QRF: an Optimization-Based Framework for Evaluating Complex Stochastic Networks

Giuliano Casale, Imperial College London, UK, g.casale@imperial.ac.uk
Vittoria De Nitto Personé, University of Rome Tor Vergata, Italy, denitto@info.uniroma2.it
Evgenia Smirni, College of William and Mary, VA, US, esmirni@cs.wm.edu

The Quadratic Reduction Framework (QRF) is a numerical modelling framework to evaluate complex stochastic networks composed of resources featuring queueing, blocking, state-dependent behavior, service variability, temporal dependence, or a subset thereof. Systems of this kind are abstracted as network of queues for which QRF supports two common blocking mechanisms: blocking-after-service and repetitive-service random-destination. State-dependence is supported for both routing probabilities and service processes. To evaluate these models, we develop a novel mapping, called Blocking-Aware Quadratic Reduction (BQR), which can describe an intractably large Markov process by a large set of linear inequalities. Each model is then analyzed for bounds or approximate values of performance metrics using optimization programs that provide different levels of accuracy and error guarantees. Numerical results demonstrate that QRF offers very good accuracy and much greater scalability than exact analysis methods.

Categories and Subject Descriptors: C.4 [**Performance of Systems**]: Modeling Techniques

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Queueing, blocking, temporal dependence, state-dependence

1. INTRODUCTION

The analysis of complex computer and communication systems often leads to describe the interaction of multiple resources composing, for example, a computer infrastructure, a network, or a distributed web application accessing database and storage. These systems display complex behaviors both in transient and stationary regimes and thus have been subject of intense research, ranging from analytical techniques to simulation. We here focus on stationary analysis, which is often the focus of design-time studies where the system is not yet operational and thus performance assessment is usually made by comparing steady-state metrics (e.g., throughputs, response times, utilizations) for specific design scenarios. Steady-state analysis is also quite common in run-time studies, where arrival and service rates are often approximately assumed constant at a reference timescale.

For the class of systems above, effects such as queueing, service blocking, state dependent behavior, processing rate variability, or temporally correlated events are quite common to observe, but models that simultaneously consider all these features are often difficult to analyze. Indeed, the literature is rich in works examining the stochastic properties of each of these features in isolation, for example by means of Markov pro-

This work was partially supported by NSF grants CCF-0937925 and CCF-1218758. The authors thank Valentina Lenti for extensive experimentation in her MSc thesis that helped in understanding better the characteristics of queueing networks with blocking.

DOI: http://dx.doi.org/10.1145/0000000.0000000

cesses [Onvural 1990; Perros 1994; Balsamo et al. 2001; Bolch et al. 2006], but the state-space explosion and the difficulty of combining multiple approximations makes it challenging to examine complex models. Thus, simulation is often preferred to evaluate models with a high degree of complexity.

In this paper, we propose an alternative evaluation methodology, called the Quadratic Reduction Framework (QRF). QRF uses liner programming to derive approximate solutions and bounds on the equilibrium solution of complex models. Compared to simulation, resorting to optimization presents a number of advantages. For example, information acquired from measurements can be directly embedded in the optimization program to speedup the model evaluation by reducing the number of unknowns or bounding them in a narrow interval. Furthermore, uncertainties on the model parameters in linear programs can be handled by robust programming [Bertsimas et al. 2011]. Lastly, sensitivity analysis techniques for linear programming can be used to greatly accelerate the evaluation of a sequence of QRF models. Due to limited space, we show in the paper only the latter case of sensitivity analysis, leaving extensions based on robust programming for future work.

Model features such as queueing or service variability are intrinsically non-linear and thus difficult to describe using a set of linear relations. One exception is when the model is Markovian, since the global balance equations that describe the process at equilibrium can be expressed as a system of linear equations. However, the size of this system grows prohibitively large as the model size grows. A more scalable approach is to represent the system using fluid approximations [Bertsimas et al. 1996], however these methods can only describe deterministic event rates, losing the benefits of a stochastic description.

Differently from fluid methods, QRF tackles the scalability challenge by generating an *incomplete* description of the underlying Markov process without giving up the stochasticity of the model. This description takes the form of hundreds or thousands of linear constraints relating marginal probabilities for the model states at equilibrium. The benefit of this approach is that marginal probabilities are fewer in number than the joint probabilities considered by the global balance equations. However, this comes at the price of limiting the analysis to metrics that can be computed as functions of second-order marginals. Also, the QRF description does not readily provide an exact solution of the model, instead it restricts the allowed values for the marginal probabilities at equilibrium. Within this region of feasibility for the marginals, an optimisation program can be formulated to compute bounds or approximations on metrics that are functions of marginals, such as utilizations, throughputs, mean and covariances of queue-lengths. Thus, while the QRF framework imposes restrictions, it is sufficiently general to compute performance metrics of first and second order that are the most frequently used in the computer and communication engineering practice.

QRF models are automatically mapped into a set of linear inequalities by a procedure that we call Blocking-Aware Quadratic Reduction (BQR). We find that this linear mapping is possible under quite general conditions. Compared to existing methodologies for evaluating queueing networks via linear programming (e.g., [Casale et al. 2010]), BQR greatly generalizes the scope of the theory by allowing state dependence and blocking, which present distinctive challenges. The main innovation is the ability to handle the significant modification of the state space of the underlying Markov process due to the introduction of these new features. We find that, despite such a broad generalization, the resulting models are still tractable and admit multiple forms of approximate analysis. In particular, we develop linear programming bounds, a maximum entropy method, and a novel minimal mutual information method to provide approximate solutions to the models. Thus, QRF offers a hierarchy of alternative approximation methods for the evaluation of complex stochastic networks, which includes both approximations and bounds.

The paper is organized as follows. In Section 2 we present the QRF approach and explain in Section 3 modelling assumptions. In Section 4 we develop the analytical characterization of QRF models and the BQR state space reduction. Section 5 discusses performance evaluation of QRF models. We illustrate actual performance of the QRF bounds in Section 6 and extend the technique to approximate analysis in Section 7. A general discussion of the applicability of QRF in comparison with simulation and exact methods is given in Section 8. Finally, we discuss some related work in Section 9 and the interested reader can find further discussion on application domains in the online Supplement.

2. METHODOLOGY

We first introduce the key ideas underlying the QRF framework by considering the special case of a network composed of M resources[1], each characterized by a state $n_i \geq 0$, $1 \leq i \leq M$, describing the number of jobs currently at resource i. For such a system, the joint probability distribution of the states at equilibrium may be written as $\pi(n_1, n_2, \ldots, n_M)$ and is normally obtained as the equilibrium distribution of a suitably-defined Markov process. Unfortunately, state space explosion poses in practice a barrier to computing and storing the joint distribution.

The Quadratic Reduction Framework (QRF) proposed in this paper addresses this limitations as follows:

- The first idea consists in describing the system in terms of a (possibly large) set of marginal probabilities, instead of joint probabilities. That is, rather than evaluating joint probabilities of the kind $\pi(n_1, n_2, \ldots, n_M)$, we focus on describing the system in terms of marginal probabilities $\pi(n_i, n_j)$, defined for each pair of queues $i, j = 1, \ldots, M$. If the system is described by a set of M^2 marginals, each taking one of $N + 1$ values, the description scales as $O((MN)^2)$, thus remaining tractable compared to the $O(N^M)$ complexity of a representation based on joint probabilities. Under this new description, the main challenge is to reformulate the global balance equations of the underlying Markov process in such a way that only marginals $\pi(n_i, n_j)$ appear in the final expressions. This means that joint probabilities $\pi(n_1, n_2, \ldots, n_M)$ can be disregarded in the analysis. This may not always be possible in general, for example it requires that all transition rates in the underlying Markov process depend on the state of no more than two stations[2]. However, we show that for the models we consider, global balance equations can always be *exactly* aggregated to relate all and only marginals of the type $\pi(n_i, n_j)$. The resulting aggregate balance equations are in general under-determined, thus insufficient to exactly determine the values of the marginals. Yet, they limit the structure of the equilibrium distribution by effectively defining a *feasible region* for the $\pi(n_i, n_j)$ marginals. As we show throughout the paper, this feasible region is often small enough to generate tight bounds or accurate approximations on the equilibrium solution of the model.
- The second main idea behind the QRF framework is to use optimization methods to generate bounds or approximations on the objective function f_{obj} in such a way that they can be expressed as a function only of the $\pi(n_i, n_j)$ marginals. Metrics of large interest in engineering studies such as server utilizations, mean throughputs,

[1]Throughout the paper, we use the terms resource, station, node and queue interchangeably.
[2]For example, if the processing rate μ of a queue depends on the state of three other stations, e.g., $\mu \equiv \mu(n_1, n_2, n_3)$, it would be then impossible to exactly describe these transitions using marginals $\pi(n_i, n_j)$.

mean response times, mean and covariances of queue lengths are all examples of valid objective functions in the QRF framework. Furthermore, we develop two approximation methods based on maximum entropy and minimum mutual information metrics, which can be used to generate good estimates for the exact value of the objective f_{obj}.

We describe in the next section the class of models to which QRF applies.

3. QRF MODELS

QRF is a methodology to evaluate complex stochastic networks where resources feature queueing, temporally correlated non-exponential service times, finite capacities, blocking, and state dependence. We describe in the next subsections the main technical assumptions underlying the QRF framework and the supported class of models.

3.1. Temporal Dependence

In order to describe events such as job departures or arrivals, QRF supports the specification of event rates by Markov-modulated processes[3], such as MMPPs or MAPs [Bolch et al. 2006]. A Markovian Arrival Process (MAP) is a continuous-time hidden-Markov process with phase-type distributed observations, we point to [Bolch et al. 2006] for background and an introduction to the (D_0, D_1) notation. The sequence of observations generated by this point processes is identically distributed, but not necessarily i.i.d. since successive inter-arrival times of observations can be correlated. A MAP is defined by an infinitesimal generator Q with K phases (i.e., states) on which state transition rates are marked as either hidden or observable. The rates in Q marked as observable are described by a non-negative matrix D_1, while hidden transitions rates are described by a matrix $D_0 = Q - D_1$. When an observable transition fires, the event associated to the MAP occurs. For example, if the MAP represents a service process, an observable transition is interpreted as a job service completion. MAPs can be easily integrated in Markov processes by tracking in the state space description also the current state of each MAP used in the model. By using existing fitting tools, such as the KPC-Toolbox [Casale et al. 2008], one can directly fit empirical time series into MAPs to represent high-variability and temporal dependence in arrival and service processes. We point to [Bolch et al. 2006; Casale et al. 2008] for further background on MAPs.

