Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

OPTIMIZATION, ERROR BOUNDS, AND WORKLOAD CHARACTERIZATION IN CLOSED PRODUCT-FORM QUEUEING

W. Cheng
R. Muntz

August 1993
CSD-930027

Optimization, Error Bounds, and Workload Characterization in Closed Product-form Queueing Networks

William C. Cheng   Richard R. Muntz*

UCLA Computer Science Department

Abstract

Product-form queueing network models have been widely used to model systems with shared resources such as computer systems (both centralized and distributed), communication networks, and flexible manufacturing systems. Closed multichain product-form networks are inherently more difficult to analyze than open networks, due to the effect of normalization. Results in optimization, error bounds, and workload characterization for closed networks in the literature are often for networks having special structures and only specific performance measures have been considered.

In this article, we present certain properties (insensitivity of conditional state probability distributions and pseudo-linearity of Markov reward functions) of closed multichain product-form networks. These properties are derived using the most basic flow balance conditions of product-form networks. Then we show how these basic properties can be applied in solving optimal routing problems, obtaining performance bounds when the model parameters are given with bounded errors, and obtaining performance bounds when similar customers are clustered together to speed up computation.

Keywords: Queueing network, product-form, closed network, quasi-reversibility, balance equation, optimization, load balancing, file allocation, error bound, clustering.

1 Introduction

Queueing network models are composed of networks of queues and customers that visit the queues. They have been widely used to model systems with shared resources such as

*This research was supported in part by NSF under contract CCR 9215064.

computer systems (both centralized and distributed), communication networks, and flexible manufacturing systems. In these applications, the queues represent the system resources, and the customers represent the users of the system. The solutions of the queueing network models can give estimates of the performance of the real world system under study. Some commonly used measures of performance are throughput, response time, server utilization, etc.

An important class of queueing network models, known as "product-form" or "separable" queueing networks, enjoys the existence of efficient computational algorithms for determining performance measures. In addition, product-form networks have been widely and successfully applied to model the aforementioned real world systems. Since these models can be efficiently solved, they are often used in studying design alternatives. Researchers have also exploited the convexity properties of certain performance measures to obtain an optimal design even more efficiently. For example, in [FRAT73], Fratta, Gerla, and Kleinrock show that, for a multichain open network, throughput is a concave function of flows, and therefore, by using a down-hill search technique, a globally optimal routing policy can be determined. In [KOBA83], Kobayashi and Gerla show that throughput is a concave function of relative arrival rates for single chain closed networks; however, for multichain closed networks, it can be easily shown that throughput is *not* a concave function of the relative arrival rates. In [TRIP88], Tripathi and Woodside show that for a multichain closed network where the routing of every customer can be be controlled independently of the routing of other customers, optimal throughput can be obtained at the vertex of the solution space, which is a convex polytope formed by the allowable range of values of the routing probabilities.

All the above results have been developed only for networks having special structures (such as assuming that the stationary state probability has certain algebraic form or restricting attention to a subset of BCMP networks [BASK75]), and only for specific performance measures. In [CHEN91], we extend the results of [TRIP88] and show that an optimal vertex solution exists for any multichain closed quasi-reversible network [KELL79] where the performance measure of interest can be expressed as a Markov reward function (i.e., a weighted sum of the state probabilities). The steps we take are basically as follows. First, we partition the state space of the network according to where a tagged customer is (i.e., which server) and show that the state probability distribution conditioned on the tagged customer being at a specific server is *independent* of the relative arrival rate of the tagged customer at that server. This result is proven without assuming any specific algebraic form for the steady-state probability distribution but by using only the most basic property of quasi-reversible queueing networks, namely, the *partial balance* condition [KELL79, NELS92] (Muntz calls it the $M \Rightarrow M$ condition [MUNT72], and Chandy and Martin call it the *local balance* condition [CHAN83]). Then we use the above result to simplify the expression for the Markov reward representation of the performance measure of interest. Lastly, we use results from exact aggregation and Courtois and Semal's bounding methods [COUR86] to show that an optimal reward can be obtained at a vertex of the solution space. In [CHEN91], we also show that the total reward, expressed as a function of the relative arrival rates of the tagged customer, is a *fractional linear function* (i.e., a ratio of linear functions). A fractional linear function is *pseudo-linear*, and when constrained linearly, can be optimized at a vertex of the solution space defined by the constraints [BAZA79].

That the state probability distribution, conditioned on the tagged customer being at a particular server, is independent of the routing of the tagged customer raises the following question: "For a network of quasi-reversible queues, is the conditional state probability distribution also independent of the mean service requirement of the tagged customer at any server?" For a delay server or a server with the processor sharing service discipline, we can let the tagged customer self-loop at the server in question and obtain the same distribution (since the distribution is independent of routing); therefore, the distribution would also be independent of the service requirement at the server (since the customer never leaves the server).

Nevertheless, the answer to the above question is negative in general. An intuitive argument against it is as follows. Consider a server where the customers are served in first-in-first-out order. The service requirements of the customer are exponentially distributed. In order for the queue to be quasi-reversible, the mean service requirements of all customer chains at this server must be identical [BASK75]. For such a queue, the conditional state probability distribution may not be independent of the mean service requirement of a tagged customer because changing the mean service requirement of a customer destroys product-form. In this article, we study conditions under which the conditional state probability distribution *is* independent of the mean service requirement of the tagged customer. A large subset of quasi-reversible networks, which we call *pseudo-reversible networks*, satisfies these conditions. (Our results on optimization and error bounds will be shown to apply to product-form networks containing first-come-first-serve service centers as is discussed in detail in Section 5.2.2.)

Knowing that the conditional state probability distribution is independent of relative arrival rates and service requirements, any measure expressible as a Markov reward function can be shown to be a fractional linear function of the model parameters (relative arrival rates, mean service requirements, and relative utilizations) of a customer, and therefore, can be optimized at a vertex of the solution space for an optimization problem, The fact that the model parameters are specified as ranges of values can result from design constraints, such as those in file allocation problems [TRIP88]. It can also come from other sources. For example, if the model parameters are known to have bounded errors or if we are interested in studying the sensitivity of a performance measure with respect to certain model parameters, we can express the parameters as ranges of values. In [GORD80], Gordon and Dowdy study the impact of parameter estimation errors in closed product-form networks. They observe that the partial derivative of the throughput of a chain with respect to a relative load for a particular server is nonpositive for any value of the relative load. If relative loads are known to have bounded errors, an upper bound on the throughput can be obtained when each relative utilization is at its smallest value, and similarly, an lower bound on the throughput can be obtained when each relative utilization is at its largest value. We will show that what Gordon and Dowdy observed in [GORD80] is a special case of a Markov reward function being a fractional linear function of the relative loads.

The fact that the conditional state probability distribution is independent of certain model parameters and that a Markov reward function can be shown to be fractionally linear in certain model parameters is not only of theoretical interest, but also has practical application in optimal routing problems and in obtaining performance bounds. However, solving large closed models can still be very expensive, and in this paper, we also consider this problem. For models having groups of customers with similar valued parameters, an approximation technique of *clustering* can be used to form clusters and associate one chain with each cluster. The customers in a chain all have the same parameters which are chosen to be representative of the cluster. By reducing the number of distinct chains, the space and time complexity of solving the model is reduced. In the literature, clustering has often been used to obtain an approximate solution of the model; to the best of our knowledge, error bounds have not been given. We will show how our basic results can be applied to obtain error bounds for clustering techniques.

The structure of this article is as follows. Section 2 defines the queueing network model and the notation used in the remainder of the article. In Section 3, we present the insensitivity results for product-form networks. Section 4 characterizes performance measures as functions of model parameters. We then apply these basic results in the areas of optimization, performance bounds, and clustering bounds in Section 5. Section 6 concludes with a summary and a discussion.

2 Product-form Queueing Network Model

The queueing network models being considered are closed multichain networks of quasi-reversible queues [KELL79] with Markovian routing for the customers. Such networks are known to have product-form steady-state probabilities. Without loss of generality, we will assume that there is only one customer per chain, unless otherwise stated.

There are K closed routing chains, and within each chain, class changes are allowed; however, the classes associated with different chains are distinct. For convenience, we will assume that a customer never visits two distinct centers in the same class. (This assumption can always be accommodated simply by renaming the classes.) Therefore, the class of a customer uniquely identifies which center it is at and which chain it belongs to. We use $\sigma(c)$ and $\varsigma(c)$ to denote the center and the chain, respectively, associated with a customer of class c. We use the term *queue* or *center* to refer to both the waiting room and the server parts of a service center.

An important property of a quasi-reversible queue is that the *partial balance* condition is satisfied when the queue is isolated from the network and driven by a multiclass Poisson workload. The *partial balance* condition is also known as the $M \Rightarrow M$ condition [MUNT72] or the *local balance* condition [CHAN83]. Let $\vec{x}$ and $\vec{y}$ denote states of a queue, $q(\vec{x}, \vec{y})$ denote the instantaneous state transition rate from $\vec{x}$ to $\vec{y}$, $\pi(\vec{x})$ denote the steady-state probability of state $\vec{x}$, and $A_c(n)$ denote the set of states where there are n class c customers in the queue. The partial balance condition is satisfied if, for any class c and any state $\vec{x} \in A_c(n)$, where $n \geq 0$,

$$\pi(\vec{x}) \sum_{\vec{y} \in A_c(n+1)} q(\vec{x}, \vec{y}) = \sum_{\vec{y} \in A_c(n+1)} \pi(\vec{y})\, q(\vec{y}, \vec{x}) \tag{1}$$

The partial balance condition balances the flow *out of* a state due to the *arrivals* of a class of customers with the flow *into* the same state due to the *departure* of the same class of customers.

Another kind of balance, *station balance* (as defined in [NELS92]), is characterized by the following equation. For any class c and any state $\vec{x} \in A_c(n)$, where $n \geq 1$,

$$\pi(\vec{x}) \sum_{\vec{y} \in A_c(n-1)} q(\vec{x}, \vec{y}) = \sum_{\vec{y} \in A_c(n-1)} \pi(\vec{y})\, q(\vec{y}, \vec{x}) \tag{2}$$

The station balance condition balances the flow *out of* a state due to the *departure* of a class of customers with the flow *into* the same state due to the *arrival* of the same class of customers.

The station balance condition defined here is weaker than the one defined in [CHAN77] where a service center is divided into *stations*, each of which can be occupied by a customer. Station balance, as defined in [CHAN77], holds if the flow out of a *station* due to the departure of the customer in the station equals the flow into the same station due to the arrival of the same class of customers. We adopt the definition of station balance of [NELS92] because it is weaker (and seems to be more natural to us) than [CHAN77]. (For a more detailed discussion, the reader is referred to [CHEN92].) Figure 1 depicts the difference between partial balance and station balance. The solid lines correspond to arrivals and the dashed lines correspond to departures of class c customers. In a way, partial balance and station balance are "mirror images" of each other.

Partial balance is a property of a quasi-reversible queue. Here we define a new term called *pseudo-reversibility*. A *pseudo-reversible* queue is a quasi-reversible queue that also satisfies the *station balance* condition described above. Not all quasi-reversible queues are pseudo-reversible. For example, a quasi-reversible queue that serves customers in the first-come-first-serve order is not pseudo-reversible.

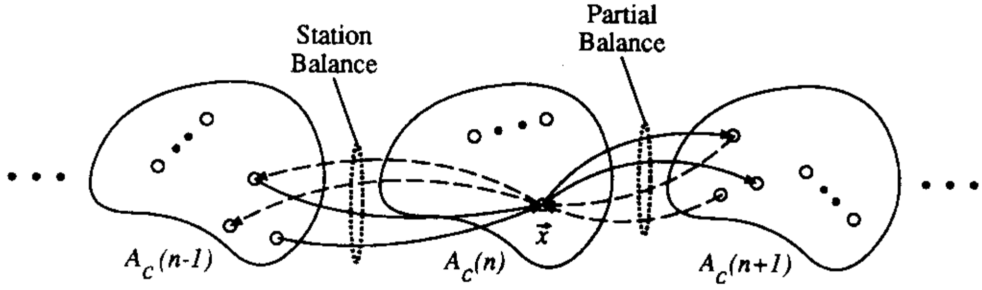


Figure 1: Partial balance vs. station balance.

For a pseudo-reversible queue, we make the following assumptions concerning the scheduling discipline of the queue and the service time distribution of the customers. (1) We assume that the service discipline has the *immediate service* characteristic, meaning that a customer has a service rate > 0 immediately after its arrival to the queue. This excludes, for example, any policy that serves the customers in first-in-first-out order. (2) We assume that a service time distribution has a rational Laplace transform; such a distribution can be represented by a network of exponential stages [COX55]. Figure 2 shows an example of an m stage Coxian distribution where the mean service time of stage i is $1/\mu_i$, and the probability that the customer goes on to stage $i+1$ when it finishes service at stage i is α_i. Algebraically, all known quasi-reversible queues satisfying assumptions (1) and (2), which we have examined, are pseudo-reversible.

