Submitted for publication.

Optimal resource capacity management for stochastic networks

A.B. Dieker
H. Milton Stewart School of ISyE, Georgia Institute of Technology, Atlanta, GA 30332, ton.dieker@isye.gatech.edu

S. Ghosh, M.S. Squillante
Mathematical Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, ghoshs@us.ibm.com, mss@us.ibm.com

We develop a general framework for determining the optimal resource capacity for each station comprising a stochastic network, motivated by applications arising in computer capacity planning and business process management. The problem is mathematically intractable in general and therefore one typically resorts to either overly simplistic analytical approximations or very time-consuming simulations in conjunction with metaheuristics. In this paper we propose an iterative methodology that relies only on the capability of observing the queue lengths at all network stations for a given resource capacity allocation. We theoretically investigate the proposed methodology for single-class Brownian tree networks, and further illustrate the use our methodology and the quality of its results through extensive numerical experiments.

Key words: capacity allocation; capacity planning; queueing networks; resource capacity management; stochastic networks; stochastic approximation
History: This paper was first submitted on ???

1. Introduction

Stochastic networks arise in many fields of science, engineering and business, where they play a fundamental role as canonical models for a broad spectrum of multi-resource applications. A wide variety of examples span numerous application domains, including communication and data networks, distributed computing and data centers, inventory control and manufacturing systems, call and contact centers, and workforce management systems. Of particular interest are strategic planning applications, the complexity of which continue to grow at a rapid pace. This in turn increases the technical difficulties of solving for functionals of stochastic networks as part of the analysis, modeling and optimization within strategic planning applications across diverse domains.

A large number of such strategic planning applications involve resource capacity management problems in which resources of different types provide services to various customer flows structured according to a particular network topology. Stochastic networks are often used to capture the dynamics and uncertainty of this general class of resource management problems, where each type of service requires processing by a set of resources with certain capabilities in a specific order and the customer processing demands are uncertain. The objective of these resource management problems is to determine the capacity allocation for each resource type comprising the stochastic network that maximize the expectation of a given financial or performance functional (or both) over a long-run planning horizon with respect to the customer workload demand and subject to constraints on either performance or financial metrics (or both). It is typically assumed that the planning horizon is sufficiently long for the underlying multidimensional stochastic process modeling the network to reach stationarity, where the multiple time scales involved in most applications of interest provide both theoretical and practical support for such a steady-state approach. The objective function is often based on rewards gained for servicing customers, costs incurred for the resource capacity allocation deployed, and penalties incurred for violating any quality-of-service agreements.

Our present study of resource capacity management problems in stochastic networks is primarily motivated by two particular application domains, although the same class of problems arise naturally in many other domains. The first application domain concerns capacity planning across a wide range of computer environments. This area has traditionally received a great deal of attention within the context of high-performance computing and Internet-based computing environments. However, with the recent proliferation of large-scale data centers and cloud computing platforms (see, e.g., Dikaiakos et al. (2009), Armbrust et al. (2009), Gartner (2012), Iyoob et al. (2012)), this application domain has become an even more important source of resource capacity management problems in practice. For these problem instances, the computer infrastructure is modeled as a stochastic network reflecting the topology of the infrastructure and the uncertainty of both the customer processing requirements and the overall demand. Various companies, such as BMC Software and IBM, provide products that address these resource capacity management problems within the context of information technology service management, data center automation, and computer performance management. The objectives of such solutions include minimizing the costs of a computing infrastructure while satisfying certain performance targets, as well as maximizing performance metrics within a given computing infrastructure budget.

The second application domain motivating our present study is business process management, which is a key emerging technology that seeks to enable the optimization of business process operations within an organization. Here, a business process generally consists of any series of activities performed within an organization to achieve a common business goal, such that revenues can be generated and costs are incurred at any step or along any flow comprising the process. A simple business process example is the processing of various types of medical claims by an insurance company. Another recent representative example is the flow of patients through the emergency department of a hospital, where the goal is to address a chronic inability of hospitals to deliver emergency services on demand in a highly dynamic and highly volatile environment in which the failure to match demand carries significant clinical risks to patients and financial risks to the hospital; see Guarisco and Samuelson (2011). For these problem instances, the business process is modeled as a stochastic network reflecting the topology of the series of activities to be performed and the uncertainty of both the processing requirements for each activity and the overall demand. Various companies, such as IBM and Oracle, provide products that address these resource capacity management problems within the context of business process modeling and optimization.

The primary state-of-the-art approaches for solving resource capacity management problems in stochastic networks from the two foregoing application domains fall into two main categories. These two sets of solution approaches also represent the state-of-the-art for a wide variety of application domains beyond our motivating applications. The first category of solution approaches is based on fairly direct applications of product-form network results, despite the fact that the underlying stochastic network does not have a product-form solution. Indeed, it is only under strong restrictions that the stationary joint distribution for the stochastic network is a product of the stationary distribution for each queue in isolation (see, e.g., Baskett et al. (1975), Harrison and Williams (1992)). This approach is often employed in computer capacity planning applications, even though the requirements for product-form solutions almost always never hold in practical capacity planning instances across a broad spectrum of computer environments. Instead, the approach is used as a simple approximation typically together with a wide range of ad hoc heuristics, which include applying product-form results for performance metrics of interest in a modified version of the original stochastic network in an attempt to address characteristics that yield a network with a non-product-form solution; e.g., refer to Menasce and Almeida (1998, 2000), Menasce et al. (2004) and the references therein. One example of this approach consists of increasing the service requirements of customers at a bottleneck resource in an attempt to have the results reflect the types of bursty arrival processes often found in computer environments. Another example consists of (artificially)

splitting the customer service requirements at a network station into multiple classes in an attempt to capture heavy tails in the service time distributions (ignoring correlation effects). Although the closed-form expressions render a direct solution for the corresponding optimization problem in a very efficient manner, the serious accuracy problems inherent in this simple approximation approach have been well established and thus are a great concern from both a theoretical and practical perspective.

The second category of state-of-the-art solution approaches is based on simulation-based optimization. Here, the literature can be broadly divided into those that use a broad spectrum of metaheuristics (e.g., tabu search, scatter search, neural networks) to control a sequence of simulation runs in order to find an optimal solution (see, e.g., Glover et al. (1999) and Chapter 20 in Nelson and Henderson (2007)), and those that apply several direct methods (e.g., stochastic approximation) which have been widely studied to address simulation-based optimization problems with a more rigorous theoretical foundation (refer to, e.g., Chapter 8 in Asmussen and Glynn (2007) and Chapter 19 in Nelson and Henderson (2007)). A great disadvantage of all these simulation-based optimization methods, however, is the often prohibitive costs in both time and resources required to obtain optimal solutions in practice for problems involving multidimensional stochastic networks. This is in large part due to the numerous parameters involved in each method that must be set via experimental tweaking for every problem instance. In fact, a recent study illustrates how simulation-based optimization can require on the order of a few days to determine optimal resource capacity levels in a much simpler class of stochastic processes than those considered in the present paper; see Heching and Squillante (2013).

The metaheuristic approach is often employed in business process management applications, where there is essentially exclusive use of simulation for the analysis and optimization of stochastic network representations of business processes; refer to, e.g., Laguna and Marklund (2004). More generally, the metaheuristic approach is employed in nearly all major simulation software products that support optimization, such as those offered by Arena and AnyLogic through partnership with companies like OptTek that provide a software control procedure which integrates various metaheuristic methods. At each step of the control procedure, there is a comparison between the simulation results for the current set of decision variables (server capacity in our application) and those for all previous settings, and then the procedure suggests another set of decision variable values for the next simulation run. Once no further improvement to the results is observed, the control procedure terminates and the best set of decision variable values found through this sequence of simulation runs are selected to be an (local) optimum. The metaheuristic approach ignores any structure of the underlying stochastic network, and therefore can be prone to accuracy concerns with respect to the true optimal solution from both a theoretical and practical perspective.

The stochastic approximation algorithm for simulation-based optimization has been extensively studied in great generality with rigorous results available on the rates of convergence under reasonable conditions for the objective function. These methods are regrettably not as common in practice as the metaheuristic approach. One stumbling block has been that the method requires the setting of certain critical parameters to "good" values in order to realize an efficient implementation, where practitioner experience demonstrates that "good" values typically depend on each instance of the problem being solved. As our results further demonstrate and quantify, this important requirement of stochastic approximation to find "good" values for certain critical parameters remains a key problem in practice for the general stochastic networks of interest in this study.

In this paper, our goal is to develop a general solution framework that provides the benefits of each of the above solution approaches while also addressing their serious limitations. Namely, we seek to realize the efficiency of analytical methods together with the accuracy of simulation-based methods within a unified framework for solving resource capacity management problems. We devise a two-phase solution framework in which a new and general form of stochastic decomposition is

derived and leveraged as part of a fixed-point iteration in the first phase to obtain a nearly optimal solution in a very efficient manner. The second phase, taking the first-phase solution as a starting point, then exploits advanced methods that deal directly with the original network to obtain an optimal solution. A good candidate for our second-phase solution is the stochastic approximation algorithm, given that it represents the only simulation-based optimization approach with a rigorous theoretical foundation. It is important to note, however, that any direct method can be used as the basis of the second phase of our framework. This second phase is much more accurate than the first phase, at the expense of much higher computational and temporal costs, but the nearly optimal starting point from the first phase allows us to leverage these detailed methods in a more surgical manner. By establishing and exploiting fundamental properties of the underlying multidimensional process of the stochastic network throughout this framework, we believe that significant improvements in computational costs and solution accuracy are possible over various state-of-the-art approaches for solving resource capacity management problems in general.

Our study includes a wide variety of numerical experiments to investigate the performance of a particular realization of our general solution framework over a broad range of problem settings. The results of these experiments clearly and convincingly demonstrate the significant benefits of our general solution framework over existing state-of-the-art approaches, namely product-form and stochastic approximation solutions. In particular, we show that the two-phase framework provides vastly superior results than product-form solutions and converges much faster than stochastic approximation approaches with respect to finding (locally) optimal solutions. Moreover, the principle new stochastic decomposition algorithm introduced as an accelerant in the first phase performs very well in producing good approximate solutions that are typically within a tiny percentage of optimality for well-behaved problem settings and within 5% of optimality in more diverse settings. This is of great advantage to the general user in practice because the algorithm has no parameters that must be tweaked to obtain such fast convergence to good approximate solutions. These high quality approximations support efficient exploration of the entire resource capacity management space across various assumptions, conditions, scenarios and workloads. Furthermore, these high quality approximations as starting points obviously benefit the method used in the second phase. In contrast, our numerical experiments clearly illustrate the difficulty encountered by users in tuning parameters to realize the best results from the sole use of the stochastic approximation algorithm.

The remainder of this paper is organized as follows. Section 2 presents our general solution framework. We then consider in Section 3 a specific instance of our general problem, namely a single-class Brownian tree network, for which we derive structural properties that play a fundamental role in an analysis of our algorithm presented in Section 4, including results on uniqueness, convergence and bounded optimality gap in the tree-network setting. A representative set of numerical experiments are provided in Section 5, which includes a brief introduction to the stochastic approximation algorithm. Concluding remarks then follow, with the appendices presenting additional technical details and some of our proofs.

Notation. All vectors in this paper are L-dimensional, where L represents the number of stations in the network. Throughout, we use boldface for vectors and identify their elements through subscripts: the i-th element of the vector $\boldsymbol{v}$ is given by v_i. We similarly use boldface for vector-valued functions regardless their domain, and write for instance $f_i(\boldsymbol{x})$ for the i-th element of $\boldsymbol{f}(\boldsymbol{x})$. We also write $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ for the inner product of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.

2. A General Approach

The complexity of solving resource capacity management problems is due in great part to the technical difficulties of solving for functionals of stationary stochastic networks. A major source of

difficulty in such analysis, modeling and optimization of stochastic networks concerns the multidimensional aspects of the underlying stochastic processes, which involve various dependencies and dynamic interactions among the different dimensions of the multidimensional process.

In this section, we present our general approach to address these complexities and difficulties in developing a novel resource capacity management solution framework. We first discuss the basic idea to provide a proper context for our approach, and then we formally present our general solution framework. Due to the highly nonlinear and possibly nonconvex nature of resource capacity management problems in general, our focus lies on finding "good" local optima.