3.2. Queueing Stations

QRF considers queueing network models with M stations and a routing matrix $P = [p_{ij}]$, $i, j, = 1, \dots, M$, such that jobs departing from queue i are directed to queue j with probability p_{ij}. This probability is allowed to be state dependent according to the assumptions described later in Section 3.5. The network is assumed to have a closed topology with N circulating jobs. The closed network assumption does not significantly restrict the applicability of the methodology[4]. The buffer capacity of queue j is denoted by $F_j \leq N$, n_j is the random variable denoting the current population at queue j and k_j is the phase of the MAP service process. Station service processes can also depend on the local state (n_j, k_j) of the queue and on the state (n_i, k_i) of the destination queue

[3]Generalizations to multi-class and batch processes appear possible but they are not contemplated in the present paper.

[4]In fact, suppose that one instead wants to describe an open network with Poisson arrivals with rate λ. Then by adding a station with exponential server having rate λ, one approximates with increasing accuracy the open model as N increases. Further discussion on the relationships between open and closed models for some classes of queueing networks can be found in [Zahorjan 1983; Whitt 1984].

at which the job will be routed. This allows us to represent features such as multiple servers, load-dependent rates, and state-dependent routing.

3.3. Finite Capacity Buffers and Blocking Mechanisms

When $n_j = F_j$, queue j becomes full and it does not accept in its waiting buffer any *new* job before a departure occurs. QRF supports Blocking After Service (BAS) and Repetitive Service-Random Destination (RS-RD) blocking mechanisms [Balsamo et al. 2001].

- *Blocking After Service (BAS).* BAS is used in models of manufacturing systems and disk I/O subsystems [Yamada et al. 2009]. In BAS, a queue i, if not empty, processes a job regardless of the job population at its destination j. When node i completes service and node j is full, node i suspends any activity (i.e., it is blocked) and the completed job waits until a departure occurs from node j. At that moment two simultaneous transitions take place: the completed/blocked job moves from i to j (since j can now accept a job, i unblocks) and the job that leaves j (which effectively unblocks server i). In a general network topology where several queues compete for sending a job towards queue j that has a full buffer, a policy regulating the order in which queues unblock has to be defined. Usually, the First Blocked First Unblocked (FBFU) policy is considered fair: first unblock the queue that was blocked first. In the remaining of this paper when we consider BAS we assume that the FBFU policy is used.
- *Repetitive Service-Random Destination (RS-RD).* RS-RD blocking is used to model congestion control in telecommunications systems [Awan et al. 2006]. In this mechanism, a queue i, if not empty, processes a job regardless of the job population at its destination j. If node j is full, the completed job is rerouted to node i where it receives a new service. During the new service, the job may select a destination that is independent from its previous one. Note that according to RS-RD blocking a node is never actually blocked, but it “wastes” its service by repeating it.

The RS-RD and BAS blocking mechanisms introduce complexity in the underlying Markov chains of queueing network models. On the one hand, RS-RD restricts the original state space, while preserving its regular structure, on the other hand BAS introduces new states describing the order in which queues progressively block once the capacity of the destination node becomes full. This information is needed to implement the FBFU rule.

To reduce the complexity of the notation, we focus on describing the framework for the BAS blocking case and illustrating the formulas only when a single queue f has finite capacity $F_f < N$ and its sending nodes behave according to BAS blocking, while all other queues $i \neq f$ have infinite capacity (i.e., $F_i = N$) such that they can accommodate all jobs in the network. The generalization to networks with several finite capacity queues is feasible and follows a similar argument, as we discuss in the online Supplement. As we show in the next section, RS-RD blocking is a simplification compared to BAS; thus we point to [De Nitto Personé et al. 2010; 2011] for further details on RS-RD and related formulas used in QRF.

Note that for some networks there exist combinations of the F_j values that lead to deadlock in presence of blocking. Such pathological situations can be examined offline prior to the application of the QRF methodology. Conditions on the network population can be checked in order to avoid models with deadlocks [Balsamo et al. 2001].

3.4. State Space

Stemming from the assumptions in the previous subsections, we now describe the state space of models supported by QRF, focusing on the case of BAS blocking. A summary of

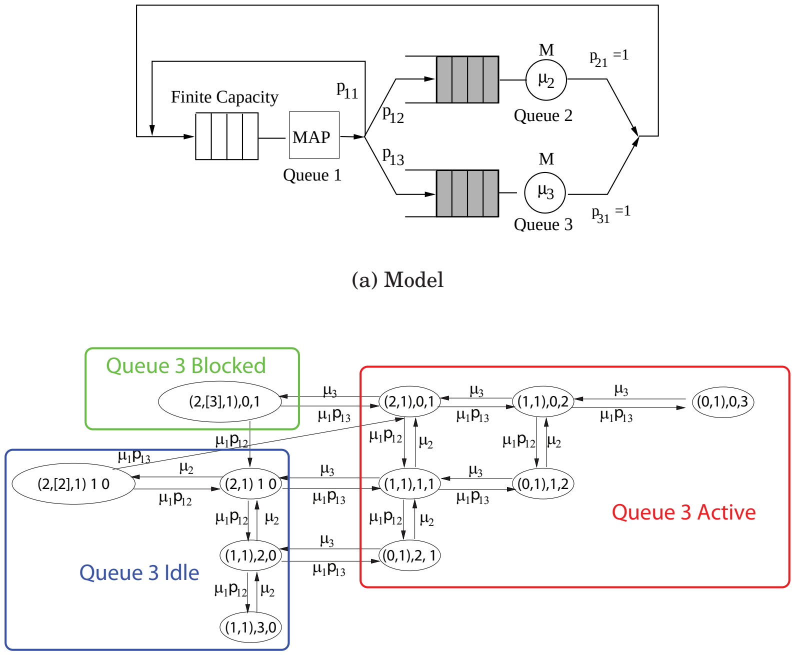
Table I. Summary of notation for queueing network models with BAS blocking

b	cardinality of the list of blocked queues $\boldsymbol{m}$
b_i	blocking state of node i
B	maximum number of queues that can block on f
f	finite capacity queue
F_i	capacity of queue i
K_i	number of phases in the MAP service process of queue i
k_i	current phase index in the MAP service process of queue i
$\boldsymbol{m}$	list of queues blocked by f
$Add(\boldsymbol{m}, j)$	list obtained by adding j to the tail of $\boldsymbol{m}$
$Head(\boldsymbol{m})$	first queue to unblock after a departure from f that is not self-routed
M	number of queues in the network
N	number of jobs in the network
n_i	number of jobs at queue i
$\pi(n_i, k_i, n_j, k_j, \boldsymbol{m})$	prob. of n_i jobs in queue i in phase k_i and n_j jobs in queue j in phase k_j and $\boldsymbol{m}$ blocked
$q_{i,j}^{k_i,k'_i} \equiv q^{k'_i}(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$	state-dependent rate of job departures from i to j when i's MAP is in phase k_i and which leave the MAP in phase k'_i when the system is in state $(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$

the main notation is given in Table I. A feasible network state in the queueing network underlying Markov process is a tuple $\boldsymbol{s} = (\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_M)$, where for queue $i \neq f$ the local state $\boldsymbol{s}_i = (n_i, b_i, k_i)$ is defined as follows: n_i is the current queue-length at station i including the job in service; b_i is the blocking state of node i (1=blocked, 0=active); k_i is the index of the active phase of the MAP service process at queue i. The service process at station i is a MAP with a total of $K_i \geq 1$ phases. The number of phases is constant and independent of the queue state. It is worth noting that if a queue is blocked, it completely stops its service activity, including MAP phase transitions. Note that this holds for BAS blocking, but not for RS-RD where a queue is never effectively blocked. As a consequence, for RS-RD the local state notation can be simplified to $\boldsymbol{s}_i = (n_i, k_i)$. In case of state dependent models, one can think to the destination queue j being selected upon service start and then the job processed according to a MAP that depends both on i and j.

Conversely, for the finite capacity queue f the state is $\boldsymbol{s}_f = (n_f, \boldsymbol{m}, k_f)$ where $\boldsymbol{m} = (m_1, m_2, \ldots, m_b)$ is a list that holds the sequence of $b = \sum_{i \neq f} b_i$ queues that can be unblocked by a departure from queue f. The index b thus denotes the current number of blocked queues and ranges in $0, \ldots, B$, where B is the number of queues $i \neq f$ that can send jobs to f. Note that it is often assumed in the literature that a finite capacity queue is never blocked by itself; we also take this assumption. The case $\boldsymbol{m} = \emptyset$ denotes that no queue is blocked by f. We denote with $Head(\boldsymbol{m})$ the head of the list, i.e., the first queue to unblock upon a departure from f that is not self-routed, and with $Add(\boldsymbol{m}, j)$ the list resulting from the addition of element j to the tail of $\boldsymbol{m}$. Finally, let E_{BAS} be the state space of the queueing network assuming BAS blocking at each node $i \neq f$ that can send jobs to f. In this state space, the Markov process transitions have rates from state $\boldsymbol{s} = (\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_M)$ to $\boldsymbol{s'} = (\mathrm{s'}_1, \mathrm{s'}_2, \ldots, \mathrm{s'}_M)$ that are uniquely defined by the rates $q_{i,j}^{k,h}$ of jobs flowing from i to j in phase k leaving queue i in phase h.

Focusing on the population random variables n_i, the state space of a blocking network is a subset of the state space of the same network but with infinite capacity queues. This is immediate because all states with $n_f > F_f$ do not exist. On the other hand, the order in which queues block needs to be explicitly accounted for in the $\boldsymbol{m}$ vector, which increases the state space cardinality. Thus, the state space of a BAS queueing network model can be smaller or bigger than in the non-blocking case.



(a) Model

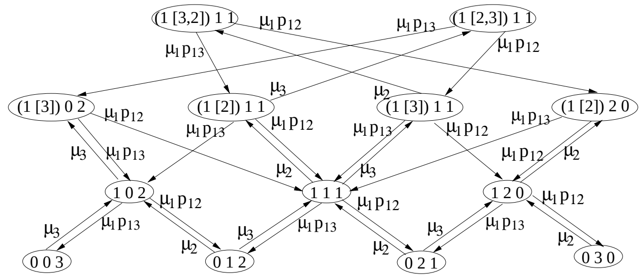
(b) State space for $N = 3$ and $F_1 = 2$ assuming queue 1 in phase 1. State notation is $((n_1, \boldsymbol{m}, k_1), n_2, n_3)$, $\boldsymbol{m} = \emptyset$ is omitted. Queue 3 states are highlighted.

