Discrete Event Dyn Syst (2014) 24:133–151
DOI 10.1007/s10626-013-0167-1

Event-based optimization of admission control in open queueing networks

Li Xia

Received: 14 April 2012 / Accepted: 2 May 2013 / Published online: 11 May 2013

Abstract In this paper, we use the event-based optimization framework to study the admission control problem in an open Jackson network. The external arriving customers are controlled by an admission controller. The controller makes decision only at the epoch when an event of customer arrival happens. Every customer in the network obtains a fixed reward for each service completion and pays a cost with a fixed rate during its sojourn time (including waiting time and service time). The complete information of the system state is not available for the controller. The controller can observe only the number of total customers in the network. Based on the property of closed form solution of Jackson networks, we prove that the system performance is monotonic with respect to the admission probabilities and the optimal control policy has a threshold form. That is, when the number of total customers is smaller than a threshold, all of the arriving customers are admitted; otherwise, all are rejected. The sufficient condition and necessary condition of optimal policy are also derived. Furthermore, we develop an iterative algorithm to find the optimal policy. The algorithm can be executed based on a single sample path, which makes the algorithm online implementable. Simulation experiments are conducted to demonstrate the effectiveness of our approach.

Keywords Admission control · Event-based optimization · Queueing network · Performance potential

This work was partly supported by the 111 International Collaboration Project (B06002), National Natural Science Foundation of China (61203039, 60736027), the Specialized Research Fund for the Doctoral Program of Higher Education (20120002120009), Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

L. Xia (✉)
Center For Intelligent and Networked Systems (CFINS),
Department of Automation and TNList, Tsinghua University,
Beijing 100084, China
e-mail: xial@tsinghua.edu.cn

1 Introduction

For most of the service systems in the real world, the resource of the system is always limited. When customers (or service requests) waiting for service become crowded, it may not be good for the system to admit more customers. In order to guarantee the service quality of current customers, operators usually adopt an admission control policy to prevent the system from congestion. Admission control problem discusses how to optimize the admission policy to achieve an optimal system performance. This topic attracts intensive research efforts since QoS (quality of service) is becoming a vital performance metric for many service systems.

The problem of admission control exists in many networking systems. For example, call admission control is an important issue in cellular networks. It aims to improve the quality of voice or data communication and the utilization of channels with the constraint of a small blocking probability (Fang 2002; Ghaderi and Boutaba 2006; Niyato and Hossain 2005; Wu et al. 2002). Admission control is also important for packet-switched networks as it can relieve the traffic congestion and maximize the throughput of the network (Gao et al. 2005; Li et al. 2004; Lima et al. 2007). Queueing model is a widely used model to formulate and analyze the admission control problem in networking systems. Admission control of queueing models attracts a fair amount of attentions in the research field of queueing theory (Gross et al. 2008; Stidham 2002). *Naor model* is a classical model to study how to control the admission of customers to achieve the *social* or *individual* optimum (Naor 1969). For simple queueing systems, such as M/M/1, it is proved that the optimal control has a monotonic structure (Hordijk and Spieksma 1989; Koole 1998; Stidham 1985). For the situation where the admission controller has incomplete information of the system, similar structure property of optimal control policy is also derived for M/M/1 queue (Lin and Ross 2001). Most of the work in the literature models the queueing system as a Markov decision process (MDP) and uses backward induction (dynamic programming) to study the structure property of optimal policy (Altman 2001; Stidham 2002). However, for queueing networks, since the system state has multiple dimensions, it is too complicated to study the optimality equation and explore the possible structure of optimal policy.

In this paper, we discuss the admission control problem in an open Jackson network with a capacity limit (Chen and Yao 2001; Jackson 1963). The externally arriving customers are controlled by an admission controller which determines the customer's admission or rejection. For the customers admitted to the network, each customer obtains a fixed reward when it finishes its service at a server. Meanwhile, each customer in the network has to pay a cost which is proportional to its sojourn time (including the waiting time and service time). We assume that there is no penalty for the customers rejected by the network.

Different from the previous studies about admission control problem, the admission controller does not know the full information about the system state, i.e., the distribution of customers among servers. The controller can only observe the number of total customers in the network to determine the admission of customers. That is, the controller is based on the partial information of the system to make decision. The controller uses admission probabilities $a(n)$'s to control the arriving customers, where n is the number of total customers currently in the network. Since n changes only when the events of customers' entrance to or departure from the network happen, the admission probabilities $a(n)$'s are adjustable only when these events happen. We call it *event-based optimization* of this admission control problem (Cao 2005, 2007). That is, we have to make decisions only at the epoch when certain events happen. However, in a standard MDP, the control policy is a mapping from the state space to the action space (Puterman 1994). The decision of MDP is made on every system state. Since the number of event categories is usually much less than that of system states, the complexity of event-based optimization is usually simpler than that of MDP.

Since the admission controller cannot observe the full information of the system, it is difficult to obtain the global optimum only with partial information. In this paper, we utilize the property of product-form solution of Jackson networks to derive a special property, which describes that the conditional probability of system state with respect to the number of total customers in the network is independent of the admission probability. With this property, we derive a very concise equation to describe the performance difference under two sets of admission probabilities. We prove that the system performance is monotonic with respect to the admission probability. We further prove that the optimal admission control policy has a threshold form, i.e., when the number of total customers n is less than a certain threshold, the controller admits all arrivals; otherwise, rejects all. Based on the difference equation, we develop an iterative algorithm to find the optimal threshold. We further implement this algorithm on a single sample path, which means that our algorithm is online implementable in the practice.

The remainder of the paper is organized as follows. In Section 2, we present the problem description and the model formulation. In Section 3, we discuss how to solve this admission control problem. The structure properties of optimal policy are derived. An efficient and iterative algorithm is also proposed to find the optimal policy. In Section 4, we conduct simulation experiments to demonstrate the effectiveness of our approach. Finally, we conclude the paper in Section 5.

2 Problem formulation of admission control

Consider a standard open Jackson network with M servers (Gross et al. 2008; Jackson 1963). The external customer arrivals to the network follow a Poisson distribution with rate λ. The arriving customers are admitted or rejected by the admission controller. The arriving customers admitted to the network join server i with a probability q_{0i}, $i = 1, 2, \ldots, M$, and $\sum_{i=1}^{M} q_{0i} = 1$. The service time of each server is exponentially distributed. We denote μ_i as the service rate of server i, $i = 1, 2, \ldots, M$. When a server is busy, the arriving customers to that server will wait in the buffer. The buffer of server is adequate. The service discipline is FCFS (first come first serve). If a customer completes its service at server i, it leaves server i and joins server j with a routing probability q_{ij}, and it exits from the network with a probability q_{i0}, $i, j = 1, 2, \ldots, M$. Without loss of generality, we assume that $q_{ii} = 0$. Obviously, we have $\sum_{j=0}^{M} q_{ij} = 1$ for all i. The number of customers at server i is denoted as n_i and the number of total customers in the network is denoted as n, $n := \sum_{i=1}^{M} n_i$. We assume that the capacity of the entire network has an upper limit denoted as N. That is, when the total number of customers $n \geq N$, any newly arriving customer from exterior will be rejected. This open Jackson network with a capacity limit is also called *semi-open* network (Chen and Yao 2001). The system state is denoted as $\boldsymbol{n} := (n_1, n_2, \ldots, n_M)$. The state space is denoted as $\mathcal{S} := \{\text{all } \boldsymbol{n} : \sum_{i=1}^{M} n_i \le N\}$.

