Automatica 54 (2015) 29–35

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Parameterized Markov decision process and its application to service rate control☆

Li Xia, Qing-Shan Jia [1]

CFINS, Department of Automation, TNLIST, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 3 September 2013
Received in revised form
15 October 2014
Accepted 19 December 2014

Keywords:
Markov decision process
Discrete event systems
Parameterized policy
Policy iteration
Service rate control

ABSTRACT

In this paper, we discuss the optimization of Markov decision processes (MDPs) with parameterized policy, where the state space is partitioned and a parameter is assigned to each partition. The goal is to find the optimal parameters which maximize the long-run average performance. The traditional policy iteration is usually inapplicable to parameterized policy because the parameter tuning at different states are correlated. With some appropriate assumptions and special conditions, we develop a modified policy iteration type algorithm to find the optimal parameters. Compared with the traditional gradient-based approaches for MDP with parameterized policy, this policy iteration type approach is much more efficient. Finally, as an example, we apply this approach to a service rate control problem in closed Jackson networks. As compared with the gradient-based approach which is trapped into local optimum, our approach is demonstrated to efficiently find the optimal service rates in global scope.

1. Introduction

Markov decision process (MDP) is a fundamental mathematical model to study the performance optimization of stochastic dynamic systems and it has been extensively studied in the literature (Feinberg & Shwartz, 2002; Guo & Hernandez-Lerma, 2009; Puterman, 1994). In the theory of MDP, the policy is a mapping from the state space to the action space. However, in many practical problems, the parameterized policy is widely used since its form is much simpler. The state transition probabilities and rewards of Markov systems will change according to the value of these parameters. The parameterized policy does not fit the standard definition of policy in MDP and the traditional approaches, such as the policy iteration, cannot be directly applied to this problem. Our target is to find the optimal parameters which maximize the average performance of Markov systems and we call it *parameterized Markov decision process* (Baxter & Bartlett, 2001; Xia & Jia, 2013).

In the literature, the gradient-based method is the main thread to optimize the parameterized policy of Markov systems. As the Markov system is stochastic, the stochastic approximation is widely used. The key problem is transformed to how to efficiently calculate or estimate the performance gradient of the system performance w.r.t. (with respect to) the parameters. Among all the research efforts for this problem, perturbation analysis (PA) is one of the successful approaches. PA was proposed by Ho and Cao (1983) and it can provide an unbiased and strongly consistent estimate of the gradient only based on single sample path when the sampled function is stochastically Lipschitz continuous (Ho & Cao, 1991). Likelihood-ratio (LR) (Glynn, 1990) and simultaneous perturbation (SP) (Spall, 1992) are other commonly used approaches to efficiently estimate the gradient with much fewer samples in simulation. Thus, these approaches are especially efficient for the problems with high dimensional parameter vectors. Along the direction of PA, Cao and Chen (1997) proposed the direct-comparison theory of MDP. This is a new sensitivity-based framework to optimize the performance of Markov systems and some efficient algorithms are also proposed (Cao, 2007; Cao & Zhang, 2004). In the society of artificial intelligence, a so-called policy gradient method was proposed (Baxter & Bartlett, 2001; Marbach & Tsitsiklis, 2001; Sutton, McAllester, Singh, & Mansour, 2000) and it can also be unified in the framework of sensitivity-based approach. However, the gradient-based methods suffer from the intrinsic deficiencies, such as the slow convergence speed, difficulty of selecting the step-size, dependence on the initial value of parameters, and being trapped into a local optimum. For example, in a service rate control problem discussed in Section 4, we will show that the gradient-based

☆ The material in this paper was partially presented at the Asian Control Conference 2013, June 23–26, 2013, Istanbul, Turkey. This paper was recommended for publication in revised form by Associate Editor Bart De Schutter under the direction of Editor Ian R. Petersen.

E-mail addresses: xial@tsinghua.edu.cn (L. Xia), jiaqs@tsinghua.edu.cn (Q.-S. Jia).

[1] Tel.: +86 10 62773006; fax: +86 10 62796115.

http://dx.doi.org/10.1016/j.automatica.2015.01.006
0005-1098/

algorithm is often trapped into a local optimum, as illustrated in Fig. 2. Although we can utilize some global search techniques (Hong & Nelson, 2006) to improve the exploration ability for global optimum, the deficiencies of gradient-based method cannot be thoroughly solved.