Fig. 1. Example network with 2 infinite capacity exponential queues and a MAP queue with finite capacity

Examples. We now give examples of the state space underlying a queueing network model with BAS blocking. For the sake of simplicity, we omit the state component b_i since it can be simply derived, i.e., $b_i = 1$ if and only if $i \in \boldsymbol{m}$. We consider an example model with $M = 3$ queues[5] where queue 1 is a finite capacity station with MAP service, queues 2 and 3 have exponential service and infinite capacities. Figure 1(a) illustrates the model with routing probabilities. The exponential queues have rates μ_2 and μ_3, the MAP completes jobs in phase 1 with rate μ_1.

The underlying Markov process for the case with $N = 3$, $F_1 = 2$ and assuming queue 1 in phase 1, is shown in Figure 1(b). For ease of illustration, MAP phase transitions are omitted, thus this partition is similar to the state space where the service at the MAP is exponential with rate μ_1. We point to [Casale et al. 2010] for figures illustrating the effects of MAP phase changes in the state space of a queueing network model. Figure 1(b) classifies the activity of queue 3 into "active" ($n_3 > 0$ and $b_3 = 0$), "idle" ($n_3 = 0$ and $b_3 = 0$), or "blocked" ($n_3 > 0$ and $b_3 = 1$). This classification is useful to understand the different rates of departure from queue 3. The states where queue 3 is active are the only states that contribute to the departure transitions out of queue 3. The state $((2, [3], 1), 0, 1)$ in the blocked subspace considers the case where queue 3 is

[5]Throughout the paper, we use small models for the sake of illustration, however there is no hard limit on the maximum model size that can be analyzed by QRF. This in practice depends on the hardware available to run the solution algorithms and on the nature of the optimization algorithm used. For example, a parallel solver would make it possible to solve models with several tens of queues.



(a) State space for $N = 3$ and $F_1 = 1$ assuming queue 1 in phase 1. State notation is $((n_1, \boldsymbol{m}, k_1), n_2, n_3)$, $\boldsymbol{m} = \emptyset$ and all phases $k_1 = 1$ are omitted.

Fig. 2. Example model 1 when several queues are blocked

blocked ($\boldsymbol{m} = [3]$) since queue 3 had previously completed a job to be sent to queue 1 while this was full. As soon as queue 1 completes a job, two simultaneous transitions take place moving the current state to $((2, 1), 1, 0)$ in the idle subspace of queue 3 if the job completed by queue 1 is routed to queue 2. The current state becomes $((2, 1), 0, 1)$ in the active subspace of queue 3 if the completed job is routed to queue 3 which thus restarts immediately service after unblocking. These simultaneous transitions are a distinctive characteristic of the state space due to BAS blocking.

To further appreciate the complexity of bound analysis for the BAS state space, Figure 2 illustrates a case where two queues can be blocked. Observe the changes in the BAS state space level compared to Figure 1. Let us now assume $F_1 = 1$, infinite capacities for queues 2 and 3, and $N = 3$ jobs in the network. For simplicity of graphical representation, the phase k_1 is omitted being always equal to 1. Figure 2 shows the totally different structure of the state space. When the system is in the state $(1, 1, 1)$ all queues are active. If queue 2 completes a job, the current state becomes $((1, [2], 1), 1, 1)$ where queue 2 is blocked ($\boldsymbol{m} = [2]$). If from this state queue 3 completes a job, the transition leads to state $((1, [2, 3], 1), 1, 1)$, where *both* queues 2 and 3 are blocked ($\boldsymbol{m} = [2, 3]$). According to the FBFU unblocking rule, when queue 1 completes a job, queue 2 is unblocked first with a transition to the state $((1, [3], 1), 1, 1)$ if the completed job is routed to queue 2, or to state $((1, [3], 1), 0, 2)$ if the completed job is routed to queue 3. This illustrates transitions that do not exist in non-blocking queueing networks and thus which require specialized characterization for bounding purposes. To obtain such a characterization, we develop in Theorem 4.5 a new class of balance conditions that is able to describe the state space in Figure 2.

3.5. Allowed State Dependence

State dependence is supported under the following assumptions. First, any state dependent rate is in the most general case a function $q^{k'_i}(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$, for a given pair of source-target stations (i, j). Next, when there are one or more queues blocked by a finite capacity station f, service rate and outgoing routing probabilities of station f can depend only on $(n_f, k_f, \boldsymbol{m}_f)$. This is because upon characterizing a queue i blocked by f, we need to express n_i and the rate at which queue i unblocks. However, this rate is the same at which f frees capacity, thus if the latter depends on the state (n_f, k_f, n_j, k_j), $j \neq i \neq f$, our analysis requires to simultaneously characterize n_i, n_j, and n_f which would be impossible with the BQR marginals that can depend on at most two random variables. A special case arises when there exist a single queue i that can send and receive jobs to or from the finite capacity queue f; in this special case all stations can be state-dependent.

4. BQR MAPPING

Denote with $\pi(\boldsymbol{s})$ the equilibrium probability for state s $\in E_{BAS}$ in the BAS queueing network model. We formulate the quadratic reduction for the BAS case as follows. We consider the following marginal probability

$$\pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_f) \tag{1}$$

which is called *BQR marginal probability* and describes the joint state of queues i and j in phases k_i and k_j while the queues in the list $\boldsymbol{m}_f$ are blocked by a finite capacity queue f. When $\boldsymbol{m}_f = \emptyset$ no queue is blocked and this is possible also when $n_f = F_f$ before any job is routed to the finite capacity station. The BQR marginals are immediately obtained by summing $\pi(\boldsymbol{s})$ over the states with the given values of n_i, k_i, n_j, k_j, and $\boldsymbol{m}_f$.

The main advantage of the BQR marginals over the original state space representation is that they scale quadratically with the total population size, which greatly improves of the global balance equations. BQR marginals also explicit the blocking list $\boldsymbol{m}_f$. In the general case with multiple finite capacity stations f_i, $i = 1, 2, \ldots$, BQR marginals would then generalize as

$$\pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_{f_1}, \boldsymbol{m}_{f_2}, \ldots) \tag{2}$$

Since the number of feasible values for each vector $\boldsymbol{m}_i$ is $b!$ when b queues are blocked by i, it is clear that blocking stations should be used parsimoniously to avoid an excessive growth in the number of possible values for the $\boldsymbol{m}_i$ vectors. Since BAS depends on the order of blocking, it does not seem possible in general to come to computational simplifications if one wants to obtain an exact representation. However, one benefit of the QRF methodology is that the inequalities it generates do not necessarily need to be all included in the optimization program used for bounding purposes. Thus, one may consider reducing computational costs by using a QRF description only for part of the state space, possibly combining this with known information on the unspecified part (e.g., bounds on the state probabilities).

4.1. Performance Indexes

Common metrics such as utilization, throughput, response times, and queue-lengths can be immediately computed from the BQR marginals. For example, the utilization of queue i is

$$U_i = \sum_{(n_j,k_j,k_i,\boldsymbol{m}_f)} \sum_{n_i \geq 1} \pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$$

whereas the effective utilization that describes the activity of a queue *excluding its blocking time* is $E_i = \sum_{k_i=1}^{K_i} E_i^{k_i}$ where the effective utilization [Balsamo et al. 2001] of phase k_i is

$$E_i^{k_i} = \sum_{(n_j,k_j,\boldsymbol{m}_f)} \sum_{n_i \geq 1} \sum_{i \notin \boldsymbol{m}_f} \pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$$

Note that $E_i = U_i$ if and only if either $i = f$ or i cannot be blocked by the finite capacity queue f due to the network topology. The effective utilization takes into account the productive utilization of a queue, that is the period of time the queue is busy and it is not blocked, so it can produce useful work. Importantly, the throughput at a resource may be obtained by Little's law as the effective utilization divided by mean service time at that resource. Other metrics such as mean queue-lengths and blocking probabilities can be readily obtained once the BQR marginals have been computed.

4.2. Basic Characterization Results

The first basic characterization result for BQR marginals in a BAS setting follows by the equilibrium of the MAP service processes. During the period where queue i is actively serving a job, the MAP service process behaves at equilibrium in the same way as the MAP considered in isolation, since we are assuming that the queue is neither idle nor blocked. This observation leads to a balance between BQR marginals for different phases.

THEOREM 4.1. *The BQR marginals satisfy at equilibrium*

$$\sum_{j=1}^{M} \sum_{\substack{h=1 \\ h\neq k \text{ if } j=i}}^{K_i} \sum_{(n_j,k_j,\boldsymbol{m}_f)} \sum_{n_i\geq 1, i\notin \boldsymbol{m}_f} q_{i,j}^{k,h} \pi(n_i,k,n_j,k_j,\boldsymbol{m}_f)$$
$$= \sum_{j=1}^{M} \sum_{\substack{h=1 \\ h\neq k \text{ if } j=i}}^{K_i} \sum_{(n_j,k_j,\boldsymbol{m}_f)} \sum_{n_i\geq 1, i\notin \boldsymbol{m}_f} q_{i,j}^{h,k} \pi(n_i,h,n_j,k_j,\boldsymbol{m}_f), \quad (3)$$

for $i = 1,\ldots,M$ and $k = 1,\ldots,K_i$.

PROOF. Consider a partitioning of the state space into two subsets: $G_{i,k}$ where queue i is in phase k and its complementary set of states $\bar{G}_{i,k}$ where queue i is in any phase $h \neq k$. By basic properties of Markov processes, the equilibrium probability flux exchanged by $G_{i,k}$ and $\bar{G}_{i,k}$ at equilibrium must be balanced. However, this is only due to phase changes that occur in the MAP, with or without an associated departure from i. The left hand side of (3) represents phase changes moving the current phase from k to any h, whereas the right hand side is the probability flux due to phase changes that move the active phase into k. Note that the condition $h \neq k$ if $j = i$ ignores self-routing of jobs that do not change the active phase. □

Another characterization result follows by observing that the total population of jobs in each state of the underlying Markov chain sums to N. This property implies that the sum of the conditional first moment of queue lengths is constant.

THEOREM 4.2. *The BQR marginals for BAS blocking satisfy the following constraints*

$$\sum_{i=1}^{M} \sum_{n_i=1}^{F_i} \sum_{k_i=1}^{K_i} n_i \pi(n_i,k_i,n_j,k_j,\boldsymbol{m}_f) = N\pi(n_j,k_j,n_j,k_j,\boldsymbol{m}_f) \quad (4)$$

for all $j = 1,\ldots,M$, $n_j = 0,\ldots,F_j$, $k_j = 1,\ldots,K_j$, and for all lists of blocked queues $\boldsymbol{m}_f$.