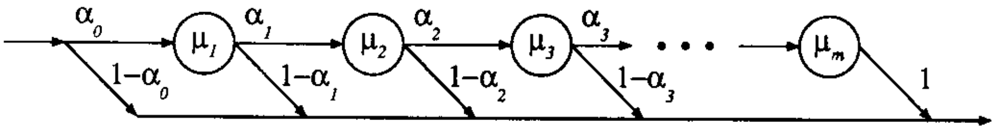


Figure 2: An m stage Coxian representation of a service time distribution

With the immediate service assumption, a model containing Coxian service time distributions can be replaced by an equivalent model with only exponential service time distributions if class changes are introduced properly. Basically, a customer finishing service at stage i and going to stage $i+1$ with probability α_i in the original model is replaced by a customer changing from class i to $i+1$ with probability α_i and returning to the same queue; the service times for classes i and $i+1$ are exponentially distributed with means $1/\mu_i$ and $1/\mu_{i+1}$, respectively. For the rest of the article, when we consider a pseudo-reversible queue satisfying assumptions (1) and (2), we assume that, without loss of generality, the service times are exponentially distributed.

For a pseudo-reversible queue, we further assume that the total rate of service for a class c customer at center $j = \sigma(c)$ can only be dependent on the state of center j (we do not allow the service rate to depend on the state of other centers); it is denoted by $\mu_c(\vec{x}_j)$, where $j = \sigma(c)$ and $\vec{x}_j$ is the state of queue j. When a class c customer has received the required amount of service, it changes its class to class d (and moves to center $\sigma(d)$) with probability p_{cd}. In this article, we will only consider the case where the service rate for a customer can be decomposed into the following form for a pseudo-reversible queue,

$$\mu_c(\vec{x}_j) \;=\; \frac{\mu_0(\vec{x}_j) f_c(\vec{x}_j)}{T_c} \qquad (3)$$

where $\mu_0(\vec{x}_j)$ denotes the total service effort supplied by server j when its state is $(\vec{x}_j)$, $f_c(\vec{x}_j)$ denotes the fraction of server j's capacity given to serve the class c customer when the state of the server is $\vec{x}_j$, and T_c is the mean *service requirement* for the class c customer. (We will use the term "service time" to also refer to the "service requirement" in the rest of the article.) We assume that $\mu_0(\vec{x}_j)$ and $f_c(\vec{x}_j)$ are given as constants (*not* functions of the service time of any particular customer). All quasi-reversible queues known to us which satisfy assumptions (1) and (2) mentioned earlier in this section have service rate equations of the form of Eq. (3). Therefore, it appears that we do not lose any generality with the above restriction in service rates.

2.1 Model Parameters

Throughout this article, we assume that certain model parameters can be varied or controlled within some range. (Applications, e.g., optimization, are described in Section 5). The model parameters being considered are the visit ratios, mean service times, and relative utilizations. When visit ratios are considered to be variables, we assume that mean service times for all chains are held constant. When mean service times are considered to be variables, we assume that visit ratios for all chains are held constant. As we will see in the following section, there is a direct relationship between the types of model parameters which can be varied and the types of performance measures for which we have effective solutions for the optimization and error bound problems.

2.2 Performance Measures

The performance measures of interest are those that can be expressed as Markov reward functions. A Markov reward function, denoted by R, is a weighted sum of state probabilities:

$$R = \sum_{\underline{\vec{x}} \in \underline{S}} \pi(\underline{\vec{x}})\, R(\underline{\vec{x}}) \tag{4}$$

where $\underline{\vec{x}} = (\vec{x}_1, \ldots, \vec{x}_J)$ denotes a state of the network of queues with J being the total number of queues, $\underline{S}$ denotes the state space of the network, $\pi(\underline{\vec{x}})$ denotes the steady-state probability of state $\underline{\vec{x}}$, and $R(\underline{\vec{x}})$ is the *reward rate* assigned to state $\underline{\vec{x}}$. Eq. (4) can express many useful mean performance measures such as server utilization, mean number of customers in a subsystem, weighted sum of customer throughputs, etc. However, it can not express measures such as mean residence time. When considering a network of *quasi-reversible* queues. we assume that $R(\underline{\vec{x}})$ is *independent* of the routing probabilities and the visit ratios of any customer and that the model parameters that can be varied are the visit ratios of the customers at various queues. It is unusual to have $R(\underline{\vec{x}})$ be dependent on visit ratios; therefore, for all practical purposes, any Markov reward function can be considered for a network of quasi-reversible queues where visit ratios can be varied.

For the more restricted case of a network of *pseudo-reversible* queues, the visit ratios of the customers may be considered fixed and the mean service times of the customers can be varied. For performance measures such as server utilization, mean number of customers, sum of state probabilities (such as those used in [DESO89] to measure steady-state availability of a repairable computer system), etc., $R(\underline{\vec{x}})$ is easily seen to be constant, relative to the mean service times. However, if the performance measure of interest is a weighted sum of customer throughputs, $R(\underline{\vec{x}})$ would be a function of some mean service time parameters. In this case, we restrict $R(\underline{\vec{x}})$ to be in a form described below.

Let $\nu_k(\underline{\vec{x}})$ denote the class of the chain k customer in state $\underline{\vec{x}}$, and let $\mathbf{1}(\cdot)$ be an indicator function which evaluates to 1 if the expression inside the parenthesis is true and to 0 otherwise. Also, let k^* denote a *distinguished* class for the chain k customer. Basically, if the performance measure of interest is a function of the throughput of the chain k customer in class c, then c must be the distinguished class for the chain k customer. We allow $R(\underline{\vec{x}})$ to be of the following form:

$$R(\underline{\vec{x}}) \;=\; a(\underline{\vec{x}}) + \sum_{k} \frac{\mathbf{1}(\nu_k(\underline{\vec{x}}) = k^*) b_k(\underline{\vec{x}})}{T_{k^*}} \tag{5}$$

where $a(\underline{\vec{x}})$ and the $b_k(\underline{\vec{x}})$'s are independent of the routing probabilities, visit ratios, and mean service times of any customer. (Please note that, for any chain, only the mean service time of *one* of its classes is allowed to appear in the reward rate equation.)

Eq. (5) looks restrictive; nevertheless, it seems that all "interesting" performance measures which are expressible as Markov reward functions can fit the form of Eq. (5). Some examples are a weighted sum of customer throughputs, a weighted sum of server utilizations, a weighted sum of mean queue lengths, and a weighted sum of state probabilities (such as availability measures for repairable computer systems [DESO89]). Mean sojourn time is *not* included because it is not expressible as a Markov reward function. The performance measures that Eq. (5) excludes seem to be obscure ones (e.g., $R(\underline{\vec{x}})$ being a polynomial function of T_c or a higher order polynomial function of $1/T_c$).

2.3 Notation

For convenience, we list below the notation which is used throughout the article. It is important to recall that there is only one customer per chain in the closed queueing model, and that a class index fully specifies the location and the chain membership of a customer.

J = number of service centers.

K = total number of closed routing chains (or customers) in the network.

C = total number of classes.

c, d = class indices.

$\varsigma(c)$ = the chain associated with the customer of class c.

$\sigma(c)$ = the service center associated with the customer of class c.

$\mathcal{C}_k$ = the set of classes visited by the chain k customer.

p_{cd} = routing probability (the probability that the customer of chain $\varsigma(c)$ in class c changes class to d when it finishes service at service center $\sigma(c)$).

θ_c = visit ratio of the chain $k = \varsigma(c)$ customer at center $\sigma(c)$, scaled such that $\theta_{c(k)} = 1$, where $c(k)$ is a specified (reference) class of chain $\varsigma(c)$. $\theta_c = \sum_{d \in \mathcal{C}_{\varsigma(c)}} \theta_d \, p_{dc}$.

T_c = mean service requirement of the chain $\varsigma(c)$ customer at center $\sigma(c)$ in class c.

ρ_c = $\theta_c T_c$ = relative utilization (or relative load) of class c.

x_{jk} = state of the chain k customer with respect to center j. Specifically, $x_{jk} = 0$ if the chain k customer is not at center j and $x_{jk} = c$ if the chain k customer is at center j in class c.

$\vec{x}_j$ = $(x_{j1}, x_{j2}, \ldots, x_{jK})$ = state of center j.

$\underline{\vec{x}}$ = $(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_J)$ = state of the network model.

$\underline{S}$ = state space of the network.

$\pi(\underline{\vec{x}})$ = steady-state probability that the network is in state $\underline{\vec{x}} \in \underline{S}$.

3 Insensitivity of Conditional State Probability Distribution

In this section, we present the insensitivity results for networks of quasi-reversible and pseudo-reversible queues. Let A_c denote the set of states where the chain $\varsigma(c)$ customer is at center $\sigma(c)$ in class c. For a state $\underline{\vec{x}} \in A_c$, let $\pi(\underline{\vec{x}}|A_c) = \pi(\underline{\vec{x}}) / \sum_{\underline{\vec{x}}' \in A_c} \pi(\underline{\vec{x}}')$ denote the steady-state probability of state $\underline{\vec{x}}$, conditioned on the chain $\varsigma(c)$ customer being in class c. Below, we show that for a network of quasi-reversible queues, for all $\underline{\vec{x}} \in A_c$, $\pi(\underline{\vec{x}}|A_c)$ is *independent* of (or *insensitive* to) the values of the visit ratios of the customer of chain $\varsigma(c)$. We also show that for a network of pseudo-reversible queues (which are also quasi-reversible by definition), the conditional state probability distribution is also *independent* of the service time of that customer in class c. (Stated in equational form, $\forall \, \underline{\vec{x}} \in A_c$, $\pi(\underline{\vec{x}}|A_c)$ is independent of the θ_d's, $\forall \, d \in \mathcal{C}_{\varsigma(c)}$ for quasi-reversible networks and $\pi(\underline{\vec{x}}|A_c)$ is, in addition, independent of the T_d's, $\forall \, d \in \mathcal{C}_{\varsigma(c)}$ for pseudo-reversible networks.)

For a product-form network, a state probability can be expressed as a product of marginal state probabilities as shown below.

$$\pi(\underline{\vec{x}}) \;=\; \frac{1}{G}\prod_{j=1}^{J}\pi_j(\vec{x}_j) \tag{6}$$

where $G = \sum_{\underline{\vec{x}}' \in \underline{S}} \pi(\underline{\vec{x}}')$ is the normalization constant and $\pi_j(\vec{x}_j)$ is the steady-state probability of state $\vec{x}_j$ for center j when it is driven by Poisson arrivals with rates θ_{c^*}'s, where $c^* \in \{\, c \,|\, 1 \leq \varsigma(c) \leq K,\ \sigma(c) = j$ and $\theta_c > 0\,\}$. For a ratio of state probabilities (e.g., as occurs in expressions for conditional state probabilities), the normalization constants cancel each other out. In order for us to prove the insensitivity results, we need to examine the relationship between the marginal state probabilities and the model parameters.

3.1 A Queue in Isolation

Consider an isolated quasi-reversible queue (or a pseudo-reversible queue, where appropriate) driven by Poisson arrivals with rates $\lambda_1, \ldots, \lambda_C$. No class change is allowed in this model. Let $\vec{x}_j$ denote a state of the queue. (For convenience in notation, we keep the j subscript in $\vec{x}_j$. This isolated queue is considered to be queue number j.) Let the (infinite) state space of this model be denoted by S, and let $A_c(n)$ denote the set of states with n class c customers. Let $\pi(\vec{x}_j|A_c(n)) = \pi(\vec{x}_j)/\sum_{\vec{x}_j' \in A_c(n)} \pi(\vec{x}_j')$ denote the state probability of state $\vec{x}_j$ conditioned on the state of the model being in some state of $A_c(n)$. We have the following lemma.

Lemma 1 *For a quasi-reversible queue, both $\pi(\vec{x}_j|A_c(0))$ and $\pi(\vec{x}_j|A_c(1))$ are* independent *of λ_c*[1]*.*

Proof: See the proofs of Lemmas 1 and 2 in [CHEN91]. □

Now we turn our attention to the service time parameter. We assume that the service rates are in the form specified in Eq. (3).

Lemma 2 *For a quasi-reversible queue, $\pi(\vec{x}_j|A_c(0))$ is* independent *of T_c.*

[1] This lemma suggests that a general proof of independence of $\pi(\vec{x}_j|A_c(n))$ is possible by induction. This is indeed true. However, the lemma is stated only for $n = 0$ and $n = 1$ because that is all that is needed here.

Proof:

The proof is very similar to the proof of Lemma 1. In the proof of Lemma 1, we have shown that $\pi(\vec{x}_j|A_c(0))$ can be obtained by truncating the state space to $A_c(0)$. (In a recent paper, Nelson calls this the *state truncation* property for queues that satisfy the partial balance condition [NELS92].) Again, let Q^* denote the rate matrix that corresponds to the truncated model; then we have $\pi(\vec{x}_j|A_c(0))\, Q^* = \vec{0}$. This situation is depicted in Figure 3. Clearly this eliminates the dependency of the transition rates in Q^* on T_c because Q^* only involves transitions within $A_c(0)$, while only transitions from states in $A_c(1)$ into states in $A_c(0)$ are dependent on T_c. Since all transition rates in Q^* are independent of T_c, the components of the solution to $\pi Q^* = \vec{0}$, namely, the $\pi(\vec{x}_j|A_c(0))$'s, are also independent of T_c. □

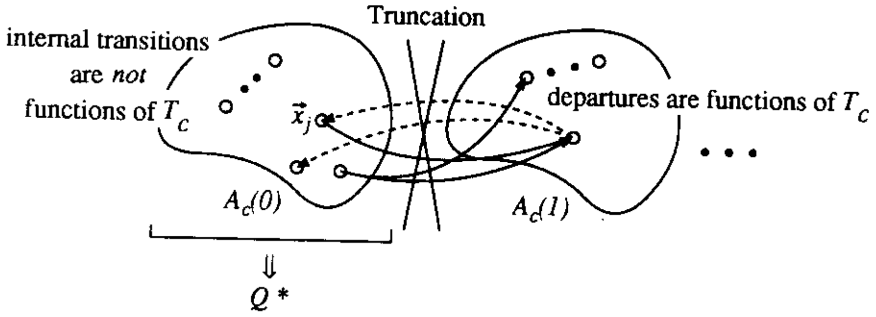


Figure 3: Truncation of the state space to $A_c(0)$.

