choice of the buffer sizes are rather arbitrary. We chose them relatively small so that it was possible to compute the buffer overflow probability  $p_{19}$  using the Markov property and direct iteration. The exact value of  $p_{19}$  turns out to be  $p_{19} = 6.8601 \times 10^{-9}$ .

The relative node utilities are:  $\beta(1)\rho_1 = 0.208$ ,  $\beta(2)\rho_2 = 0.042$ ,  $\beta(3)\rho_3 = 0.013$ ,  $\beta(4)\rho_4 = 0.0004$ ,  $\beta(5)\rho_5 = 0.0008$ . Node 1 is clearly the most strained node and the loads on the rest of the nodes are spread. Following the same reasoning as in Section 5 we take  $\epsilon = 0.3$  and  $\delta = 0.1$ . The IS simulation now proceeds as before. One uses  $\bar{p}(\cdot|x) = \bar{p}^*(\cdot|x)$  given in (17) for the IS change of measure. There turns out to be only eight effective gradients (out of a maximum of 32). The results of five consecutive estimations using the subsolution based IS algorithm are displayed in Table 3. Once again, the estimation results are close to the exact value  $p_{19} = 6.8601 \times 10^{-9}$  and the formal 95% confidence intervals are tight and happen to contain the exact value.

## 7 Discussion

The goal of the present paper was to extend the IS algorithms in [12], which looked at tandem Jackson networks, to more general networks. We thought tree networks were an interesting generalization and a comparison with the algorithms in [12] will reveal that the tree networks require much more sophisticated subsolutions and IS algorithms for asymptotic optimality. [7] proves a further generalization to arbitrary stable Jackson networks. In this section we would like to discuss how the results in [7] relate to our results.

Let  $p_{i,j} = \mu_{i,j}/\mu_i$  denote the routing probability from node  $i$  to  $j$ , where  $j$  is allowed to take the value 0. In the notation of the present paper, the IS algorithm in [7] can be described as follows. Define the effective rate for the boundary  $b$  as:

$$M_i(b) \doteq \begin{cases} \mu_i, & \text{if } b(i) = 1, \\ \min \left( \mu_i, \sum_{k:i \rightarrow k} \frac{p_{i,k}\Lambda_i}{\Lambda_k} M_k(b) + \mu'_{i,0} \right), & \text{if } b(i) = 0. \end{cases} \quad (20)$$

As before if a node is nonempty under  $b$ , i.e.,  $b(i) = 1$ , then its effective rate is just the service rate  $\mu_i$ . If it is empty, one now takes a *weighted* sum of the effective rates of its neighbors, as before this sum is min'ed with  $\mu_i$ . The weight of  $M_k(b)$  is the fraction of the  $k^{\text{th}}$  node's traffic in the fluid model that is coming from node  $i$ . This fraction is always 1 for a tree network and thus for such networks (20) reduces to (11). Once the effective rates are defined as above one proceeds as in subsection 4.1.

We note that (11) is a recursive formula: one can start from the leaves of the network and go up and compute all effective gradients using (11). In the case of general Jackson networks (20) is an equation that needs to be solved; as observed in [7], it can be solved by reducing it to a linear equation, which is a generalization of (13). It can also be directly solved using (20) itself and an iterative method.

Another contribution of [7] is the identification of the large deviation decay rate  $\gamma$  of  $p_n$  for any exit boundary  $\mathcal{S}$  for which such a rate exists. In the notation of the present paper, [7, Proposition 3.1] asserts that

$$\gamma = \inf_{x \in \mathcal{S}} -\langle q, x \rangle$$

where  $q$  is the effective or simple gradient of  $b = (1, 1, 1, \dots, 1)$ . As noted in [7] this implies that the IS change of measure given by (20), or (11) for the case of tree networks, is asymptotically optimal for any buffer structure  $\mathcal{S}$  for which there is a large deviation decay rate.