2.1. Basic idea

Given our goal of developing a solution approach that provides the efficiencies of analytical methods together with the accuracies of simulation-based methods, we devise a general two-phase solution framework for optimal resource capacity management problems in stochastic networks.

The first phase is based on a fixed-point iteration approach where observed queue lengths at the current iterate determine the resource capacity allocation for the next iterate. If one of the queue lengths is disproportionally high in the current iterate, the next iterate will allocate more resource capacity at the corresponding station. This process repeats, forming the basis of an efficient fixed-point iteration that renders a nearly (locally) optimal solution to the resource capacity management problem. Depending on the stochastic network setting in which our general solution framework is applied, the required queue length information can be obtained from (a combination of) advanced analytical (e.g., Baskett et al. (1975), Dieker and Gao (2011), Harrison and Williams (1992), Pollett (2009)), numerical (e.g., Dai and Harrison (1992), Saure et al. (2009)) or simulation-based (e.g., Asmussen and Glynn (2007), Nelson and Henderson (2007)) methods. As a result, our first-phase approach can be applied to stochastic networks that are analytically intractable as long as they can be simulated.

Our iterative algorithm updates resource allocations based on the square root of the observed queue lengths, as motivated and formalized mathematically in the next subsection. Roughly speaking, our updating rule is derived from an appropriate separable functional form for performance metrics of each station in the network, such as expected steady-state queue length or expected steady-state sojourn time at the queue. The functional form is given by $\tau/(\beta - \lambda)$, where λ and β are the arrival and service rates for the queue and τ is a function of various characteristics of the arrival and service processes at all stations in the network. This particular functional form naturally arises in all known queueing formulas; refer to, e.g., Kingman (1962) for an enlightening example of how it appears and see the next subsection for a more detailed discussion.

This first phase of our methodology is similar in spirit to a traditional approach from the field of applied probability, which approximates the complex multidimensional stochastic process through a set of simpler processes of reduced dimensionality together with fixed-point equations that capture the dependencies and dynamic interactions among the dimensions; see, e.g., Squillante (2011). A classical example of this basic approach is the well-known Erlang fixed-point approximation in which the multidimensional Erlang formula is replaced by a system of nonlinear equations in terms of the one-dimensional Erlang formula; refer to, e.g., Kelly (1991). Our approach similarly uses a functional form for the one-dimensional queueing processes as the basis of a multidimensional approximation, but our fixed-point iteration critically relies on the multidimensional queueing dynamics as captured through the means of, for instance, numerical or simulation methods. As a result, since we do not rely on explicit queueing formulas that hold under restrictive assumptions, our approach requires effectively no underlying assumptions and promises to be widely applicable.

The resource allocation decisions from the first phase subsequently serve as a starting point for the second phase. This second phase is based on general search methods that deal directly with the original stochastic network to further improve upon the first-phase starting point and

obtain a locally optimal solution. When the original stochastic network has a product-form solution (see, e.g., Baskett et al. (1975), Harrison and Williams (1992)), then the results of our first-phase algorithm lead to the desired optimal resource capacities after exactly one iteration and the second phase of our general solution framework is not needed.

2.2. Mathematical formalization

We now formalize our approach in a setting where the goal is to minimize the sum of the weighted expected steady-state queue lengths in a stochastic network subject to a budgetary constraint. We gear the discussion towards application of our approach to generalized Jackson networks (refer to, e.g., Chen and Yao (2001)) and their Brownian counterparts (see, e.g., Harrison and Williams (1987)); other settings are discussed in the next subsection.

Some additional notation is needed. We write $\boldsymbol{\gamma}$ for the effective arrival rate vector and $\boldsymbol{\beta}$ for the vector of service rates. Further parameters of the network, such as the routing matrix and the exact external interarrival and service distributions, need not be specified to present our approach and thus we do not introduce them here. We write $Z_i^{\boldsymbol{\beta}}$ for the steady-state queue length at the i-th station (alternatively one can similarly study sojourn times). The dependence on $\boldsymbol{\beta}$ is made explicit since we are interested in comparing a functional of the steady-state vector $\boldsymbol{Z}^{\boldsymbol{\beta}}$ as we change the service-rate vector $\boldsymbol{\beta}$. Assume that each unit of resource capacity at station i costs c_i, comprising a cost vector $\boldsymbol{c}$, and that we have a total budget of C for allocating resources in the network.

We aim to minimize the expected steady-state queue lengths weighted by a vector $\boldsymbol{w}$, subject to the constraints that we cannot spend more than the budget C and that the queueing system is stable. This leads to the following optimization problem:

$$\text{(OPT)} \qquad \begin{aligned} \min_{\boldsymbol{\beta}\in(0,\infty)^L} \quad & \sum_{i=1}^{L} w_i \mathbb{E} Z_i^{\boldsymbol{\beta}} \\ \text{s.t.} \quad & \langle \mathbf{c}, \boldsymbol{\beta} \rangle \le C, \\ & \beta_i > \gamma_i,\ i = 1, \ldots, L. \end{aligned}$$

Throughout, we shall assume $\langle \mathbf{c}, \boldsymbol{\gamma} \rangle < C$ so that the above mathematical program is feasible. One can expect the solution to satisfy $\langle \mathbf{c}, \boldsymbol{\beta} \rangle = C$; see Section 4.2 for a result in this direction.

After defining

$$\tau_i(\boldsymbol{\beta}) = (\beta_i - \gamma_i) \mathbb{E} Z_i^{\boldsymbol{\beta}}, \tag{1}$$

the objective function takes the form $t(\boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{\tau}(\boldsymbol{\beta}))$, where for $\boldsymbol{\beta} - \boldsymbol{\gamma}, \boldsymbol{w}, \boldsymbol{\tau} > 0$ we have

$$t(\boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{\tau}) = \sum_{k=1}^{L} w_k \frac{\tau_k}{\beta_k - \gamma_k}. \tag{2}$$

For a queue in a single-class product-form network, $\boldsymbol{\tau}$ is known to be equal to λ; see for instance Baskett et al. (1975). Furthermore, τ equals $\lambda(c_A^2 + c_S^2)/2$ in a single-class Brownian product-form network of GI/GI/1 queues, where c_A^2 and c_S^2 denote the second-order variation terms for the arrival and service process, respectively; see Harrison and Williams (1992). In general stochastic networks, however, $\boldsymbol{\tau}(\boldsymbol{\beta})$ is mathematically intractable.

Our approach relies on the idea that $\boldsymbol{\beta} \mapsto t(\boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{\tau}(\boldsymbol{\beta}))$ can be reasonably approximated locally by $\boldsymbol{\beta} \mapsto t(\boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{\tau}(\bar{\boldsymbol{\beta}}))$ in the neighborhood of a given point $\bar{\boldsymbol{\beta}}$. Through this functional form, the i-th summand in the approximating objective function only depends on $\boldsymbol{\beta}$ through the one-dimensional quantity β_i, thus effectively "decomposing" the objective function. The explicit incorporation of $\beta_i - \gamma_i$ in the denominator is motivated by the aforementioned natural occurrences of this functional form, which includes product-form results that effectively arise from one-dimensional queueing

formulas. We note that the idea of locally approximating the objective function is a well-known principle in trust-region based optimization; refer to, e.g., Conn et al. (2000). Our approach, however, is very different from traditional trust-region methods in that we exploit structural properties of the stochastic network through a global functional-form decomposition whose (few) unknown parameters are estimated locally, whereas trust-region methods take a black-box approach in which the parameters of an arbitrary polynomial approximation of the objective function are fitted locally.

The following lemma, which is readily proved by applying standard Lagrangian methods, then becomes an essential ingredient in our analysis. Kleinrock (1964) and Wein (1989) used this result in their work on capacity allocation for product-form networks.

LEMMA 1. *The minimum of $t(\boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{\tau})$ over the feasible region in (OPT) is $\boldsymbol{\beta}^*(\boldsymbol{w}, \boldsymbol{\tau})$, where for $\ell = 1, \ldots, L$*

$$\beta_\ell^*(\boldsymbol{w}, \boldsymbol{\tau}) = \gamma_\ell + (C - \langle \boldsymbol{c}, \boldsymbol{\gamma} \rangle) \frac{\sqrt{w_\ell \tau_\ell / c_\ell}}{\sum_{k=1}^L \sqrt{w_k \tau_k c_k}}.$$

As an extension of the idea that queue lengths may be approximated locally by functions of the form (2), we propose to use the capacity allocation $\boldsymbol{\beta}^*$ determined through the following system of nonlinear equations as the outcome of the first phase of our approach: For $\ell = 1, \ldots, L$,

$$\beta_\ell^* = \gamma_\ell + (C - \langle \boldsymbol{c}, \boldsymbol{\gamma} \rangle) \frac{\sqrt{w_\ell \tau_\ell(\boldsymbol{\beta}^*) / c_\ell}}{\sum_{i=1}^L \sqrt{w_i \tau_i(\boldsymbol{\beta}_i^*) c_i}}. \tag{3}$$

Section 4 shows that this system of equations is guaranteed to have a unique solution for a certain class of stochastic networks, but we leave open the question of existence and uniqueness for other settings.

In an attempt to numerically find a vector $\boldsymbol{\beta}^*$ satisfying (3), assuming existence, we propose the fixed-point iteration scheme with iterates $\{\boldsymbol{\beta}^{(k)} : k \geq 0\}$ given by

$$\beta_\ell^{(k+1)} = \gamma_\ell + (C - \langle \boldsymbol{c}, \boldsymbol{\gamma} \rangle) \frac{\sqrt{w_\ell \tau_\ell(\boldsymbol{\beta}^{(k)}) / c_\ell}}{\sum_{i=1}^L \sqrt{w_i \tau_i(\boldsymbol{\beta}^{(k)}) c_i}}. \tag{4}$$

This can be rewritten in the following insightful way. In view of (1), we find that (4) implies

$$\frac{\beta_i^{(k+1)} - \gamma_i}{\beta_j^{(k+1)} - \gamma_j} = \sqrt{\frac{\beta_i^{(k)} - \gamma_i}{\beta_j^{(k)} - \gamma_j} \times \frac{w_i \mathbb{E} Z_i^{\boldsymbol{\beta}^{(k)}} / c_i}{w_j \mathbb{E} Z_j^{\boldsymbol{\beta}^{(k)}} / c_j}}, \tag{5}$$

which lies at the heart of the first phase of our approach because this equation establishes an important connection with a resource capacity iteration scheme based on observed queue length information. Since we must allocate at least capacity γ_i to station i, $(\beta_i - \gamma_i)/(\beta_j - \gamma_j)$ is the ratio of "additional" resource capacities allocated to station i and j, respectively. Equation (5) expresses this ratio in terms of the ratio of mean queue lengths, so that more capacity is allocated in the next iterate to stations with disproportionally long queue lengths in the current iterate.

The right-hand side of (5) can be interpreted as the geometric mean of two fractions, and thus we can think of (5) as a "slowed down" version of the iterative scheme

$$\frac{\tilde{\beta}_i^{(k+1)} - \gamma_i}{\tilde{\beta}_j^{(k+1)} - \gamma_j} = \frac{w_i \mathbb{E} Z_i^{\tilde{\boldsymbol{\beta}}^{(k)}} / c_i}{w_j \mathbb{E} Z_j^{\tilde{\boldsymbol{\beta}}^{(k)}} / c_j}. \tag{6}$$

From (5) it becomes evident that we hope for our iterative scheme to converge to $\boldsymbol{\beta}^*$ satisfying

$$\frac{\beta_i^* - \gamma_i}{\beta_j^* - \gamma_j} = \frac{w_i \mathbb{E} Z_i^{\boldsymbol{\beta}^*} / c_i}{w_j \mathbb{E} Z_j^{\boldsymbol{\beta}^*} / c_j}.$$

We found numerical examples with unique fixed points for which the iteration process of (5) produces a converging sequence whereas the iteration process of (6) produces an oscillating sequence. Hence, slowing down the iterative scheme can improve its convergence properties. This is in fact a well-studied phenomenon in the literature on fixed-point iteration processes, where taking an (arithmetic) average often produces better results; see, e.g., Mann (1953), Ishikawa (1974) and the vast body of subsequent work in this area. The geometric average in our case arises from the assumed functional form given in (2). As quantified in Lemma 1, a further consequence of this functional form is that the iterates in (5) avoid the boundary of the feasible region, unlike the iterates of (6).