Below, we restrict ourselves to *pseudo-reversible* queues because station balance is needed.

Lemma 3 *For a pseudo-reversible queue,* $\pi(\vec{x}_j|A_c(1))$ *is* independent *of* T_c.

Proof:

From global balance, we have, $\forall\ \vec{x}_j \in A_c(1)$,

$$\pi(\vec{x}_j) \sum_{\vec{x}_j{}' \in A_c(1)} q(\vec{x}_j, \vec{x}_j{}') \;+\; \pi(\vec{x}_j) \sum_{\vec{y}_j \in A_c(0)} q(\vec{x}_j, \vec{y}_j) \;+\; \pi(\vec{x}_j) \sum_{\vec{z}_j \in A_c(2)} q(\vec{x}_j, \vec{z}_j) \;=$$
$$\sum_{\vec{x}_j{}' \in A_c(1)} \pi(\vec{x}_j{}') q(\vec{x}_j{}', \vec{x}_j) \;+\; \sum_{\vec{y}_j \in A_c(0)} \pi(\vec{y}_j) q(\vec{y}_j, \vec{x}_j) \;+\; \sum_{\vec{z}_j \in A_c(2)} \pi(\vec{z}_j) q(\vec{z}_j, \vec{x}_j)$$

The second and the third terms on each side of the equation cancel out due to station balance and partial balance, respectively. We then have:

$$\pi(\vec{x}_j) \sum_{\vec{x}_j' \in A_c(1)} q(\vec{x}_j, \vec{x}_j') = \sum_{\vec{x}_j' \in A_c(1)} \pi(\vec{x}_j') q(\vec{x}_j', \vec{x}_j) \tag{7}$$

Eq. (7) corresponds to the global balance equation if the state space is truncated to $A_c(1)$. Let $\pi^*(\vec{x}_j)$ denote the solution for the truncated model. Since none of the parameters of the truncated model are functions of T_c, $\pi^*(\vec{x}_j)$ is independent of T_c.

Clearly, $G\pi(\vec{x}_j)$ is a solution to the truncated model, where G is the normalizing constant, i.e., $G = 1/\sum_{\vec{x}_j' \in A_c(1)} \pi(\vec{x}_j')$. Therefore, $\pi^*(\vec{x}_j) = G\pi(\vec{x}_j) = \pi(\vec{x}_j|A_c(1))$, and we have that $\pi(\vec{x}_j|A_c(1))$ is independent of T_c. □

Since the relative load (ρ_c) is just the product of the visit ratio (θ_c) and the service time (T_c), the following is a direct consequence of Lemmas 1, 2, and 3. For a quasi-reversible queue, $\pi(\vec{x}_j|A_c(0))$ is *independent* of ρ_c, and for a pseudo-reversible queue, $\pi(\vec{x}_j|A_c(1))$ is *independent* of ρ_c.

Below, we show that $\pi(\underline{\vec{x}}|A_c)$, the conditional state probability in a closed queueing model, is independent of the visit ratios of the chain $\varsigma(c)$ customer for a network of quasi-reversible queues. We also show that $\pi(\underline{\vec{x}}|A_c)$ is independent of the service times and the relative loads of the chain $\varsigma(c)$ customer for a network of pseudo-reversible queues.

3.2 Closed Queueing Network Model

Lemma 4 *For a network of quasi-reversible queues with Markovian routing, $\pi(\underline{\vec{x}}|A_c)$ is* independent *of θ_d, $\forall\, d \in \mathcal{C}_{\varsigma(c)}$.*

Proof:

See the proof of Lemma 3 in [CHEN91] for the case where $d = c$. For $d \in \mathcal{C}_{\varsigma(c)}$ and $d \neq c$, it is quite straight forward. From the definition of $\pi(\underline{\vec{x}}|A_c)$ and by application of Eq. (6), we have,

$$\pi(\underline{\vec{x}}|A_c) = \frac{\pi(\underline{\vec{x}})}{\sum_{\underline{\vec{x}}' \in A_c} \pi(\underline{\vec{x}}')} = \frac{\prod_{j=1}^{J} \pi_j(\vec{x}_j)}{\sum_{\underline{\vec{x}}' \in A_c} \prod_{j=1}^{J} \pi_j(\vec{x}_j')} \tag{8}$$

Since $\underline{\vec{x}} \in A_c$ and each chain can only have one customer, when the chain $\varsigma(c)$ customer is in class c, there can be no customer in class d. Therefore, the expression for any marginal state probability in Eq. (8) must not contain any θ_d's. Therefore, $\pi(\underline{\vec{x}}|A_c)$ is independent of θ_d. □

In general, the unconditioned state probability is a function of θ_c and the θ_d's, for $d \neq c$. However, the θ_d's only appear in the normalization constant, and the dependencies cancel out in the equation for the conditional state probability. In addition to appearing in the normalization constant, θ_c also appears in each $\pi_j(\vec{x}_j)$ and $\pi_j(\vec{x}_j{}')$. Again, the dependencies cancel out. The next lemma is similar to the above, but it is for networks of pseudo-reversible queues.

Lemma 5 *For a network of pseudo-reversible queues with Markovian routing, $\pi(\underline{\vec{x}}|A_c)$ is* independent *of T_d, $\forall\, d \in C_{\varsigma(c)}$.*

Proof:

The proof is almost the same as the proof for Lemma 4, except that the visit ratios are constants here, and Lemmas 2 and 3 are used to show that for any center j, $\pi_j(\vec{x}_j{}')/\pi_j(\vec{x}_j)$ is independent of T_d due to the fact that the respective conditional state probabilities are independent of T_d. □

3.3 Discussion

It is important to note that, throughout the proofs in this section, we do *not* assume any specific algebraic form for the steady-state probability distribution. We use the most basic property of quasi-reversible and pseudo-reversible queues, namely, the *partial balance* and the *station balance* conditions. With pseudo-reversible queues, the results are restricted to queues with the immediate service characteristic, customers with service time distributions that have rational Laplace transforms, and certain form of Markov reward functions. Nevertheless, as we have discussed in the presentation above, we do not lose generality in practice.

It is also important to note that even though any model with multiple customers per chain can be converted to a model with a single customer per chain, we assume that the visit ratios of a customer are *independent* of the visit ratios of another customer in Lemma 4 and that the service times of each customer are *independent* of the service times of other customers in Lemma 5. Therefore, if one starts with multiple customers per chain, these results do not apply. However, as we will see in Sections 5.2 and 5.3, such a restriction can be relaxed for certain applications.

4 Pseudo-linearity of Markov Reward Functions

In this section, we show that, for a network of quasi-reversible queues, the mean time the chain k customer spends in any class $c \in \mathcal{C}_k$ (per visit to class c) is independent of chain k customer's routing parameters ($\theta_{c'}$'s, $\forall c' \in \mathcal{C}_k$). The mean time the chain $\varsigma(c)$ customer spends in class c is referred to as the *mean sojourn time* or the *mean waiting time* (including the customer's service time) in class c, and it is denoted by W_c (the notation l_c is used in [CHEN91]). We also show that, for a network of pseudo-reversible queues, the mean sojourn time is *proportional* to T_c.

Using the above facts, we then show that a Markov reward function is *marginally pseudo-linear* in the model parameters of a customer. The adjective "marginally" means that the model parameters of other customers are held constant. The model parameters being considered are visit ratios for networks of quasi-reversible queues and service times and relative utilizations for networks of pseudo-reversible queues.

Throughout this section, we will focus on an (arbitrarily chosen) chain k customer. For ease of exposition, let $\mathcal{C}_k = \{1, 2, \ldots, C_k\}$. The state space $\underline{S}$ can be partitioned according to the class of the chain k customer, and the partition is denoted $\{A_1, A_2, \ldots, A_{C_k}\}$. Let $P(A_c) = \sum_{\underline{\vec{x}} \in A_c} \pi(\underline{\vec{x}})$ denote the steady-state probability that the chain k customer is in class c. Using the definition of the conditional state probability, $\pi(\underline{\vec{x}}|A_c) = \pi(\underline{\vec{x}})/P(A_c)$, the steady-state reward rate can be rewritten as follows:

$$R = \sum_{\underline{\vec{x}} \in \underline{S}} \pi(\underline{\vec{x}}) R(\underline{\vec{x}}) = \sum_{c=1}^{C_k} \sum_{\underline{\vec{x}} \in A_c} \pi(\underline{\vec{x}}) R(\underline{\vec{x}}) = \sum_{c=1}^{C_k} \left[P(A_c) \sum_{\underline{\vec{x}} \in A_c} \pi(\underline{\vec{x}}|A_c) R(\underline{\vec{x}}) \right] \qquad (9)$$

We will first examine the relationship between the mean waiting time and the model parameters.

4.1 Mean Waiting Time Equations

Lemma 6 *For a network of quasi-reversible queues, W_c is* independent *of the routing of the chain k customer.*

Proof:

Let λ_c denote the throughput of the chain k customer in class c. Using Eq. (9), λ_c can be expressed as,

$$\lambda_c = P(A_1) \sum_{\vec{\underline{x}} \in A_1} \pi(\vec{\underline{x}}|A_1) R'(\vec{\underline{x}}) + \ldots + P(A_{C_k}) \sum_{\vec{\underline{x}} \in A_{C_k}} \pi(\vec{\underline{x}}|A_{C_k}) R'(\vec{\underline{x}})$$

where $R'(\vec{\underline{x}})$ is the throughput (departure) rate of the chain k customer in state $\vec{\underline{x}}$ (and therefore, independent of the routing of the chain k customer). Clearly, if $\vec{\underline{x}} \notin A_c$, $R'(\vec{\underline{x}}) = 0$, and the above equation reduces to,

$$\lambda_c = P(A_c) \sum_{\vec{\underline{x}} \in A_c} \pi(\vec{\underline{x}}|A_c) R'(\vec{\underline{x}}) \tag{10}$$

Since there is only a single customer per chain, $P(A_c)$ is just the mean queue length of the class c customer at queue $\sigma(c)$. From Little's result, $\lambda_c W_c = P(A_c)$, and we have,

$$W_c = \left[\sum_{\vec{\underline{x}} \in A_c} \pi(\vec{\underline{x}}|A_c) R'(\vec{\underline{x}}) \right]^{-1} \tag{11}$$

Since $\pi(\vec{\underline{x}}|A_c)$ and $R'(\vec{\underline{x}})$ are independent of the routing of chain k (confer Lemma 4), W_c is independent of the routing of chain k. □

Lemma 6 is consistent with the *Arrival Theorem* [LAVE80], which is used in the celebrated Mean Value Analysis (MVA) algorithm [REIS80]. The Arrival Theorem basically states that, in a product-form network, a customer arriving to a queue sees the network (with itself removed) in equilibrium. This would also imply that the mean waiting time at the queue is independent of the routing of the customer.

Lemma 7 *For a network of pseudo-reversible queues, W_c is* proportional *to T_c.*

Proof:

In Eq. (11), $R'(\underline{\vec{x}})$ measures the throughput rate of the class c customer when the state is $\underline{\vec{x}}$. In other words, $R'(\underline{\vec{x}})$ is the service rate of the class c customer in state $\underline{\vec{x}}$, which we have assumed can be written as $R''(\underline{\vec{x}})/T_c$, where $R''(\underline{\vec{x}})$ is independent of T_c (confer Eq. (3)). Then we have,

$$W_c = \frac{T_c}{\sum_{\underline{\vec{x}}\in A_c} \pi(\underline{\vec{x}}|A_c)R''(\underline{\vec{x}})} \tag{12}$$

The denominator of Eq. (12) is independent of T_c (confer Lemma 5); therefore, W_c is *proportional* to T_c. □

Many familiar waiting time equations for product-form queues are in the form indicated by Lemma 7. For example, for a delay server, $W_c = T_c$. For a fixed-rate server,

$$W_c(\vec{N}) = T_c\left[1 + L_{\sigma(c)}(\vec{N} - \vec{e}_{\varsigma(c)})\right]$$

where $\vec{N}$ denotes the network population vector, and L_j is the mean queue length at queue j [REIS80]. For a load-dependent server,

$$W_c = T_c \sum_{m=1}^{N} \frac{m}{\mu_{\sigma(c)}(m)} P_{\sigma(c)}(m-1|\vec{N} - \vec{e}_{\varsigma(c)})$$

where N is the total number of customers in the network, $\mu_j(m)$ is the service rate of queue j when it has m customers, and $P_j(m|\vec{N})$ is the stationary probability that queue j has m customers when the network population vector is $\vec{N}$ [REIS81].

Now we show that the Markov reward functions we are considering (see Section 2.2) are *marginally pseudo-linear* in the model parameters of a customer.