Therefore, a question follows naturally: Can we use the policy iteration to solve the parameterized MDP since the policy iteration is much more efficient than the gradient-based method and it can find the global optimum? In this paper, we study a special category of parameterized MDP, where the state space is partitioned and an action (parameter tuning) is assigned to each partition. We use the direct-comparison theory to develop a policy iteration type algorithm for such parameterized MDP. The key idea of direct-comparison theory is the difference equation, which quantifies the performance difference of Markov systems under any two policies or parameter settings (Cao, 2007). Difference equation gives a straightforward perspective to study the relation between the system performance and parameters. The performance difference may provide more sensitivity information than the performance gradient. With the difference equation as a basis, we can clearly analyze the optimization of parameterized MDP and obtain the sufficient conditions to develop the policy iteration algorithm. This gives us a new direction to study the parameterized MDP, besides the traditional gradient-based methods. Finally, as an example, we study a service rate control problem in closed Jackson networks to illustrate our approach. Numerical experiments are conducted to demonstrate the algorithm efficiency.

2. Problem formulation

Consider a discrete time Markov chain $\boldsymbol{X} := \{X_t,\ t = 0, 1, 2, \ldots\}$, where X_t is the system state at time epoch t. The state space $\mathcal{S}$ is assumed finite. Without loss of generality, we denote $\mathcal{S} := \{1, 2, \ldots, S\}$, where S equals the size of the state space. The Markov chain is controlled by a parameterized policy (Baxter & Bartlett, 2001; Bhatnagar, Sutton, Ghavamzadeh, & Lee, 2009) and the parameters are denoted as a vector $\boldsymbol{\theta} := (\theta_1, \theta_2, \ldots, \theta_k)$ in a k-dimensional real number space $\mathbb{R}^k$. The parameters $\boldsymbol{\theta}$ affect the state transition probability and the reward function. We denote the state transition probability as $p^{\boldsymbol{\theta}}(s'|s)$ and the reward function as $f^{\boldsymbol{\theta}}(s)$, $s, s' \in \mathcal{S}$.

In many cases, the effects of parameters $(\theta_1, \theta_2, \ldots, \theta_k)$ on transition probabilities $p(\cdot|s)$ and $f(s)$ are decomposable. That is, if we change the value of one parameter, say θ_i, it affects only part of transition probabilities $p(\cdot|s)$'s and reward functions $f(s)$'s. For example, consider a service rate control problem in a closed Jackson network with 3 servers and 6 customers (the detailed formulation of a closed Jackson network can be referred to Section 4 and Gordon & Newell, 1967). We want to optimize the service rates of server 1. Thus, the parameter θ_i is the load-dependent service rate $\mu_{1,i}$, $i = 1, 2, \ldots, 6$. The system state s is a vector representing the queue length (include the customer being served) of these 3 servers. Suppose the reward function is $f^{\boldsymbol{\theta}}(s) = s(1) + \mu_{1,s(1)}$, where $s(1)$ is the first element of state vector s (i.e., the queue length of server 1). If we change the value of parameter θ_2, i.e., $\mu_{1,2}$, the transition probabilities $p(\cdot|s)$'s and reward function $f(s)$'s are affected only when $s \in \{(2, 0, 4), (2, 1, 3), (2, 2, 2), (2, 3, 1), (2, 4, 0)\}$, where the queue length of server 1 is 2. For other system states, such as $s = (1, 3, 2)$ or $s = (4, 1, 1)$, the change of parameter $\mu_{1,2}$ will not affect the value of $p(\cdot|s)$ or $f(s)$. Therefore, we can have the following definition to partition the state space $\mathcal{S}$.

Definition 1. $\mathcal{S}_i$ is defined as the set of states s whose transition probabilities $p(\cdot|s)$ and reward function $f(s)$ are affected by θ_i, $i = 1, 2, \ldots, k$.

Different parameters θ_i's have different $\mathcal{S}_i$'s and we have the following assumption

Assumption 1. $\mathcal{S}_i$'s are mutually exclusive, i.e., $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ when $i \neq j$ and $i, j = 1, 2, \ldots, k$.