PROOF. Since the sum of the total population in a state is constant, we can write

$$\sum_{\boldsymbol{s}\in S} \sum_{i=1}^{M} n_i \pi(\boldsymbol{s}) = N \sum_{\boldsymbol{s}\in S} \pi(\boldsymbol{s}) \quad (5)$$

for any partition of states $S \subseteq E_{BAS}$, where we omit $n_i = 0$ since the corresponding term in the left-hand side summation is zero. Define S as the set of states where the blocked queue list is $\boldsymbol{m}$ and queue j has population n_j in phase k_j, thus the right hand side becomes $N\pi(n_j, k_j, \boldsymbol{m}_f, n_j, k_j, \boldsymbol{m}_f)$. Denote by $\boldsymbol{s}_k$ the components of $\boldsymbol{s}$ different from n_i, k_i, n_j, k_j, $\boldsymbol{m}_f$. We can equivalently rewrite the above expression as

$$\sum_{i=1}^{M}\sum_{n_i=1}^{F_i}\sum_{k_i=1}^{K_i} n_i \sum_{\boldsymbol{s}_k} \pi(n_i, k_i, n_j, k_j, \boldsymbol{s}_k, \boldsymbol{m}_f) = N\pi(n_j, k_j, \boldsymbol{m}_f) \tag{6}$$

However, the inner summation on $\boldsymbol{s}_k$ gives the BQR marginal $\pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$ which proves the theorem. □

We remark that a higher-order extension of the above theorem holds as well, but it cannot be represented explicitly using the BQR marginals since a order-k formula requires the joint probability of k queue-length terms, and BQR can express such relations only for $k \leq 2$.

The next theorem can be seen as an extension of Theorem 4.2 as it defines a relation between the sum of mean queue-lengths of all queues and the utilization of queue i when a given queue j is in phase k_j.

THEOREM 4.3. *The sum of the mean queue-lengths of all queues conditioned on queue j being in phase k_j satisfies*

$$\sum_{w=1}^{M}\sum_{n_w=1}^{F_w}\sum_{k_w=1}^{K_w}\sum_{n_j=0}^{F_j}\sum_{\boldsymbol{m}_f} n_w \pi(n_w, k_w, n_j, k_j, \boldsymbol{m}_f) \geq N \sum_{n_i=1}^{F_i}\sum_{k_i=1}^{K_i}\sum_{n_j=0}^{F_j}\sum_{\boldsymbol{m}_f} \pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_f) \tag{7}$$

for all $1 \leq i \leq M$, $1 \leq j \leq M$, $1 \leq k_j \leq K_j$.

We point the interested reader to [De Nitto Personé et al. 2010, Thm. 4] for a complete derivation.

4.3. Marginal Balance Conditions

The theorems in the previous section provide a characterization of basic properties of utilization and queue-lengths in the BQR marginal representation. However, these properties loosely depend on the inter-dependencies between stations, such as the flows of jobs between queues and the rules of BAS blocking. Thus, they can constrain in a limited manner the uncertainty about the model's stationary distribution. A more precise characterization of BAS blocking and job flows is provided by the following marginal balance conditions. Such conditions express, using the BQR marginals, the probability flux balance resulting from cuts of the Markov chain separating states with a marginal population n_i from states where queue i has population $n_i + 1$.

THEOREM 4.4 (MARGINAL BALANCE). *The arrival flow of queue i when the local queue-length is of n_i jobs, $0 < n_i \leq F_i - 1$, is in equilibrium with the departure flow* *when the queue-length is n_i+1, i.e.,*

$$\sum_{\substack{j=1\\ j\neq i\\ j\neq f}}^{M}\sum_{n_j=1}^{F_j}\sum_{k=1}^{K_j}\sum_{h=1}^{K_j}\sum_{v=1}^{K_i}\sum_{\boldsymbol{m}_f:j\notin\boldsymbol{m}_f} q_{j,i}^{k,h}\pi(n_j,k,n_i,v,\boldsymbol{m}_f)+\sum_{n_f=1}^{F_f}\sum_{k=1}^{K_f}\sum_{h=1}^{K_f}\sum_{v=1}^{K_i}\sum_{\boldsymbol{m}_f:Head(\boldsymbol{m}_f)\neq i} q_{f,i}^{k,h}\pi(n_f,k,n_i,v,\boldsymbol{m}_f)$$

$$=\sum_{\substack{j=1\\ j\neq i\neq f}}^{M}\sum_{n_j=0}^{F_j}\sum_{k=1}^{K_i}\sum_{h=1}^{K_i}\sum_{v=1}^{K_j}\sum_{\boldsymbol{m}_f:i\notin\boldsymbol{m}_f} q_{i,j}^{k,h}\pi(n_i{+}1,k,n_j,v,\boldsymbol{m}_f)+\sum_{n_f=0}^{F_f-1}\sum_{k=1}^{K_i}\sum_{h=1}^{K_i}\sum_{v=1}^{K_f} q_{i,f}^{k,h}\pi(n_i{+}1,k,n_f,v,\emptyset)$$

$$+\sum_{\substack{w=1\\ w\neq f\\ w\neq i}}^{M}\sum_{k=1}^{K_i}\sum_{v=1}^{K_f}\sum_{p=1}^{K_f}\sum_{\boldsymbol{m}_f:Head(\boldsymbol{m}_f)=i} q_{f,w}^{v,p}\pi(n_i+1,k,F_f,v,\boldsymbol{m}_f), \quad (8)$$

for $i=1,\ldots,M$ and $i\neq f$. When $i=f$ the expression becomes

$$\sum_{\substack{j=1\\ j\neq f}}^{M}\sum_{n_j=1}^{F_j}\sum_{k=1}^{K_j}\sum_{h=1}^{K_j}\sum_{v=1}^{K_f} q_{j,f}^{k,h}\pi(n_j,k,n_f,v,\emptyset)=\sum_{\substack{j=1\\ j\neq f}}^{M}\sum_{n_j=0}^{F_j-1}\sum_{k=1}^{K_f}\sum_{h=1}^{K_f}\sum_{v=1}^{K_j} q_{f,j}^{k,h}\pi(n_j,v,n_f{+}1,k,\emptyset), \quad (9)$$

for each $n_f=0,\ldots,F_f-1$. Furthermore, for $n_i=0$ (equiv. $n_f=0$ when $i=f$) the above balances admit a stronger form where they hold true for each phase $k=1,\ldots,K_i$ considered in isolation.

PROOF. Under the assumptions made, the probability flux exchanged between states with n_i and with n_i+1 jobs in queue i must be in equilibrium at steady state. This is because we do not consider batch arrivals or departures, resets, or other types of events that modify queue lengths by more than a single unit at a time.

It is then sufficient to show that all the events leading to probability fluxes between these two partitions can be written using BQR marginals. Since the expression of these marginals are given in the theorem statement, we focus here on enumerating all the possible transitions between states with n_i and n_i+1 jobs.

First, consider $i\neq f$, we have the following events:

— Event LHS.1: The left hand side of equation (8) includes all the departures from any non-empty queue j (i.e., $n_j>0$) towards queue i. After these departures, the population of i becomes n_i+1, except in the case where $j=f$ and $Head(\boldsymbol{m}_f)=i$, i.e., queue i is unblocked by the departure from f. In this case, queue i is waiting for free space in f and, because of the simultaneous transitions, the population in i remains equal to n_i.

— Event LHS.2: As a consequence of the above discussion, when $j=f$, the condition $Head(\boldsymbol{m}_f)\neq i$ must be also true, this corresponds to the second term of the left side of (8).

The right hand side of the equation considers all departures from queue i with population equal to n_i+1. After these departures, the population of queue i becomes n_i. These departures include the following events:

— Event RHS.1: Transitions from i towards any queue j, $j\neq i\neq f$. Note that these transitions are always possible because queue j does not have finite capacity; this is the first term of the right side.

— Event RHS.2: The second term represents transitions from queue i to the finite capacity queue f. These require i to be unblocked and $n_f<F_f$ since otherwise i would

block and its population would not decrease from $n_i + 1$ to n_i. As a consequence $\boldsymbol{m}_f = \emptyset$.

— Event RHS.3: Transitions from node f to any other node w, $w \neq f$, $w \neq i$ when f is full and node i is the first blocked one, that is $Head(\boldsymbol{m}) = i$. These transitions trigger a simultaneous transition from queue i, thus decrease its population to n_i. This is the third term on the right side of (8).

Equation (9) can be similarly proved and has a simpler form since queue f cannot block. □

The above equations show several differences compared to marginal balances for queueing network models with infinite capacity [Casale et al. 2010]. In addition to the obvious condition on the stations contributing to the throughput flow being active, i.e., $j \notin \boldsymbol{m}_f$, the last term in the right hand side of (8) describes the departures from station i following an unblocking event in station f that frees a buffer slot which i has priority to use since $Head(\boldsymbol{m}) = i$. Thus, this term captures the fundamental behavior of a departure from f that unblocks queue i. Interestingly, the departure flow from i is regulated in this case by the rate of departure of f, thus showing a case of non-product-form behavior where the throughput of a station directly depends on the rate of another station.

We now introduce a new class of balance conditions that describe the throughput rate while queue f is full. These balances are related to cuts of the Markov chain underlying the queueing network that separate the states shown in Figure 2 into partitions where $\boldsymbol{m}_f$ has different length b, i.e., different number of blocked queues. Intuitively, as queue f enters into an extended period of time during which it remains full, the queues feeding f progressively block leading to changes in the composition of the list $\boldsymbol{m}$. The next theorem summarizes the balance between the rate of change of m due to queue blocking and the corresponding rate of unblocking events due to departures from f.