Finally, we would like to point out a parametrization that seems most natural for (20). Define  $\mathbf{M}_i \doteq 1/\rho_i$  and  $\mathbf{M}_i(b) \doteq M_i(b)/\Lambda_i$ . The first is the ordinary service to arrival ratio of node  $i$ . The second can be thought of as

the effective service to arrival ratio of the same node when the system is on boundary  $b$ . By convention let  $\mathbf{M}_0(b) = 1$ , i.e., the service to arrival ratio of the outside of the system is 1. In terms of these new variables (20) is simply:

$$\mathbf{M}_i(b) \doteq \begin{cases} \mathbf{M}_i, & \text{if } b(i) = 1 \\ \min(\mathbf{M}_i, \sum_{k:i \rightarrow k} p_{i,k} \mathbf{M}_k(b)), & \text{if } b(i) = 0, \end{cases} \quad (21)$$

where  $k = 0$  value is allowed in the summation to denote the outside of the system. If node  $i$  is empty, its effective service to arrival ratio is taken to be the average of the effective ratios of the nodes that are directly connected to  $i$ . The average is taken with respect to the routing probabilities. As before the ordinary service to arrival ratio is an upperbound on the effective one. So if the average exceeds the ordinary, the effective ratio is set to the ordinary ratio.

The effective gradient  $q$  for  $b$  will have components  $-2 \log \mathbf{M}_i(b)$ . And the change of measure  $\bar{p}_b(q)$  is:

$$\bar{\mu}_{i,j} = \begin{cases} \mu_{i,j}, & \text{if node } i \text{ is empty} \\ \mu_{i,j} \frac{\mathbf{M}_j(b)}{\mathbf{M}_i(b)}, & \text{if node } i \text{ is nonempty,} \end{cases}$$

and this is renormalized so that  $\bar{\mu}_{i,j}$  sum to 1. One can use (21) directly to compute the IS algorithm.

## A Proof of Lemma 4.2

Before we begin, a convention: the decay rate  $\gamma$  depends on the buffer structure. We used  $\gamma_1$  for the shared buffer ( $\mathcal{S}_1$ ) and  $\gamma_2$  for the individual buffers for each node ( $\mathcal{S}_2$ ). In the proofs we will simply write  $\gamma$ .

**Lemma A.1.** *Let  $q$  be the simple gradient associated with boundary  $b$ . Then  $H_{\bar{b}}(q) = 0$  for any  $\bar{b} \geq b$ .*

*Proof.* We first prove that  $H_b(q) = 0$ , or equivalently  $N_b(q) = 1$ . Directly from the definitions (5), (13) one sees that  $N_b(q) = 1$  if and only if

$$\sum_{i:b(i)=1} \left( \sum_{j:i \rightarrow j} m_j(b) + \mu'_{i,0} \right) + m_1(b) = \lambda + \sum_{i:b(i)=1} \mu_i.$$

The definition of  $\mu'_{i,0}$  directly imply that  $\sum_{i=1}^d \mu'_{i,0} = \lambda$ . The above display follows from this fact and (13).

Next fix a  $\bar{b} > b$ . We will show that  $N_{\bar{b}}(q) = 1$ .

$$N_{\bar{b}}(q) - N_b(q) = \sum_{i:\bar{b}(i)-b(i)=1, i \rightarrow j} \mu_{i,j} e^{\frac{q(i)-q(j)}{2}} + \sum_{i:\bar{b}(i)-b(i)=1} \mu_{i,0} e^{q(i)/2} - \sum_{i:\bar{b}(i)-b(i)=1} \mu_i \quad (22)$$



Fix  $i$  such that  $\bar{b}(i) - b(i) = 1$  and let  $C$  denote the terms contributed by the index  $i$  in the first two sums. Our goal is now to show that  $C = \mu_i$ . This will imply that first two sums and the last sum in (22) cancel each other and that  $N_{\bar{b}}(q) = N_b(q)$ . Because  $b(i) = 0$  we have that

$$m_i(b) = \sum_{j:i \rightarrow j} m_j(b) + \mu'_{i,0}. \quad (23)$$

Then

$$C = \mu_{i,0} e^{q(i)/2} + \sum_{j:i \rightarrow j} \mu_{i,j} e^{\frac{q(i)-q(j)}{2}} = \mu_{i,0} \frac{\Lambda_i}{m_i(b)} + \sum_{j:i \rightarrow j} \mu_{i,j} \frac{\Lambda_i}{m_i(b)} \frac{m_j(b)}{\Lambda_j}$$