4.2 Reward Functions

For a network of *quasi-reversible* queues, $R(\underline{\vec{x}})$ is assumed to be independent of the routing of the chain k customer. Lemma 4 shows that $\pi(\underline{\vec{x}}|A_c)$ is also independent of the routing of the chain k customer. Therefore, the sum, $\sum_{\underline{\vec{x}}\in A_c} \pi(\underline{\vec{x}}|A_c)R(\underline{\vec{x}})$ is independent of the routing of the chain k customer. We use R_c to denote the above sum. R_c is referred to as the *conditional reward rate* of the chain k customer in class c. Eq. (9) can then be rewritten as,

$$R = \sum_{c=1}^{C_k} P(A_c) R_c \tag{13}$$

We have the following lemma.

Lemma 8 *For a network of quasi-reversible queues, a Markov reward function R is a* fractional linear function *of the visit ratios of the chain k customer, if the visit ratios of all other chains are held constant.*

Proof:

Using Little's result, we have that $P(A_c) = \lambda_c W_c$. However, $\sum_{c'=1}^{C_k} \lambda_{c'} W_{c'} = 1$, and therefore,

$$P(A_c) = \frac{\lambda_c W_c}{\sum_{c'=1}^{C_k} \lambda_{c'} W_{c'}}$$

Since $\lambda_c / \lambda_{c'} = \theta_c / \theta_{c'}$, the above equation becomes,

$$P(A_c) = \frac{\theta_c W_c}{\sum_{c'=1}^{C_k} \theta_{c'} W_{c'}} \tag{14}$$

and therefore,

$$R = \frac{\sum_{c=1}^{C_k} \theta_c W_c R_c}{\sum_{c'=1}^{C_k} \theta_{c'} W_{c'}} \tag{15}$$

From Lemma 6, W_c and the $W_{c'}$'s for $c' \in \mathcal{C}_k$ are all independent of the routing of the chain k customer. Therefore, R in the above equation is a fractional linear function of the visit ratios of the chain k customer. □

For a network of *pseudo-reversible* queues, Eq. (14) is still valid. Substituting Eq. (12) into Eq. (14) and denoting the denominator of Eq. (12) by $1/\alpha_c$, we have,

$$P(A_c) = \frac{T_c \theta_c \alpha_c}{\sum_{c'=1}^{C_k} T_{c'} \theta_{c'} \alpha_{c'}} \tag{16}$$

where α_c is independent of T_c, as shown in Lemma 7. The total reward can be written as,

$$R = \sum_c P(A_c) R_c = \frac{\sum_{c=1}^{C_k} T_c \theta_c \alpha_c R_c}{\sum_{c'=1}^{C_k} T_{c'} \theta_{c'} \alpha_{c'}} \tag{17}$$

If $R(\vec{\underline{x}})$ is independent of the T_c's, the R_c's will also be independent of the T_c's, and R can easily be seen to be a *fractional linear function* of the mean service times of the chain k customer. However, this excludes interesting performance measures such as a weighted sum of customer throughputs. Therefore, we allow $R(\vec{\underline{x}})$ to take the the following form (as mentioned in Section 2.2):

$$R(\vec{\underline{x}}) \;=\; a(\vec{\underline{x}}) + \sum_{l=1}^{K} \mathbf{1}(\nu_l(\vec{\underline{x}}) = l^*)\frac{b_l(\vec{\underline{x}})}{T_{l^*}} \tag{18}$$

where $a(\vec{\underline{x}})$ and the $b_l(\vec{\underline{x}})$'s are constants. Furthermore, since we are focusing on the chain k customer, the mean service times of other chains are considered constant. We have the following lemma.

Lemma 9 *For a network of pseudo-reversible queues, if the reward rate for a state can be written as shown in Eq. (18), then R is a* fractional linear function *of the mean service times of the chain k customer, if the mean service times of all other chains are held constant.*

Proof:

Eq. (18) can be rewritten as,

$$R(\vec{\underline{x}}) \;=\; a(\vec{\underline{x}}) \;+\; \sum_{l=1,\, l\neq k}^{K} \left[\mathbf{1}(\nu_l(\vec{\underline{x}}) = l^*)\frac{b_l(\vec{\underline{x}})}{T_{l^*}}\right] \;+\; \mathbf{1}(\nu_k(\vec{\underline{x}}) = k^*)\frac{b_k(\vec{\underline{x}})}{T_{k^*}} \tag{19}$$

Since there is only one distinguished class per chain, the second term on the right hand side of Eq. (19) is constant with respect to T_{k^*}. Combining the first two terms on the right hand side of Eq. (19) and calling the combined term $a_k(\vec{\underline{x}})$, we have,

$$R(\vec{\underline{x}}) \;=\; a_k(\vec{\underline{x}}) + \mathbf{1}(\nu_k(\vec{\underline{x}}) = k^*)\frac{b_k(\vec{\underline{x}})}{T_{k^*}}$$

where $a_k(\vec{\underline{x}})$ is a constant with respect to the mean service times of the chain k customer. Notice that if $\vec{\underline{x}} \in A_c$, then $\nu_k(\vec{\underline{x}}) = c$; combining this with the definition of R_c, we have,

$$\begin{aligned} R_c &= \sum_{\vec{\underline{x}}\in A_c} \pi(\vec{\underline{x}}|A_c)R(\vec{\underline{x}}) \\ &= \sum_{\vec{\underline{x}}\in A_c} \pi(\vec{\underline{x}}|A_c)a_k(\vec{\underline{x}}) \;+\; \mathbf{1}(c = k^*)\frac{1}{T_{k^*}} \sum_{\vec{\underline{x}}\in A_c} \pi(\vec{\underline{x}}|A_c)b_k(\vec{\underline{x}}) \\ &= a_c + \mathbf{1}(c = k^*)\frac{b_c}{T_{k^*}} \end{aligned}$$

where a_c and b_c are independent of T_{k^*}. Since k^* is just one of the classes of chain k $(1 \leq k^* \leq C_k)$, we can substitute the above equation for R_c into Eq. (17) and get,

$$R = \frac{\theta_{k^*}\alpha_{k^*}b_{k^*} + \sum_{c=1}^{C_k}(T_c\theta_c\alpha_c a_c + \theta_c\alpha_c b_c)}{\sum_{c'=1}^{C_k} T_{c'}\theta_{c'}\alpha_{c'}} \tag{20}$$

Clearly, R is a fractional linear function of the mean service times of the chain k customer. □

If we are interested in R as a function of the relative utilizations, we have the following lemma.

Lemma 10 *For a network of pseudo-reversible queues, if the reward rate for a state is constant with respect to both the visit ratios and the mean service times, then R is a* fractional linear function *of the relative utilizations of the chain k customer, if the relative utilizations of all other chains are held constant.*

Proof:

This is easily seen to be true from Eq. (17). □

5 Applications of the Basic Results

In the previous section, we have shown that a Markov reward function is *marginally pseudo-linear* in the model parameters for a product-form queueing network. A pseudo-linear function is both pseudo-concave and pseudo-convex. Such a function of one variable is illustrated in Figure 4. (For a detailed characterization of pseudo-linear functions. the reader is referred to [BAZA79].) One important characteristic of a pseudo-linear function is that if $f(x)$ is pseudo-linear in the interval $[x_1, x_2]$, the minimal (or maximal) value of f in that interval must occur at an extreme point, namely, either x_1 or x_2. In higher dimensions, the scalar x is replaced by a vector $\vec{x}$, the interval is replaced by a convex polytope $\mathcal{F}$, and if $\vec{x}$ can be varied within the convex polytope, then f can be minimized (or maximized) at a vertex of the convex polytope. In the sections below, we discuss various ways in which the restriction of parameters to a convex polytope can arise and how the *basic results* (the insensitivity results of Section 3 and the pseudo-linearity results of Section 4) can apply. In Section 5.1, we

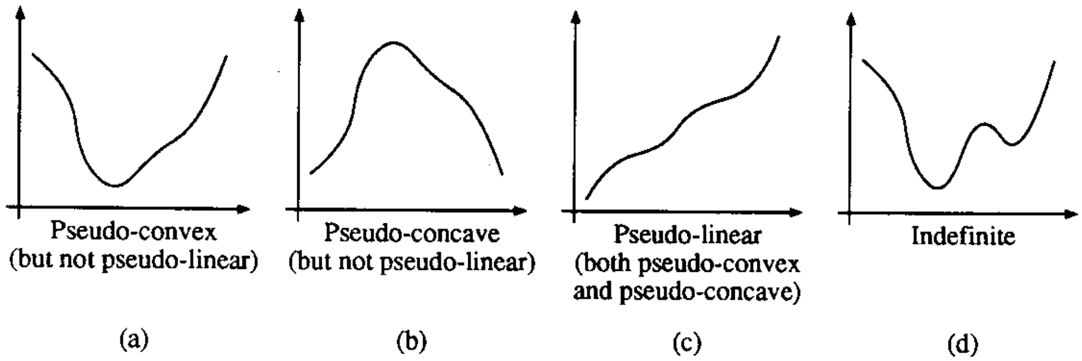


Figure 4: Non-pseudo-linear and Pseudo-linear Functions.

look at optimization problems where a set of design constraints determines the shape of the convex polytope. In Section 5.2, we look at the problem of obtaining performance bounds when the model parameters have bounded errors. In this case, the errors in the model parameters define the convex polytope. In Section 5.3, we look at the problem of obtaining performance bounds for models with a large number of similarly behaving customer chains. In this case, we artificially create the convex polytope so that the performance bounds can be obtained relatively inexpensively. A numerical example is given at the end of Section 5.3.

5.1 Optimization Problems

In [CHEN91], we investigate the problem of optimally routing customers in closed quasi-reversible networks. The results of [CHEN91] are briefly reviewed in Section 5.1.1 below.

5.1.1 Review of Previous Results

In [CHEN91], the queueing network models studied are closed multichain networks of quasi-reversible queues with Markovian routing for the customers. Only one customer is allowed in each chain (therefore, the words *customer* and *chain* are used interchangeably unless otherwise stated). The transition behavior of a chain k customer is described by a stochastic routing matrix $\mathbf{P}_k = [\![p^k_{cd}]\!]$. (Solving $\vec{\theta}_k\mathbf{P}_k = \vec{\theta}_k$ gives chain k's visit ratios $\vec{\theta}_k$). Since we are interested in optimal routing problems, not all the elements in $\mathbf{P}_k$ are fixed; we allow $p^k_{cd} = a^k_{cd} + v^k_{cd}$ (but subject to the constraints that $\sum_d p^k_{cd} = 1, \; \forall c$) where a^k_{cd} is a nonnegative constant and v^k_{cd} is a nonnegative variable. The problem of optimal routing is to find an assignment for all the v_{cd}^k's for all chains such that a given Markov reward function is optimized. In this case, each v_{cd}^k is a decision variable in the optimal routing problem. To simplify the discussion, we assume that all the a_{cd}^k's are zero. (Our results easily generalize to the case where a_{cd}^k can be nonzero.) The decision variables are all distinct and independent except that they are constrained to keep $\mathbf{P}_k$ stochastic. Let $R = \vec{\pi} \cdot \vec{R}$ be the vector notation for Eq. (4), and let $\mathbf{Q}$ denote the state transition rate matrix. The optimization problem can be specified (without loss of generality, assuming that we are minimizing the given reward function) as,

$$\begin{aligned} \text{min:} \quad & \vec{\pi} \cdot \vec{R} \\ \text{s.t.:} \quad & \vec{\pi}\, \mathbf{Q} \; = \; \vec{0} \end{aligned} \tag{21}$$

Clearly, Eq. (21) has a nonlinear objective function and linear constraints in the decision variables. It is not clear that Eq. (21) has an optimal vertex solution; however, in [CHEN91], we show hat it *does* have an optimal vertex solution in the following fashion.

Let the routing variables be fixed for all chains except for a tagged chain, say, chain k. The rate matrix $\mathbf{Q}$ can be partitioned into C_k parts according the class of the chain k customer. If we apply exact aggregation [COUR86] to the partitioned $\mathbf{Q}$ and obtain $\mathbf{Q}^{Ag}$, solving $\vec{w}\,\mathbf{Q}^{Ag} = \vec{0}$ gives the steady-state probability distribution of the chain k customer in all its classes.

Let's further assume that all the routing variables for chain k are fixed except for those that are associated with the routing probabilities when the chain k customer finishes service in a particular class, say, class c. All these routing variables are on the row of $\mathbf{Q}^{Ag}$ which corresponds to class c. The problem of finding an assignment for the v_{cd}^k's that minimizes (or maximizes) the total reward can be formulated as the optimization problem below.

$$\begin{aligned} \text{min:} \quad & R \\ \text{s.t.:} \quad & \textstyle\sum_d v_{cd}^k \; = \; 1 \\ & v_{cd}^k \geq 0 \quad \forall\, d \end{aligned} \tag{22}$$

Let $\mathbf{Q}^{Ag}(d)$ denote the matrix obtained from $\mathbf{Q}^{Ag}$ by setting v_{cd}^k to 1 and setting $v_{cd'}^k$ to 0 for all $d' \neq d$, let $\vec{w}(d)$ denote the solution to $\vec{w}(d)\,\mathbf{Q}^{Ag}(d) = \vec{0}$, and let its corresponding reward be denoted by $R(d)$. Using a lemma from [COUR86], it is shown in [CHEN91] that $\vec{w}$ is in the convex hull of $\{\vec{w}(d), d \in \mathcal{C}_k \text{ and } d \neq c\}$, or equivalently, $\vec{w} = \sum_d \beta_d \vec{w}(d)$ where

$\sum_d \beta_d = 1$. Combining this fact and Eq. (13), we obtain that,

$$\min_d \{R(d)\} \;\leq\; R \;\leq\; \max_d \{R(d)\} \tag{23}$$

Eq. (23) states that the optimal total reward can be obtained at a vertex of the solution space which is defined by the constraints in Eq. (22).