THEOREM 4.5. *The BQR marginals for states where the finite capacity queue f is full satisfy*

$$\sum_{\substack{j=1\\j\neq f}}^{M}\sum_{n_j=1}^{F_j}\sum_{k=1}^{K_j}\sum_{h=1}^{K_j}\sum_{v=1}^{K_f}\sum_{\substack{\boldsymbol{m}_f:j\notin\boldsymbol{m}_f\\ \sum_i b_i=b}} q^{k,h}_{j,f}\pi(n_j,k,F_f,v,\boldsymbol{m}_f) = \sum_{\substack{j=1\\j\neq f}}^{M}\sum_{n_j=0}^{F_j}\sum_{k=1}^{K_f}\sum_{h=1}^{K_f}\sum_{v=1}^{K_j}\sum_{\substack{\boldsymbol{m}_f:\\ \sum_i b_i=b+1}} q^{k,h}_{f,j}\pi(n_j,v,F_f,k,\boldsymbol{m}_f), \tag{10}$$

for all number of blocked queues $b = 0, \ldots, B-2$. When the number of blocked queues is $b = B-1$ we can write the stronger condition

$$\sum_{\substack{j=1\\j\neq f}}^{M}\sum_{n_j=1}^{F_j}\sum_{k=1}^{K_j}\sum_{h=1}^{K_j}\sum_{v=1}^{K_f} q^{k,h}_{j,f}\pi(n_j,k,F_f,v,\boldsymbol{m}^j_f) = \sum_{\substack{j=1\\j\neq f}}^{M}\sum_{\substack{w=1\\w\neq i}}^{M}\sum_{k=1}^{K_f}\sum_{v=1}^{K_f}\sum_{\substack{\boldsymbol{m}^j_f:\\ \sum_i b_i=b+1}} q^{k,v}_{f,w}\pi(F_f,k,F_f,k,Add(\boldsymbol{m}^j_f,j)), \tag{11}$$

where $\boldsymbol{m}^j_f$ is any blocking list with $b = B-1$ blocked queues satisfying $j \notin \boldsymbol{m}^j_f$, and $Add(\boldsymbol{m}^j_f, j)$ is a list obtained by adding queue j at the tail of $\boldsymbol{m}^j_f$.

PROOF. Consider a partitioning of the state space E_{BAS} into the following two subsets: H_b where there are up to $b \geq 0$ blocked queues and H_{b+1}, where the finite capacity queue is full and there are $b+1$ or more blocked queues on f. The left hand side of (10) represents the probability flux flowing through the state space cut associated to departures from station j to f that block j thus adding an entry at the end of $\boldsymbol{m}_f$. Conversely, the right hand side of (10) is the probability flux of departures from f such that at least a station $j \neq f$ gets unblocked thus reducing by one entry the list $\boldsymbol{m}_f$. Since no more than one queue gets blocked or unblocked at a time, it follows that the balance fully characterizes the probability flux balance across the cut that separates H_b from H_{b+1} which proves the equation.

Equation (11) considers the case where only a single queue j (in addition to f) is left unblocked, for any feasible choice of j. In this case, we know that only a departure event from j can increase the blocked queue list $\boldsymbol{m}_f^j$ to $Add(\boldsymbol{m}_f^j, j)$. Thus, we can apply the same argument used to prove (10) by focusing on the cut that separates in E_{BAS} the partition having blocking list $Add(\boldsymbol{m}_f^j, j)$ from the rest of the chain, which completes the proof. □

5. BOUNDABLE APPROXIMATIONS

The fundamental idea behind the proposed approximations and bounds is to use the BQR marginals to formulate an educated guess of the values of the BQR marginals. We here describe our methodology for BAS networks, while the application to RS-RD blocking follows easily by considering $\boldsymbol{m} = \emptyset$.

To determine an approximate marginal distribution for the model, we assume the values $\pi(n_i, k_i, n_j, k_j, \boldsymbol{m})$ as unknowns in an optimization program $\mathcal{O}$. This optimization program takes the form

$$\begin{aligned} \mathcal{O}: \qquad \min f_{obj}(\boldsymbol{\pi}^G) \qquad & \text{s.t.} \\ \boldsymbol{A}\boldsymbol{\pi}^G \leq \boldsymbol{b} & \\ \boldsymbol{C}\boldsymbol{\pi}^G \leq \boldsymbol{d} & \end{aligned}$$

where $\boldsymbol{\pi}^G$ is the vector of the current guesses for all the BQR marginals $\pi(n_i, k_i, n_j, k_j, \boldsymbol{m})$, f_{obj} is a (possibly nonlinear) objective function to be optimized, and the constraints are of two types. A first group of constraints, $\boldsymbol{A}\boldsymbol{\pi}^G \leq \boldsymbol{b}$, is the set of all equations and inequalities developed in the BAS (or RS-RD) characterizations, including the specialized marginal balances for $n_i = 0$. The coefficient matrix $\boldsymbol{A}$ and the vector of known terms $\boldsymbol{b}$ are readily obtained by the corresponding coefficients and known terms in the equations. Notice that such equations are all linear constraints, mainly equalities that can be easily rewritten as a pair of inequalities after standard transformations[6]. A second group of linear constraints, $\mathbf{C}\boldsymbol{\pi}^G \leq \boldsymbol{d}$, imposes obvious conditions that describe in the optimization program the feasible values of the terms $\pi^G(n_i, k_i, n_j, k_j, \boldsymbol{m}) \in \boldsymbol{\pi}^G$ in order to specify a valid BQR marginal distribution. These constraints impose, for instance, that the unknowns of the linear program are probabilities, hence numbers ranging in $[0, 1]$, or that a queue can only be in a single state at a time hence, e.g., $\pi^G(n_i, k_i, n_i + c, k_i, \boldsymbol{m}) = 0$, $\forall c \neq 0$. A summary of these basic conditions is given in Table II.

Let $\boldsymbol{\pi}_{opt}^G$ be the guess $\boldsymbol{\pi}^G$ which provides the optimal value for the objective function f_{obj}. The crucial property of the optimization program $\mathcal{O}$ is that its constraints are satisfied by the exact BQR marginal distribution $\boldsymbol{\pi}$. It then follows that the exact solution $f_{obj}(\boldsymbol{\pi})$ is always a *feasible* solution for the optimization program $\mathcal{O}$, although it may not be necessarily the optimal one $f_{obj}(\boldsymbol{\pi}_{opt}^G)$. This property leads to the following approximation and bounding techniques.

[6] We point the reader to a standard textbook on linear programming for further details.

5.1. BQR Bounds

First, suppose that f_{obj} defines a performance metric of interest, such as the utilization of station i

$$f_{obj}(\boldsymbol{\pi}^G) = \sum_{k_i=1}^{K_i} \sum_{n_i \geq 1} \pi^G(n_i, k_i, n_i, k_i, \boldsymbol{m})$$

or its average queue-length

$$f_{obj}(\boldsymbol{\pi}^G) = \sum_{k_i=1}^{K_i} \sum_{n_i \geq 1} n_i \pi^G(n_i, k_i, n_i, k_i, \boldsymbol{m})$$

Then, by construction, minimizing $\mathcal{O}$ returns a lower bound $f_{obj}(\boldsymbol{\pi}^G_{opt}) = \min f_{obj}(\boldsymbol{\pi}^G) \leq f_{obj}(\boldsymbol{\pi})$, since $\boldsymbol{\pi}^G = \boldsymbol{\pi}$ is a feasible solution of the optimization program. Similarly, solving $\mathcal{O}$ as a maximization problem returns an upper bound $f_{obj}(\boldsymbol{\pi}^G_{opt}) = \max f_{obj}(\boldsymbol{\pi}^G) \geq f_{obj}(\boldsymbol{\pi})$. Noting that utilizations and queue-lengths are linear functions of $\boldsymbol{\pi}^G$, it then follows that $\mathcal{O}$ can be solved efficiently as a linear optimization program. Such a solution provides upper and lower bounds on the performance metrics of a queueing network model. Note that such values are guaranteed to be bounds by construction if the optimizer returns a *global* optimum for $\mathcal{O}$, as it is always the case for linear programs. The approach similarly applies to other metrics that can be expressed as a function of the BQR marginals.

6. NUMERICAL VALIDATION

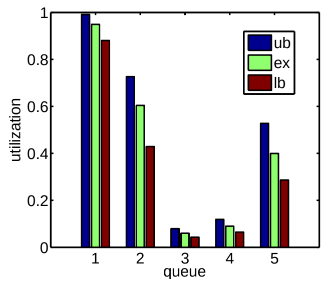
We illustrate the accuracy of the BAS and RS-RD bounds on a set of case studies having different level of complexities, number of queues, and network topology. Throughout the experiments, we use a combination of exponential service processes and nonrenewal autocorrelated MAPs. We use the GLPK linear programming solver to compute bounds. For simplicity of comparison, we always use a short-range dependent MAP process with two-phases having representation [Bolch et al. 2006]

$$D_0 = \begin{bmatrix} -1.016212022108574 & 0 \\ 0 & -0.015702871508448 \end{bmatrix} \tag{12}$$

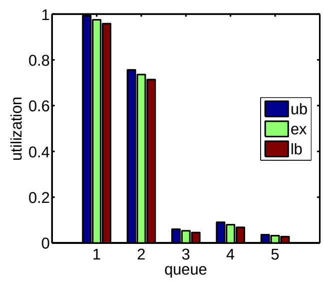
$$D_1 = \begin{bmatrix} 1.016186165025678 & 0.000025857082896 \\ 0.001569887597955 & 0.014132983910493 \end{bmatrix} \tag{13}$$

This yields a process with moments $E[X] = 1$, $E[X^2] = 4$, $E[X^3] = 400$, and positive autocorrelation function $\rho_k = \frac{1}{3}(\frac{9}{10})^k$ such that $\rho_1 = 0.300$, $\rho_2 = 0.270$, $\rho_3 = 0.243, \ldots$. On a laptop computer, the hardest case study execution times were less than 5 seconds for the BQR bounds, about 300 seconds for the nonlinear programs used for point approximation (maximum entropy and minimum mutual information). Note that we used a single CPU core, however nonlinear solvers running on multi-core machines are usually 8-10 times faster, thus the nonlinear solution can be significantly accelerated.

Due to limited space, in the following we present a case study on a central server model with state dependent routing. We present further results in the online Supplement where the interested reader can find several case studies and also a comparison with a different approximation technique proposed in the literature.



(a) Queue 5 rate: 0.30



(b) Queue 5 rate: 3.33

Fig. 3. QRF Bounds under state-dependent routing.

6.1. Bounding Problem: State Dependent Routing

In this problem we wish to study a state dependent model. The network has a central-server topology with $M = 5$, $N = 5$, and routing matrix

$$P^{-} = \begin{bmatrix} 0 & 0.1000 & 0.2000 & 0.3000 & 0.4000 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \end{bmatrix}$$

when $n_1 \leq 2$ and

$$P^{+} = \begin{bmatrix} 0 & 1.0000 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \end{bmatrix}$$

otherwise. Furthermore, for $j = 2, \ldots, 5$, we assume that the rate of jobs processed at queue i to be routed at j is multiplied by $(1+n_j)^{-1}$, thus the routing out of queue 1 will tend to prioritize the destination with the least number of jobs. The model has been chosen in this way to illustrate a model with a sharp state-dependent behavior where the steady state solution is significantly different from the case without dependence.