The admission control is executed as follows. When an external customer arrives at the network, it will be admitted to the network with an admission probability $a(n)$, $n = 0, 1, \ldots, N-1$. In other words, we can observe only the partial information about the network, i.e., the number of total customers currently in the network n, to adjust the admission probabilities. The full information of the system state $\boldsymbol{n}$ is not observable for the controller. Obviously, $0 \le a(n) \le 1$, $n = 0, 1, \ldots, N-1$, and $a(N) = 0$. Figure 1 is an illustration of the queueing network with admission control.

When a customer finishes its service at a server, it gets a fixed reward R. Meanwhile, each customer has to pay a cost C per unit of its sojourn time (including the waiting time and service time). The reward function is defined as the profit rate (total rewards minus costs per unit time). Thus, the reward function is denoted as

$$f(\boldsymbol{n}) := R \sum_{i=1}^{M} 1_{n_i>0}\mu_i - C \sum_{i=1}^{M} n_i, \tag{1}$$

where $1_{n_i>0}$ is an indicator function defined as, if $n_i > 0$, $1_{n_i>0} = 1$; otherwise, $1_{n_i>0} = 0$. Please note, the results in this paper are still valid for other $f(\boldsymbol{n})$'s if $f(\boldsymbol{n})$ is irrelevant to $a(n)$'s. We denote $\boldsymbol{n}_t$ as the system state at time t. The long-run average performance of the system is written as follows.

$$\eta = E\{f(\boldsymbol{n}_t)\} = \lim_{T\to\infty} \frac{1}{T} \int_0^T f(\boldsymbol{n}_t)dt, \tag{2}$$

where the second equality holds for ergodic systems.

We observe that when the number of total customers increases and the network becomes more crowded, the waiting time of customers will increase and the cost of customers will also increase. On the other hand, when the network becomes more crowded, the servers will become more busy and the rewards obtained by the network will also increase. Thus, there is a tradeoff between the total costs

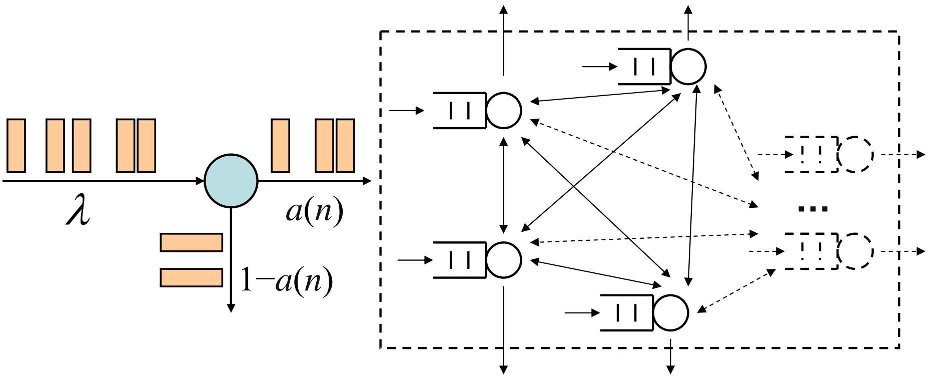


Fig. 1 An open Jackson network with admission control

and rewards. For simplicity, we denote the row vector of admission probabilities as $\boldsymbol{a} := (a(0), a(1), \ldots, a(N-1))$. Since $a(n) \in [0, 1]$, the value domain of $\boldsymbol{a}$ is an N-dimensional real number space $[0, 1]^N$. The goal of admission control problem is to find the optimal admission probabilities $\boldsymbol{a}^* := (a^*(0), a^*(1), \ldots, a^*(N-1))$, which maximize the long-run average performance of the entire network. That is,

$$\boldsymbol{a}^* = \arg \max_{\boldsymbol{a} \in [0,1]^N} \{\eta\}. \tag{3}$$

Since this Jackson network with controlled admission probabilities is a Markov system, it seems that we can use MDP theory to solve this admission control problem. However, we further observe that this problem is different from the traditional MDP. In this admission control problem, the decision is made only at the epochs when the events of external arrivals happen. At the epochs of customer transitions among servers inside the network, we cannot make admission decision. This is different from the traditional MDP where we have to make decision at each state transition epoch. Moreover, the policy in this problem is parameterized (parameters $\boldsymbol{a}$). This is not a standard policy in MDP theory where we require the action (admission probability) be chosen independently at every state. For example, for two states $\boldsymbol{n} = (3, 2, 5)$ and $\boldsymbol{n}' = (2, 6, 2)$, the actions (admission probability $a(10)$) should be identical at these two states since the numbers of total customers at these two states are both 10. This violates the requirement of "state-independent action" in standard MDP and we cannot directly apply the classical MDP methods, such as policy iteration, to solve this problem.

In this paper, we use the framework of event-based optimization to formulate this admission control problem (Cao 2005; Jia 2011). We denote a transition from state u to v as a symbol $\langle u, v \rangle$. An event is defined as a set of transitions satisfying certain common properties. The total event space is defined as $\mathcal{E} := \{\varnothing, \langle u, v \rangle : u, v \in \mathcal{S}\}$, where $\varnothing$ denotes null element for the sake of completeness. A certain category of events e is a subset of $\mathcal{E}$. For example, the event of an arriving customer admitted by the network is defined as $e_a := \{\langle \boldsymbol{n}, \boldsymbol{n}_{+i} \rangle : \boldsymbol{n} \in \mathcal{S}, i = 1, 2, \ldots, M\}$, where $\boldsymbol{n}_{+i}$ is called the neighboring state of $\boldsymbol{n}$ with definition $\boldsymbol{n}_{+i} := (n_1, \ldots, n_i + 1, \ldots, n_M)$, $i = 1, 2, \ldots, M$. Similarly, an event of an arriving customer rejected by the network is defined as $e_r := \{\langle \boldsymbol{n}, \boldsymbol{n} \rangle : \boldsymbol{n} \in \mathcal{S}\}$. Therefore, the event of external customer arrival can be defined as $e_A := e_a \cup e_r$. We further denote $e_A(n)$ as the events of external arrivals and the number of total customers at the arrival epoch equals n, $n = 0, 1, \ldots, N-1$. We denote $\mathcal{E}_A := \{\text{all } e_A(n) : n = 0, 1, \ldots, N-1\}$ as the event space of customer arrivals. Therefore, the event-based optimization of this admission control problem aims to find a mapping from the space $\mathcal{E}_A$ to $[0, 1]$, which makes the corresponding system performance η maximal.

In the event-based optimization framework, we make decision only when certain events (e.g., customer arrivals e_A happens in this paper) happen. However, in the MDP framework, we have to make decision at every state. The size of event space is usually much smaller than that of state space, e.g., $|\mathcal{E}_A| = N \ll |\mathcal{S}| = \binom{M+N}{N}$ in this paper. That is, the event-based optimization has a much smaller search space than that of MDP. The detailed analysis results are presented in the next section.