Assumption 1 means that the state space $\mathcal{S}$ can be partitioned by parameters $\boldsymbol{\theta}$ and every state's transition probability $p(\cdot|s)$ and reward function $f(s)$ are controlled by only one parameter θ_i, where $s \in \mathcal{S}_i$. Therefore, we can further denote $p^{\boldsymbol{\theta}}(\cdot|s)$ and $f^{\boldsymbol{\theta}}(s)$ as $p^{\theta_i}(\cdot|s)$ and $f^{\theta_i}(s)$ respectively, where $s \in \mathcal{S}_i$.

Usually, the partition results of $\mathcal{S}_i$'s are not affected by the value of parameters $\boldsymbol{\theta}$. This is determined by the problem structure. That is, we have the following assumption.

Assumption 2. The value change of parameters $\boldsymbol{\theta}$ does not affect the structure of $\mathcal{S}_i$'s, $i = 1, 2, \ldots, k$.

For completeness, we further define $\mathcal{S}_0$ as the set of states whose transition probability and reward function are not affected by the parameters $\boldsymbol{\theta}$. With Assumptions 1 and 2, we see that the state space $\mathcal{S}$ is partitioned as a series of subsets $\mathcal{S}_i$'s. That is, $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_k$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, $i, j = 0, 1, \ldots, k$. The transition probability $p^{\theta_i}(\cdot|s)$ and the reward function $f^{\theta_i}(s)$ are affected only by the parameter θ_i and independent of other θ_j's, where $s \in \mathcal{S}_i$. Still use the aforementioned service rate control problem as an example. The parameter θ_i is the load-dependent service rate $\mu_{1,i}$, $i = 1, 2, \ldots, 6$. The state space $\mathcal{S}$ is partitioned as a series of subsets according to θ_i's. That is $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_6$, where $\mathcal{S}_0 = \{(0, 0, 6), (0, 1, 5), (0, 2, 4), (0, 3, 3), (0, 4, 2), (0, 5, 1), (0, 6, 0)\}$, $\mathcal{S}_1 = \{(1, 0, 5), (1, 1, 4), (1, 2, 3), (1, 3, 2), (1, 4, 1), (1, 5, 0)\}$, $\ldots$, $\mathcal{S}_5 = \{(5, 0, 1), (5, 1, 0)\}$, $\mathcal{S}_6 = \{(6, 0, 0)\}$.

The steady state probability of the Markov system staying at state s is denoted as $\pi(s)$ and $\boldsymbol{\pi} := (\pi(1), \pi(2), \ldots, \pi(S))$ is a row vector. The long-run average performance of the Markov system is denoted as η. To reflect the effect of parameter $\boldsymbol{\theta}$, we rewrite $\boldsymbol{\pi}$ and η as $\boldsymbol{\pi}^{\boldsymbol{\theta}}$ and $\eta^{\boldsymbol{\theta}}$, respectively. For ergodic chains, $\eta^{\boldsymbol{\theta}}$ can be written as follows.

$$\eta^{\boldsymbol{\theta}} = \lim_{T\to\infty} \frac{1}{T} E\left\{\sum_{t=0}^{T-1} f^{\boldsymbol{\theta}}(X_t)\right\}, \tag{1}$$

which is independent of the initial state X_0. Obviously, we can rewrite the above definition as

$$\eta^{\boldsymbol{\theta}} = \sum_{s\in\mathcal{S}} \pi^{\boldsymbol{\theta}}(s) f^{\boldsymbol{\theta}}(s) = \boldsymbol{\pi}^{\boldsymbol{\theta}} \boldsymbol{f}^{\boldsymbol{\theta}}, \tag{2}$$

where $\boldsymbol{f}^{\boldsymbol{\theta}} := (f^{\boldsymbol{\theta}}(1), f^{\boldsymbol{\theta}}(2), \ldots, f^{\boldsymbol{\theta}}(S))^T$ is a corresponding column vector. We denote $\boldsymbol{P}^{\boldsymbol{\theta}}$ as the corresponding transition probability matrix. We have $\boldsymbol{P}^{\boldsymbol{\theta}}\boldsymbol{e} = \boldsymbol{e}$, $\boldsymbol{\pi}^{\boldsymbol{\theta}}\boldsymbol{P}^{\boldsymbol{\theta}} = \boldsymbol{\pi}^{\boldsymbol{\theta}}$, and $\boldsymbol{\pi}^{\boldsymbol{\theta}}\boldsymbol{e} = 1$, where $\boldsymbol{e}$ is an S-dimensional column vector of 1.