Station 1 has MAP service (12), all other stations are exponential. All mean service rates are equal to 1, except for queue 5 where is 0.3. Station 5 is also the only finite capacity queue with capacity $F_5 = 3$. Results are shown in Figure 3(a). The bounds are quite accurate in capturing the bottleneck queue utilization. The solution is accurate for stations 3 and 4 as well. For stations 2 and 5, the analysis is more difficult but still with acceptable accuracy. However, the results are consistent and quite centered around the exact value, allowing to obtain good approximate estimates by averaging the two bounds.

Figure 3(b) illustrates results for a similar state-dependent model where we set the mean service rate for queue 5 to 3.333 and we remove the scaling factors $(1+n_j)^{-1}$, but keep the state dependent condition for the choice between P^{-} and P^{+}. For this model, the finite capacity station is considerably fast, thus less probability mass is put on the states where queue 1 is blocked. In this case, the approximation is extremely accurate.

7. APPROXIMATE MODEL SOLUTION

The second main application of the optimization program $\mathcal{O}$ is the approximation of the values of the BQR marginal probabilities. In this section, we define objective functions that obtain accurate approximations, noticeably also on cases where the BQR bounds are not tight. We here introduce two approximation techniques: a *maximum entropy method* (MEM) and a new approximation based on a *minimal mutual information* (MMI) principle that we introduce in this paper. It is important to remark that, since the BQR bounds can always be generated regardless of these approximations, the gap between upper and lower bounds provides an independent assessment on the maximum error for MEM and MMI and makes the errors in these approximations always bounded. Furthermore, the objective functions are non-linear, hence one should consider a local optimum obtained by a nonlinear solver. In the case of entropy maximization, since entropy is convex, interior point methods can efficiently find the unique global optimum.

7.1. Maximum Entropy Method (MEM)

MEM searches for a set of BQR marginals that maximizes the information content of the distribution as defined by the entropy function H. To simplify notation, for the rest of this section let $\pi^G(n_i, n_j) \equiv \pi^G(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$. MEM optimizes in $\mathcal{O}$ the objective function

$$\max H = \max \left(- \sum_{n_i, n_j, k_i, k_j, \boldsymbol{m}_f} \pi^G(n_i, n_j) \log \pi^G(n_i, n_j) \right)$$

The values of performance indexes such as utilizations and queue-lengths are then obtained directly from the BQR marginal distribution that maximizes H. The rationale behind a maximum entropy solution is that it is known to be exact in a number of queueing models, noticeably in exponential single-class closed queueing networks [Ferdinand 1970]. Notice that a well-known maximum entropy method for queueing networks has already been developed in [Kouvatsos 1993] based on the analysis of the $GI/GI/1$ queue. However, the MEM technique we propose differs substantially from the one in [Kouvatsos 1993]. First, the method is able for the first time to consider the state of *all* the queues in the network simultaneously, instead of recursively evaluating queues one at a time as in [Kouvatsos 1993]. Importantly, our technique is also able to consider the impact of autocorrelation in job flows introduced by MAPs, which is ignored in the analysis of the $GI/GI/1$ queue. Indeed, this is a critical aspect of a queueing network model that cannot be ignored, being responsible for dynamic bottleneck switch effects, even at equilibrium, that significantly affect the model solution [Casale and Smirni 2009]. Finally, and perhaps most importantly, our MEM solution is subject to satisfying the very large set of constraints developed for BAS and RS-RD, whereas the one in [Kouvatsos 1993] considers a small subset of such constraints. Therefore, our approximation is more heavily constrained. Instead, the main limitation of the proposed MEM compared to the one in [Kouvatsos 1993] is that our method requires numerical optimization, whereas [Kouvatsos 1993] is based on simple closed-form formulas.

7.2. Minimum Mutual Information (MMI)

In addition to the MEM method, we introduce the MMI criterion as a new technique for approximating an unknown probability distribution of a queueing network. For a

BQR marginal probability distribution, MMI considers the following objective function

$$\min\left(\sum_{n_i,n_j,k_i,k_j,\boldsymbol{m}_f} \pi^G(n_i,n_j)\log\frac{\pi^G(n_i,n_j)}{\pi^G(n_i,n_i)\pi^G(n_j,n_j)}\right)$$

where the argument of the minimization is the mutual information of $\pi^G(n_i,n_j)$. This quantity characterizes how much the knowledge about n_i reduces our uncertainty about n_j. Noting that for a product-form model the knowledge of the state of a queue provides few insights on the state of the other stations, for instance in a closed model it only provides an upper bound $n_j \leq N - n_i$ that becomes progressively looser as N and M increase. This provides some intuition that the MMI solution may be interpreted as a product-form-type approximation for a queueing network model and its minimization corresponds to a distribution describing the situation in which n_i and n_j are maximally independent. Clearly, in networks with blocking the mutual information is not in general minimal, since blocking yields a strong dependence between the behavior of two (or even more than two) queues. However, the fundamental assumption of the proposed method is that the blocking is already characterized by our BQR marginal balances, hence MMI deals only with allocating the portion of the probability mass that remains unconstrained. We illustrate this concept below in a "toy" example. Notice also that the MMI approach is expected to be accurate especially in heavy load, where closed networks progressively approach the behavior of open models due to the formation of bottleneck stations whose service process, being continuously busy, acts similarly to an exogenous arrival process for the rest of the network. Open networks are typically less inter-dependent than their closed counterparts, thus we expect MMI to perform better in heavy load. It should be noted that, differently from MEM, the mutual information is a non-convex objective, thus the optimization program will be in general non-convex.

7.3. Illustrating Example

To better understand the properties of MEM and MMI, consider the following illustrating example. We use the MINOS solver for nonlinear programs required to evaluate the MEM and MMI approximations. The model is composed of three queues with exponential service rate $\mu_1 = \mu_2 = 1$, $\mu_3 = 2$. The routing matrix is

$$P = \begin{bmatrix} 0 & 0.50 & 0.50 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{14}$$

which is a special case for the topology shown in Figure 1. Buffer capacities are $F_1 = 1$, $F_2 = F_3 = N$, with $N = 3$ being the job population; the blocking mechanism is RS-RD. Despite its apparent simplicity, for this model the BQR bounds provide the following estimates of upper bounds (U_k^{max}) and lower bounds (U_k^{min}) on the exact utilizations (U_k) of queue k:

	queue 1	queue 2	queue 3
U_k^{max}	0.5000	0.7500	0.9524
U_k	0.4828	0.4483	0.8966
U_k^{min}	0.4762	0.3333	0.7500

In this example, the utilization of queue 2 is loosely captured by the BQR bounds that leave a gap of about 42% between the upper and lower limits. In this example, for the upper bound

$$\pi^G_{opt}(n_1=1,k_1=1,n_2=1,k_2=1) = 0.5000, \pi^G_{opt}(n_1=1,k_1=1,n_3=1,k_3=1) = 0.1905$$

and for the lower bound

$$\pi^G_{opt}(n_1 = 1, k_1 = 1, n_2 = 1, k_2 = 1) = 0.0, \pi^G_{opt}(n_1 = 1, k_1 = 1, n_3 = 1, k_3 = 1) = 0.0$$

Due to the fact that this model has just 3 jobs, it is possible to see that the two marginals both describe the probability of the same state $(n_1 = 1, n_2 = 1, n_3 = 1, k_1 = 1, k_2 = 1, k_3 = 1)$ in the original queueing network. Thus, in the original model

$$\pi(n_1 = 1, k_1 = 1, n_2 = 1, k_2 = 1) = \pi(n_1 = 1, k_1 = 1, n_3 = 1, k_3 = 1) = 0.1379.$$

which illustrates the error incurred by using QRF on this model. To correct this problem, one has to add to $\mathcal{O}$ the following additional constraint

$$\pi(n_j, k_j, n_i, k_i) = \pi(n_j, k_j, n_t = N - n_j - n_i, k_t),$$

for all choices of the stations $i \neq j \neq t$ and their states. This provides the optimal solution $\boldsymbol{\pi}^G_{opt} = \boldsymbol{\pi}$. This imposes that, in a model with $M = 3$ queues, there are at most two degrees of freedom in assigning the populations n_i and n_j at the queues, since the population at the last queue will be automatically set to $n_t = N - n_i - n_j$. This constraint is obvious but its integration in the BQR marginal characterization requires in general a cubic number of equations for a model with $M = 3$ which is not consistent with the approach that we have pursued; furthermore, for a model with $M \geq 4$ these constraints cannot be imposed using the BQR marginals, since one would need to express the state of $M - 1$ queues simultaneously. This example highlights some consequences of the structural limitation of BQR marginals; this limitation is that they cannot represent correctly the allocations of jobs (or the active phases) on more than two queues simultaneously.

We have then obtained the MEM and MMI solutions for the above model and found them as follows

	queue 1	queue 2	queue 3
U_k^{mem}	0.4887	0.5515	0.8464
U_k^{mmi}	0.4818	0.4316	0.9046
U_k	0.4828	0.4483	0.8966

that are much closer to the exact distribution that the BQR bound solution. Further, we have now for MEM

$$\pi^G_{mem}(n_1 = 1, k_1 = 1, n_2 = 1, k_2 = 1) = 0.2618, \quad (15)$$
$$\pi^G_{mem}(n_1 = 1, k_1 = 1, n_3 = 1, k_3 = 1) = 0.0907, \quad (16)$$

and for MMI

$$\pi^G_{mmi}(n_1 = 1, k_1 = 1, n_2 = 1, k_2 = 1) = 0.1180, \quad (17)$$
$$\pi^G_{mmi}(n_1 = 1, k_1 = 1, n_3 = 1, k_3 = 1) = 0.1455, \quad (18)$$

which provide a substantial consistency improvement compared to the BQR bounds, especially for the novel MMI method.

In the following, we apply the proposed approximation solution to a full mesh topology with BAS blocking. We refer to the online Supplement for a case study considering RS-RD blocking and a case study for a central-server topology, quite similar to the one used for the Bounding Problem in Section 6.1.