2.3. Discussion

Our two-phase framework only relies on the capability of evaluating the queue lengths at all stations comprising the stochastic network under any resource capacity allocation, and thus it holds great promise to perform well in many stochastic network optimization problems beyond the setting of generalized Jackson networks and their Brownian analogs. The key approximation in our framework consists of the separable functional form $\tau/(\beta-\lambda)$, which constitutes a near universal phenomenon in stochastic networks under a wide range of queueing dynamics (e.g., processor sharing networks and multi-class networks under a variety of scheduling policies). As a result, we expect that resource capacity management optimization through an iterative algorithm based on ratios of observed queue lengths and slow-down through geometric means is promising for many different settings, such as several variants of the setting discussed in the previous subsection. For instance, one could have a discrete decision space in which to allocate a number of servers to each station, and then use Lemma 1 in conjunction with a local search algorithm over the discrete parameter space to generate an iterative scheme. In other examples, the feasible region may be less explicit and only described through a stability condition (since it may not always be possible to determine this region, one could equip the queue-length evaluation procedure with an explosion check), in which case the iterative scheme could require new insights to perform an optimization of the approximate objective function over this space. Another interesting variant is the dual formulation of the problem, as discussed in the introduction, where the aim is to minimize the total expenditure subject to a bound on the sum of the weighted expected steady-state queue lengths.

3. Single-class Brownian Tree Networks

We next introduce single-class Brownian tree networks, presenting several of their structural properties that play a key role in the analysis of our algorithm. The premise of Brownian network models is that they approximate the queue-length (or waiting-time) dynamics of stochastic networks, relying on a central limit theorem scaling. Such an approximation is often rigorously justified in heavy traffic; refer to, e.g., Reiman (1984), Harrison and Williams (1987). The heavy-traffic assumption is particularly relevant in the context of resource capacity management problems, where it implies the often realistic assumption that systems are operated close to the system capacity. As in the central limit theorem, interarrival and service distributions are approximated in the Brownian model using only their first two moments. Hence, a Brownian network model can be thought of as a two-moment approximation of the underlying stochastic network. This idea lies at the heart of the so-called QNET method proposed in Harrison and Nguyen (1990). Even though the queue-length process of a generalized Jackson network is typically non-Markovian, the queue-length process of a Brownian network forms a Markov process known as reflected (or regulated) Brownian motion. Further background on Brownian tree networks is provided in Appendix A, together with proofs for the two lemmas presented in this section.

Our specific focus in this section lies on single-class Brownian tree networks, which arise from an underlying generalized Jackson network (see, e.g., Chen and Yao (2001)) with a tree network

topology. Within this class of models, we are able to derive several qualitative properties of our framework. The network topology is represented by a rooted tree $G=(V,E)$ comprised of $L=|V|$ vertices, at which customers are served by a (Brownian) server. We identify the root as station 1. For $i>1$, we write $\pi(i)$ to denote the label of the unique station adjacent to i that is closer to the root. Customers are served at station i at rate β_i, meaning that the mean service time there equals $1/\beta_i$; we say β_i is the *service capacity* at station i. After having received their required service, customers are routed from station i to station j with probability p_{ij}. The network topology imposes the restriction that $(i,j)\in E$ if and only if $p_{ij}>0$. We refer to Appendix A for second order (variance) parameters of this model, which are not used below.

A path is defined to be a sequence of vertices such that from each of its vertices there is an edge in E to the next vertex in the sequence. We write $\mathcal{P}_i=(1,\ldots,\pi(i),i)$ for the unique path from the root to station i. Given a vector $\boldsymbol{v}\in\mathbb{R}^L$, we write $\boldsymbol{v}_i$ for the vector obtained by restricting $\boldsymbol{v}$ to those elements in the path $\mathcal{P}_i$. For instance, $\boldsymbol{\beta}_i$ stands for $(\beta_1,\ldots,\beta_{\pi(i)},\beta_i)$. Due to the specific tree structure studied here, the queue length at station i, $Z_i^{\boldsymbol{\beta}}$, only depends on $\boldsymbol{\beta}$ through the upstream capacity vector $\boldsymbol{\beta}_i$. We abuse notation slightly and write $Z_i^{\boldsymbol{\beta}_i}$ instead of $Z_i^{\boldsymbol{\beta}}$, similarly using $Z_j^{\boldsymbol{\beta}_i}$ for $j\in\mathcal{P}_i$.

For notational convenience, we set $p_{\pi(1),1}=0$. For any $j\in\mathcal{P}_i$, define

$$q_{ji}=\prod_{k\in\mathcal{P}_i,k\notin\mathcal{P}_j} p_{\pi(k),k},$$

where an empty product should be interpreted as 1, so that $q_{ii}=1$. We write $\gamma_i=\sum_{k\in\mathcal{P}_i} q_{ki}\lambda_k$ for the effective arrival rate at station i, where λ_i denotes the external arrival rate at station i. Since our interest in this paper is solely on stable networks, we impose throughout that $\beta_i>\gamma_i$ for $i=1,\ldots,L$.

The following functions play a key role in the analysis of our algorithm for Brownian tree networks. We define $h_i:(0,\infty)^{|\mathcal{P}_i|}$, for $i=1,\ldots,L$, through

$$h_i(\boldsymbol{x}_i)=\sum_{j\in\mathcal{P}_i} q_{ji}\mathbb{E}Z_j^{\boldsymbol{\gamma}_i+\boldsymbol{x}_i},$$

with the convention $h_{\pi(1)}=0$. This definition implies

$$\mathbb{E}Z_i^{\boldsymbol{\beta}}=h_i(\boldsymbol{\beta}_i-\boldsymbol{\gamma}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{\beta}_{\pi(i)}-\boldsymbol{\gamma}_{\pi(i)}), \tag{7}$$

and thus the original optimization problem (OPT) is readily reformulated in terms of these h_i functions for tree networks as follows.

$$\begin{aligned}(\mathrm{OPT-BTN}) \qquad \min_{\boldsymbol{\beta}\in(0,\infty)^L} \quad & \sum_{i=1}^{L} w_i\left[h_i(\boldsymbol{\beta}_i-\boldsymbol{\gamma}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{\beta}_{\pi(i)}-\boldsymbol{\gamma}_{\pi(i)})\right]\\ \text{s.t.} \quad & \langle \mathbf{c},\boldsymbol{\beta}\rangle\le C,\\ & \beta_i>\gamma_i,\ i=1,\ldots,L.\end{aligned}$$

This formulation is advantageous since the functions h_i enjoy several useful structural properties, which are described in the next two lemmas.

LEMMA 2. *For any $i=1,\ldots,L$, the function $h_i:(0,\infty)^{|\mathcal{P}_i|}\to\mathbb{R}_+$ is:*

(i) convex on its domain;

(ii) nonincreasing in each coordinate; and

(iii) strictly decreasing in the last coordinate x_i unless the degeneracy condition of deterministic service times at stations $\pi(i)$ and i, deterministic routing to station i, and no external arrivals at station i holds.

The monotonicity result in (ii) implies that the mean queue length at the i-th station $\mathbb{E}Z_i^{\boldsymbol{\beta}}$ decreases in the service capacity β_i. A more precise statement of the degeneracy condition in (iii) is given by Σ_{ii}, defined in (18) of Appendix A, being equal to 0. The lemma condition of $\Sigma_{ii} > 0$ appearing in (iii) prevents the capacity allocation problem at the i-th station from being degenerate, i.e., additional capacity does not lead to lower (Brownian) queue lengths due to the deterministic setting stated in the lemma.

In view of Lemma 2, a wide variety of generic techniques are available to study (OPT-BTN). The convexity property implies that (OPT-BTN) is a so-called difference of convex functions (DC) programming problem, as studied in for instance An and Tao (2005). It also shows that (OPT-BTN) becomes a convex program under certain settings of the weights. Notice that the convex function h_i has coefficient $(w_i - \sum_{j=1}^{L} w_j p_{ij})$ in the objective function of (OPT-BTN). Thus, if the weights are constant $(w_1 = \ldots = w_L)$ or more generally the weights are non-increasing $(w_1 \geq \ldots \geq w_L)$, then the problem (OPT-BTN) is convex.

Another important property is that h_i is homogeneous of degree -1, which is a consequence of the Brownian scaling property and precisely stated in the following lemma.

LEMMA 3. *For any $i = 1, \ldots, L$, $\boldsymbol{x}_i > 0$, and $\delta > 0$, we have $\delta h_i(\delta \boldsymbol{x}_i) = h_i(\boldsymbol{x}_i)$.*

4. Analysis of our Approximation Algorithm for Brownian Tree Networks

This section analyzes the algorithm of Section 2 in the context of the Brownian tree networks of Section 3. In particular, we prove that there is a unique fixed point in this case, and that (a minor modification of) our algorithm converges to this fixed point. We also establish that the optimality gap remains bounded in the budget C, further proving a desirable property of the step sizes taken in our iterative capacity allocation procedure.

4.1. Existence and Uniqueness of a Fixed Point

Our approximation to the optimal capacity allocation is defined as a solution of the fixed-point equations (3). In this section, we establish the existence and uniqueness of such a fixed point, which is therefore a feasible solution for (OPT).

Standard techniques for proving uniqueness of a fixed point typically involve bounds on (first-order) derivatives, or on a spectral radius in the present multivariate setting. Since we were unable to derive such results for the τ_i functions, we take a different approach. Our approach is to rewrite the fixed-point equations in an appropriate form, and then use the structural properties of the h_i functions from the preceding section to show the existence and uniqueness of the fixed point.

Before proceeding, we need some additional notation that is used throughout this section. Abusing notation as before, define

$$x_i(\boldsymbol{\beta}) = x_i(\boldsymbol{\beta}_i) = (\beta_i - \gamma_i)/(\beta_1 - \gamma_1) \tag{8}$$

and $\boldsymbol{x}(\boldsymbol{\beta}) = (x_1(\boldsymbol{\beta}), \ldots, x_L(\boldsymbol{\beta}))$, so that, in particular, $x_1(\boldsymbol{\beta}) = 1$ for all $\boldsymbol{\beta}$. We write $\boldsymbol{x}_i(\boldsymbol{\beta}_i)$ for $(x_1(\boldsymbol{\beta}_1), \ldots, x_{\pi(i)}(\boldsymbol{\beta}_{\pi(i)}), x_i(\boldsymbol{\beta}_i))$. Note the difference between the vector $\boldsymbol{x}_i(\boldsymbol{\beta}_i)$ and its i-th element $x_i(\boldsymbol{\beta}_i)$. Lastly, we introduce the set S_{L-1}, via the following lemma, for our main result below.