3 Optimal admission control policy and algorithms

In this section, we derive the difference equation for the admission control problem and identify the special properties of optimal policy. We also develop an iterative algorithm to efficiently find the optimal control policy. We further implement the algorithm online based on the system sample path.

3.1 Difference equation and optimality properties

First, we give a brief introduction about the theory of performance potential, which is proposed by Cao (2007). Difference equation is the key idea of this theory and it quantifies the performance difference of Markov systems under any two parameter settings or policies. This theory provides a straightforward way to find the optimal policy of Markov systems.

Consider a Markov process $\boldsymbol{X} = \{X_t, t \geq 0\}$, where $X_t \in \mathcal{S}$ is the system state at time t and $\mathcal{S}$ is the state space. Without loss of generality, we assume $\mathcal{S}$ is finite and its size is S. We further write $\mathcal{S} = \{1, 2, \dots, S\}$. The infinitesimal generator (or called transition rate matrix) of this Markov process is denoted as matrix $\boldsymbol{B}$. The steady-state distribution is denoted as a row vector $\boldsymbol{\pi} := (\pi(1), \pi(2), \dots, \pi(S))$. We further denote the reward function $\boldsymbol{f}$ as a column vector, i.e., $\boldsymbol{f} := (f(1), f(2), \dots, f(S))^T$. Obviously, we have $\boldsymbol{Be} = \boldsymbol{0}$, $\boldsymbol{\pi B} = \boldsymbol{0}$, and $\boldsymbol{\pi e} = 1$, where $\boldsymbol{e}$ is an S-dimensional column vector whose elements are all 1. The long-run average performance of the system is denoted as η and we have $\eta = \boldsymbol{\pi f}$.

In an MDP, system parameters or policies will affect the element values of infinitesimal generator $\boldsymbol{B}$ and reward function $\boldsymbol{f}$. When the system parameters (or policies) change, the corresponding infinitesimal generator and reward function will change and they are denoted as $\boldsymbol{B}'$ and $\boldsymbol{f}'$, respectively. The difference of the long-run average performance η associated with these two parameter settings is quantified by the following difference equation (Cao 2007).

$$\eta' - \eta = \boldsymbol{\pi}' \left[(\boldsymbol{B}' - \boldsymbol{B})\boldsymbol{g} + (\boldsymbol{f}' - \boldsymbol{f}) \right], \tag{4}$$

where the row vector $\boldsymbol{\pi}'$ is denoted as the steady-state distribution of the system with $\boldsymbol{B}'$ and $\boldsymbol{f}'$, the S-dimensional column vector $\boldsymbol{g}$ is called the *performance potential* of the system with $\boldsymbol{B}$ and $\boldsymbol{f}$. The element of $\boldsymbol{g}$, $g(u)$ and $u \in \mathcal{S}$, is defined as

$$g(u) = \lim_{T \to \infty} E \left\{ \int_0^T [f(X_t) - \eta] dt \Big|_{X_0 = u} \right\}, \tag{5}$$

which can be estimated from the system sample path of $\boldsymbol{X}$ (Cao 2007).

For the admission control problem described in Section 2, this system has Markovian property. The change of admission probabilities induces the change of $\boldsymbol{B}$ of the Markov system, thus it also induces the performance change of the Markov system. Below, we use the performance difference Eq. 4 to analyze the change of system performance which is affected by the admission probabilities $\boldsymbol{a}$.

Since the reward function is defined as Eq. 1, we can see that $\boldsymbol{f}$ does not change under different policies (different admission probabilities $\boldsymbol{a}$). With the property of the open Jackson network, we know that the structure of $\boldsymbol{B}$ is as follows (Xia et al. 2009). For a particular state $\boldsymbol{n} \in \mathcal{S}$, we have $B(\boldsymbol{n}, \boldsymbol{n}) = -a(n)\lambda - \sum_{i=1}^{M} 1_{n_i > 0}\mu_i$;

$B(\boldsymbol{n}, \boldsymbol{n}_{+i}) = a(n)\lambda q_{0i}$, for $i = 1, \ldots, M$; $B(\boldsymbol{n}, \boldsymbol{n}_{-i,+j}) = 1_{n_i>0}\mu_i q_{ij}$, for $i = 1, \ldots, M$ and $j = 0, \ldots, M$; $B(\boldsymbol{n}, :) = 0$, for all the other situations. Note that $\boldsymbol{n}_{-i,+j} := (n_1, \ldots, n_i - 1, \ldots, n_j + 1, \ldots, n_M)$ is called the neighboring state of state $\boldsymbol{n}$.

When the admission probabilities change from $\boldsymbol{a}$ to $\boldsymbol{a}'$, the corresponding infinitesimal generator will change from $\boldsymbol{B}$ to $\boldsymbol{B}'$, while the reward function $\boldsymbol{f}$ has no changes. With the structure of $\boldsymbol{B}$ as above, we see that only the terms related to $a(n)$'s have changes. Thus, we rewrite the difference Eq. 4 as follows.

$$\begin{aligned} \eta' - \eta &= \boldsymbol{\pi}'(\boldsymbol{B}' - \boldsymbol{B})\boldsymbol{g} \\ &= \sum_{n=0}^{N-1}[a'(n) - a(n)]\lambda \sum_{\boldsymbol{n}\in\mathcal{S}_n} \pi'(\boldsymbol{n}) \left\{ \sum_{i=1}^{M} q_{0i}[g(\boldsymbol{n}_{+i}) - g(\boldsymbol{n})] \right\} \\ &= \sum_{n=0}^{N-1} \pi'(n)[a'(n) - a(n)]\lambda \sum_{\boldsymbol{n}\in\mathcal{S}_n} \pi'(\boldsymbol{n}|n) \left\{ \sum_{i=1}^{M} q_{0i}[g(\boldsymbol{n}_{+i}) - g(\boldsymbol{n})] \right\}, \end{aligned} \tag{6}$$

where $\pi'(n)$ is the marginal probability that the number of total customers in the network equals n and $\mathcal{S}_n$ is a set of states where the number of total customers in the network equals n. Obviously, we have $\pi'(n) = \sum_{\boldsymbol{n}\in\mathcal{S}_n} \pi'(\boldsymbol{n})$ and

$$\pi'(\boldsymbol{n}|n) = \frac{\pi'(\boldsymbol{n}, n)}{\pi'(n)} = \frac{\pi'(\boldsymbol{n})}{\sum_{\boldsymbol{n}\in\mathcal{S}_n} \pi'(\boldsymbol{n})}, \quad \forall \boldsymbol{n} \in \mathcal{S}_n. \tag{7}$$

Below, we study a special property of conditional probability $\pi(\boldsymbol{n}|n)$ in the Jackson network. As described in Section 2, our queueing system is an open Jackson network with a capacity limit N. Actually, this semi-open network is equivalent to a closed Jackson network with $M + 1$ servers and N customers, where the external customer arrival process can be modeled as a virtual server (Chen and Yao 2001). This virtual server is indexed as server 0 and the number of customers in server 0 is defined as $n_0 := N - \sum_{i=1}^{M} n_i = N - n$. Therefore, the number of customers in this equivalent network always equals N. The service rate of server 0 is denoted as $\mu_{0,n_0} = a(n)\lambda = a(N - n_0)\lambda$, which is load-dependent. We further study the conditional probability $\pi(\boldsymbol{n}|n)$ with this equivalent closed Jackson network.