In [CHEN91], we show that the above result can easily be extended to the case where all the routing variables are allowed to vary for all chains. This is called the *Generalized Vertex Allocation Theorem.* Knowing that an optimal solution exists at a vertex of the solution space reduces the search space from a continuum to a finite number of points; however, solving the optimization problem can still be computationally prohibitive. In [CHEN91], we further study a special case where the total reward is a fractional linear function of the control variable and a gradient based algorithm can be used to obtain the optimal solution.

5.1.2 Discussion of the Previous Results

Optimal routing problems for closed queueing networks are closely related to stochastic optimization problems. If we allow state-dependent routing, where the routing decision of a customer can depend on the state of other queues (still having only one customer per chain), the optimal routing problem can easily be transformed into a stochastic optimization problem with infinite horizon and no discounting, and such a problem is known to have an optimal vertex solution [BERT87]. However, this type of routing is considered to be *dynamic routing* in the queueing literature. In [CHEN91], the routing of the customers is *static*, meaning that the routing decision must be *independent* of the state of other queues. A corresponding problem in stochastic optimization will constrain a set of states to have the same branching probabilities, and for such a type of problem we can not find any general results in the stochastic optimization literature stating that such a problem would have an optimal vertex solution. We have in fact observed experimentally that a small deviation from the quasi-reversible models we consider always leads to optimal solutions not obtainable at a vertex of the solution space.

5.1.3 Other Optimization Applications

In [CHEN91], the control variables for the optimization problem are the routing probabilities of the customer chains. If the control variables are instead the visit ratios of the customers, which are constrained linearly, then such an optimization problem would be formulated as,

$$\begin{array}{lll} \text{min:} & R & \\ \text{s.t.:} & \forall\, k, & \mathbf{A}_k\vec{\theta}_k \;\geq\; \vec{b}_k, \text{ and} \\ & & \vec{\theta}_k \geq \vec{0} \end{array} \tag{24}$$

where $\mathbf{A}_k$ is a matrix of arbitrary constants and $\vec{b}_k$ is a vector of arbitrary constants with respect to the visit ratios of the chain k customer. From Lemma 8, R is a fractional linear function of the visit ratios of any chain k customer for a quasi-reversible network, and therefore, Eq. (24) is a *fractional linear program* (characterized by the fact that the objective function is fractional linear and the constraints are linear), which is known to have an optimal vertex solution [BAZA79]. The maximizing version of the fractional linear program of Eq. (24) also has an optimal vertex solution. Using an argument similar to the one used to prove the *Generalized Vertex Allocation Theorem*, when the visit ratios of all chains are considered, an optimal reward can be obtained at a vertex of the search space.

In an optimal routing problem for a closed queueing model, the control variables are often the routing probabilities, which indirectly determine the visit ratios. (For open models, the control variables are often the routing probabilities *or* the visit ratios.) Even though it might be unnatural to be able to control the visit ratios directly in a closed model, as we will see in the section below, for certain applications, being able to solve such an optimization problem is useful for to obtaining bounds.

If the control variables are the mean service times (or relative utilizations) of the customers, which are constrained linearly, a similar optimization problem can be formulated as,

$$\begin{array}{lll} \text{min:} & R & \\ \text{s.t.:} & \forall\, k, & \mathbf{A}'_k\vec{T}_k \;\geq\; \vec{b}'_k, \text{ and} \\ & & \vec{T}_k \geq \vec{0} \end{array} \tag{25}$$

where elements of matrix $\mathbf{A}'_k$ and vector $\vec{b}'_k$ are arbitrary constants with respect to the mean service times (or relative utilizations) of the chain k customer. From Lemma 9, R is a fractional linear function of the mean service times (or relative utilizations) of any chain k customer for a pseudo-reversible network. Consequently, the optimal reward can be obtained at a vertex of the search space when the mean service times (or relative utilizations) of all chains are considered.

In certain optimization problems for closed queueing models, the mean service times of the customers are controlled indirectly through the speeds of the servers. Our results are not applicable in these cases since we require that the service times of different chains are independent of each other. Even though it might be unnatural to be able to control the service times directly, as we will see in the sections below, for certain applications, being able to solve such an optimization problem is useful for to obtaining bounds.

5.2 Error Bounds in the Presence of Parameter Uncertainties

Studies of the impact of parameter estimation errors in queueing network models are often done in the context of either *sensitivity analysis* or *workload characterization.* In sensitivity analysis, the sensitivity of a performance measure with respect to a certain parameter is defined in terms of the partial derivative of the measure with respect to that parameter [DESO88, MCKE84]. For example, the partial derivative of the throughput of, say, chain c customers at center j with respect to the visit ratios of chain c at center j can be expressed in terms of the throughput, the visit ratio, the mean queue length, and the mean queue length seen by an arriving customer of chain c at center j. In [GORD80], such an expression is used to bound the elasticity of throughputs with respect to visit ratios for multichain closed networks. Elasticity is defined as the ratio of the percent change in throughputs for a given percent change in visit ratios. In [DESO88] and [STRE86], it is shown that some partial derivatives can be easily obtained using recursive expressions similar to those used in the MVA algorithm [REIS80]. However, partial derivatives only give the rate of change of the performance measures for infinitesimal changes in the parameter values. If a performance measure possesses the convexity property, error bounds on the performance measures can be computed from partial derivatives. Figure 5(a) depicts the situation where the convexity of the performance measure f and the partial derivative can be used to obtain bounds. The straight line which is tangent to f in Figure 5(a) represents the partial derivative of

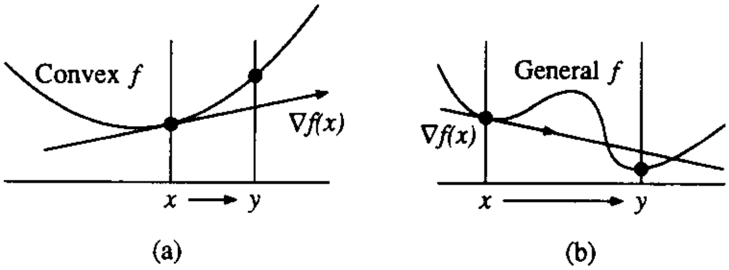


Figure 5: Relationship between partial derivative, convexity, and bounds.

f evaluated at point x, and it is denoted by $\nabla f(x)$. Since f is convex, for any point y, we have that $f(y) \geq f(x) + \nabla f(x)(y - x)$; therefore, $f(x) + \nabla f(x)(y - x)$ is a lower bound of f at point y. Similarly, when f is concave, the partial derivative can be used to obtain an upper bound. Figure 5(b) depicts the situation where f does not have the convexity property. In this example, f is convex around point x. However, for different values of y along the line $\nabla f(x)$, $f(x) + \nabla f(x)(y - x)$ can be either smaller than $f(y)$ or larger than $f(y)$. Therefore, the partial derivative can not be used to obtain bounds. The convexity property usually exists only in open networks [FRAT73, DESO84] or single chain closed networks [KOBA83, SURI83, SURI85, TAY85]. For multichain closed networks, examples can be easily constructed to show that most performance measures do not enjoy the convexity property. In general, it is very difficult to obtain error bounds for multichain closed networks.

In the area of workload characterization, the parameters of a queueing network model are often obtained through measurements, and measured data inherently introduces errors. The uncertainties in model output can be categorized as modeling errors or parameter estimation errors. Modeling errors arise when a queueing network model is not a good representation of the system it is trying to model. For example, suppose we know that a particular server in the system serves the arriving jobs in first-in-first-out order, and we know that the service times for the jobs are exponentially distributed, but each job class has a different mean. In this case, a modeling error is introduced if we model the server by a product-form fixed rate server with a certain mean service time for all the customers. In contrast, if all jobs have the same mean service requirement at the server in the above example, but we do not know the exact value of the mean service requirement, a parameter estimation error is introduced if the value for the mean service requirement we assign is in error. Most work in this area tries to answer the question of how to minimize modeling errors [FERR83, SERA85], i.e., how to choose a "good" model structure. Some work in characterizing the parameter estimation errors is done in the context of sensitivity analysis, e.g., [GORD80] and [TAY85]. We assume that the underlying workload can be accurately modeled by a closed multichain product-form network and concentrate on the case where the parameters have estimation errors that can arise due to measurement errors or due to to the fact that the workload of the system is not constant with respect to time. In these cases, one would like to know the impact of the errors on the accuracy of the solution of the queueing network model.

5.2.1 Optimization vs. Error Bounds

In Section 5.1, we considered the optimization problem where the ranges of values for the routing probabilities of closed queueing networks are given as constraints which define the search space. Solving the optimization problem gives the set of values for the routing probabilities that optimize the objective function. Suppose that the values for the routing probabilities are not given exactly but are known to have errors; we can view this as if the routing probabilities are given in terms of a range of values that each can assume. In this case, if we formulate two optimization problems (one maximizing and one minimizing the same objective function) and let the ranges of values for the routing probabilities define the search space, solving the optimization problems would give error bounds on the performance measure of interest. From this point of view, there is not much difference between solving an optimization problem and computing error bounds. This idea can be easily extended to the case where the values of the visit ratios (or the mean service times or relative utilizations) are known within certain bounds for closed quasi-reversible (or pseudo-reversible) networks.

It is important to note that, the above procedure is valid, as long as the model parameters that contain errors are only associated with pseudo-reversible queues, the model *can* contain other quasi-reversible queues which are not pseudo-reversible (such as a first-come-first-server queue).

5.2.2 First-come-first-serve Queues

Although queues that serve customers in the first-come-first-serve (FCFS) order are quasi-reversible but not pseudo-reversible, the approach described in this section for obtaining error bounds is still applicable with small modifications[2] if the service time[3] or the relative utilization of a FCFS queue is known to have bounded error.

If a quasi-reversible FCFS queue is replaced with a fixed-rate pseudo-reversible queue, such as a processor-sharing (PS) queue, the most common performance measures of the modified model (throughputs, mean queue length, server utilization, etc.) are identical to those of the original model [BASK75]. Therefore, if the service time (or relative utilization) of a quasi-reversible FCFS queue has bounded errors, say, $\tau_j^{\mathrm{min}} \leq T_j \leq \tau_j^{\mathrm{max}}$, we can substitute a PS queue, and for every chain k that visits the FCFS queue in the original model, add the constraint that $\tau_j^{\mathrm{min}} \leq T_{jk} \leq \tau_j^{\mathrm{max}}$. Solving the optimization problems as described previously for this modified model gives error bounds for the original model for the following two reasons. Firstly, the constraint region in the modified model contains the constraint region in the original model. Secondly, for the most common performance measures mentioned above, for every possible solution in the original model, there is a *feasible* solution in the modified problem that has identical performance measures.

5.2.3 Multiple Customers per Chain

So far we have assumed that every chain has only one customer. In general, a model with multiple customers per chain can be represented by an equivalent model with only a single customer per chain. By equivalence, we mean that all the performance characteristics are identical. We also have assumed that the model parameters of a customer are independent of those of other customers. Allowing multiple customers per chain means that model parameters of customers in the same chain must be identical. This destroys the independence. For an optimal routing problem, if we assume that all the customers are independent when

[2] The modifications work in this case because we know the algebraic form of the marginal state probability for FCFS queues. It does not work in general for non-pseudo-reversible queues.

[3] The service times of all customers on a quasi-reversible FCFS queue must be identical. Therefore, one can express errors (or sensitivity) in the speed of the server as errors in the service time.

in fact they are dependent, we will get erroneous results. However, for error bounds computation, if we assume that all the customers are independent, the bounds obtained are still *valid.* Since we have loosened the constraints, the bounds will be looser.

If there exist dependencies among model parameters of different customers other than the equality type as in the case of multiple customers per chain, the above argument is still valid. We can still assume that the model parameters among customers are independent, and use the optimization method to obtain bounds.

In the next section, we further exploit the idea that computing bounds and solving optimization problems are just different ways of viewing the same problem.

5.3 Bounds on the Error Introduced by Clustering of Chains

There are a variety of techniques for solving closed product-form queueing network models approximately. Many of them are based on the MVA equations [BARD79, CHAN82, HSIE89, SCHW79, ZAHO88] or the decomposition methods [COUR77, DESO84a]. *Clustering* is a statistical method for grouping together objects that are "similar" in their behavior [MACQ67, SEBE84]. Clustering can be applied to queueing networks to group jobs that have similar resource usage patterns and replace them with a set of identical jobs [ARTI85, DOWD92, TRIP85, ZAHO80]. Since no error bounds can be computed from approximate solution methods, considerable effort needs to be spent on validating an approximation method (against a simulation or, when feasible, an exact solution) in order to gain confidence. Nevertheless, there would be no guarantee that the method would work well when a new model is solved. Therefore, performance bounds are often more desirable than approximation methods. In this section, we concentrate on obtaining bounds for closed product-form networks with a large number of chains.