LEMMA 4. *Writing $S_{L-1} = \{\boldsymbol{\beta} \in (\gamma_1, \infty) \times \cdots \times (\gamma_L, \infty) : \langle \boldsymbol{c}, \boldsymbol{\beta} \rangle = C\}$, the mapping $\boldsymbol{\beta} \in S_{L-1} \mapsto \boldsymbol{x}(\boldsymbol{\beta}) \in \{1\} \times \mathbb{R}_+^{L-1}$ is one-to-one.*

Proof. It suffices to show that there is a unique $\boldsymbol{\beta} \in S_{L-1}$ corresponding to a given vector $\boldsymbol{x} \in \mathbb{R}_+^L$ with $x_1 = 1$. By the definition of x_i, we have $\beta_i = \gamma_i + x_i(\beta_1 - \gamma_1)$ and therefore $\langle \boldsymbol{c}, \boldsymbol{\beta} \rangle = \langle \boldsymbol{c}, \boldsymbol{\gamma} \rangle + (\beta_1 - \gamma_1)\langle \boldsymbol{c}, \boldsymbol{x} \rangle$. Setting the right-hand side equal to C yields β_1, which in turn fixes $\beta_2, \ldots, \beta_L$ through $x_i = (\beta_i - \gamma_i)/(\beta_1 - \gamma_1)$ for $i = 2, \ldots, L$. $\square$

The first step is to observe that the fixed-point system (3) in terms of $\boldsymbol{\beta}$ can be written as a system of equations in terms of $\boldsymbol{x}_i(\boldsymbol{\beta}_i)$. For $i=1,\ldots,L$, we obtain from (1), (7) and the homogeneity of h (Lemma 3) that

$$\begin{aligned}\tau_i(\boldsymbol{\beta}) &= (\beta_i-\gamma_i)\mathbb{E}Z_i^{\boldsymbol{\beta}}\\ &= x_i(\boldsymbol{\beta}_i)(\beta_1-\gamma_1)[h_i(\boldsymbol{\beta}_i-\boldsymbol{\gamma}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{\beta}_{\pi(i)}-\boldsymbol{\gamma}_{\pi(i)})]\\ &= x_i(\boldsymbol{\beta}_i)[h_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i))-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)}(\boldsymbol{\beta}_{\pi(i)}))]. \qquad (9)\end{aligned}$$

In particular, $\tau_i(\boldsymbol{\beta})$ is a function solely of $\boldsymbol{x}_i(\boldsymbol{\beta}_i)$ (assuming all network parameters are fixed except for the service capacities). Namely, the tree structure yields that $\tau_i(\boldsymbol{\beta})$ only depends on $\boldsymbol{\beta}_i$, and the Brownian nature of the network further reduces the dependency to only $\boldsymbol{x}_i(\boldsymbol{\beta}_i)$. Hence, abusing notation, we can write

$$\tau_i(\boldsymbol{x}_i)=x_i[h_i(\boldsymbol{x}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})]. \qquad (10)$$

We next rewrite the fixed-point system in terms of $\boldsymbol{x}_i$. If $\boldsymbol{\beta}^*$ solves the system of fixed-point equations (3), we readily obtain

$$\frac{C-\langle \boldsymbol{c},\boldsymbol{\gamma}\rangle}{c_i(\beta_i^*-\gamma_i)}\sqrt{w_i\tau_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i^*))c_i}=\sum_{k=1}^{L}\sqrt{w_k\tau_k(\boldsymbol{x}_k(\boldsymbol{\beta}_k^*))c_k},$$

implying that $\sqrt{w_i\tau_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i^*))c_i}/(c_i(\beta_i^*-\gamma_i))$ does not depend on i. Thus, we have

$$\frac{\sqrt{w_i\tau_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i^*))c_i}}{c_i(\beta_i^*-\gamma_i)}=\frac{\sqrt{w_1\tau_1c_1}}{c_1(\beta_1^*-\gamma_1)},$$

which is equivalent to

$$\frac{\tau_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i^*))}{x_i(\boldsymbol{\beta}_i^*)}=\tau_1\frac{c_i/w_i}{c_1/w_1}x_i(\boldsymbol{\beta}_i^*).$$

The next step is to introduce the h_i functions by noting that (9) implies

$$h_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i^*))-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)}(\boldsymbol{\beta}_{\pi(i)}^*))=\frac{\tau_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i^*))}{x_i(\boldsymbol{\beta}_i^*)}.$$

We then conclude that $\boldsymbol{\beta}^*$ satisfies (3) if and only if, for $i=1,\ldots,L$,

$$h_i(\boldsymbol{x}_i(\boldsymbol{\beta}_i^*))-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)}(\boldsymbol{\beta}_{\pi(i)}^*))=\tau_1\frac{c_i/w_i}{c_1/w_1}x_i(\boldsymbol{\beta}_i^*). \qquad (11)$$

We now prove that there is exactly one solution $\boldsymbol{x}^*$ to (11), which immediately shows that there is exactly one solution $\boldsymbol{\beta}^*$ of (3) on the boundary of the feasible set.

THEOREM 1. *The system of fixed-point equations (3) has exactly one solution on the set S_{L-1}.*

Proof. We first show that there is exactly one solution to the fixed-point equations in $(x_2,\ldots,x_L)\in(0,\infty)^{L-1}$, i.e., to the equations

$$h_i(\boldsymbol{x}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})=\tau_1\frac{c_i/w_i}{c_1/w_1}x_i \qquad (12)$$

for $i=1,\ldots,L$. To see this, first consider all equations corresponding to stations at a distance of 1 from the root. These equations are of the form $h_i(1,x_i)=p_{1i}\tau_1+\tau_1\frac{c_i/w_i}{c_1/w_1}x_i$. Since the left-hand side is nonincreasing from $+\infty$ in $x_i>0$ and the right-hand side is strictly increasing in x_i, there is exactly one x_i-value for which equality holds.

Next assume that the x_ℓ corresponding to stations at a distance of $n-1$ from the root have been determined from (12), and consider a station i at a distance of n from the root. As before, the left-hand side of (12) is nonincreasing in x_i while the right-hand side is strictly increasing in x_i. Thus, there is a unique x_i for which equality holds. $\square$

4.2. Optimality Gap

We next study the relationship between the solution to (OPT-BTN) and the proposed fixed-point approximation. Using the homogeneity of h_i from Lemma 3 and the definition of x_i in (8), one readily finds that the objective function of (OPT-BTN) becomes

$$\min_{\beta_1>\gamma_1, x_2>0,\ldots,x_L>0} \quad \frac{1}{\beta_1-\gamma_1}\sum_{i=1}^{L} w_i[h_i(\boldsymbol{x}_i) - p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})]$$

subject to $(\beta_1-\gamma_1)[c_1+\sum_{\ell=2}^{L} c_\ell x_\ell] \le C - \langle \boldsymbol{c},\boldsymbol{\gamma}\rangle$. Since the objective function is decreasing in $\beta_1-\gamma_1$, the constraint will be binding in any optimal solution. After casting the resulting problem back into the form of (OPT-BTN), we immediately obtain the following lemma.

LEMMA 5. *Any solution $\boldsymbol{\beta}$ to (OPT-BTN) satisfies $\langle \boldsymbol{c},\boldsymbol{\beta}\rangle = C$.*

Upon substituting the constraint $(\beta_1-\gamma_1)[c_1+\sum_{\ell=2}^{L} c_\ell x_\ell] = C - \langle \boldsymbol{c},\boldsymbol{\gamma}\rangle$ into the objective function, we see that the solution of (OPT-BTN) is equivalent to

$$\min_{x_1=1,x_2,\ldots,x_L} \frac{\langle \boldsymbol{c},\boldsymbol{x}\rangle}{C-\langle \boldsymbol{c},\boldsymbol{\gamma}\rangle}\sum_{i=1}^{L} w_i[h_i(\boldsymbol{x}_i) - p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})]. \tag{13}$$

Define the *optimality gap* as the ratio of the objective function at the allocation given by the fixed-point approximation and by the solution to (OPT-BTN). Now we can present the main result of this subsection.

PROPOSITION 1. *Our fixed-point approximation has the following properties when applied to Brownian tree networks.*

(i) As $C\to\infty$, the optimality gap remains bounded. In fact, it has a limit in $[1,\infty)$.

(ii) The difference between the fixed-point approximation and the optimizing argument in (OPT-BTN) is $O(C)$ in each coordinate as $C\to\infty$.

Proof. For (i), we use the fact that the solution of (OPT-BTN) is of order $1/C$ as $C\to\infty$, as shown in (13). The fixed-point solution approximates (13) by evaluating the objective function at the fixed point $\boldsymbol{x}^*$ in $\boldsymbol{x}$-space. Thus, it is also of order $1/C$ by construction.

For (ii), we note that the one-to-one correspondence from Lemma 4 between points $\boldsymbol{x}$ with $x_1=1$ and points $\boldsymbol{\beta}\in S_{L-1}$ satisfies

$$\beta_i = \gamma_i + (C-\langle \boldsymbol{c},\boldsymbol{\gamma}\rangle)\frac{x_i}{\langle \boldsymbol{c},\boldsymbol{x}\rangle}.$$

Since neither a true optimal solution for (13) nor the fixed-point approximation depends on C in $\boldsymbol{x}$-space, the corresponding quantities in $\boldsymbol{\beta}$-space are of order C. □

The argument in the proof shows that part (ii) of this proposition can be further refined by saying that, unless the fixed-point approximation is exact (as in the product-form case), it is *exactly* order C away from the optimal capacity allocation.

4.3. Analysis of the Iterates

This subsection analyzes the iterates of our fixed-point algorithm within the context of Brownian tree networks. In particular, we show that the relative step size of our algorithm is larger when the current iterate is further away from the fixed point, which can be a key advantage of our fixed-point approximation since generic optimization algorithms do not possess this property (as it would require knowledge of the target point). Based on this result, we also discuss how our algorithm can be modified slightly to guarantee that it converges to the unique fixed point.

In order to exploit the structural properties of h_i, it is convenient to study the iterates $\{\boldsymbol{\beta}^{(k)}\}$ through $\{\boldsymbol{x}^{(k)} = \boldsymbol{x}(\boldsymbol{\beta}^{(k)})\}$ as defined in (8); the two sequences are in one-to-one correspondence in view of Lemma 4. The system of fixed-point equations (4) then becomes

$$x_i^{(k+1)} = \sqrt{\frac{w_i \tau_i(\boldsymbol{x}_i^{(k)})/c_i}{w_1 \tau_1 / c_1}}. \tag{14}$$

Namely, the fixed-point iterates in (4) can be equivalently described through (14), where the definition of $\tau_i(\boldsymbol{x}_i)$ is as given in (10). Note that $x_1^{(k)} = 1$ for any $k \geq 1$. We also write $\boldsymbol{x}^*$ for $\boldsymbol{x}(\boldsymbol{\beta}^*)$.

4.3.1. Relative Step Sizes. We now show that the iterates move in the direction of the fixed point, and that the magnitude of the step is larger when the iterate is further away from the fixed point. First, we consider the iterates corresponding to children of the root.

LEMMA 6. *Let i be a child of the root. If $x_i^{(k)} < x_i^*$, then we have*

$$\frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k)}} \geq \sqrt{\frac{x_i^*}{x_i^{(k)}}} - 1 > 0. \tag{15}$$

Similarly, if $x_i^{(k)} > x_i^*$, we have

$$\frac{x_i^{(k)} - x_i^{(k+1)}}{x_i^{(k)}} \geq 1 - \sqrt{\frac{x_i^*}{x_i^{(k)}}} > 0. \tag{16}$$

Proof. Write $\alpha_i = c_1 w_i / (c_i w_1 \tau_1)$. If $x_i^{(k)} < x_i^*$, we have $h_i(1, x_i^{(k)}) - p_{1i}\tau_1 \geq h_i(1, x_i^*) - p_{1i}\tau_1 = \alpha_i^{-1} x_i^*$ by Lemma 2. In view of (14) and (10), this leads to

$$x_i^{(k+1)} = \sqrt{\alpha_i x_i^{(k)} \left[h_i(1, x_i^{(k)}) - p_{1i}\tau_1 \right]} \geq \sqrt{x_i^* x_i^{(k)}} > x_i^{(k)}.$$

Similarly, $x_i^{(k)} > x_i^*$ implies $x_i^{(k+1)} \leq \sqrt{x_i^* x_i^{(k)}} < x_i^{(k)}$. □

An analogous result holds for an arbitrary station in the network, but the formulation is slightly more intricate. The formal statement of the result is given in the following proposition, which is illustrated in Figure 1, and whose proof is provided in Appendix B.

PROPOSITION 2. *Let $i > 1$ be fixed. Suppose that $\lim_{k\to\infty} \boldsymbol{x}_{\pi(i)}^{(k)} = \boldsymbol{x}_{\pi(i)}^*$. Then for any $\eta > 0$, there exists a $\delta = \delta(\eta) > 0$ such that (15) holds for $\boldsymbol{x}_i^{(k)} \in \mathcal{L}_i^\eta$ and (16) holds for $\boldsymbol{x}_i^{(k)} \in \mathcal{U}_i^\eta$, where*

$$\mathcal{L}_i^\eta = [x_1^* \pm \delta(\eta)] \times \cdots \times [x_{\pi(i)}^* \pm \delta(\eta)] \times (0, x_i^* - \eta),$$
$$\mathcal{U}_i^\eta = [x_1^* \pm \delta(\eta)] \times \cdots \times [x_{\pi(i)}^* \pm \delta(\eta)] \times (x_i^* + \eta, \infty).$$

The convergence assumption on $\boldsymbol{x}_{\pi(i)}^{(k)}$ prevents us from applying this lemma inductively, since the statement of the proposition does not exclude the possibility that the iterates overshoot the fixed point indefinitely.