It is well known that the steady-state distribution of closed Jackson network has the following product-form solution (Gross et al. 2008).

$$\pi(\boldsymbol{n}) = \frac{1}{G} \frac{v_0^{n_0}}{\prod_{k=1}^{n_0} \mu_{0,k}} \prod_{i=1}^{M} \left(\frac{v_i}{\mu_i} \right)^{n_i}, \tag{8}$$

where v_i is the visit ratio of server i and G is a normalization constant which can be obtained with the law of total probability. Visit ratio v_i is determined by the following traffic equations.

$$v_i = \sum_{j=0}^{M} v_j q_{ji}, \quad i = 0, \ldots, M. \tag{9}$$

Since $n_0 = N - n$ and $\mu_{0,k} = a(N-k)\lambda$, we substitute them into Eq. 8 and obtain

$$\pi(\boldsymbol{n}) = \frac{1}{G}\frac{1}{\prod_{k=1}^{N-n} a(N-k)}\left(\frac{v_0}{\lambda}\right)^{N-n}\prod_{i=1}^{M}\left(\frac{v_i}{\mu_i}\right)^{n_i}$$
$$= \frac{1}{G}\frac{1}{\prod_{k=n}^{N-1} a(k)}\left(\frac{v_0}{\lambda}\right)^{N-n}\prod_{i=1}^{M}\left(\frac{v_i}{\mu_i}\right)^{n_i}, \tag{10}$$

Therefore, the conditional probability $\pi(\boldsymbol{n}|n)$ can be written as

$$\pi(\boldsymbol{n}|n) = \frac{\pi(\boldsymbol{n})}{\sum\limits_{\boldsymbol{n}\in\mathcal{S}_n}\pi(\boldsymbol{n})} = \frac{\frac{1}{\prod_{k=n}^{N-1}a(k)}\left(\frac{v_0}{\lambda}\right)^{N-n}\prod_{i=1}^{M}\left(\frac{v_i}{\mu_i}\right)^{n_i}}{\sum\limits_{\boldsymbol{n}\in\mathcal{S}_n}\frac{1}{\prod_{k=n}^{N-1}a(k)}\left(\frac{v_0}{\lambda}\right)^{N-n}\prod_{i=1}^{M}\left(\frac{v_i}{\mu_i}\right)^{n_i}}. \tag{11}$$

For any state $\boldsymbol{n} \in \mathcal{S}_n$, the number of total customers n is a fixed constant. Therefore, the first parts of both numerator and denominator in Eq. 11 are common factors and they can be canceled. Therefore, Eq. 11 is rewritten as

$$\pi(\boldsymbol{n}|n) = \frac{\prod_{i=1}^{M}\left(\frac{v_i}{\mu_i}\right)^{n_i}}{\sum\limits_{\boldsymbol{n}\in\mathcal{S}_n}\prod_{i=1}^{M}\left(\frac{v_i}{\mu_i}\right)^{n_i}}. \tag{12}$$

Please note that in the above analysis we focus on the steady-state distribution. The probabilities $\pi(\boldsymbol{n})$ and $\pi(\boldsymbol{n}|n)$ require the system state $\boldsymbol{n}$ be ergodic. We do not discuss the situation that $\boldsymbol{n}$ is transient since the transient behavior has no effect on the long-run average performance η. From Eq. 12, we observe that $\pi(\boldsymbol{n}|n)$ has no relation to the admission probabilities $a(n)$. Therefore, for the ergodic state $\boldsymbol{n}$, the conditional probability $\pi(\boldsymbol{n}|n)$ always remains constant when the admission probabilities $a(n)$'s have changes. That is,

$$\pi'(\boldsymbol{n}|n) = \pi(\boldsymbol{n}|n), \text{ for different } a(n)\text{'s}, n = 0, \ldots, N-1. \tag{13}$$

Property 13 describes a special property of Jackson networks and it can greatly simplify the optimization of this admission control problem. By applying Eq. 13, we can rewrite the difference Eq. 6 as below.

$$\eta' - \eta = \sum_{n=0}^{N-1}\pi'(n)[a'(n) - a(n)]D(n), \tag{14}$$

where $D(n)$ is called *aggregated performance potential rate* which is defined as

$$D(n) := \lambda\sum_{\boldsymbol{n}\in\mathcal{S}_n}\pi(\boldsymbol{n}|n)\left\{\sum_{i=1}^{M}q_{0i}[g(\boldsymbol{n}_{+i}) - g(\boldsymbol{n})]\right\}. \tag{15}$$

We see that $D(n)$ depends only on the system with the current policy. $D(n)$ can be estimated based on the current sample path. Details will be discussed later in Section 3.2.

Equation 14 is very concise and it quantifies the change of system performance when the admission probabilities $\boldsymbol{a}$ have changes. From the right-hand side of Eq. 14, we find that the performance change equals the summation of a series of products of three terms, $\pi'(n)$ is a marginal probability of the perturbed system and it is always positive for ergodic states; $a'(n) - a(n)$ is the change of admission probability and it is a known parameter; $D(n)$ is the aggregated performance potential rate of the current system and it can be calculated or estimated based on the current system sample path. In summary, Eq. 14 gives a clear picture about the influence of admission probabilities on the system performance. Below, we use Eq. 14 to analyze this admission control problem. Some theorems and special properties of this optimization problem are derived with a straightforward way.

First, we study the relation between the system performance η and the admission probabilities $a(n)$'s. We obtain the following theorem.

Theorem 1 *System performance η is monotonic with respect to the admission probabilities $a(n)$'s, $n = 0, 1, \ldots, N-1$.*

Proof Without loss of generality, we consider the situation that only one admission probability has changes. Suppose that one particular admission probability $a(k)$ changes to $a'(k)$ and other $a(n)$'s, $n \neq k$, remain the same. With Eq. 14, we obtain the change of the system performance as below.

$$\eta' - \eta = \pi'(k)[a'(k) - a(k)]D(k). \tag{16}$$

Reversely, we consider the situation where the admission probability is changed from $a'(k)$ to $a(k)$ and other $a(n)$'s remain the same. Similarly, we rewrite the equation of performance difference (Eq. 14) as below.

$$\eta - \eta' = \pi(k)[a(k) - a'(k)]D'(k). \tag{17}$$

Comparing Eqs. 16 and 17, we obtain the following equation

$$\frac{D'(k)}{D(k)} = \frac{\pi'(k)}{\pi(k)} > 0, \tag{18}$$

since $\pi(k)$ and $\pi'(k)$ are probabilities which are always positive. Therefore, Eq. 18 means that the sign of $D(k)$ remains the same no matter what value the admission probability $a(k)$ is.