Knowing that the cost of solving a closed network is an exponential function of the number of chains, reducing the number of chains can drastically reduce the cost of solving the model. This makes the technique of clustering very attractive, since it can potentially reduce the number of chains by replacing a number of different jobs by a set of identical jobs. We illustrate the potential benefit of clustering by the following example. Figure 6(a) shows a central-server model of a computer system. There are 48 users in the system. All the

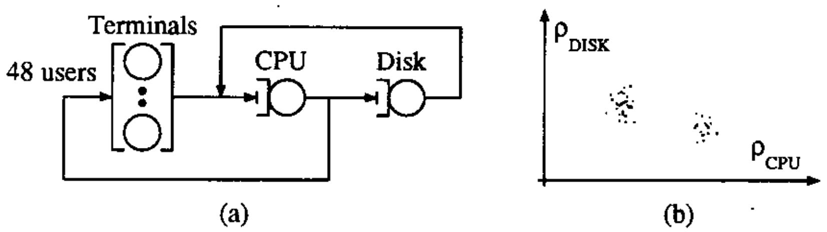


Figure 6: Central server model example.

users are observed to have identical mean think time at the terminals; however, their relative loads are different at the CPU and the disk. Figure 6(b) plots the relative loads at the CPU and the disk for each user. The loading suggests that the original model should have 48 chains with one customer per chain. In this case, $(N+1)^K$ (the main factor in the space and time complexities for solving a K-chain closed queueing network model with N customers each) evaluates to 0.28×10^{15} (with $N = 1$ here). Therefore, solving the model exactly is infeasible. Figure 6(b) emphasizes that the loading of the customers forms two clusters. The clusters are depicted in Figure 7(a). In Figure 7(b), 30 of the users are clustered together to form one chain and the other 18 users are clustered together to form another chain. Let's further assume that the modeler is interested in obtaining the actual CPU utilization when three identical customers are added to the network, as shown in Figure 7(c). The three chain model of Figure 7(c) will hopefully be a good approximation of the actual model (which has 49 chains). The factor mentioned above evaluates to a manageable $31 \times 19 \times 4 = 2356$ for Figure 7(c).

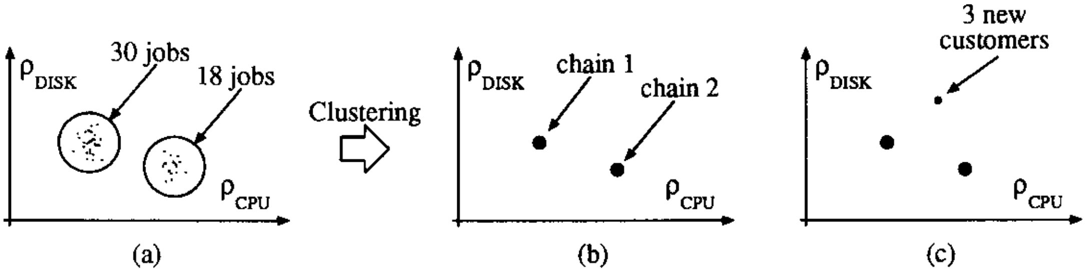


Figure 7: Clustering of 48 jobs into 2 chains.

The main problem with clustering is that, like any approximation method, it is difficult to determine how good the approximation is. It is therefore desirable to be able to obtain error bounds when clustering is applied.

In this section, we show how error bounds can be obtained when clustering is applied. The key is to apply the basic results in a novel way. First, we define the model and the notation used in the remainder of the section. Then we present how our basic results can be used to obtain bounds when clustering is applied.

5.3.1 Closed Queueing Network Model

The queueing network models considered here are multichain networks of quasi-reversible or pseudo-reversible queues (depending on the parameters being considered) with Markovian routing for the customers. We are interested in obtaining bounds on performance measures which can be expressed as Markov reward functions. The procedure described in this section is valid for any performance measure expressible as a Markov reward function.

Let N_k denote the number of customers in chain k, and let $\vec{y}_k$ denote the *resource vector* for chain k. (In general, a resource vector is an n-tuple characterizing a workload element [ARTI85].) We only consider *homogeneous* resource vectors in which all components of a resource vector refer to the same type of model parameter. For a network of quasi-reversible queues, $\vec{y}_k$ denotes a vector of visit ratios for a set of centers. For a network of pseudo-reversible queues, $\vec{y}_k$ denotes either a vector of service requirements or relative utilizations at a set of centers. For ease of exposition, we assume that, 1) customers do not change class and 2) $\forall\, k$, $\|\vec{y}_k\| = J$ (the number of elements in $\vec{y}_k$ is J). Both of these conditions can be relaxed; however, the notation will get unnecessarily complex. Given these assumptions, $\vec{y}_k$ can be written as $(y_{1k}, y_{2k}, \ldots, y_{Jk})$. Let $Y = \{\vec{y}_k\} = \{\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_K\}$ denote the set of all resource vectors or the *workload* of the system, and let $R(Y)$ denote the total reward for workload Y. We also refer to $R(Y)$ as the *exact solution* of the model.

The K customer chains are partitioned into M clusters, denoted by B_1,B_2,...,B_M. We will use the notation $\xi(k)$ to denote the cluster to which the chain k customer belongs. Usually, the partition is formed by heuristically grouping chains with "similar" resource vectors. There exist many statistical techniques for partitioning objects according to the numerical values of their resource vectors [SEBE84]. Different partitionings of the chains into clusters may drastically affect the spread of the performance bounds. However, we are not as concerned with the way the partitions are obtained as we are with determining the error bounds for a given partitioning. We assume that the partitions are given as part of the problem specification. Let Y_m denote the set of resource vectors of cluster m, or $Y_m = \{\vec{y}_k \mid k \in B_m\}$. Since the B_m's are mutually exclusive, we can also denote Y by $\{Y_1, Y_2, \ldots, Y_M\}$.

In order to take advantage of the fact that the computational complexity of the exact solution procedure decreases combinatorially as the number of chains is reduced, customers in the same cluster are "aggregated" into a single chain. The resource vector of the aggregated chain for cluster m is denoted by $\vec{y}(m)$. In general, $\vec{y}(m)$ need not have any relationship to the resource vectors of the chains in cluster m. Nevertheless, a reasonable $\vec{y}(m)$ would be a *convex combination* of the resource vectors of the chains in cluster m. Or,

$$\vec{y}(m) \;=\; \sum_{k \in B_m} r_k N_k \vec{y}_k$$

where $r_k \geq 0$ is a constant weighting factor for chain k, $\sum_{k \in B_m} r_k N_k = 1$. For example, if $\vec{y}(m)$ corresponds to the *centroid* of cluster m, then $r_k = 1 / \sum_{k' \in B_m} N_{k'}$. When all clusters are replaced by their respective aggregates, the resource vector is denoted by $\hat{Y}$,

$$\hat{Y} \;=\; \{\,\vec{y}(m)\,\} \;=\; \{\,\underbrace{\vec{y}(1), \ldots, \vec{y}(1)}_{\|B_1\|},\, \underbrace{\vec{y}(2), \ldots, \vec{y}(2)}_{\|B_2\|},\, \ldots,\, \underbrace{\vec{y}(M), \ldots, \vec{y}(M)}_{\|B_M\|}\,\} \tag{26}$$

We will call $\hat{Y}$ the *aggregated workload* of the system and call $R(\hat{Y})$ a *clustering approximation* of the model for a given aggregated workload $\hat{Y}$.

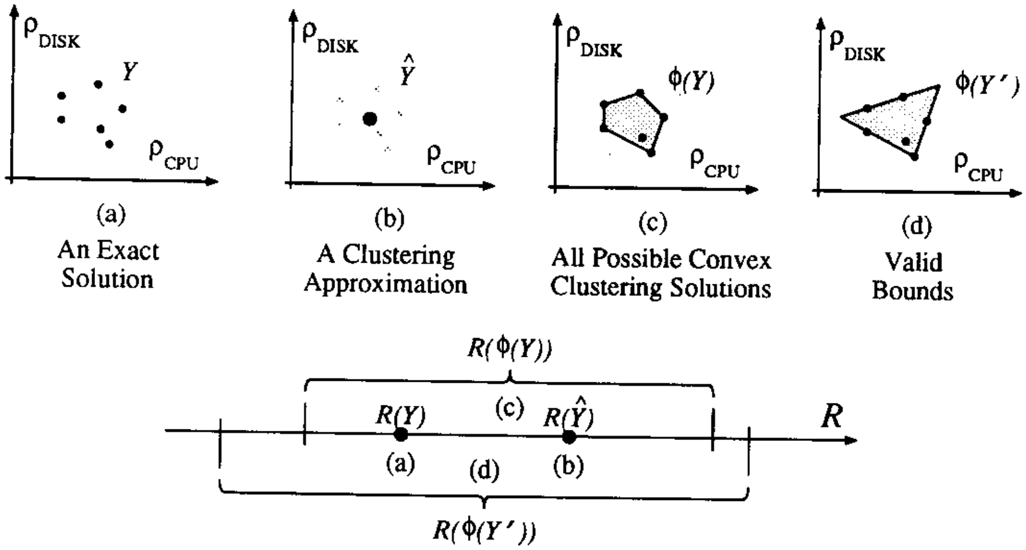
We assume, as before and without loss of generality, that each of the original chains has only one customer (or $N_k = 1$, $\forall\, k$), and use "chain" and "customer" interchangeably, unless otherwise stated, to simply our notation. With this assumption, for any cluster m, $\vec{y}(m) = \sum_{k \in B_m} r_k \vec{y}_k$ and $\sum_{k \in B_m} r_k = 1$.

In the following section, we describe an algorithm for computing bounds for $R(Y)$ for a class of clustering schemes in which the resource vector of an aggregated chain is constrained to be a convex combination of the resource vectors of the original chains within the same cluster.

5.3.2 Application of the Basic Results

In this section, we describe how error bounds can be obtained when clustering of customers is applied in the manner described in Section 5.3.1. To be space efficient, we only describe the main ideas. For a comprehensive treatment of the approach, the reader is referred to [CHEN92]. The approach presented in this section does *not* produce bounds for a *specific* clustering method, e.g., the centroid method, but rather it produces bounds for *all* possible clustering methods where the resource vector of the aggregated chain for a cluster is a convex combination of resource vectors of the original chains within the same cluster. This approach is easily extended to obtaining bounds for any specific clustering method in Section 5.3.4.

Figure 8 illustrates the relationship between the exact solution, a clustering approximation, and two different bounds for a simple example where there is a single cluster with 6 resource vectors of cardinality 2 (represented as points in the two dimensional space) within the cluster. The total reward evaluated at various workloads and the workload ranges are illustrated in the lower half of Figure 8. Part (a) of Figure 8 shows the original 6 resource vectors and their corresponding total reward $R(Y)$. In part (b), the 6 resource vectors are aggregated into one with the aggregated workload denoted by $\hat{Y}$. In part (c), all possible convex combinations of the original resource vectors are shown as the shaded area; the *con-*



(a) An Exact Solution

(b) A Clustering Approximation

(c) All Possible Convex Clustering Solutions

(d) Valid Bounds

Figure 8: Relationship between exact solution, clustering approximation, and bounds.

vex hull of Y is denoted by $\phi(Y)$. The set of values of R for all possible $\hat{Y} \in \phi(Y)$ is denoted by $R(\phi(Y))$ and is shown to include both $R(Y)$ of part (a) and $R(\hat{Y})$ of part (b). The reason why this is true is described below.

Part (c) of Figure 8 represents the following two optimization (one minimizing and one maximizing) problems. Consider adding the six customers (chains) one at a time, each one is constrained to having its resource vector lie within $\phi(Y)$. The objective functions are the same as the original reward function. Clearly, both Y and $\hat{Y}$ are in the search space of both optimization problems. Therefore,

$$\min_{Z \in \phi(Y)} R(Z) \;\le\; R(Y),\; R(\hat{Y}) \;\le\; \max_{Z \in \phi(Y)} R(Z) \tag{27}$$

For a detailed proof, the reader is referred to [CHEN92].

In part (d) of Figure 8, the convex hull of Y is relaxed to have a triangular shape which includes the convex hull of Figure 8(c), and it is denoted by $\phi(Y')$, where Y' is the set of vertices of the triangular area, and $\phi(Y) \subset \phi(Y')$. Using the same argument as described above, we can easily see that $R(\phi(Y)) \subset R(\phi(Y'))$. In other words, if we can compute the upper and lower bounds of $R(\phi(Y'))$, then we would obtain bounds for the total reward for any convex clustering of the original resource vectors. Bounds obtained from part (d) are looser than the ones from part (c); however, in many cases, the bounds for part (d) can be obtained at a relatively low cost. This is the topic of the following section. For a discussion on how to obtain the convex hull of Figure 8(c), the reader is referred to the references in [AURE91]. For a discussion on how to obtain the regions of Figure 8(d), the reader is referred to [CHEN92].

5.3.3 Low Cost Clustering Bounds

In Section 4, we showed that the reward function R is *marginally* fractional linear in the resource vectors. If the resource vectors of all but one chain are fixed, then we can use efficient greedy decent algorithms to compute the bounds. In obtaining clustering bounds, the resource vectors of all chains are not fixed; in this case, R is *marginally* fractional linear, but *not* fractional linear. In general, all vertex solutions must be examined in order to find the bounds. Consider the following model with only one cluster having 48 customers and the resource vectors having only two components. If the convex hull has V vertices, then there is a total of V^{48} possible vertex solutions. Even for $V=3$ (the best case in two dimensions), there are almost 0.8×10^{24} vertex solutions. However, if we examine carefully what these vertex solutions correspond to, the number of vertex solutions can be drastically reduced.