At this point the facts  $\Lambda_j = \Lambda_i \frac{\mu_{i,j}}{\mu_i}$  and  $\frac{\mu_{i,0} \Lambda_i}{\mu_i} = \mu'_{i,0}$  and (23) and simple arithmetic yield  $C = \mu_i$ . Thus the difference in (22) is zero, i.e.,  $N_{\bar{b}}(q) = N_b(q) = 1$ . This finishes the proof of this lemma.  $\square$

**Lemma A.2.** *Let  $q$  be the effective gradient associated with boundary  $b$ . Then  $H_{b'}(q) \geq 0$  for all  $b' \geq b$ .*

*Proof.*  $H_{b'}(q) \geq 0$  if and only if  $N_{b'}(q) \leq 1$ . By Lemma 4.1 there exists  $\bar{b} \geq b$  such that  $q$  is the simple gradient associated with  $\bar{b}$ . Then by Lemma A.1  $N_{b'}(q) = 1$  for all  $b' \geq \bar{b}$ . Now take any  $b'$  such that  $b' < \bar{b}$  and  $b' \geq b$ . Because  $\bar{b} > b' \geq b$  we have

$$\begin{aligned} & N_{\bar{b}}(q) - N_b(q) \\ &= \sum_{i:\bar{b}(i)-b(i)=1} \left( \sum_{j:i \rightarrow j} \mu_{i,j} e^{\frac{q(i)-q(j)}{2}} + \mu_{i,0} e^{q(i)/2} \right) - \sum_{i:\bar{b}(i)-b(i)=1} \mu_i \\ &= \sum_{i:\bar{b}(i)-b(i)=1} \left( \sum_{j:i \rightarrow j} \mu_{i,j} \frac{\Lambda_i}{M_i(b)} \frac{M_j(b)}{\Lambda_j} + \mu_{i,0} \frac{\Lambda_i}{M_i(b)} \right) - \sum_{i:\bar{b}(i)-b(i)=1} \mu_i \\ &= \sum_{i:\bar{b}(i)-b(i)=1} \left( \mu_i \frac{\sum_{j:i \rightarrow j} M_j(b) + \mu'_{i,0}}{M_i(b)} \right) - \sum_{i:\bar{b}(i)-b(i)=1} \mu_i \end{aligned} \quad (24)$$

Now by the construction of  $\bar{b}$ ,  $\bar{b}(i) - b(i) = 0$  if and only if  $M_i(b) = \mu_i \leq \sum_{i \rightarrow j} M_j(b) + \mu'_{i,0}$ . The last display and (24) imply  $N_{\bar{b}}(q) \geq N_b(q)$ . Because  $N_{\bar{b}}(q) = 1$  (because  $q$  is the simple gradient associated with boundary  $b$ ) this finishes the proof of this lemma.  $\square$

*Proof of Lemma 4.2.* The proof of this lemma is similar to the proof of Theorem 4.31 in [16]. For small positive real numbers  $\delta, \epsilon$  let  $W^{\epsilon, \delta}$  be defined as in (15). For ease of notation we will drop the superscript  $(\epsilon, \delta)$  and write  $W$ . We would like to prove the following: there is a constant  $C_1$  that only depends on the parameter system such that for all  $x \in \mathbb{R}_+^d$   $H_b(DW(x)) \geq -C_1 \exp(-\epsilon/\delta)$ ,

where  $b$  defined in (6) is the boundary corresponding to  $x$ . Let  $E$  be the set of effective gradients  $q$  such that there is a boundary  $b' \leq b$  with effective gradient  $q$ . Define  $q' = \sum_{q_l \in E} w_l^{\epsilon, \delta}(x) q_l$ , where  $w_l^{\epsilon, \delta}$  are the weights defined in (16). Once again to ease notation, we drop the superscript  $(\epsilon, \delta)$ . Its definition directly implies that  $H_b$  is concave and Lipschitz continuous. By Lemma A.2 we have that  $H_b(q) \geq 0$  for  $q \in E$ . This fact and the concavity of  $H_b$  and  $H_b(0) = 0$  imply that  $H_b(q') \geq 0$ . This, (16) and the Lipschitz continuity of  $H_b$  give

$$\begin{aligned} H_b(DW(x)) &= H_b(q') + H_b(DW(x)) - H_b(q') \geq |H_b(DW(x)) - H_b(q')| \\ &\geq K|q' - DW(x)| = -K \sum_{q^l \in E^c} w_l(x) |q^l|. \end{aligned}$$

The last inequality follows from (16) and the triangle inequality. Therefore to prove the first part of Lemma 4.2 it is enough to prove  $w_l(x) \leq \exp(-\epsilon/\delta)$ , for  $l$  such that  $q_l \in E^c$ .