With the difference Eq. 16, we can easily obtain the following equation

$$\frac{\Delta\eta}{\Delta a(k)} = \pi'(k)D(k), \tag{19}$$

where $\Delta\eta = \eta' - \eta$ and $\Delta a(k) = a'(k) - a(k)$. When $a'(k) \to a(k)$, we derive the following derivative equation

$$\frac{d\eta}{da(k)} = \pi(k)D(k). \tag{20}$$

Since $\pi(k)$ is always positive and the sign of $D(k)$ remains the same under different values of $a(k)$, the sign of derivative $\frac{d\eta}{da(k)}$ will also remain unvaried. Therefore, η is monotonic with respect to $a(k)$ and the theorem is proved. □

Theorem 1 describes an important property of this admission control problem. This monotonicity property is straightforwardly derived based on the difference equation and it can help simplify the optimization process. With Theorem 1, we directly obtain the following corollary.

Corollary 1 *The optimal admission probability is either* 0 *or* 1. *That is,* $a^*(n) \in \{0, 1\}$, $n = 0, 1, \ldots, N-1$.

Proof This corollary follows directly Theorem 1. We give a simple proof as follows to present the main idea. With Theorem 1, we see that η is monotonic with respect to $a(n)$'s. If η is monotonically increasing with respect to $a(n)$, then $a^*(n)$ is 1. Otherwise, if η is monotonically decreasing with respect to $a(n)$, $a^*(n)$ is 0. Therefore, $a^*(n) \in \{0, 1\}, n = 0, 1, \ldots, N-1$. □

With Corollary 1, the optimization algorithm has to consider only the boundary values of admission probabilities. That is, the value domain of $a(n)$ is reduced from a real number interval [0, 1] to a two-element set {0, 1}. Also, the value domain of $\boldsymbol{a}$ is reduced from an infinite N-dimensional space $[0, 1]^N$ to a finite space with size 2^N. This is a great reduction of the search space for optimization algorithms.

Based on the special feature of the admission controlled queueing network, we further study the structure of optimal policy which is described by the following theorem.

Theorem 2 *There exists a threshold* $\theta^* \in \{0, 1, \ldots, N\}$, *which has* $a^*(n) = 1$, $n < \theta^*$; $a^*(n) = 0$, $n \geq \theta^*$. *That is, threshold-type control policy is an optimal policy.*

Proof Since the optimal admission probability can be 0 or 1, suppose $a^*(k) = 0$ at a particular k. For the situation where $n > k$, n will decrease to reach k with a positive probability because $\mu_i > 0$ and the servers always drain the customers out of the network. That is, n will reach k for sure with a finite time. Once n reaches k, it will never increase because $a^*(k) = 0$ and all of the external arrivals will be rejected. This means that all of the states in $\mathcal{S}_{n>k}$ are transient. Since the steady state probabilities of transient states are 0, these transient states have no effects on the system long-run average performance η. Therefore, we can set the admission probabilities at these transient states as 0, i.e., $a^*(n) = 0$ for $n > k$. In summary, if $a^*(k) = 0$, then $a^*(n) = 0$ for all $n \geq k$. Therefore, it is obvious that there exists a threshold θ^*, which has $a^*(n) = 0$ for $n \geq \theta^*$ and $a^*(n) = 1$ for $n < \theta^*$. The theorem is proved. □

With Theorem 2, we see that the optimal policy has a threshold-type structure. Therefore, our optimization algorithm needs to only consider the threshold-type policy and the optimization goal is to find an optimal threshold θ^* which maximizes the average performance η. This structure of optimal policy further reduces the search space of optimization algorithms. We need to identify the optimal threshold θ^* only from the set $\{0, 1, \ldots, N\}$. Therefore, with Theorems 1 and 2, the total search space of $\boldsymbol{a}$ is reduce from an infinite N-dimensional space $[0, 1]^N$ to a finite space $\{0, 1, \ldots, N\}$. This is a significant reduction of the optimization complexity.

In order to efficiently find the optimal threshold θ^*, it is helpful to figure out the relation between the threshold and the system performance. Below, we further study how the system performance is changed when the threshold policy has changes.

Suppose that the threshold of admission control policy is changed from θ to θ', where $\theta, \theta' = 0, 1, \ldots, N$. Without loss of generality, we assume $\theta > \theta'$. This change of threshold from θ to θ' means that the admission probabilities are changed from $a(n) = 1$ to $a'(n) = 0$ for $n = \theta', \ldots, \theta - 1$, and all the other admission probabilities are the same for this two policies. Substituting these $a(n)$'s and $a'(n)$'s into Eq. 14, we obtain the following equation

$$\eta' - \eta = \sum_{n=\theta'}^{\theta-1} -\pi'(n) D(n). \tag{21}$$

For the system with the new threshold θ', since the admission probabilities are 0 when $n \geq \theta'$, all the states in $\mathcal{S}_{n>\theta'}$ are transient. Therefore, $\pi'(n) = 0$ if $n > \theta'$. We rewrite Eq. 21 and derive the following difference equation when the threshold is changed from θ to θ', $\theta > \theta'$.

$$\eta' - \eta = -\pi'(\theta') D(\theta'), \tag{22}$$

where $\pi'(\theta')$ is the marginal probability of the system with new threshold θ'. The equation above describes a very concise relation of the threshold and the system performance.

Since $\pi'(\theta')$ is always positive, we only have to know the sign of $D(\theta')$, which is a parameter of the current system with threshold θ. If $D(\theta') < 0$, then $\eta' > \eta$ and θ' is a better threshold than θ; otherwise if $D(\theta') > 0$, then $\eta' < \eta$ and we should not adopt this new threshold θ'. Therefore, we can use this difference equation to guide the optimization process.

Note that Eq. 22 requires $\theta > \theta'$. Below, we discuss the situation $\theta < \theta'$. Similarly, the threshold changed from θ to θ' means that the admission probability is changed from $a(n) = 0$ to $a'(n) = 1$, for $n = \theta, \ldots, \theta' - 1$. We substitute these $a(n)$'s and $a'(n)$'s into Eq. 14 and obtain

$$\eta' - \eta = \sum_{n=\theta}^{\theta'-1} \pi'(n) D(n). \tag{23}$$

For the system with the new threshold θ', the states in $\mathcal{S}_{\theta \leq n < \theta'}$ are recurrent and $\pi'(n)$'s are positive, $n = \theta, \ldots, \theta' - 1$. However, for the system with the current threshold θ, the states in $\mathcal{S}_{\theta < n < \theta'}$ are transient. Therefore, with Eq. 15, the states involved in $D(n)$ are transient for $n = \theta, \ldots, \theta' - 1$. This characteristic seriously affects the estimation of $D(n)$ based on the sample path, because the transient states may not be visited frequently enough for estimation. Therefore, difference Eq. 22 is more applicable than Eq. 23 because all the states involved in $D(n)$ of Eq. 22 are recurrent and they can be visited frequently in sample paths to obtain an accurate estimate. In the rest of the paper, we will use Eq. 22 to do optimization.

With Eqs. 22 and 23, we derive the following theorem about the sufficient condition of the optimal threshold θ^*.