Continuing with the above example with $K=48$, $J=2$, $M=1$, and $V=3$, Figure 9 shows a triangular area that contains the convex hull of the resource vectors of the 48 customers. Solving the optimization problems by constraining the resource vector of each customer to be inside the triangular area, we get the bounds described in Eq. (27). Furthermore, in an optimal vertex solution, the resource vector of any customer is replaced by a resource vector that corresponds to one of the vertices of the triangular area. This is illustrated in Figure 9 where a customer's resource vector is "moved" to one of the vertices. Instead of thinking about which of the 3 vertices each customer "moves" to in order to obtain the optimal solution, we can think of deciding, "How many customers should be at vertices 1, 2, and 3?" The number of possible solutions to this question is $\binom{48+3-1}{3-1}=1225$. Furthermore, in order to solve each combination of the customer populations, a three-chain closed network needs to be analyzed. In this case, it becomes feasible to find the optimal solutions, and therefore, obtain the clustering bounds. This method easily extends to models with multiple clusters (but at higher costs).

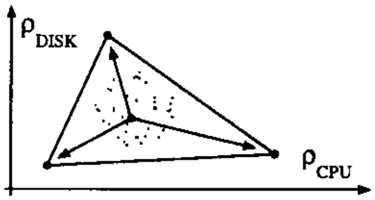


Figure 9: Moving a customer to a vertex.

5.3.4 Bounds for Specific Clustering Methods

The method described above produces bounds for all possible clustering methods where the resource vector of the aggregated chain for a cluster is a convex combination of resource vectors of the original customers within the same cluster. If a specific clustering method, e.g., the centroid method, is specified, then we can obtain tighter bounds, at an additional computational expense. The additional cost comes from needing to solve queueing models with one additional chain. The idea is illustrated in Figure 10 below. Parts (a) and (b) of Figure 10 are identical to parts (a) and (b) of Figure 8. In Figure 10(c), the triangle of Figure 8(d) is further divided into three sections by the point that represents the specific clustering method. In Section 5.3.2, we mentioned that the optimization procedure for obtaining the bounds which corresponds to Figure 8(d) can be thought of as adding customers one at a time and deciding which one of the vertices of the triangle the customer would end up. In this case, when we add the customer, we need to consider which point the customer corresponds to. For example, if we are adding the customer which corresponds to point 1 (or point 2) in Figure 10, it can choose among the three vertices of the top region in Figure 10(c). If we are adding the customer which corresponds to point 3, it can choose among the three vertices of the right region, and so on. It is clear that the bounds are still valid because the exact solution and the clustering approximation are still in the search space of the optimization problems. Also, it is clear that the bounds are tighter since the new constraints are tighter than the previous ones. However, a four chain queueing model needs to be solved in this case to obtain a vertex solution. The number of vertex solutions that need to be computed depends on how the resource vectors are distributed.

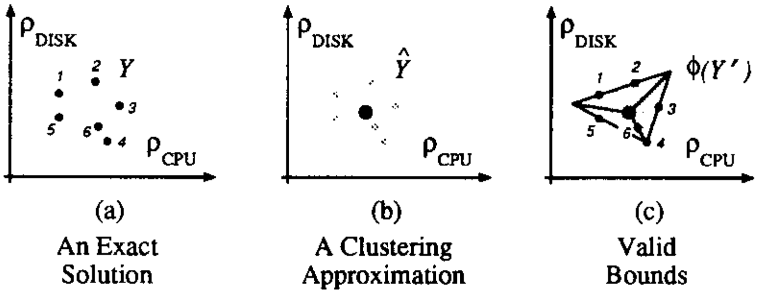


Figure 10: Specific Clustering Method Applied

5.3.5 A Numerical Example

In this section, we use a numerical example to illustrate our approach. The model considered is a closed network of 12 fixed rate servers. There are 48 customers. Their relative loads are identical on 10 of the servers (numbered from 3 to 12), and their values are show in Table 1. The two servers where the relative loads for the customers are different are referred as

Server	3	4	5	6	7	8	9	10	11	12
ρ_{jk}	0.2251	0.7750	0.2875	0.4472	0.6258	0.9276	0.1762	0.2465	0.7261	0.3625

Table 1: Relative loads for all customers; $3 \leq j \leq 12$.

servers 1 and 2. The relative loads for the customers on servers 1 and 2 are shown in Table 2 (assuming that the reference server is server 1). We are interested in obtaining bounds on the total throughput of the customers at server 1. The customers are modeled as closed chains with 1 customer in each chain. The workload of the model (resource vectors of the customers) is plotted in Figure 11(a). By clustering all the customers into one chain and using the *centroid* of the original model as the load vector for the clustered model, we get a single chain model with its load vector having a value of (ρ_1, ρ_2) = (0.2500, 0.6579). The total workload for the clustered model is illustrated in Figure 11(b). (ρ_j, for $3 \leq j \leq 12$ remains the same as in Table 1.) Using the MVA algorithm to solve the clustered model (12 servers, 1 chain with 48 customers) gives a throughput of 1.07743. On a SPARC10,

0.2171, 0.6750	0.2096, 0.6870	0.2577, 0.6786	0.2692, 0.6369	0.2874, 0.6745	0.2446, 0.6354
0.2733, 0.6260	0.2394, 0.6777	0.2845, 0.6576	0.2716, 0.6083	0.2456, 0.6110	0.2545, 0.6391
0.2569, 0.6959	0.2868, 0.6163	0.2276, 0.6260	0.2924, 0.6436	0.2789, 0.6128	0.2082, 0.6941
0.2026, 0.6154	0.2382, 0.6155	0.2529, 0.6877	0.2431, 0.6264	0.2314, 0.6770	0.2107, 0.6771
0.2705, 0.6219	0.2762, 0.6412	0.2649, 0.6930	0.2502, 0.6687	0.2436, 0.6608	0.2577, 0.6633
0.2463, 0.6632	0.2138, 0.6961	0.2144, 0.6447	0.2325, 0.6953	0.2358, 0.6398	0.2101, 0.6955
0.2985, 0.6576	0.2866, 0.6150	0.2909, 0.6651	0.2064, 0.6955	0.2966, 0.6786	0.2805, 0.6571
0.2283, 0.6963	0.2579, 0.6437	0.2375, 0.6923	0.2029, 0.6769	0.2723, 0.6591	0.2426, 0.6642

Table 2: Relative loads (ρ_1, ρ_2) for 48 customers.

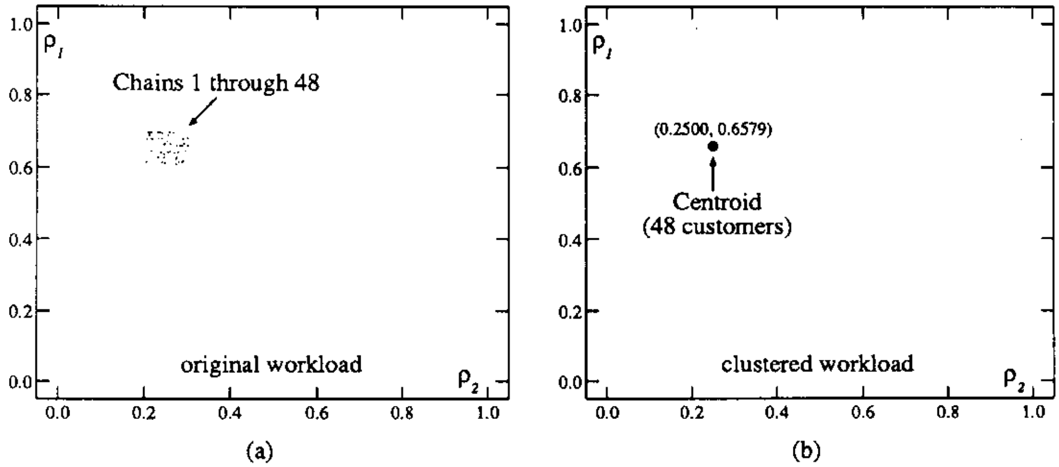
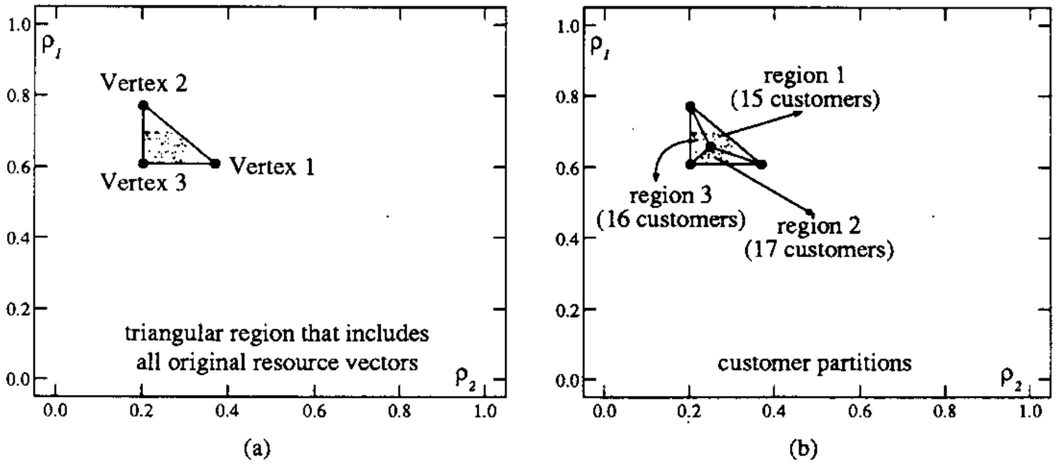


Figure 11: Relative loads for 48 chains at servers 1 and 2.

running SunOS 4.1.3, solving the above model takes 0.1 second (including the time to load the program from the UNIX shell).

In order to use our approach to obtain the clustering bounds, we need to construct a three-vertex region containing the resource vectors of the 48 customers. Using a simple algorithm described in Appendix A, we obtained three vertices having relative loads (0.3669, 0.6083), (0.2026, 0.7726), and (0.2026, 0.6083) (these points are labeled as vertices 1, 2, and 3, respectively, in Figure 12(a)). Using the approach illustrated in Figure 10(c), these three vertices and the centroid form three regions. There are 15, 17, and 16 customers fall into regions 1, 2, and 3 respectively, as shown in Figure 12(b). There are 18360 ways of distributing the 48 customers among the 4 vertices (while keeping the number of customers inside each region the same as specified above). For every way of distributing the customers, we need to solve a model with 12 servers and 4 chains (the total number of customers is 48). If we use the approach of Section 5.3.3, we need to solve 1225 3-chain models. We decide to solve the smaller models first. If the bounds are too loose, we would then attempt to solve the larger models.



(a)

(b)

Figure 12: Clustering bounds example.

After solving all 1225 models, we find that a minimum throughput of 1.07650 is achieved when all customers have their resource vectors located at Vertex 2. A maximum throughput of 1.07753 is achieved when all customers have their resource vectors located at Vertex 3. Since we do not know the exact solution for the original model, we define the percent error for a reward function R as,

$$Err(R) \;=\; \frac{R_{\max} - R_{\min}}{R_{\max} + R_{\min}} \times 100$$

Using the above formula, an error of 0.05% is achieved using our approach. With such a tight bound, it is unnecessary to solve the larger models. It took almost two days to solve all 1225 3-chain models on the SUN4 mentioned above. The numerical results are summarized in Table 3.

Method	K	Models		N_{V1}	N_{V2}	N_{V3}	N_{centroid}	Λ_1	Error	Time
centroid	1	1		--	--	--	48	1.07743	unknown	0.1 second
general-bound	3	1225	min	0	48	0	--	1.07650	0.05%	1.9 days
			max	0	0	48	--	1.07753		
centroid-bound	4	18360	min	?	?	?	?	?	$\leq$ 0.05%	?
			max	?	?	?	?	?		

Table 3: Summary of the numerical results.

The number of models need to be solved grows exponentially as the number of chains increases; in addition, the cost for solving each model increases. Therefore, one should first compute the the less expensive bounds. If the bounds are too loose, one can then compute the bounds for a specific clustering method.

5.3.6 Discussion

Many bounding methods in the queueing literature are based on the MVA equations, and therefore, they can only obtain bounds for a few (although important) mean performance measures. They also have the restriction that they are only valid for restricted types of product-form networks. The approach developed in this section has wide applicability to arbitrary closed product-form queueing networks and to a wide variety of performance measures.

It should also be noted that the procedure developed in this section drastically cuts down on the number of closed chains, so it becomes feasible to compute the error bounds. Even though there may be many models (each with a small number of chains) to solve, with the exponential speed increase in the microcomputer technology and the ease of solving independent models in parallel, our approach can become more attractive.

6 Conclusion

In this article, we show that for a closed multichain quasi-reversible network with a single customer per chain, the state probability distribution, conditioned on a customer being at any server in any class, is *independent* of the routing behavior of that customer. Such a result is obtained from the most basic property of quasi-reversible queues, namely, *partial balance*. We also define a class of restricted quasi-reversible queues, which we call *pseudo-reversible* queues; a pseudo-reversible queue satisfies the *station balance* condition. From the station balance condition, we derive the result that for a closed multichain pseudo-reversible network with a single customer per chain, the state probability distribution, conditioned on a customer being at any server in any class, is *independent* of the mean service requirements of that customer.