4.3.2. Convergence of the Iterates. Since Lemma 6 and Proposition 2 do not exclude the possibility of oscillations occurring under our fixed-point iteration algorithm, the iterates may not converge. Even though we have not encountered any instance in which the iterates do not converge throughout our many numerical experiments, it is worthwhile to explore how the algorithm can be slightly modified to ensure that the iterates converge to the fixed point.

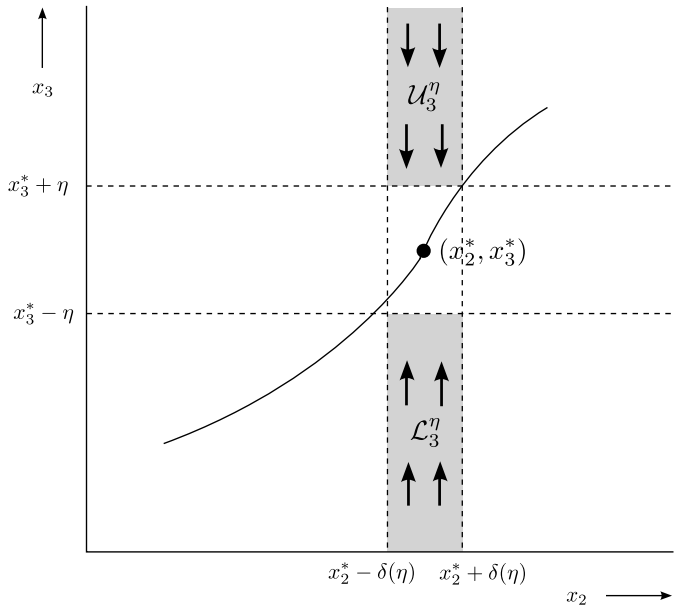


Figure 1 Illustration of Proposition 2 for three queues in series. The arrows indicate that iterates move in the specified direction.

In our modified algorithm, we additionally maintain an interval $I_i^{(k)}$ within which x_i^* as well as all modified iterates $x_i^{(k+1)}, x_i^{(k+2)}, \ldots$ will lie. The idea is easiest to explain when station i is a child of the root, in which case the argument relies on Lemma 6. If the sequence $\{x_i^{(k)}\}$ is a monotone sequence, then it must converge to x_i^* in view of Lemma 6 and no modifications are needed; let us therefore suppose that the sequence is not monotone. The left endpoint of $I_i^{(k)}$ is defined as the largest iterate among $x_i^{(0)}, \ldots, x_i^{(k)}$ that is smaller than x_i^*, and the right endpoint is similarly defined as the smallest iterate exceeding x_i^*. Then, if $x_i^{(k+1)}$ calculated from (14) lies outside of $I_i^{(k)}$, we overwrite $x_i^{(k+1)}$ with the center of the interval and continue the algorithm on the subinterval containing x_i^*. By adding this bisection step, the length of the interval $I_i^{(k)}$ shrinks as $k \to \infty$. To ensure that the interval length shrinks to zero, we add the additional requirement (again enforced by bisection) that either the left endpoint grows by a factor $1+\xi$ or the right endpoint shrinks by a factor $1-\xi$ in each iteration, where $\xi > 0$ is a parameter that is of smaller order than the desired level of accuracy for the fixed-point approximation. Similarly, if i is not a child of the root, then Proposition 2 shows that the modified iterates $\{x_i^{(k)}\}$ must converge.

5. Numerical Experiments

In this section we describe an extensive collection of numerical experiments performed to evaluate our general two-phase solution framework, introduced in Section 2, under a variety of stochastic network settings. A primary goal is to gain an understanding of how well the first phase of the framework, employing our fixed-point iteration algorithm, approximates a locally optimal solution to various instances of the stochastic optimization problem (OPT). This evaluation is based on comparing and contrasting against both the second phase of our framework and an approach based solely on the stochastic approximation (SA) algorithm, each of which identifies local optima. In the setting of convex optimization problems, we obviously have unique globally optimal solutions and, from our results in Section 4.1, the fixed point identified in the first phase of our approach is unique for convex instances of (OPT) in the Brownian tree-network setting. Hence, in convex settings, we are interested in the quality of the fixed-point approximation vis-à-vis the globally optimal solution to the problem (OPT) identified by the second phase of our approach. For settings when

the optimization problem is possibly non-convex and may therefore have multiple local optima, we also investigate whether the limit point(s) of our first-phase iterates changes under different starting points for the algorithm.

The key step in our fixed-point approximation method is the estimation of the expected queue lengths under the capacity values for the current iterate. A simulation-based implementation of queue-length estimation in the fixed-point iteration under the original stochastic network setting yields a consistent estimation. Our experimental results are therefore generated from such a simulation-based implementation. We note, however, that a numerical solution for the fixed-point queue-length estimation under a Brownian approximation of the original stochastic network can be used as the basis of the first-phase solution within our framework, providing results comparable to simulation in a much more efficient manner when the Brownian network is a reasonable approximation. The second phase of our framework then searches for a locally optimal solution close to the limit point, by starting the simulation-based optimization from the limit point identified in the first phase. We chose to use the SA algorithm because of its rigorous theoretical foundation, but note that any direct method can be used as the basis of the second-phase solution within our framework.

The remainder of this section is organized as follows. After providing a brief overview of the SA algorithm, Section 5.2 details the settings over which the numerical experiments were performed. A summary of the observations from these experiments is then discussed in Section 5.3.

5.1. Stochastic Approximation (SA)

Denote the objective function of (OPT) by $z(\boldsymbol{\beta}) \stackrel{\triangle}{=} \min_{\boldsymbol{\beta}\in(0,\infty)^L} \sum_{i=1}^{L} w_i \mathbb{E}Z_i^{\boldsymbol{\beta}}$. We then consider the following iterative simulation algorithm to solve the optimization problem (OPT):

$$\boldsymbol{\beta}^{(n+1)} = \boldsymbol{\beta}^{(n)} - \epsilon_n \mathbf{K} \left(\mathbf{W}^{(n+1)} - \frac{\langle \mathbf{W}^{(n+1)}, \mathbf{c} \rangle}{\langle \mathbf{c}, \mathbf{c} \rangle} \mathbf{c} \right), \tag{17}$$

where the variable $\mathbf{W}^{(n+1)}$ is an estimator of the gradient of $z(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, the scaling matrix $\mathbf{K}$ is taken by common practice to be the identity matrix $\mathbf{I}$, and the term in parentheses is the projection of the gradient estimate $\mathbf{W}^{(n+1)}$ onto the hyperplane $\{\langle \mathbf{c}, \boldsymbol{\beta} \rangle = C\}$. (Recall from Lemma 5 that the optimal solution satisfies the budget inequality constraint strictly in the Brownian tree-network setting.) This iterative scheme, known as the SA algorithm, has been well studied in the literature; refer to Asmussen and Glynn (2007), Kushner and Yin (2003). The SA algorithm is effectively the "stochasticization" of a Newton-type iterative optimization (or root-finding) algorithm. Assuming that the estimates $\mathbf{W}^{(n+1)}$ are all generated with the same sample size (i.e., independent of the iteration number n) and that the ϵ_n-sequence satisfies $\epsilon_n \to 0$ and $\sum_n \epsilon_n \to \infty$, it follows from known results (see Kushner and Yin (2003)) that the iterates $\boldsymbol{\beta}^{(n)}$ of (17) converge to the set of local minimizers of (OPT). Furthermore, the rate of convergence is $O(n^{-1/2})$ when the gradient estimator $\mathbf{W}^{(n+1)}$ is consistent, with a slower convergence rate otherwise. Unbiased methods of gradient estimators can be constructed for special stochastic network settings, e.g., tandem (Ho et al. (1983)) and Jackson-like (Glasserman (1991)) networks, which improve the asymptotic rate of convergence over biased methods such as those based on finite difference. However, finite-difference methods were chosen for our comparisons because of the wider applicability of finite-difference methods to general stochastic network settings. In addition, the difficulty in choosing the parameters is unaffected by the choice of the estimator.

We modify the generic finite-difference gradient estimation method for our numerical experiments. In particular, we employ a central-difference gradient estimator:

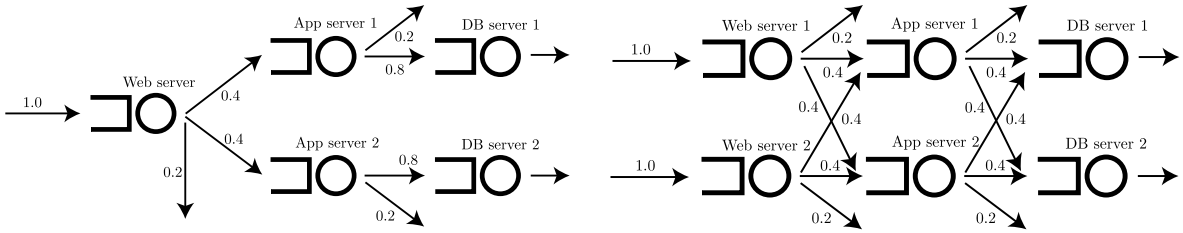
$$\mathbf{W}_i^{(n+1)} = \frac{z(\boldsymbol{\beta} + h_{n+1}\, \mathbf{e}_i) - z(\boldsymbol{\beta} - h_{n+1}\, \mathbf{e}_i)}{2h_{n+1}}, \qquad \forall i = 1, \ldots, L,$$

where h_{n+1} is the difference increment sequence and $\mathbf{e}_i$ the vector of all zeros except for a one in the i-th element. Such an estimator is known to converge at the best possible rate of $O(n^{-1/3})$ under a certain optimal choice for the ϵ_n and h_n. Observe the much slower optimal order of convergence for this scheme compared to that of the unbiased gradient-estimator. When such a finite-difference gradient estimator is used, the resulting SA algorithm is called a *Kiefer-Wolfowitz* scheme.

This scheme suffers from several significant sources of potential errors and inefficiencies when employed to solve (OPT). First, the quantities $z(\boldsymbol{\beta})$ that need to be evaluated are steady-state measures, and standard batch-means techniques for estimating such measures suffer from an additional source of bias due to initial transience. Second, although the SA theory specifies an optimal order-of-magnitude result for ϵ_n and h_n, the specific choice for these constants can be problematic. This is especially true when the gradient estimates are created from a fixed sample size, which is assumed in the theoretical results on convergence. (Note that, in this context, sample size is counted in terms of batches needed by the standard steady-state batch-means techniques.) In our experiments, we use an equivalent simulation termination criterion that tracks the standard confidence interval (CI) for the estimator of the objective function $z(\boldsymbol{\beta})$ and checks if the interval satisfies a desired relative size. This makes the sample size sensitive to the current iterate $\boldsymbol{\beta}^{(n)}$ via the variance of the gradient estimator $\mathbf{W}^{(n)}$. However, since the variance can be expected to be well-behaved in the interior of the budget-defined simplex, the sample sizes at each step remain independent of the iteration count n of the SA algorithm, and thus the asymptotic analysis of Kushner and Yin (2003) continues to hold. Our experiments demonstrate that if the sample size is chosen to be too small, then the "noise" in the iterate sequence is high (in the sense that iterates show little systematic improvement in solution quality with increasing n) and the SA algorithm needs to execute a large number of iterations n before the decreasing step-size sequence ϵ_n can ensure convergence. On the other hand, if the sample size is chosen to be too large, the algorithm expends a great deal of computer time in each step and is very slow to converge. Our experience here suggests that the SA iteration scheme works best in practice when started with a small sample size which is then slowly increased with the iteration count n. Table 1 describes the tradeoff observed for one specific experimental setting. This graduated-increase approach however does not fit any framework of analysis for SA algorithms, and a convergence analysis of such a scheme is beyond the scope of this paper.