Theorem 3 *θ^* is the optimal threshold if it satisfies, $D(n) \geq 0$, for $n = 0, \ldots, \theta^* - 1$; $D(n) \leq 0$, for $n = \theta^*, \ldots, N - 1$, where $D(n)$ is the aggregated performance potential rate of the system with threshold θ^*.*

Proof This theorem can be directly derived from the difference Eqs. 22 and 23. With Eq. 22, since $D(n) \geq 0$ for $n = 0, \ldots, \theta^* - 1$, the performance of the system with any threshold $\theta' < \theta^*$ is not better than that with threshold θ^*. With Eq. 23, since $D(n) \leq 0$ for $n = \theta^*, \ldots, N - 1$, the performance of the system with any threshold $\theta' > \theta^*$ is not better than that with threshold θ^*. Therefore, the system performance with threshold θ^* is maximal and the theorem is proved. □

Furthermore, we derive the following theorem about the necessary condition of the optimal threshold θ^*.

Theorem 4 *If θ^* is the optimal threshold, it has to satisfy, $D(n) \geq 0$ for $n = 0, \ldots, \theta^* - 1$ and $D(n) \leq 0$ for $n = \theta^*$, where $D(n)$ is the aggregated performance potential rate of the system with threshold θ^*.*

Proof We use a contradiction method to prove this theorem from two aspects. First, we assume that there exists an integer k, $0 \leq k < \theta^*$, which has $D(k) < 0$. We choose a new threshold as $\theta' = k$. With Eq. 22, we directly have $\eta' > \eta^*$, where η^* is the system performance corresponding to the optimal threshold η^*. Therefore, this is contradictory and the assumption is not correct. Second, we assume that $D(n) > 0$ when $n = \theta^*$. With Eq. 23, it is obvious that $\eta' > \eta^*$ when $\theta' = \theta^* + 1$. We also obtain a contradictory conclusion and the assumption is not correct. Therefore, the theorem is proved. □

Theorems 3 and 4 give the sufficient condition and the necessary condition of the optimal threshold θ^*, respectively. The difference Eqs. 22 and 23 play an important role in these theorems. It is worth pointing out that all the related results in this subsection are still valid when the reward function $\boldsymbol{f}$ is independent of the policy. When $\boldsymbol{f}$ is different with different policy, these theorems may be incorrect. We need further investigations to study the more general situations of this admission control problem.

3.2 Optimization algorithms and online implementation

Based on the difference Eq. 22, we develop an iterative algorithm to find the optimal threshold of the admission control problem. First, we have to calculate the values of $D(n)$'s in the difference equation. With Eqs. 15 and 5, we find that $D(n)$'s only depend on the behavior of the system with the current threshold. Therefore, we can observe the sample path of the current system to estimate $D(n)$'s. With $D(n)$'s and Eq. 22, we can easily find a better threshold θ' to improve the system performance. Running the system with the new threshold θ' and observing its sample path, we repeat this process and continue improving the system performance until we find the optimal threshold θ^*.

With a clear description of the performance change provided by Eq. 22, we develop the following iterative algorithm to find the optimal threshold θ^*.

Algorithm 1 Iterative algorithm to find the optimal threshold θ^*

Initialization

- Choose an initial threshold as $\theta^{(0)} = N$, set the space of feasible thresholds as $\Theta = \{0, 1, \ldots, N-1\}$ and $k = 0$.

Evaluation

- For the current system with threshold $\theta^{(k)}$, calculate or estimate the aggregated performance potential rate $D(n), n = 0, 1, \ldots, \theta^{(k)} - 1$.

Reduction

- If $D(n) \geq 0$, remove the element n from the space Θ, where $n = 0, 1, \ldots, \theta^{(k)} - 1$.

Stopping Rule

- If $\Theta = \varnothing$, set $\theta^* = \theta^{(k)}$ and stop;
- Otherwise, choose the new threshold as $\theta^{(k+1)} = \max\{\Theta\}$ and remove the element $\theta^{(k+1)}$ from the space Θ, set $k := k+1$ and go to step 2.

Step 3 identifies the inferior thresholds and removes them from the search space Θ. This can greatly reduce the search space and make the optimization algorithm efficient. The validity of this reduction is guaranteed by Eq. 22 and Theorem 4. With Eq. 22, we can see that the system performance is strictly improved after each iteration. Since the search space is finite, the algorithm will converge to the optimal threshold after a finite number of iterations.

With the description of the above algorithm, the worst situation of this algorithm is that we have to enumerate every possible θ, $\theta = 0, 1, \ldots, N-1$. Thus, the complexity of this algorithm is N times the complexity of estimating $D(n)$ for the worst case. Moreover, the significant reduction of searching space benefits from the difference Eq. 22. The correctness of Eq. 22 requires the condition 13 , which is not always valid in the practice. When Eq. 13 does not hold for other situations, we cannot derive the similar difference equation and optimization algorithms. How to handle these situations is our further research topic.

As we mentioned before, this problem of finding the optimal θ^* is not a standard MDP. The classical algorithms in MDP, such as value iteration and policy iteration, cannot solve this problem because of the constraints of decision epochs and parameterized policy (Puterman 1994). We may resort to some other parameter optimization approaches, such as the gradient-based approach with infinitesimal perturbation analysis (IPA) (Ho and Cao 1991) or policy gradient approach in MDP (Marbach and Tsitsiklis 2001). However, compared to these approaches, Algorithm 1 provides a more direct and efficient way to optimize the parameter, because it can strictly improve the system performance in each iteration based on Eq. 22. Algorithm 1 is guaranteed to find the global optimum, while the gradient-based optimization approaches may be trapped in the local optimum. This can be roughly explained that the performance difference has more sensitivity information than that of the performance gradient.

To implement Algorithm 1, we have to obtain the values of $D(n)$'s. For the system with a small size, $D(n)$'s can be theoretically calculated based on Eqs. 15 and 5, where $g(\boldsymbol{n})$'s can be obtained by solving linear equations (Cao 2007). However, for a medium or large size system, the system state space will explode exponentially and the theoretical calculation of $g(\boldsymbol{n})$'s becomes untreatable. Fortunately, we notice that the number of $D(n)$'s to be estimated is small and it increases linearly with respect to the system size. Hence, it is feasible to estimate $D(n)$'s based on the sample path, even for large-scale systems. The estimation method is introduced as follows.

From the definition (Eq. 15), we see that $D(n)$'s have relation to $\pi(\boldsymbol{n}|n)$ and $g(\boldsymbol{n})$, which are unknown variables. As we will see later, the information of $\pi(\boldsymbol{n}|n)$ is easy to obtain from the sample path. Below, we introduce how to estimate $g(\boldsymbol{n}) - g(\boldsymbol{n}')$ based on the sample path, $\boldsymbol{n}, \boldsymbol{n}' \in \mathcal{S}$, which is a prior work in the literature (Cao 2007).