From the insensitivity results, we then show that for a quasi-reversible network, the mean waiting time of a customer at a server is *independent* of the routing of that customer. We also show that for a pseudo-reversible network, the mean waiting time of a customer at a server is *proportional* to the mean service time of that customer at the same server. Furthermore, we show that for a quasi-reversible network, a Markov reward function on the network states is a *marginally pseudo-linear* function of the visit ratios of a customer. Similarly, for a pseudo-reversible network, a (more restrictive) Markov reward function on the network states is a *marginally pseudo-linear* function of the mean service times of a customer.

We then show that these basic results can be applied to several important applications. We show that for optimization problems, the search space can be restricted to a finite number of points which correspond to the vertices of the solution space of the original problem. For models where the parameters are known to have bounded errors, the basic results can be applied to obtain upper and lower bounds on performance measures. The key is to exploit the relationship between solving optimization problems and obtaining error bounds. We also apply the basic results in a novel way to obtain bounds when clustering techniques are applied. In this case, the key idea is to create a search space for an optimization problem from the original model. If this search space is created properly, bounds can be obtained in a relatively inexpensive way. Finally, we show that certain sensitivity analysis results in the literature can be generalized using these basic results.

Appendix A

An Algorithm for Obtaining Vertices for A Cluster

In this appendix, we describe a simple algorithm for obtaining $L+1$ vertices for a cluster of customers having resource vectors with cardinalities of L. This algorithm is used to obtain 3 vertices for the example described in Section 5.3.5.

Let $\vec{y}_k = (y_{1k}, y_{2k}, \ldots, y_{Lk})$ be the resource vector of customer k, where K is the number of customers being considered for clustering. We first look at a *constrained* problem of obtaining L vectors in an L dimensional space where the convex hull of the solution contains all the resource vectors, subject to the constraint that $\sum_j y_{jk} = 1, \forall k$. Then we show that any unconstrained problem in an L dimensional space can be transformed into a constrained problem in an $L+1$ dimensional space. Then we can use the method for solving the constrained problem to obtain $L+1$ vectors whose convex hull contains the transformed resource vectors. Then we can apply a reverse transformation to obtain the solution to the unconstrained problem.

Let $u_j = \min_k y_{jk}$ for $1 \leq j \leq L$ be the minimum of the j^{th} component of the resource vectors of all the customers. Clearly, $\sum_{j=1}^{L} u_j \leq 1$. Let $\vec{u} = (u_1, u_2, \ldots, u_L)$, and let $u^* = 1 - \sum_{j=1}^{L} u_j$ be the "deficiency" of vector $\vec{u}$. Let's further define $\vec{x}_j$ to be the vector $\vec{u}$ with all the "deficiency" added to component j; or equivalently,

$$\vec{x}_j = \vec{u} + u^* \vec{e}_j \tag{28}$$

where $\vec{e}_j$ is the unit vector with the j^{th} component equal to 1 and zeroes everywhere else. Let $X = \{\vec{x}_1, \ldots, \vec{x}_L\}$, and as before, $Y = \{\vec{y}_1, \ldots, \vec{y}_K\}$. We have the following lemma.

Lemma 11 *If the components of X are obtained using Eq. (28), then $\phi(Y) \subset \phi(X)$.*

Proof:

Showing that $\phi(Y) \subset \phi(X)$ is equivalent to showing that $\vec{y}_k \in \phi(X), \forall k$. This is also equivalent to showing that every $\vec{y}_k$ can be expressed as a convex combination of the $\vec{x}_j$'s.

Since $u_j = \min_k y_{jk}$, we have that $y_{jk} \geq u_j$. Therefore, we can write,

$$y_{jk} = u_j + \alpha_j u^* \tag{29}$$

where $\alpha_j \geq 0$. Summing over all j, we have

$$\sum_j y_{jk} = \sum_j (u_j + \alpha_j u^*)$$

Simplifying the above equation, we get,

$$1 = (1 - u^*) + u^* \sum_j \alpha_j$$

Therefore, $\sum_j \alpha_j = 1$. Eq. (29) in vector form becomes,

$$\begin{aligned}\vec{y}_k &= \vec{u} + \sum_j (\alpha_j u^* \vec{e}_j) \\ &= \vec{u} \sum_j \alpha_j + \sum_j (\alpha_j u^* \vec{e}_j) \\ &= \sum_j \alpha_j \vec{x}_j\end{aligned}$$

where Eq. (28) is used in the last step. We have just shown that, for any k, $\vec{y}_k$ can be expressed as a convex combination of the $\vec{x}_j$'s. □

The next step is to show how to transform an unconstrained problem in an L dimensional space to a constrained problem in an $L+1$ dimensional space. We still use $\vec{y}_k = (y_{1k}, y_{2k}, \ldots, y_{Lk})$ to denote the resource vector of customer k. Summing the components of the resource vector for each chain and let $y^* = \max_k \sum_j y_{jk}$ be the maximum of the sums. Every resource vector $\vec{y}_k$ can be augmented with $v_k = y^* - \sum_j y_{jk}$. Let

$$\vec{z}_k = \left(\frac{y_{1k}}{y^*}, \ldots, \frac{y_{Lk}}{y^*}, \frac{v_k}{y^*} \right) \tag{30}$$

The cardinality of $\vec{z}_k$ is $L+1$, and the components of $\vec{z}_k$ are all nonnegative and sum to 1. Let $Z = \{\vec{z}_1, \ldots, \vec{z}_K\}$. Given a set of resource vectors Y, Z is uniquely determined. Furthermore, there is a one-to-one mapping between a point in the Y space and a point in the Z space. To transform a point in the Y space to a point in the Z space, the same procedure for obtaining $\vec{z}_k$ can be followed. To transform a point in the Z space to a point in the Y space, just truncate the last component and multiply each of the remaining component by y^*.

Let's summarize how a convex hull having $L+1$ vertices can be obtained for a given set of K resource vectors, each with cardinality L.

Step 1: Compute y^* and the v_k's. Transform each resource vector $\vec{y}_k$ to $\vec{z}_k$ using Eq. (30).

Step 2: Compute u^* and the u_j's. Obtain $L+1$ points in the Z space using Eq. (28).

Step 3: Use the reverse transform (truncation) to transform each $\vec{x}_j$ back into the Y space to obtain the $L+1$ points whose convex hull contains the resource vectors of all the chains.

References

[ARTI85] H. P. Artis. Workload characterization using SAS PROC FASTCLUS. In G. Serazzi, editor, *Workload Characterization of Computer Systems and Computer Networks*, pages 21–32. North-Holland, 1985.

[AURE91] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[BARD79] Y. Bard. Some extensions to multiclass queueing network analysis. In M. Arato, A. Butrimenko, and E. Gelenbe, editors, *Performance of Computer Systems*, pages 51-61. North-Holland, 1979.

[BASK75] F. Baskett, K. M. Chandy, R. R. Muntz, and F. Palacios. Open, closed and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April, 1975.

[BAZA79] M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms.* John Wiley and Sons, Inc., 1979.

[BERT87] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models.* Prentice Hall, 1987.

[CHAN77] K. M. Chandy, J. H. Howard Jr., and D. F. Towsley. A characterization of product-form queueing networks. *Journal of the ACM*, 24(2):250–263, April 1977.

[CHAN82] K. M. Chandy and D. Neuse. Linearizer: A heuristic algorithm for queueing network models of computer systems. *Communications of the ACM*, 25:126–134, 1982.

[CHAN83] K. M. Chandy and A. J. Martin. A characterization of product-form queueing networks. *Journal of the ACM*, 30:286–299, April 1983.

[CHEN91] W. C. Cheng and R. R. Muntz. Optimal routing for closed queueing networks. *Performance Evaluation*, 13(1):3–17, September 1991.

[CHEN92] W. C. Cheng. Optimization, performance bounds, and approximations in queueing networks. Technical Report Ph.D. Dissertation, University of California at Los Angeles, July 1992.

[COUR77] P. J. Courtois. *Decomposability – Queueing and Computer System Applications.* Academic Press, New York, 1977.

[COUR86] P. J. Courtois and P. Semal. Computable bounds for conditional steady-state probabilities in large Markov chains and queueing models. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):926–937, September, 1986.

[COX55] D. R. Cox. A use of complex probabilities in the theory of stochastic processes. In *Cambridge Phil. Soc.*, volume 51, pages 313–319, 1955.

[DESO84] E. de Souza e Silva and M. Gerla. Load balancing in distributed systems with multiple classes and site constraints. In *Proceedings of PERFORMANCE'84 and 1984 ACM SIGMETRICS Conf.*, pages 17–33, 1984.

[DESO84a] E. de Souza e Silva, S. S. Lavenberg, and R. R. Muntz. A perspective on iterative methods for the approximate analysis of closed queueing networks. In G. Iazeola, P. J. Courtois, and A. Hordijk, editors, *Mathematical Computer Performance and Reliability*, pages 225–244. North Holland, 1984.

[DESO88] E. de Souza e Silva and R. R. Muntz. Simple relationships among moments of queue lengths in product form queueing networks. *IEEE Transactions on Computers*, 37(9):1125–1129, September 1988.

[DESO89] E. de Souza e Silva and S. S. Lavenberg. Calculating joint queue length distributions in product form queueing networks. *Journal of the ACM*, 36:194–207, 1989.

[DESO90] E. de Souza e Silva and R. R. Muntz. Queueing networks: Solutions and applications. In H. Takagi, editor, *Stochastic Analysis of Computer and Communication Systems*, pages 319–399. Elsevier Science Publishing Company, Inc., 1990.

[DOWD92] L. W. Dowdy, B. M. Carlson, A. T. Krantz, and S. K. Tripathi. Single class bounds of multi-class networks. *Journal of the ACM*, 39(1):188–213, 1992.

[FERR83] D. Ferrari, G. Serazzi, and A. Zeigner. *Measurement and Tuning of Computer Systems.* Prentice Hall, 1983.

[FRAT73] L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method – an approach to store-and-forward communication network design. *Networks*, 3:97–133, 1973.

[GORD80] K. D. Gordon and L. W. Dowdy. The impact of certain parameter estimation errors in queueing network models. In *Proceedings of 1980 ACM SIGMETRICS Conf.*, pages 3–9, 1980.

[HSIE89] C. T. Hsieh and S. S. Lam. PAM - a noniterative approximate solution method for closed multichain queueing networks. *Performance Evaluation*, 9:119–133, 1989.

[KELL79] F. P. Kelly. *Reversibility and Stochastic Networks.* John Wiley and Sons, 1979.

[KOBA83] H. Kobayashi and M. Gerla. Optimal routing in closed queueing networks. *ACM Transactions on Computer Systems*, 1:294–310, 1983.

[LAVE80] S. S. Lavenberg and M. Reiser. Stationary state probabilities of arrival instants for closed queueing networks with multiple types of customers. *Journal of Applied Probability*, 17:1048–1061, 1980.

[MACQ67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[MCKE84] J. McKenna and D. Mitra. Asymptotic expansions and integral representations of moments of queue lengths in closed Markovian networks. *Journal of the ACM*, 31:346–360, 1984.

[MUNT72] R. R. Muntz. Poisson departure processes and queueing networks. Technical Report RC 4145, IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., 1972.

[NELS92] R. Nelson. The mathematics of product form queueing networks. *to appear in ACM Computing Surveys*, 1992.

[REIS80] M. Reiser and S. S. Lavenberg. Mean value analysis of closed multichain queueing networks. *Journal of the ACM*, 27:313–322, 1980.

[REIS81] M. Reiser. Mean-value analysis and convolution method for queue-dependent servers in closed queueing networks. *Performance Evaluation*, 1:7–18, 1981.

[SCHW79] P. Schweitzer. Approximate analysis of multiclass closed networks of queues. In *International Conference on Stochastic Control and Optimization*, Amsterdam, 1979.

[SEBE84] G. A. F. Seber. *Multivariate Observations.* John Wiley and Sons, Inc., 1984.

[SERA85] G. Serazzi, editor. *Workload Characterization of Computer Systems and Computer Networks.* North-Holland, 1985.

[STRE86] J. Strelen. A generalization of mean value analysis for higher moments: Moment analysis. In *Proceedings of PERFORMANCE'86 and 1986 ACM SIGMETRICS Conf.*, pages 129–140, 1986.

[SURI83] R. Suri. Robustness of queueing network formulas. *Journal of the ACM*, 30:564–594, 1983.

[SURI85] R. Suri. A concept of monotonicity and its characterization for closed queueing networks. *Operations Research*, 33(3):606–624, May 1985.

[TAY85] Y. C. Tay and R. Suri. Error bounds for performance prediction in queueing networks. *ACM Transactions on Computer Systems*, 3(3):227–254, Auguest 1985.

[TRIP85] S. K. Tripathi and L. W. Dowdy. Workload representation and its impact on the performance prediction using queueing network models. In G. Serazzi, editor, *Workload Characterization of Computer Systems and Computer Networks*, pages 159–178. North-Holland, 1985.

[TRIP88] S. K. Tripathi and C. M. Woodside. A vertex-allocation theorem for resources in queueing networks. *Journal of the ACM*, 35(1):221–230, January 1988.

[ZAHO80] J. Zahorjan. The approximate solution of large queueing network models. Technical Report Ph.D. Dissertation, University of Toronto, 1980.

[ZAHO88] J. Zahorjan, D. E. Eager, and H. M. Sweillam. Accuracy, speed, and convergence of approximate mean value analysis. *Performance Evaluation*, 8:255–270, 1988.