Table II. Constraints imposing basic properties of $\pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_f)$ included in the optimization programs

$\pi(n_i, k_i, n_j, k_j, \boldsymbol{m}_f) \geq 0,$	$\forall_{i=1}^{M} \forall_{n_i=0}^{F_i} \forall_{k_i=1}^{K_i} \forall_{j=1}^{M} \forall_{n_j=0}^{F_i} \forall_{k_j=1}^{K_j} \forall_{\boldsymbol{m}_f}$
$\sum_{n_j=0}^{F_j} \sum_{k_j=1}^{K_j} \sum_{\boldsymbol{m}_f} \pi(n_j, k_j, n_j, k_j, \boldsymbol{m}_f) = 1,$	$\forall_{j=1}^{M}$
$\pi(n_j, k, n_j, h, \boldsymbol{m}_f) = 0,$	$\forall_{j=1}^{M} \forall_{k=1}^{K_j} \forall_{h=1,h\neq k}^{K_j} \forall_{n_j=0}^{F_j} \forall_{\boldsymbol{m}_f}$
$\pi(n_j, k, n'_j, h, \boldsymbol{m}_f) = 0,$	$\forall_{j=1}^{M} \forall_{k=1}^{K_j} \forall_{n_j=0}^{F_j} \forall_{h=1}^{K_j} \forall_{n'_j=0, n'_j \neq n_j}^{F_j} \forall_{\boldsymbol{m}_f}$
$\pi(n_j, k, n_i, h, \boldsymbol{m}_f) = 0$	$\forall_{j=1}^{M} \forall_{k=1}^{K_j} \forall_{n_j=0}^{F_j} \forall_{i=1,i\neq j}^{M} \forall_{h=1}^{K_i} \forall_{n_i=N-n_j+1}^{F_i} \forall_{\boldsymbol{m}_f}$
$\sum_{n_f=0}^{F_f-1} \sum_{h=1}^{K_f} \pi(n_j, k, n_f, h, \boldsymbol{m}_f) = 0,$	$\forall_{j=1,f\neq j}^{M} \forall_{k=1}^{K_j} \forall_{n_j=0}^{F_j} \forall_{\boldsymbol{m}_f : \boldsymbol{m}_f \neq \emptyset}$
$\pi(n_j = 0, k, n_i, h, \boldsymbol{m}_f) = 0,$	$\forall_{j=1}^{M} \forall_{k=1}^{K_j} \forall_{i=1}^{M} \forall_{h=1}^{K_i} \forall_{n_i=0}^{F_i} \forall_{\boldsymbol{m}_f : j \in \boldsymbol{m}_f}$
$\pi(n_j, k, n_i, h, \boldsymbol{m}_f) = 0,$	$\forall_{j=1}^{M} \forall_{k=1}^{K_j} \forall_{n_j=F_j+1}^{N} \forall_{i=1}^{M} \forall_{h=1}^{K_i} \forall_{n_i=0}^{F_i} \forall_{\boldsymbol{m}_f}$
$\pi(n_j, k, n_i, h, \boldsymbol{m}_f) = 0$	$\forall_{j=1}^{M} \forall_{k=1}^{K_j} \forall_{n_j=1}^{F_j} \sum_{\substack{i=1 \\ i\neq j\neq f}}^{M} \forall_{h=1}^{K_i} \forall_{n_i=N-nj-F_f+1}^{F_i} \forall_{\boldsymbol{m}_f : Head(\boldsymbol{m}_f)=j}$
$\pi(n_j, k, n_i, h, \boldsymbol{m}_f) = \pi(n_i, h, n_j, k, \boldsymbol{m}_f),$	$\forall_{j=1}^{M} \forall_{n_j=0}^{F_j} \forall_{k=1}^{K_j} \forall_{i=1}^{M} \forall_{n_i=0}^{F_i} \forall_{h=1}^{K_i} \forall_{\boldsymbol{m}_f}$
$\pi(n_j, k, n_j, k, \boldsymbol{m}_f) = \sum_{n_i=0}^{N-n_j} \sum_{h=1}^{K_i} \pi(n_j, k, n_i, h, \boldsymbol{m}_f),$	$\forall_{j=1,j\neq i}^{M} \forall_{k=1}^{K_j} \forall_{n_j=0}^{F_j} \forall_{i=1}^{M} \forall_{\boldsymbol{m}_f}$
$\pi(n_j, k, n_j, k, \boldsymbol{m}_f) = 0,$	$\forall_{j=1}^{M} \forall_{k=1}^{K_j} \forall_{n_j=0}^{F_j} : N - n_j > \sum_{\substack{y=1 \\ y\neq j}}^{M} F_y$
$\pi(n_j, k, n_i, h, \boldsymbol{m}_f) = 0,$	$\forall_{i}^{M} \forall_{j=1,j\neq i}^{M} \forall_{k=1}^{K_j} \forall_{n_j=0}^{F_j} \forall_{h=1}^{K_i} \forall_{n_i=0}^{F_i} : N - n_j - n_i > \sum_{\substack{y=1 \\ y\neq j\neq i}}^{M} F_y$

7.4. Approximation Problem: BAS Blocking

Let us consider a full mesh topology model with $M = 5$ queues, BAS blocking and routing matrix

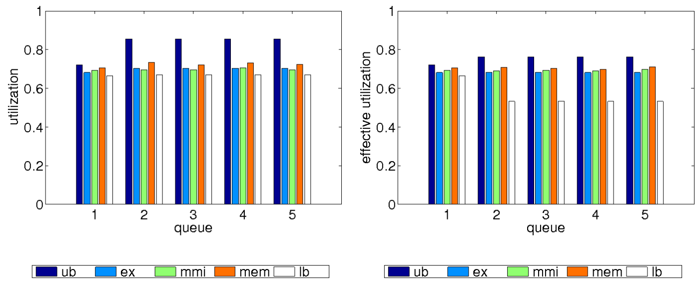
$$P = \begin{bmatrix} 0 & 0.2500 & 0.2500 & 0.2500 & 0.2500 \\ 0.2500 & 0 & 0.2500 & 0.2500 & 0.2500 \\ 0.2500 & 0.2500 & 0 & 0.2500 & 0.2500 \\ 0.2500 & 0.2500 & 0.2500 & 0 & 0.2500 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 & 0 \end{bmatrix}$$

Furthermore, station capacities are $F_1 = 5$, $F_2 = F_3 = F_4 = F_5 = N$ so that only station 1 has finite capacity. The population is $N = 10$ jobs. Hence, all stations can be blocked except for station 1. Service processes are identical short-range dependent MAPs given in (12). The results in Figure 4 indicate that the bounds are very effective in capturing the performance of the finite capacity queue 1, while more uncertainty is left on queues $2-5$ where the gap between upper and lower bounds is approximately up to 20%. In spite of such uncertainty, MEM and MMI again find very accurate results, again within a few percent of the exact results, with MMI again being slightly better than MEM. This is a relevant result, since despite its apparent simplicity, the number of possible combinations of $\boldsymbol{m}_f$ vectors is 64 for each state in which queue 1 is full, which is significant. Hence, this experiment suggests that the MEM and MMI approximation are effective also on cases where the portion of the state-space due to the BAS precedence constraints is non-negligible.

8. APPLICABILITY

Having shown in the previous sections the accuracy of QRF, we now investigate the computational requirements of the method, illustrating cases where the QRF methodology is superior to both exact analysis and simulation.

The analysis of complex networks with blocking may be performed either by simulation or by state-based methods. QRF belongs to the latter group and represents an alternative to exact analysis of the underlying Markov process, which requires the specification of an infinitesimal generator. As the size of the model is increased,



(a) Utilization (b) Effective Utilization

Fig. 4. Approximation Problem - A model with BAS blocking and mesh topology (lb = lower bound, ub = upper bound, ex = exact)

both simulation and state-based methods will feature different scalability and accuracy properties. It is unavoidable for state-based methods to hit a computational bottleneck earlier than simulation. However, methods such as hierarchical modeling and decomposition can be used in these cases to make state-based methods competitive to simulation. Although we did not explore this case in the paper, hierarchical modeling appears to be applicable. Instead, decomposition may be difficult to apply since it would break the equality constraints for the probability mass on which QRF relies.

Focusing on models that can be analyzed by QRF, the specific advantages of this method compared to simulation and exact analysis become apparent when multiple models are evaluated to optimize a goal function, as routinely done in design exploration and control. In these scenarios, simulation is expensive to apply to compare hundreds or thousands of alternatives, since each simulation requires at least a few seconds. Similarly, the cost of exact analysis tend to be prohibitive unless the model is very small. Conversely, QRF offers a major computational advantage since efficient techniques are available in ordinary linear programming solvers to re-optimize a linear program after a perturbation. This allows QRF to offer greater computational efficiency than simulation when evaluation a sequence of models.

For example, in QRF bounds when one solves the model for upper or lower performance bounds using the simplex algorithm, it is possible to save the basis at optimality. This provides an advanced basis to start the re-optimization, thus bypassing the phase-I of the simplex algorithm, and provides a feasible solution that is typically close to the new optimal solution. In practice, re-optimization often requires just a very small number of iterations compared to solving the initial problem. Furthermore, the shadow prices of the initial problem can be used to bound the optimal value of the new problem without the need to run an optimization routine.

To illustrate these properties, we compare in an example problem simulation (sim), exact analysis (exact/CTMC), and the QRF method. We here considered in each problem a parallel network with $N = 32$ jobs, $M-1$ queues, fed by a queue with finite capacity $F = 16$ and RS-RD blocking. Service times are exponentially distributed with rates all equal to 1 in the parallel queues and equal to 100 in the finite capacity queue. Routing to the parallel queues is done with identical probabilities and completed jobs return to the finite capacity queue. Table III illustrates the results. Results obtained with the AMPL/Gurobi 2.0 solver for QRF running on a quad-core machine with 2.6GHz AMD Opteron 2218; simulation results are obtained by the Java Modelling Tools simulator [Bertoli et al. 2009] to compute utilizations and mean queue-lengths in the model, thus

Table III. Comparative analysis - Single model

N	M	QRF time [s]	QRF cols	QRF rows	QRF ub	QRF lb	sim time [s]	exact/CTMC order	exact/CTMC time [s]
32	4	1	3086	507	0.9411	0.9406	14	5729	20
32	8	46	30248	2412	0.8421	0.8407	28	timeout	>1800
32	12	176	35965	4806	0.7619	0.7601	42	timeout	>1800
32	16	286	65807	8526	0.6956	0.6935	58	timeout	>1800

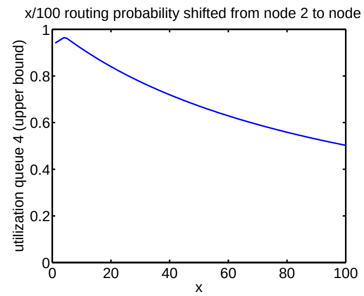


Fig. 5. Comparative analysis - Multiple models. Sensitivity of queue 4 utilization for $M = 4$. QRF generates the plot in $32s$. Simulation requires about $1400s$.

making the amount of information provided by QRF and simulation similar. For all the experiments, the gap between QRF bounds is less than 1% and we report the time to solve for the upper bound on the utilization of one of the parallel nodes (queue 4). The results indicate that exact methods become quickly intractable, and even if one considers just 4 queues the method is an order of magnitude slower than QRF. This suggests that QRF is generally to be preferred to exact analysis if the problem is not too small. The largest model in the table, which has 16 queues, has approximately 10^{23} states in the Markov chain, therefore it is clearly intractable with exact methods; however, QRF is still applicable and solves the model in less than 5 minutes.