From the definition (Eq. 5), we see that $g(\boldsymbol{n}) - g(\boldsymbol{n}')$ can be written as follows.

$$g(\boldsymbol{n}) - g(\boldsymbol{n}') = E\left\{ \int_0^{T(\boldsymbol{n}',0)} [f(X_t) - \eta]dt \Big|_{X_0=\boldsymbol{n}} \right\}, \tag{24}$$

where $T(\boldsymbol{n}', 0)$ is the time when the system state X_t first reaches $\boldsymbol{n}'$. We define $T(\boldsymbol{n}', t_0) := \min\{t : X_t = \boldsymbol{n}', t > t_0\}$, which indicates the time when the system state X_t first reaches $\boldsymbol{n}'$ after t_0. We further define $s_{\boldsymbol{n}}^0 := 0$ and $s_{\boldsymbol{n}}^k := \min\{t : X_t = \boldsymbol{n}, t > s_{\boldsymbol{n}}^{k-1}\}$, $k = 1, 2, \ldots$. Hence, $s_{\boldsymbol{n}}^k$ indicates the time when X_t reaches $\boldsymbol{n}$ at the kth time. $s_{\boldsymbol{n}}^k$ is also called the regenerative point of the state $\boldsymbol{n}$ in Markov process. For ergodic process, Eq. 24 can be rewritten and estimated as follows (Cao 2007).

$$g(\boldsymbol{n}) - g(\boldsymbol{n}') = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \int_{s_{\boldsymbol{n}}^k}^{T(\boldsymbol{n}', s_{\boldsymbol{n}}^k)} [f(X_t) - \eta]dt. \tag{25}$$

The time-average performance η can be estimated as

$$\eta = \lim_{T\to\infty} \frac{1}{T} \int_0^T f(X_t)dt. \tag{26}$$

Based on Eq. 25, we further study how to estimate $D(n)$'s. First, we rewrite Eq. 15 as below.

$$D(n) = -\lambda \sum_{\boldsymbol{n}\in\mathcal{S}_n} \pi(\boldsymbol{n}|n) \left\{ \sum_{i=1}^{M} q_{0i}[g(\boldsymbol{n}) - g(\boldsymbol{n}_{+i})] \right\}. \tag{27}$$

Since λ is a known constant, we focus on the estimation of left terms. Similarly, we define $T(\mathcal{S}_n, t_0) := \min\{t : X_t \in \mathcal{S}_n, t > t_0\}$, which indicates the time when X_t first reaches the state set $\mathcal{S}_n$ after t_0. We further define $s_n^0 := 0$ and $s_n^k := \min\{t : X_t \in \mathcal{S}_n, t > s_n^{k-1}\}$, $k = 1, 2, \ldots$. We see that s_n^k is the kth time when X_t reaches the state set $\mathcal{S}_n$. Using Eq. 25, we can rewrite Eq. 27 as follows.

$$D(n) = -\lim_{K\to\infty} \frac{\lambda}{K} \sum_{k=1}^{K} \int_{s_n^k}^{T(\mathcal{S}_{n+1}, s_n^k)} [f(X_t) - \eta]dt. \tag{28}$$

Therefore, we can use Eqs. 26 and 28 to estimate $D(n)$'s. From Eq. 28, we see that $\pi(\boldsymbol{n}|n)$ and q_{0i} disappear. This is because these information is already reflected from the sample path. Hence, we only have to observe the system state X_t and the associated reward $f(X_t)$.

Since $D(n)$'s can be estimated from the sample path, we can implement Algorithm 1 with an online manner. A typical scenario of using Algorithm 1 can be described as follows.

Procedure 1 Online implementation procedure of Algorithm 1

- Execute the Initialization step of Algorithm 1.
- At the current threshold policy, observe the information X_t and $f(X_t)$ from the sample path and estimate $D(n)$'s with Eqs. 26 and 28.
- Execute the Reduction step of Algorithm 1 to generate a new threshold policy.
- Run the system under the newly generated policy and repeat the step 2 and 3 above, until Algorithm 1 stops.

4 Simulation

First, we use a simple example to demonstrate the effectiveness of our admission control approach.

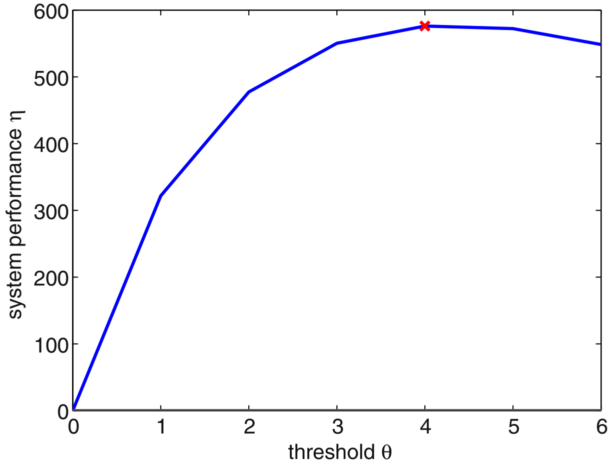
Consider a semi-open Jackson network with $M = 3$ servers and the capacity of the network is $N = 6$. The Poisson arrival rate is $\lambda = 50$. The service rates of servers are $\mu_1 = 40$, $\mu_2 = 50$, and $\mu_3 = 80$, respectively. The routing probabilities are $q_{01} = 0.3$, $q_{02} = 0.4$, $q_{03} = 0.3$, $q_{10} = 0.2$, $q_{12} = 0.4$, $q_{13} = 0.4$, $q_{20} = 0.4$, $q_{21} = 0.3$, $q_{23} = 0.3$, $q_{30} = 0.3$, $q_{31} = 0.2$, $q_{32} = 0.5$. The reward for each service completion is $R = 10$. The cost rate per unit time is $C = 100$.

We apply Algorithm 1 to find the optimal threshold of the admission control problem. The algorithm only iterates three times to find the optimal threshold $\theta^* = 4$, corresponding to the maximal performance $\eta^* = 576.1$. To verify the correctness of θ^*, we numerically calculate the system performance under various thresholds and obtain the theoretical curve of η with respect to θ, $\theta = 0, 1, \ldots, N$. The curve is illustrated in Fig. 2. It is verified that $\theta^* = 4$ is truly the optimal threshold.

From the optimization process of Algorithm 1, we see that the threshold is improved from 6, 5 to 4. When $\theta = 4$, with step 3 of Algorithm 1, we find that $D(n) \geq 0$, for all $n < 4$. This means that the performance with smaller thresholds is worse than that with $\theta = 4$. Therefore, all the left search space is removed and the algorithm stops at $\theta = 4$. From the process described above, we see that the algorithm does not need to simulate the situations with $\theta < 4$ because these situations are excluded through step 3 of Algorithm 1. This efficiently reduces the optimization complexity. This advantage comes from the difference Eq. 22.

The experiment above is simply for demonstration and the values of $D(n)$'s are calculated theoretically. For a system with medium or large size, it is inefficient to theoretically calculate $D(n)$'s because of the exponentially-exploded state space. We have to resort to the estimation techniques. Below, we conduct an experiment with a larger system size to demonstrate the estimation of $D(n)$'s and the online implementation of Algorithm 1.

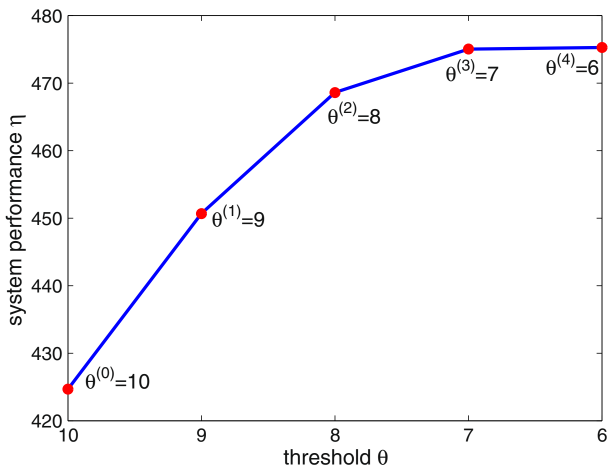
Fig. 2 The curve of system performance with respect to thresholds



Consider a semi-open Jackson network with $M = 8$ servers and the network capacity is $N = 10$. The external customer arrival rate is $\lambda = 50$. All the service rates of servers are the same as 60. $R = 5$ and $C = 100$. The routing probabilities are chosen randomly and their values are not listed here for the limit of space, which does not affect the demonstration purpose of the experiment.