The value domain of parameter θ_i can be a real-number interval denoted as $\mathbb{D}_i$, $i = 1, 2, \ldots, k$. Thus, the value domain of $\boldsymbol{\theta}$ is denoted as $\mathbb{D} := \mathbb{D}_1 \times \mathbb{D}_2 \times \cdots \times \mathbb{D}_k$, where $\times$ is the Cartesian product. Our goal is to find the optimal parameter $\boldsymbol{\theta}^*$ which maximizes the average performance of the parameterized MDP. This optimization problem is mathematically formulated as below.

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\mathbb{D}} \{\eta^{\boldsymbol{\theta}}\}. \tag{3}$$

Assumptions 1 and 2 limit our study to a special category of parameterized MDP, where the state space is partitioned and a parameter to be tuned is assigned to each partition. Please note, our parameterized MDP problem is different from another parameter optimization problem called LSPI (Least Squares Policy Iteration) in MDP. LSPI aims to find the optimal parameters (weights) of the basis functions to approximate the value function (Lagoudakis & Parr,

2003). Policy approximation techniques, such as grid (Busoniu, De Schutter, Babuska, & Ernst, 2010), can also be used to handle the representation difficulty of continuous policy space therein. However, the purpose of these techniques is different from that of our parameterized MDP. Of course, as a future work, we may combine LSPI techniques to handle the issue of large policy space in our parameterized MDP.

3. Analysis and optimization algorithm

The direct-comparison theory was proposed by Cao (2007), Cao and Chen (1997) and we give a brief introduction as below.

Suppose the policy of a Markov system is changed from $\mathcal{L}$ to $\mathcal{L}'$, the corresponding transition probability matrix and the reward function are changed from $\boldsymbol{P}$ and $\boldsymbol{f}$ to $\boldsymbol{P}'$ and $\boldsymbol{f}'$, respectively. The difference of the system average performance η and η' is

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

where $\boldsymbol{\pi}'$ is the vector of steady state distribution under policy $\mathcal{L}'$. In (4), $\boldsymbol{g}$ is a column vector called *performance potential* and its element, $g(s)$, $s \in \mathcal{S}$, is defined as follows.

$$g(s) = E\left\{\sum_{t=0}^{\infty} f(X_t) - \eta \Big| X_0 = s\right\}. \tag{5}$$

We see that $g(s)$ quantifies the long-term accumulated deviation of the system reward from the average performance. Different initial state s has a generally different accumulated deviation. $g(s)$ is also called the relative value function or bias in the MDP theory (Puterman, 1994). Extending (5) at $t = 0$ and substituting it recursively, we obtain

$$g(s) = f(s) - \eta + \sum_{s'=1}^{S} p(s'|s)g(s'). \tag{6}$$

We can further rewrite the above equation in a matrix form and obtain the following Poisson equation

$$(\boldsymbol{I} - \boldsymbol{P} + \boldsymbol{e}\boldsymbol{\pi})\boldsymbol{g} = \boldsymbol{f}. \tag{7}$$

Performance potential $\boldsymbol{g}$ can be calculated with Poisson equation (7), or be estimated from the sample path with the definition (5) or other variations (see Chapter 3.1 of the book of Cao, 2007). Another feasible way to approximately calculate $\boldsymbol{g}$ can be referred to the series expansion method (Heidergott, Hordijk, & Leder, 2010; Heidergott, Hordijk, & Uitert, 2007), which is originally studied to calculate the steady state distribution of perturbed Markov processes.

Difference equation (4) gives a clear description of the relation between the system performance and policies. Since the elements of $\boldsymbol{\pi}'$ are always positive for ergodic chains, if we choose a proper $\mathcal{L}'$ which makes the values of elements of the column vector represented by the square bracket of (4) positive, then we have $\eta' > \eta$ and the system performance is improved. This is exactly the intuitive explanation of the policy iteration in MDP. Below, we follow this idea to study the parameterized MDP problem (3).