By its definition (16)  $w_l$  equals

$$\begin{aligned} w_l(x) &= \frac{\exp\{-W_l^\epsilon(x)/\delta\}}{\sum_{j=1}^L \exp\{-W_j^\epsilon(x)/\delta\}} = \frac{\exp\{(\alpha_l \epsilon - \langle q_l, x \rangle)/\delta\}}{\sum_{j=1}^L \exp\{(\alpha_j \epsilon - \langle q_j, x \rangle)/\delta\}} \\ &\leq \frac{\exp\{(\alpha_l \epsilon - \langle q_l, x \rangle)/\delta\}}{\exp\{(\alpha_{j_0} \epsilon - \langle q_{j_0}, x \rangle)/\delta\}}, \end{aligned} \quad (25)$$

where  $q_{j_0}$  is an effective gradient to be selected. By Definition 4.1,  $\alpha_l$  is one plus the number of 0's in the the boundary (bitmap)  $r$  whose simple gradient equals  $q_l$ . Form the bitmap  $\tilde{r}$  from  $r$  as follows: if  $r(i) = 1$  but  $b_x(i) = 0$  then set  $\tilde{r}(i) = 0$  otherwise set  $\tilde{r}(i) = r(i)$ . By this construction  $\tilde{r} \leq b_x$  and  $\tilde{r} < r$ . The last inequality is strict, because otherwise we would have  $b_x = r$  which would imply, by Lemma A.2,  $H_{b_x}(q_l) \geq 0$  which in turn contradicts  $q_l \notin E$ . Let  $q_{j_0}$  be the effective gradient associated with the bitmap  $\tilde{r}$ .  $\tilde{r} \leq b_x$  and Lemma A.2 imply that  $H_{b_x}(q_{j_0}) \geq 0$ . This implies that  $q_{j_0} \in E$  and consequently  $q_{j_0} \neq q_l \in E^c$ . These facts and the strict inequality  $\tilde{r} < r$  imply that  $\alpha_{j_0} - \alpha_l \geq 1$ .

Furthermore, remember  $x$  is such that  $x_i = 0$  if  $b_x(i) = 0$ . The bitmaps  $r$  and  $\tilde{r}$  differ only at such  $i$ . Then the effective gradients of these bitmaps, namely  $q_l$  and  $q_{j_0}$  will also differ only at such  $i$ . This means  $\langle q_l, x \rangle = \langle q_{j_0}, x \rangle$ . These considerations and (25) imply  $w_l(x) \leq \exp(-\epsilon/\delta)$  and hence the first part of Lemma 4.2.

By its definition

$$W(0) = -\delta \log \sum_{l=1}^L \exp\left\{-\frac{2\gamma - \alpha_l \epsilon}{\delta}\right\} = 2\gamma - \epsilon \left( \frac{\delta}{\epsilon} \log \sum_{l=1}^L \exp\left\{\frac{\alpha_l}{\delta/\epsilon}\right\} \right)$$

This proves the second part of Lemma 4.2.

Now let us prove the third part. Let  $q_L$  be the effective gradient of the boundary  $1 = (1, 1, 1, \dots, 1, 1)$ . For  $x \in \mathbb{R}_+^d$  with  $x_1 + x_2 + \dots + x_d = 1$  we have the following estimate:

$$W(x) = -\delta \log \sum_{l=1}^L \exp \left\{ -\frac{1}{\delta} (2\gamma - \alpha_l \epsilon + \langle q_l, x \rangle) \right\} \leq \langle q_L, x \rangle + 2\gamma - \alpha_L \epsilon.$$