Table III also highlights that state-based methods incur a computational bottleneck earlier than simulation. However, this is a penalizing scenario for QRF against simulation, since we focus on evaluating a single model. To illustrate the benefit of QRF in multiple model evaluations, assume that we want to increase the number of jobs routed to queue 3, reducing by an equal proportion the routing probability to queue 2. This kind of evaluation is common if one uses optimization to determine load balancing probabilities. After solving the initial model, which requires $1s$, QRF can re-optimize quickly after each change of the routing probabilities, solving a single model in about $300ms$. Figure 5 illustrates the result of the evaluation on 100 successive re-optimizations for different values of the routing probabilities. In order to perform the same analysis with simulation, one would need to run 100 simulations, taking approximately $1400s$. Therefore, QRF is an order of magnitude faster than simulation in this example. For models with more queues, similar advantages would apply for QRF against simulation, provided that the initial QRF model size allows to obtain a solution and the optimal basis.

9. RELATED WORK

The analysis of complex stochastic networks via optimization methods has been subject of intense research, especially in the area of stochastic optimal control [Kumar and Kumar 1994; Bertsimas et al. 1994; Bertsimas and Niño-Mora 1996]. Previous works in this area have characterized the region of achievable performance for open and closed queueing network models and related scheduling and stability problems.

The studies in [Kumar and Kumar 1994; Morrison and Kumar 2002] consider Markov chains which have a polyhedral translation invariance property. The polyhedra can be described by sets of linear equalities and inequalities and performance bounds are obtained by optimizing over these sets. In particular, [Kumar and Kumar 1994] considers open and closed re-entrant lines with Poisson arrivals and exponential service. By assuming stability, the authors consider priority scheduling policies and provide performance bounds. For the open basic re-entrant line, the authors also consider finite buffer with Repetitive Service. In this case they provide throughput bounds. In [Morrison and Kumar 2001], the authors improve the bounds in [Kumar and Kumar 1994] for closed re-entrant lines with exponential assumptions, while in [Morrison and Kumar 2002], the authors extend the system characteristics to include unreliable machines, setups and batch tools for just open re-entrant lines with exponential assumptions. [Bertsimas et al. 1994] considers the control of jobs sequencing and routing in open and closed multiclass queueing networks to determine an optimal policy to minimize a linear combination of sojourn times. The work takes exponentiality assumptions, but results may generalize to phase-type distributions. In [Bertsimas 1995], the author reviews the approach to consider several problems, including closed multi class queueing networks with the aim to optimize on sequencing and routing decisions. They assume stability conditions and the objective is to maximize the throughput. Again, the presented case is limited to exponential assumptions, but the method is reported as possible to extend to phase-type distributions.

In [Bertsimas and Niño-Mora 1996], the authors present a general polyhedral framework for formulating and solving scheduling problems focusing on the multiclass $M/G/1$ queue. The results are then used in [Glazebrook and Nino-Mora 1999] to investigate stability for open Markovian multiclass network with single servers. Finally, in [Ansell et al. 1999] the authors consider a M/M/1 queue with two job classes with priority scheduling and investigate backlogs of low priority jobs. They use the approach in [Bertsimas et al. 1994] and they provide bounds for parameterized heuristic policies, by imposing constraints on the second moments of the queue lengths.

Similarly to the above works, QRF follows the idea of bounding through an optimization-based search. However, the classes of models addressed by previous work and QRF are significantly different. Stochastic optimal control works have focused on scheduling for systems with re-entrant lines under priority preemptive disciplines and renewal inter-arrival and service times. Instead, QRF focuses on network with non-preemptive scheduling, Markov-modulated non-renewal arrival and service processes, finite capacities and blocking. We thus believe that QRF methods represent a significant novelty in the area of optimisation-based approximations.

10. CONCLUSION

In this paper, we have introduced QRF, a novel framework for quantitative analysis of complex stochastic networks featuring blocking, high variability, temporal dependence, and state dependence. The focus is on system at equilibrium and the main technical achievement is a novel methodology to cast the global balance equations of a intractably large Markov process into a tractable set of linear inequalities for systems with blocking and state dependence. We have also proposed convex and non-convex point approximations that can estimate stationary behavior more accurately than using QRF bounds. To aid implementation of the method, a prototype release of the QRF bounds code can be downloaded at http://www.doc.ic.ac.uk/~gcasale/content/tomacs_qrf.tgz.

REFERENCES

P.S. Ansell, K.D. Glazebrook, I. Mitrani, and J. Nino-Mora. 1999. A semidefinite programming approach to the optimal control of a single server queueing system with imposed second moment constraints. *Journal of the Operational Research Society* 50 (1999), 765–773. DOI:10.1057/palgrave.jors.2600764

I. Awan, A. Yar, and M. E. Woodward. 2006. Analysis of Queueing Networks with Blocking under Active Queue Management Scheme. In *ICPADS*. IEEE Computer Society, 61–68. DOI:10.1109/ICPADS.2006.25

S. Balsamo, V. De Nitto Personé, and R. Onvural. 2001. *Analysis of queueing networks with blocking*. Kluwer Academic Publishers, Boston, MA.

M. Bertoli, G. Casale, and G. Serazzi. 2009. JMT: performance engineering tools for system modeling. *SIGMETRICS Performance Evaluation Review* 36, 4 (2009), 10–15. DOI:10.1145/1530873.1530877

D. Bertsimas. 1995. The achievable region method in the optimal control of queueing systems; formulations, bounds and policies. *Queueing Systems* 21, 3-4 (1995), 337–389. DOI:10.1007/BF01149167

D. Bertsimas, D. B. Brown, and C. Caramanis. 2011. Theory and Applications of Robust Optimization. *SIAM Rev.* 53, 3 (2011), 464–501. DOI:10.1137/080734510

D. Bertsimas, D. Gamarnik, and J. N. Tsitsiklis. 1996. Stability Conditions for Multiclass Fluid Queueing Networks. *IEEE Trans. Autom. Control* 41, 11 (1996), 1618–1631. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=543999

D. Bertsimas and J. Niño-Mora. 1996. Conservation Laws, Extended Polymatroids and Multiarmed Bandit Problems: A Polyhedral Approach to Indexable Systems. *Mathematics of Operations Research* 21, 2 (1996), 257–306. DOI:10.1287/moor.21.2.257

D. Bertsimas, I. Ch. Paschalidis, J. N. Tsitsiklis and N. John. 1994. Optimization of Multiclass Queueing Networks: Polyhedral and Nonlinear Characterizations of Achievable Performance. *Ann. Appl. Probab.* 4, 1 (1994), 43–75. DOI:10.1214/aoap/1177005200

G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. 2006. *Queueing Networks and Markov Chains*. 2nd ed., John Wiley and Sons.

G. Casale, N. Mi, and E. Smirni. 2010. Model-Driven System Capacity Planning under Workload Burstiness. *IEEE Trans. Computers* 59, 1 (2010), 66–80. DOI:10.1109/TC.2009.135

G. Casale and E. Smirni. 2009. MAP-AMVA: Approximate mean value analysis of bursty systems. In *Dependable Systems Networks, 2009. DSN '09. IEEE/IFIP International Conference on*. 409–418. DOI:10.1109/DSN.2009.5270309

G. Casale, E. Z. Zhang, and E. Smirni. 2008. KPC-Toolbox: Simple Yet Effective Trace Fitting Using Markovian Arrival Processes. In *QEST*. IEEE Computer Society, 83–92. DOI:10.1109/QEST.2008.33

V. De Nitto Personé,G. Casale, and E. Smirni.. 2010. *Analysis of Blocking Networks with Temporal Dependence*. Technical Report. Tech. Rep. RR-10.83, Dept. Comp. Science, University of Rome Tor Vergata.

V. De Nitto Personé, G. Casale, and E. Smirni. 2011. Approximate analysis of blocking queueing networks with temporal dependence. In *Proc. of IEEE/IFIP DSN*. IEEE, 574–585. DOI:10.1109/DSN.2011.5958269

A. E. Ferdinand. 1970. A Statistical Mechanical Approach to Systems Analysis. *IBM Journal of Research and Development* 14, 5 (Sept. 1970), 539–547. DOI:10.1147/rd.145.0539

K.D. Glazebrook and J. Nino-Mora. 1999. A linear programming approach to stability, optimisation and performance analysis for Markovian multiclass queueing networks. *Annals of Operations Research* 92, 0 (1999), 1–18. DOI:10.1023/A:1018922412074

D. Kouvatsos. 1993. Maximum Entropy Analysis of Queueing Network Models. *Lecture Notes in Computer Science* 729 (1993), 245–290. DOI:10.1007/BFb0013856

S. Kumar and P. R. Kumar. 1994. Performance bounds for queueing networks and scheduling policies. *IEEE Trans. Autom. Control* 39 (1994), 1600–1611. DOI:10.1109/9.310033 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=310033&isnumber=7517

J. R. Morrison and P. R. Kumar. 2001. New Linear Program Performance Bounds for Closed Queueing Networks. *Discrete Event Dynamic Systems* 11, 4 (2001), 291–317. DOI:10.1023/A:1011217024661

J. R. Morrison and P. R. Kumar. 2002. Linear Programming Performance Bounds for Markov Chains With Polyhedrally Translation Invariant Probabilities and Applications to Unreliable Manufacturing Systems and Enhanced Wafer Fab Models *(ASME 2002 International Mechanical Engineering Congress and Exposition)*. 69–78. DOI:10.1115/IMECE2002-39274

R. O. Onvural. 1990. Survey of Closed Queueing Networks with Blocking. *Comput. Surveys* 22, 2 (June 1990), 83. DOI:10.1145/78919.78920

H. G. Perros. 1994. *Queueing Networks with Blocking*. Oxford University Press, Inc., New York, NY, USA.

W. Whitt. 1984. Open and closed models for networks of queues. *AT&T Bell Laboratories Technical Journal* 63, 9 (1984), 1911–1979. DOI:10.1002/j.1538-7305.1984.tb00084.x

T. Yamada, N. Mizuhara, H. Yamamoto, and M. Matsui. 2009. A performance evaluation of disassembly systems with reverse blocking. *Computers & Industrial Engineering* 56, 3 (2009), 1113–1125. DOI:10.1016/j.cie.2008.09.029

J. Zahorjan. 1983. Workload representations in queueing models of computer systems. *ACM perf. eval. rev. (SIGMETRICS)* 12 (1983), 70–81. DOI:10.1145/800040.801396