5.2. Stochastic Network Configurations

We present results for two stochastic network configurations, namely a five-station tree network depicted in Figure 2(a) and a six-station feedforward network with a non-tree structure depicted in Figure 2(b). Both network structures arise naturally in a variety of computer architectures that serve Internet traffic, as well as various canonical business processes. In each configuration, there are three tiers of service for the processing of incoming requests, which we characterize in the context of a generic data center to clarify the presentation. The servers comprising the first tier (e.g., web servers) provide initial processing (e.g., access to web pages that may require updating of timely information such as stock prices). If a request requires additional processing (and the request is deemed to be legitimate), then the first-tier server either routes the request to a specific server comprising the second tier (e.g., application servers) or performs a form of load balancing of the arriving requests among these second-tier servers. The second tier of servers in turn either completes the processing of the request in its entirety using locally available information or provides another level of processing before forwarding the request for additional service by the next tier of servers (e.g., database servers). Probabilistic routing is often used to model the flow of traffic through these feedforward stochastic networks, and the specific probabilities used in our representative experiments presented below are given in Figure 2. All interarrival times and service times follow



(a) The Tree Network (b) The Non-Tree Feedforward Network

Figure 2 Network structures explored by the experiments. The parameters of the probabilistic routing are given.

2-stage Coxian distributions. Each server is assumed to have unit cost. The total capacity budget is set to 5 units for the tree-network configuration and 10 units for the feedforward configuration.

Three settings are defined for each network configuration by varying the weights assigned to each server. Recall from the discussion following Lemma 2 that the Brownian tree network version (OPT-BTN) of the optimization problem (OPT) is convex if the server weights satisfy $w_{\pi(i)} \geq w_i$. We therefore consider three settings that consist of (i) unit weights, (ii) weights satisfying the non-increasing condition, and (iii) weights arbitrarily assigned to yield a (possibly) non-convex instance of (OPT-BTN). For each of the resulting six model settings, numerical experiments are conducted over multiple combinations of coefficients of variation (CoVs) for the arrival and service processes, none of which satisfy the product-form requirements. The tree network is analyzed in Section 3, where we establish that the optimization problem (OPT-BTN) is convex when the server-weights w_i satisfy the non-increasing condition, and thus this problem has a unique globally optimal solution. Experimental results suggest that the iterates (4) of our fixed-point approximation method have a unique limit point for more general stochastic networks than those satisfying the conditions of Theorem 1. Under these circumstances, the test then becomes a straightforward comparison of the quality of the limit point obtained by iterations (4) against the global optimal solution identified by the SA algorithm in the second phase when started from the identified limit point.

For the same network configuration when the weights w_i yield a difference-of-convex functions for the objective function of (OPT-BTN), the problem may have multiple local optima and our fixed-point iteration algorithm itself may have multiple limit points. This is a possibility for the final four model settings. To test the hypothesis that multiple limit points could exist, we execute the first-phase algorithm from a collection of starting server capacity values sampled uniformly from the feasible set. In addition, we execute the SA algorithm from each of these same starting points to investigate if the algorithm identifies multiple local optima for the original problem (OPT).

5.3. Observations

Figure 3 plots the relative optimality gap for the limit points identified by the first phase of our framework based on the fixed-point iteration (4) in comparison with the locally optimal solution identified by the second phase of our framework phase based on the SA algorithm starting from the first-phase limit point. This two-phase framework was applied to each of the six model settings under multiple CoV combinations for the interarrival and service time distributions. It is evident from the results for each setting, plotted in Figure 3, that the second-phase SA algorithm is able to improve upon the solution quality by at most 5% for convex problem instances, with the majority of such improvements limited to 1.0–1.5%. The relative performance of our fixed-point iteration is reduced a bit for the non-convex case, where the worst-case improvement provided by the second phase rises to about 10% with the average-case improvement in the 3–5% range. In contrast, the

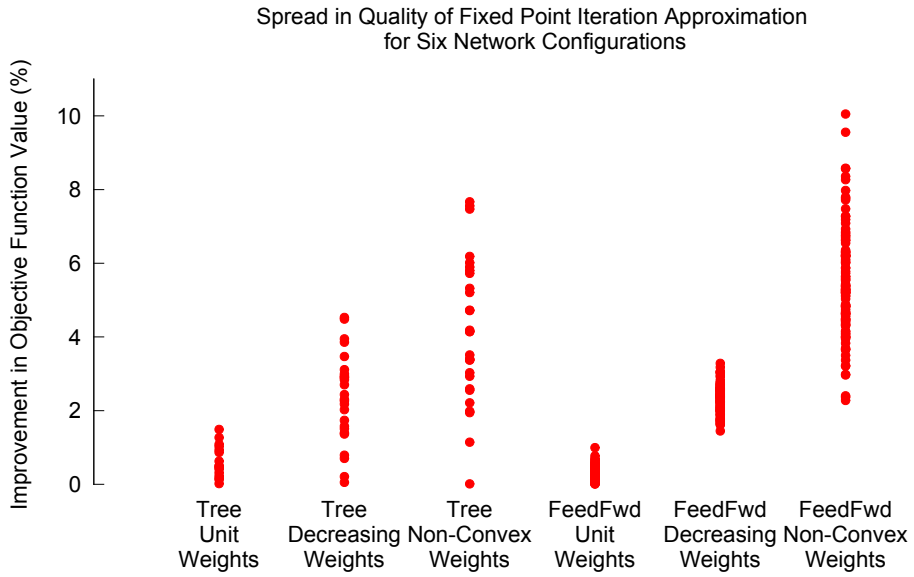


Figure 3 Quality of fixed-point iteration. The relative optimality gap is within 5%, typically 1% for settings where the Brownian network is convex. For settings where the Brownian network is possibly non-convex (simply referred to as non-convex in this diagram), the relative gap can be as much as 10%, while the average is around 4–5%.

objective function value for the optimal capacity allocation obtained using a corresponding product-form approximation (i.e., setting all CoVs to 1) was observed to have a relative optimality gap in the range of 75% to 350%, clearly indicating the very poor quality of such simplistic assumptions.

	Sim Termination Criterion	**Num Converged**	**Average Num Iterations**	**Total Sim Time**
1	Fixed at 5×10^{-2}	12	20.42	9.44×10^{8}
2	Geometric Decrease: 5×10^{-2} to 5×10^{-3} in 20 iters.	15	18.47	2.18×10^{9}
3	Fixed at 5×10^{-3}	16	19.19	5.90×10^{9}

Table 1 Setting stochastic approximation parameters. Not all 20 trials ran to convergence because the maximum iteration count, set to 25, was reached.

The simulation experiments for both the fixed-point iteration and the SA algorithm were terminated with the criterion that the relative CI of the estimator of the objective function $z(\cdot)$ falls below a desired value. (The simulations for the fixed-point iteration were also executed with an alternative stopping rule for relative CI gaps on the estimation of the average individual queue lengths, and the algorithm was found to be indifferent to the stopping rule chosen.) An implementation of the SA algorithm requires the user to select "good" parameter values, and a critical parameter is the stopping criterion for the simulation runs. Table 1 further underlines the computational savings from the fixed-point approximation method in the first phase of our framework, before employing the SA algorithm in the second phase, by illustrating the difficulty faced in making this stopping criterion choice. Multiple runs of the fixed-point approximation and the SA algorithm were initialized with 20 uniformly sampled starting points for a particular network setting from the "Feedforward non-convex weights" set in Figure 3. When the fixed-point approximation of the first phase is executed with a simulation stopping target of 5×10^{-3} as the relative

CI width, the method terminated after an average of 4.75 iterations over the 20 trials, with an average total simulation-time of 1.25×10^8 time-units. The SA algorithm was executed from the same 20 starting points with three different termination criteria: the first executes all simulations with relative CI-width to match 5×10^{-2}, the second gradually strengthens this requirement in a geometric sequence to 5×10^{-3} over the first 20 iterations, and the last executes all simulations to match 5×10^{-3}. Table 1 shows that the weaker CI bound helps the method converge faster, but the iterations of the SA algorithm tend to wander and fewer trials complete within the maximum count criterion. Strengthening the CI bound increases the number of successfully completed trials and decreases the average number of iterations required, but each trial takes much longer to execute in the aggregate. A practical parameter choice seem to be the approach that gradually strengthens the CI requirement and strikes a good balance between the run length and the accuracy.

Algorithm	**Solution Returned in β-space**						**Num. Trials Converge Here**	**Objective Value Estimate**	**Percentage Improvement**
First phase (Fixed-point)	1.59	1.61	1.19	1.20	2.26	2.15	20	59.17	–
Second phase	1.51	1.68	1.20	1.25	2.37	2.33	14	53.85	9.88%
(SA)	1.42	1.36	1.13	1.16	2.51	2.46	4	58.34	1.42%

Table 2 Performance of two-phase procedure over a non-convex setting with multiple locally optimal solutions. Note that the SA algorithm in the second-phase did not converge within reasonable computational budget for two of the trials.

Finally, in settings that do not satisfy the sufficiency conditions for a convex objective function in (OPT-BTN), we observe that the iterative approximation (3) *always produces a unique limit point* independent of the starting point. These results suggest that the uniqueness of our first-phase limit point holds more generally than the network conditions of Section 4. This is in stark contrast to the SA algorithm executed over the same randomly chosen starting points, which in some instances produces multiple local optima. In these cases, one expects to find locally optimal solutions spanning a range of solution quality, as illustrated by Table 2 which provides results from such a problem setting. Here, the fixed-point approximation and the SA algorithm were executed from 20 uniformly sampled starting points. While the fixed-point approximation always produces the same limit point, the overall two-phase framework finds two locally optimal solutions due to the fact that the SA algorithm can find more than one local optima even if it is started close to the same limit point. The quality improvement between the two local solutions does not exceed 10%, and thus one expects the locally optimal solution from our two-phase framework to be "close" to the seemingly unique fixed point identified in the first phase. In comparison, when the SA algorithm was executed from the same 20 starting points, many additional local optima were obtained that were of both better and worse quality than the local optima identified by our two-phase framework.

6. Conclusions

In this paper we developed a general framework for determining the optimal resource capacity allocation at each station comprising a stochastic network, motivated by computer capacity planning and business process management applications. These problems are well known to be very difficult from both a mathematical and practical perspective. Our solution framework is based on an iterative methodology that relies only on the capability to observe the queue lengths at all network stations under any given resource capacity allocation. We theoretically investigated this methodology for single-class Brownian tree networks, and further demonstrated the benefits of our methodology through extensive numerical experiments. The latter show that the first phase of our

methodology renders approximations to locally optimal solutions that are within 5% of optimality on average. In addition to these solution-quality benefits, our framework does not require the fine-tuning of parameters and appears to be insensitive to the chosen simulation stopping criterion. Our methodology further provides reductions in computation of multiple orders of magnitude over a purely simulation-optimization approach based on stochastic approximation. In fact, regardless of the parameter settings considered, all of the stochastic approximation algorithms required orders of magnitude more iterations than our methodology to converge to an optimal solution.

Appendix A: Properties of Brownian Tree Networks

This appendix briefly reviews elements of the construction of a single-class Brownian tree network as it arises from a generalized Jackson network with a tree network topology. Further details on constructing single-class Brownian networks in general can be found, for instance, in Harrison and Williams (1987). We also establish in this appendix several important properties of the networks of interest, including proofs of Lemmas 2 and 3 from Section 3. Both queue-length dynamics and steady-state behavior are considered, and thus we use slightly different notation from the body of the paper. In particular, we write $Z^{\boldsymbol{\beta}}(t)$ for the queue-length vector at time t, and use $Z^{\boldsymbol{\beta}}(\infty)$ instead of $Z^{\boldsymbol{\beta}}$ for the corresponding steady-state vector.