By definition  $q_L(i) = 2 \log \frac{\mu_i}{\Lambda_i}$ . This and (10) imply that the last line is less than  $-\alpha_L \epsilon$ . This finishes the proof of the third part of Lemma 4.2. It only remains to prove the last part. Differentiating the first expression in (16) gives:  $\frac{\partial^2 W}{\partial x_j \partial x_i}(x) = \sum_{l=1}^L \frac{\partial w_l}{\partial x_j}(x) q_l(i)$ . Differentiating the second expression in (16) gives:  $\frac{\partial w_l}{\partial x_j}(x) = \frac{1}{\delta} w_l(x) \left( \sum_{k=1}^L w_k(x) (q_k(j) - q_l(j)) \right)$ . These imply the bound in part 4 of Lemma 4.2, which is what we wanted to prove.  $\square$

## References

- [1] Pieter-Tjerk De Boer and Victor F. Nicola. Adaptive state-dependent importance sampling simulation of markovian queueing networks. *European Transactions on Telecommunications*, 13:303–315, 2001.
- [2] Sandeep Juneja Cheng-Shang Chang, Philip Heidelberger and Perwez Shahabuddin. Effective bandwidth and fast simulation of atm intree networks. *Performance Evaluation*, 20:45–66, 1994.
- [3] Paul Dupuis and Hui Wang. Importance sampling, large deviations and differential games. *Stochastics and Stochastic Reports*, 76(6):481–508, 2004.
- [4] Paul Dupuis and Hui Wang. Adaptive importance sampling for uniformly recurrent markov chains. *Annals of Applied Probability*, 15(1):1–38, 2005.
- [5] Paul Dupuis and Hui Wang. Subsolutions of an isaacs equation and efficient schemes for importance sampling: Convergence analysis. 2005. Preprint available at <http://www.dam.brown.edu/people/huiwang>.
- [6] Paul Dupuis and Hui Wang. Subsolutions of an isaacs equation and efficient schemes for importance sampling: Examples and numerics. 2005. Preprint available at <http://www.dam.brown.edu/lcds/publications>.
- [7] Paul Dupuis and Hui Wang. Importance sampling for jackson networks. *preprint*, 2008.
- [8] Paul Glasserman and Shing-Gang Kou. Analysis of an importance sampling estimator for tandem queues. *ACM Transactions on Modeling and Computer Simulation*, 5:22–42, 1995.

- [9] S. Juneja and V. Nicola. Efficient simulation of buffer overflow probabilities in jackson networks with feedback. *ACM Transactions on Modeling and Computer Simulation*, 15:281–315, 2005.
- [10] D. Koroese and V. Nicola. Efficient simulation of jackson networks. *ACM Transactions on Modeling and Computer Simulation*, 12:119–141, 2002.
- [11] S. Parekh and Jean Walrand. A quick simulation method for excessive backlogs in networks of queues. *IEEE Transactions on Automatic Control*, 34:54–66, 1989.
- [12] Ali Devin Sezer Paul Dupuis and Hui Wang. Dynamic importance sampling for queueing networks. *Annals of Applied Probability*, 17(4):1306–1346, 2007.
- [13] D. P. Kroese Pieter-Tjerk De Boer and R. Y. Rubenstein. A fast cross-entropy method for estimating buffer overflows in queueing networks. *Management Science*, 50:883–895, 2004.
- [14] John S. Sadowsky. Large deviations theory and efficient simulation of excessive backlogs in a gi/gi/m queue. *IEEE Transactions on Automatic Control*, 36:1383–1394, 1991.
- [15] Ali Devin Sezer. Asymptotically optimal importance sampling for jackson networks with a tree topology, preprint. Available at <http://arxiv.org/abs/0708.3260> .
- [16] Ali Devin Sezer. *Dynamic Importance Sampling for Queueing Networks, Ph.D. thesis*. Brown University Division of Applied Mathematics, 2005. Preprint available at <http://www.dam.brown.edu/people/sezer>.
- [17] David Siegmund. Importance sampling in the monte carlo study of sequential tests. *The Annals Statistics*, 4:673–684, 1976.
- [18] Lei Wei and Honghui Qi. An efficient importance sampling method for rare event simulation in large scale tandem networks. *Proceedings of the 2002 Winter Simulation Conference*, pages 580–587, 2002.