As we know, the size of the system state space is $|\mathcal{S}| = \binom{M+N}{N}$. Compared to the network in the previous experiment, $|\mathcal{S}|$ of this network is greatly increased from 84 to 43758. Therefore, even for this medium-size system, the state space is too large to conduct the theoretical calculation of $D(n)$'s. However, we notice that the number of $D(n)$'s we need to estimate is only $N = 10$, which is very small compared to the state space. Hence, it is feasible for us to use Eqs. 26 and 28 to estimate $D(n)$'s from the system sample path and implement Algorithm 1 online as described in

Fig. 3 The online iteration procedure for searching the optimal threshold



Procedure 1. The sample path is long enough to guarantee the estimation accuracy of $D(n)$'s. The optimization procedure is illustrated in Fig. 3.

From Fig. 3, we see that the algorithm iterates five times and stops at $\theta^* = 6$ as its output of the optimal threshold. This experiment demonstrates that we can use the estimation techniques to estimate $D(n)$'s from the system sample path. The number of estimation parameters is linear to the system size. Thus, the "curse of dimensionality" is avoided in this problem. From the experiments, we see that when the system capacity N increases, the optimal threshold θ^* will also increase. This can be easily explained since a larger system capacity N offers more choice for the selection of threshold θ^*. Furthermore, from step 3 of Algorithm 1, we see that we only need to know the sign of $D(n)$, instead of its precise value. The estimation accuracy of a sign is usually better than that of a value. Therefore, this feature can help our algorithm effectively approach to a near optimum, with certain tolerance of estimation noise. Moreover, during the online implementation of Procedure 1, we see that we do not have to know some prior system information, such as q_{ij}'s, which can be directly reflected from the sample path. Thus, the complete system information is not required in our approach and this is an advantage during the practical application.

5 Conclusion

In this paper, we give an efficient approach to optimize the admission control problem of an open Jackson network. The control decision is made only at the epoch of customer arrival. We formulate this problem with the framework of event-based optimization. Based on a very concise difference equation which describes the system performance under any two policies, we prove that the system performance is monotonic with respect to the admission probabilities and the optimal control policy has a threshold form. We further develop an iterative algorithm to find the optimal threshold and implement the algorithm online based on the system sample path.

The optimization goal in this study is to maximize the total profit of the entire network. Thus, our approach finds a social optimum. If the admission decision is made by each arriving customer independently, the associated optimization result is called an individual optimum which is generally different from the social optimum. In our future research, we will further study the individual optimization from a perspective of game theory.

References

Altman E, Jimenez T, Koole G (2001) On optimal call admission control in resource-sharing system. IEEE Trans Commun 49:1659–1668

Cao XR (2005) Basic ideas for event-based optimization of Markov systems. Discrete Event Dyn Syst Theory Appl 15:169–197

Cao XR (2007) Stochastic learning and optimization – a sensitivity-based approach. Springer, New York

Chen H, Yao D (2001) Fundamentals of queueing networks, performance, asymptotics, and optimization. Springer, New York

Fang Y, Zhang Y (2002) Call admission control schemes and performance analysis in wireless mobile networks. IEEE Trans Veh Technol 51:371–382

Gao D, Cai J, Ngan KN (2005) Admission control in IEEE 802.11e wireless LANs. IEEE Netw 19:6–13

Ghaderi M, Boutaba R (2006) Call admission control in mobile cellular networks: a comprehensive survey. Wirel Commun Mob Comput 6:69C93
Gross D, Shortle JF, Thompson JM, Harris CM (2008) Fundamentals of queueing theory, 4th edn. Wiley, Hoboken
Ho YC, Cao XR (1991) Perturbation analysis of discrete event systems. Kluwer, Boston, MA
Hordijk A, Spieksma F (1989) Constrained admission control to a queueing system. Adv Appl Probab 21:409–431
Jackson JR (1963) Jobshop-like queueing systems. Manage Science 10:131–142
Jia QS (2011) On solving event-based optimization with average reward over infinite stages. IEEE Trans Automat Contr 56:2912–2917
Koole G (1998) Structural results for the control of queueing systems using event-based dynamic programming. Q Syst 30:323–339
Li T, Iraqi Y, Boutaba R (2004) Pricing and admission control for QoS-enabled interne. Comput Netw 46:87–110
Lima SR, Carvalho P, Freitas V (2007) Admission control in multiservice IP networks: architectural issues and trends. IEEE Commun Mag 45:114–121
Lin KY, Ross SM (2001) Admission control with incomplete information of a queueing system. Oper Res 51:645–654
Marbach P, Tsitsiklis JN (2001) Simulation-based optimization of Markov reward processes. IEEE Trans Automat Contr 46(2):191–209
Naor P (1969) The regulation of queue size by levying tolls. Econometrica 37:15–24
Niyato D, Hossain E (2005) Call admission control for QoS provisioning in 4G wireless networks: issues and approaches. IEEE Netw 19:5–11
Puterman ML (1994) Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, New York
Stidham J (1985) Optimal control of admission to a queueing system. IEEE Trans Automat Contr 30:705–713
Stidham J (2002) Analysis, design and control of queuing systems. Oper Res 50:197–216
Wu S, Wong KY, Li B (2002) A dynamic call admission policy with precision QoS guarantee using stochastic control for mobile wireless networks. IEEE/ACM Trans Netw 10:257–271
Xia L, Cao XR (2012) Performance optimization of queueing systems with perturbation realization. Eur J Oper Res 218:293–304
Xia L, Chen X, Cao XR (2009) Policy iteration for customer-average performance optimization of closed queueing systems. Automatica 45(7):1639–1648

Li Xia received the B.E. degree in automation, in July 2002, and the Ph.D. degree in control science and engineering, in July 2007, both from Tsinghua University, Beijing, China. From 2007 to 2009, he was a staff research member in IBM China Research. From 2009 to 2011, he was a postdoctoral research fellow in the King Abdullah University of Science and Technology (KAUST), Saudi Arabia. Presently, he is an assistant professor in the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua University.

He was a visiting scholar in the Hong Kong University of Science and Technology and a visiting scholar in University of Waterloo, Canada. He serve/served as a reviewer of international journals including IEEE Transactions on Automatic Control, IEEE Transactions on Automation Science and Engineering, European Journal of Operational Research, IIE Transactions, Journal of Optimization Theory and Applications, etc. He is an IEEE member and a member of Operation Research Society of China. His research interests include the methodology research in stochastic analysis and optimization, queueing theory, discrete event systems, and the application research in building energy, scheduling and optimization of smart grid, wireless sensor networks, production systems, etc.