A Brownian tree network relies on an L-dimensional Brownian motion $\{X(t)\}$ with zero mean and covariance structure determined by

$$\mathbb{C}\mathrm{ov}(X_i(t), X_j(t)) = \Sigma_{ij} t,$$

where Σ_{ij} is given by, for $i = 1, \ldots, L$,

$$\Sigma_{ii} = \lambda_i c_{A,i}^2 + \gamma_i c_{B,i}^2 + \gamma_{\pi(i)} p_{\pi(i),i}(1 - p_{\pi(i),i}) + \gamma_{\pi(i)} p_{\pi(i),i}^2 c_{B,\pi(i)}^2, \tag{18}$$

and, for $i \neq j$,

$$\Sigma_{ij} = -\left[\gamma_i c_{B,i}^2 p_{ij} + \gamma_j c_{j,B}^2 p_{ji} + \gamma_{\pi(i)} p_{\pi(i),i} p_{\pi(j),j}(1 - c_{B,\pi(i)}^2) 1_{\{\pi(i)=\pi(j)\}}\right].$$

Here the notation introduced in Section 3 is used, and the parameters $c_{A,i}^2$ and $c_{B,i}^2$ correspond to the squared coefficient of variation of the external arrival process at station i and the squared coefficient of variation of the service times at station i, respectively, in the underlying (pre-limit) generalized Jackson network.

We write $Z^{\boldsymbol{\beta}}$ for the vector-valued process of queue lengths in the Brownian network under the service-rate vector $\boldsymbol{\beta}$. By construction, as in Harrison and Williams (1987), the process $Z^{\boldsymbol{\beta}}$ arises from the solution of a high-dimensional Skorokhod reflection problem with X plus a drift term (dependent on $\boldsymbol{\beta}$) as input. More precisely, following the convention that $I_{\pi(1)}(t) = \gamma_{\pi(1)} = \beta_{\pi(1)} = 0$, $(Z^{\boldsymbol{\beta}}, I)$ is the (unique) process satisfying:

- $Z_i^{\boldsymbol{\beta}}(t) = X_i(t) + [(\beta_i - \gamma_i) - p_{\pi(i),i}(\beta_{\pi(i)} - \gamma_{\pi(i)})]t + I_i(t) - p_{\pi(i),i} I_{\pi(i)}(t) \geq 0$ for any $i = 1, \ldots, L$, $t \geq 0$;
- I_i is continuous and nondecreasing, with $I_i(0) = 0$, for $i = 1, \ldots, L$;
- $\int_0^\infty Z_i(t) dI_i(t) = 0$ for $i = 1, \ldots, L$.

Using the explicit solution to this Skorokhod problem, we can establish the following lemma. Similar results appear in various places, but we provide a proof here for completeness.

LEMMA 7. *For $i = 1, \ldots, L$ and $\boldsymbol{x}_i > 0$, we have*

$$h_i(\boldsymbol{x}_i) = \mathbb{E}\left[\sup_{0 \leq t_i \leq t_{\pi(i)} \leq \cdots \leq t_1 < \infty} \sum_{j \in \mathcal{P}_i} q_{ji}\left(X_j(t_j) - [x_j - p_{\pi(j),j} x_{\pi(j)}]t_j\right)\right].$$

Proof. We first note that it is possible to verify the integrability of the random variable on the right-hand side for any $\boldsymbol{x}_i \in (0,\infty)^{|\mathcal{P}_i|}$, though such details are beyond the scope of this paper. A key ingredient would be Borell's inequality, which implies the integrability of the supremum of a centered Gaussian process with bounded variance (see, e.g., Adler (1990)).

It is well known that the solution to the Skorokhod problem associated with a (Brownian) tree network can be found by recursively solving one-dimensional Skorokhod problems; see, e.g., Dębicki et al. (2007). Assuming $Z^{\boldsymbol{\beta}}(0) = 0$, this leads to the identity, for $t \geq 0$,

$$I_i(t) = -\inf_{0 \leq t_1 \leq \cdots \leq t_{\pi(i)} \leq t_i \leq t} \sum_{j \in \mathcal{P}_i} q_{ji} \left(X_j(t_j) - [x_j - p_{\pi(j),j} x_{\pi(j)}] t_j \right).$$

Time reversal shows that

$$\sum_{j \in \mathcal{P}_i} q_{ji} Z_j^{\boldsymbol{\beta}}(t) \stackrel{d}{=} \sup_{0 \leq t_i \leq t_{\pi(i)} \leq \cdots \leq t_1 \leq t} \sum_{j \in \mathcal{P}_i} q_{ji} \left(X_j(t_j) - [x_j - p_{\pi(j),j} x_{\pi(j)}] t_j \right),$$

where $\stackrel{d}{=}$ denotes equality in distribution. Upon letting $t \to \infty$, we obtain by monotone convergence

$$\lim_{t \to \infty} \mathbb{E}\left[\sum_{j \in \mathcal{P}_i} q_{ji} Z_j^{\boldsymbol{\beta}}(t) \right] = h_i(\boldsymbol{x}_i),$$

from which we also conclude that $\{\sum_{j \in \mathcal{P}_i} q_{ji} Z_j^{\boldsymbol{\beta}}(t)\}$ is uniformly integrable. Combining this with the ergodic theorem for reflected Brownian motion yields

$$\lim_{t \to \infty} \frac{1}{t} \sum_{k=1}^{t} \mathbb{E}\left[\sum_{j \in \mathcal{P}_i} q_{ji} Z_j^{\boldsymbol{\beta}}(k) \right] = \mathbb{E}\left[\sum_{j \in \mathcal{P}_i} q_{ji} Z_j^{\boldsymbol{\beta}}(\infty) \right].$$

The claim then follows from the preceding two displays. □

Lemma 7 forms the basis for our proof of Lemma 2, which establishes structural properties of the h_i functions. We present this proof next, noting that condition (iii) of the lemma is defined more precisely below in terms of (18) than in Section 3.

Proof of Lemma 2. Fix i throughout. For given $0 \leq t_i \leq t_{\pi(i)} \leq \cdots \leq t_1 < \infty$, the quantity

$$\sum_{j \in \mathcal{P}_i} q_{ji} \left(X_j(t_j) - [x_j - p_{\pi(j),j} x_{\pi(j)}] t_j \right) \tag{19}$$

appearing in the definition of $h_i(\boldsymbol{x}_i)$ is convex in $\boldsymbol{x}_i$ (in fact, it is an affine function). As (the expected value of) a pointwise supremum of convex functions, h_i is convex in its argument as well. To see that it is nonincreasing, note that the drift term in (19) equals

$$-\sum_{j \in \mathcal{P}_i} q_{ji} [x_j - p_{\pi(j),j} x_{\pi(j)}] t_j = -\sum_{j \in \mathcal{P}_i} q_{ji} x_j (t_j - t_{s(j)}),$$

where $s(j)$ is the successor of j on the path $\mathcal{P}_i$, with $t_{s(i)} = 0$ by convention.

It remains to show (iii), defined here to be that h_i is strictly decreasing in the last coordinate x_i unless the degeneracy condition $\Sigma_{ii} = 0$ holds. We first introduce some notation, writing for $\boldsymbol{x}_i \in (0,\infty)^i$

$$Y_i^{\boldsymbol{x}_i}(\boldsymbol{t}_i) = \sum_{j \in \mathcal{P}_i} q_{ji} \left(X_j(t_j) - [x_j - p_{\pi(j),j} x_{\pi(j)}] t_j \right)$$

and

$$\overline{Y}_i^{\boldsymbol{x}_i} = \sup_{0 \leq t_i \leq t_{\pi(i)} \leq \cdots \leq t_1 < \infty} Y_i^{\boldsymbol{x}_i}(\boldsymbol{t}_i).$$

For some arbitrary $\epsilon > 0$, we also set

$$\overline{Y}_i^{\epsilon,\boldsymbol{x}_i} = \sup_{\epsilon \le t_i \le t_{\pi(i)} \le \cdots \le t_1 < \infty} Y_i^{\boldsymbol{x}_i}(\boldsymbol{t}_i),$$
$$\overline{Y}_{\epsilon,i}^{\boldsymbol{x}_i} = \sup_{t_i \le \epsilon, 0 \le t_i \le t_{\pi(i)} \le \cdots \le t_1 < \infty} Y_i^{\boldsymbol{x}_i}(\boldsymbol{t}_i),$$

so that $\overline{Y}^{\boldsymbol{x}_i} = \max(\overline{Y}_i^{\epsilon,\boldsymbol{x}_i}, \overline{Y}_{\epsilon,i}^{\boldsymbol{x}_i})$. Now suppose $x_i > x_i' > 0$, and write $\boldsymbol{x}_i' = (x_1', \ldots, x_{\pi(i)}', x_i') = (\boldsymbol{x}_{\pi(i)}, x_i')$. We first argue that $\{\overline{Y}_i^{\boldsymbol{x}_i} > \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}\} \subseteq \{\overline{Y}_i^{\boldsymbol{x}_i} > \overline{Y}_i^{\boldsymbol{x}_i'}\}$. Indeed, let the supremum in the definition of $\overline{Y}_i^{\boldsymbol{x}_i}$ be attained at $(t_i^*, t_{\pi(i)}^*, \ldots, t_1^*)$; it is easy to see that the supremum is attained almost surely. If $\overline{Y}_i^{\boldsymbol{x}_i} > \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}$, then $t_i^* > 0$ and thus we have

$$\begin{aligned}\overline{Y}_i^{\boldsymbol{x}_i'} &\ge Y_i^{\boldsymbol{x}_i'}(\boldsymbol{t}_i^*) = \sum_{j \in \mathcal{P}_i} q_{ji} X_j(t_j^*) - \sum_{j \in \mathcal{P}_i} q_{ji} x_j'(t_j^* - t_{s(j)}^*) \\ &> \sum_{j \in \mathcal{P}_i} q_{ji} X_j(t_j^*) - \sum_{j \in \mathcal{P}_i} q_{ji} x_j(t_j^* - t_{s(j)}^*) = \overline{Y}_i^{\boldsymbol{x}_i}.\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}&\mathbb{P}\left[\overline{Y}_i^{\boldsymbol{x}_i} > \overline{Y}_i^{\boldsymbol{x}_i'}\right] \ge \mathbb{P}\left[\overline{Y}_i^{\boldsymbol{x}_i} > \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}\right] \\ &\ge \mathbb{P}\left[\sup_{0 \le t_i \le \epsilon \le t_{\pi(i)} \le \cdots \le t_1 < \infty} \sum_{j \in \mathcal{P}_i} q_{ji}\left(X_j(t_j) - [x_j - p_{\pi(j),j} x_{\pi(j)}]t_j\right) > \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}\right] \\ &= \mathbb{P}\left[\sup_{0 \le t_i \le \epsilon}\left(X_i(t_i) - [x_i - p_{\pi(i),i} x_{\pi(i)}]t_i\right) + Y_{\pi(i)}^{\epsilon,\boldsymbol{x}_{\pi(i)}} > \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}\right] \\ &\ge \mathbb{P}\left[\overline{Y}_{\pi(i)}^{\epsilon,\boldsymbol{x}_{\pi(i)}} = \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}\right],\end{aligned}$$

where the last inequality relies on the observation that $\sup_{0 \le t_i \le \epsilon}\left(X_i(t_i) - [x_i - p_{\pi(i),i} x_{\pi(i)}]t_i\right) > 0$ almost surely; this uses $\Sigma_{ii} > 0$ and can, for instance, be deduced from the law of the iterated logarithm. In view of $\overline{Y}_i^{\boldsymbol{x}_i} \ge \overline{Y}_i^{\boldsymbol{x}_i'}$, we find that $\mathbb{E}\overline{Y}_i^{\boldsymbol{x}_i} > \mathbb{E}\overline{Y}_i^{\boldsymbol{x}_i'}$ after proving that $\overline{Y}_{\pi(i)}^{\epsilon,\boldsymbol{x}_{\pi(i)}} = \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}$ with positive probability. To see this, observe that, with $\widehat{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}$ being an independent copy of $\overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}$ and $\boldsymbol{\epsilon}_i = (\epsilon, \ldots, \epsilon)$, we have

$$\begin{aligned}\mathbb{P}\left[\overline{Y}_{\pi(i)}^{\epsilon,\boldsymbol{x}_{\pi(i)}} = \overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}\right] &= \mathbb{P}\left[\overline{Y}_{\epsilon,\pi(i)}^{\boldsymbol{x}_{\pi(i)}} \le \overline{Y}_{\pi(i)}^{\epsilon,\boldsymbol{x}_{\pi(i)}}\right] \\ &= \mathbb{P}\left[\overline{Y}_{\epsilon,\pi(i)}^{\boldsymbol{x}_{\pi(i)}} - Y_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}(\boldsymbol{\epsilon}_i) \le \overline{Y}_{\pi(i)}^{\epsilon,\boldsymbol{x}_{\pi(i)}} - Y_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}(\boldsymbol{\epsilon}_i)\right] \\ &\ge \mathbb{P}\left[\overline{Y}_{\epsilon,\pi(i)}^{\boldsymbol{x}_{\pi(i)}} - Y_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}(\boldsymbol{\epsilon}_i) < M, \widehat{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}} > M\right] \\ &= \mathbb{P}\left[\overline{Y}_{\epsilon,\pi(i)}^{\boldsymbol{x}_{\pi(i)}} - Y_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}(\boldsymbol{\epsilon}_i) < M\right] \mathbb{P}\left[\overline{Y}_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}} > M\right] \\ &\ge \mathbb{P}\left[\overline{Y}_{\epsilon,\pi(i)}^{\boldsymbol{x}_{\pi(i)}} - Y_{\pi(i)}^{\boldsymbol{x}_{\pi(i)}}(\boldsymbol{\epsilon}_i) < M\right] \\ &\qquad \times \sup_{0 \le t_i \le t_{\pi(i)} \le \cdots \le t_1 < \infty} \mathbb{P}\left[\sum_{j \in \mathcal{P}_i} q_{ji}\left(X_j(t_j) - [x_j - p_{\pi(j),j} x_{\pi(j)}]t_j\right) > M\right].\end{aligned}$$

The first probability is bounded away from 0 for large M since $\overline{Y}_{\epsilon,\pi(i)}^{\boldsymbol{x}_{\pi(i)}} < \infty$ with probability 1, and the supremum over Gaussian tails is readily seen to be bounded away from 0 as well. □

Lastly, we now prove the homogeneity property of the h_i functions.

Proof of Lemma 3. Observe that

$$\sup_{0\le t_i\le t_{\pi(i)}\le\cdots\le t_1<\infty}\sum_{j\in\mathcal{P}_i} q_{ji}\left(\delta X_j(t_j)-[x_j-p_{\pi(j),j}x_{\pi(j)}]\delta^2 t_j\right)$$
$$=\sup_{0\le t_i\le t_{\pi(i)}\le\cdots\le t_1<\infty}\sum_{j\in\mathcal{P}_i} q_{ji}\left(\delta X_j(\delta^{-2}t_j)-[x_j-p_{\pi(j),j}x_{\pi(j)}]t_j\right),$$

and that $\{\delta X(\delta^{-2}t)\}$ has the same distribution as $\{X(t)\}$ by the Brownian scaling property. □

Appendix B: Proof of Proposition 2

By assumption, for any $\epsilon>0$, $x_\ell^{(k)}\in[x_\ell^*\pm\epsilon]$ for sufficiently large k and all stations $\ell\in\mathcal{P}_{\pi(i)}$. We write $\alpha_i=c_1w_i/(c_iw_1\tau_1)$. The contour $\mathcal{C}_i=\{\boldsymbol{x}_i\in(0,\infty)^{|\mathcal{P}_i|}: h_i(\boldsymbol{x}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})=\alpha_i^{-1}x_i^*\}$ is continuous in any neighborhood of $\boldsymbol{x}_i^*$, since each h_ℓ is a continuous function on $(0,\infty)^\ell$ by convexity (refer to Lemma 2); see (Rockafellar 1970, Theorem 10.1). Moreover, since $h_i(\boldsymbol{x}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})$ is strictly increasing in x_i by Lemma 2, a suitable version of the implicit function theorem (see, e.g., Kumagai (1980)) shows that there exists a continuous function $g_i:(0,\infty)^{|\mathcal{P}_i|-1}\to(0,\infty)$ such that $\mathcal{C}_i$ coincides with the graph $\{(\boldsymbol{x}_{\pi(i)},g_i(\boldsymbol{x}_{\pi(i)})):\boldsymbol{x}_{\pi(i)}\in(0,\infty)^{|\mathcal{P}_i|-1}\}$, where g_i is defined through

$$\begin{aligned}g_i(\boldsymbol{x}_{\pi(i)}) &= \inf\{x_i>0: h_i(\boldsymbol{x}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})=\alpha_i^{-1}x_i^*\}\\ &= \sup\{x_i>0: h_i(\boldsymbol{x}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})=\alpha_i^{-1}x_i^*\}.\end{aligned}$$

For arbitrary $\eta>0$, by continuity of g_i, we can find some $\delta(\eta)$ such that the range of g_i on $[x_1^*\pm\delta(\eta)]\times\cdots\times[x_{\pi(i)}^*\pm\delta(\eta)]$ is included in $[x_i^*\pm\eta]$; refer to Figure 1. We may assume without loss of generality that $\delta(\eta)$ is strictly increasing in η.

Next we prove that, for an iterate in the shaded areas of Figure 1, the subsequent iterate moves in the direction indicated by the arrows. If $x_i>g_i(\boldsymbol{x}_{\pi(i)})$, we have $h_i(\boldsymbol{x}_i)-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)})<\alpha_i^{-1}x_i^*$ by construction. In view of the identity

$$x_i^{(k+1)}=\sqrt{\alpha_i x_i^{(k)}\left[h_i(\boldsymbol{x}_i^{(k)})-p_{\pi(i),i}h_{\pi(i)}(\boldsymbol{x}_{\pi(i)}^{(k)})\right]},$$

the inequality $x_i^{(k)}>g_i(\boldsymbol{x}_{\pi(i)}^{(k)})$ is equivalent to

$$x_i^{(k+1)}<\sqrt{x_i^*x_i^{(k)}}.$$

Thus, we obtain $x_i^{(k+1)}<\sqrt{x_i^*x_i^{(k)}}$ for $x^{(k)}\in\mathcal{U}_i^\eta$. Similarly, the inequality $x_i^{(k+1)}>\sqrt{x_i^*x_i^{(k)}}$ holds on the set $\mathcal{L}_i^\eta$. The claims readily follow from these observations.

Acknowledgments

ABD has been supported in part by an IBM postdoctoral fellowship and NSF grant EEC-0926308.

References

Adler, R. J. 1990. *An introduction to continuity, extrema, and related topics for general Gaussian processes*. Institute of Mathematical Statistics, Hayward, CA.

An, Le Thi Hoai, Pham Dinh Tao. 2005. The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Ann. Oper. Res.* **133** 23–46.

Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, I. Stoica A. Rabkin, M. Zaharia. 2009. Above the Clouds: A Berkeley View of Cloud Computing. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf.

Asmussen, Søren, Peter W. Glynn. 2007. *Stochastic simulation: algorithms and analysis*. Springer.

Baskett, F., K. M. Chandy, R. R. Muntz, F. Palacios-Gomez. 1975. Open, closed and mixed networks of queues with different classes of customers. *J. ACM* **22** 248–260.

Chen, Hong, David D. Yao. 2001. *Fundamentals of queueing networks*, vol. 46. Springer-Verlag, New York.

Conn, Andrew R., Nicholas I. M. Gould, Philippe L. Toint. 2000. *Trust-region methods*. SIAM, Philadelphia, PA.

Dai, J. G., J. M. Harrison. 1992. Reflected Brownian motion in an orthant: numerical methods for steady-state analysis. *Ann. Appl. Probab.* **2** 65–86.

Dębicki, K., A. B. Dieker, T. Rolski. 2007. Quasi-product forms for Lévy-driven fluid networks. *Math. Oper. Res.* **32** 629–647.

Dieker, A. B., X. Gao. 2011. Sensitivity analysis for diffusion processes constrained to an orthant. ArXiv:1107.2871.

Dikaiakos, M. D., D. Katsaros, P. Mehra, G. Pallis, A. Vakali. 2009. Cloud Computing: Distributed Internet Computing for IT and Scientific Research. *IEEE Internet Computing* **13**(5) 10–13.

Gartner. 2012. Cloud Computing Planning Guide.

Glasserman, P. 1991. *Gradient Estimation Via Perturbation Analysis*. Kluwer, Netherlands.

Glover, Fred, James Kelly, Manuel Laguna. 1999. New advances for wedding optimization and simulation. *Proceedings of the 1999 Winter Simulation Conference*. 255–260.

Guarisco, J., D.A. Samuelson. 2011. Rx for the ER: Service delivery model greatly improves emergency department performance. *OR/MS Today* **38**(5) 30–35.

Harrison, J. M., V. Nguyen. 1990. The QNET method for two-moment analysis of open queueing networks. *Queueing Systems Theory Appl.* **6**(1) 1–32.

Harrison, J. M., R. J. Williams. 1987. Brownian models of open queueing networks with homogeneous customer populations. *Stochastics* **22** 77–115.

Harrison, J. M., R. J. Williams. 1992. Brownian models of feedforward queueing networks: quasireversibility and product form solutions. *Ann. Appl. Probab.* **2** 263–293.

Heching, Aliza R., Mark S. Squillante. 2013. Stochastic decision-making in information technology services delivery. *Decision Making in Service Industries: A Practical Approach*, chap. 1. Taylor and Francis, 3–36.

Ho, Y.C., Xiren Cao, Christos Cassandras. 1983. Infinitesimal and finite perturbation analysis for queueing networks. *Automatica* **19**(4) 439–445.

Ishikawa, Shiro. 1974. Fixed points by a new iteration method. *Proceedings of the American Mathematical Society* **44**(1) 147–150.

Iyoob, I.M., E. Zarifoglu, A.B. Dieker. 2012. Cloud computing Operations Research. Submitted to Service Science.

Kelly, Frank P. 1991. Loss networks. *Ann. Appl. Probab.* **1**(3) 319–378.

Kingman, J. F. C. 1962. On queues in heavy traffic. *J. Roy. Statist. Soc. Ser. B* **24** 383–392.

Kleinrock, L. 1964. *Communication Nets; Stochastic Message Flow and Delay*. McGraw-Hill Book Company, New York.

Kumagai, S. 1980. An implicit function theorem: comment. *J. Optim. Theory Appl.* **31**(2) 285–288. doi: 10.1007/BF00934117.

Kushner, H. J., G. G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY.

Laguna, M., J. Marklund. 2004. *Business Process Modeling, Simulation and Design*. Prentice Hall.

Mann, W. Robert. 1953. Mean value methods in iteration. *Proceedings of the American Mathematical Society* **4** 506–510.

Menasce, Daniel A., Virgilio A.F. Almeida. 1998. *Capacity Planning for Web Performance: Metrics, Models, and Methods*. Prentice Hall.

Menasce, Daniel A., Virgilio A.F. Almeida. 2000. *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*. Prentice Hall.

Menasce, Daniel A., Virgilio A.F. Almeida, Larry W. Dowdy. 2004. *Performance by Design: Computer Capacity Planning by Example*. Pr.

Nelson, B.L., S.G. Henderson, eds. 2007. *Handbooks in OR and MS: Simulation*. Elsevier Science.

Pollett, P. K. 2009. Optimal capacity assignment in general queueing networks. *Optimization: structure and applications*, *Springer Optim. Appl.*, vol. 32. Springer, New York, 261–272.

Reiman, Martin I. 1984. Open queueing networks in heavy traffic. *Math. Oper. Res.* **9**(3) 441–458.

Rockafellar, R. T. 1970. *Convex analysis*. Princeton University Press, Princeton, N.J.

Saure, D., P. Glynn, A. Zeevi. 2009. A linear programming algorithm for computing the stationary distribution of semi-martingale reflected Brownian motion. Tech. rep., Graduate School of Business, Columbia University.

Squillante, Mark S. 2011. Stochastic analysis and optimization of multiserver systems. Danilo Ardagna, Li Zhang, eds., *Run-time Models for Self-managing Systems and Applications*, chap. 1. Springer, 1–24.

Wein, L. M. 1989. Capacity allocation in generalized Jackson networks. *Oper. Res. Lett.* **8**(3) 143–146.