For the parameterized MDP formulated in Section 2, suppose the parameter is changed from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}'$. With (4), we have

$$\eta^{\boldsymbol{\theta}'} - \eta^{\boldsymbol{\theta}} = \boldsymbol{\pi}^{\boldsymbol{\theta}'}[(\boldsymbol{P}^{\boldsymbol{\theta}'} - \boldsymbol{P}^{\boldsymbol{\theta}})\boldsymbol{g}^{\boldsymbol{\theta}} + (\boldsymbol{f}^{\boldsymbol{\theta}'} - \boldsymbol{f}^{\boldsymbol{\theta}})]. \tag{8}$$

Since we assume $\mathcal{S}_i$'s are mutually exclusive, we decompose the matrix form of the above equation as follows.

$$\eta^{\boldsymbol{\theta}'} - \eta^{\boldsymbol{\theta}} = \sum_{i=1}^{k}\sum_{s\in\mathcal{S}_i} \boldsymbol{\pi}^{\boldsymbol{\theta}'}(s)\left\{\sum_{s'=1}^{S}\left[p^{\theta_i'}(s'|s) - p^{\theta_i}(s'|s)\right] \times g^{\boldsymbol{\theta}}(s') + \left[f^{\theta_i'}(s) - f^{\theta_i}(s)\right]\right\}. \tag{9}$$

In the above difference equation, we see that p^{θ_i}, $p^{\theta_i'}$, f^{θ_i}, and $f^{\theta_i'}$ are known and $g^{\boldsymbol{\theta}}$ can be calculated or estimated based on the sample path of the system with parameter $\boldsymbol{\theta}$. But $\boldsymbol{\pi}^{\boldsymbol{\theta}'}$ is unknown. If we want to calculate $\boldsymbol{\pi}^{\boldsymbol{\theta}'}$ under various parameters $\boldsymbol{\theta}'$, its complexity is equivalent to that of exhaustively enumerating the system performance under every possible $\boldsymbol{\theta}'$. Fortunately, we find the following observations which can help solve this difficulty.

We use the symbol τ_i to denote the event that the current system state s is in $\mathcal{S}_i$, i.e., $s \in \mathcal{S}_i$, $i = 1, 2, \ldots, k$. We define the probability of event τ_i happening as $\pi^{\boldsymbol{\theta}}(\tau_i) := \sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}}(s)$. Similarly, for the system with a new parameter $\boldsymbol{\theta}'$, we define $\pi^{\boldsymbol{\theta}'}(\tau_i) := \sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}'}(s)$. Therefore, with the definition of conditional probability, we have

$$\pi^{\boldsymbol{\theta}'}(s) = \pi^{\boldsymbol{\theta}'}(\tau_i)\pi^{\boldsymbol{\theta}'}(s|\tau_i), \quad s \in \mathcal{S}_i. \tag{10}$$

We further define the aggregated performance potentials $G^{\boldsymbol{\theta}}(s, \theta_i)$ and $G^{\boldsymbol{\theta}}(s, \theta_i')$ as below.

$$\begin{aligned} G^{\boldsymbol{\theta}}(s, \theta_i) &:= \sum_{s'=1}^{S} p^{\theta_i}(s'|s)g^{\boldsymbol{\theta}}(s') \\ G^{\boldsymbol{\theta}}(s, \theta_i') &:= \sum_{s'=1}^{S} p^{\theta_i'}(s'|s)g^{\boldsymbol{\theta}}(s') \end{aligned} \tag{11}$$

Since $p^{\theta_i'}$ is a known parameter and $g^{\boldsymbol{\theta}}(s')$ can be calculated or estimated from the sample path of the system with $\boldsymbol{\theta}$, we can view $G^{\boldsymbol{\theta}}(s, \theta_i)$ and $G^{\boldsymbol{\theta}}(s, \theta_i')$ as known parameters. Substituting (10) and (11) into (9), we have

$$\eta^{\boldsymbol{\theta}'} - \eta^{\boldsymbol{\theta}} = \sum_{i=1}^{k}\pi^{\boldsymbol{\theta}'}(\tau_i)\sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}'}(s|\tau_i)\left\{\left[G^{\boldsymbol{\theta}}(s, \theta_i') + f^{\theta_i'}(s)\right] - \left[G^{\boldsymbol{\theta}}(s, \theta_i) + f^{\theta_i}(s)\right]\right\}. \tag{12}$$

Since $\pi^{\boldsymbol{\theta}'}(\tau_i)$ is always positive for ergodic Markov chains, we directly have the following observation.

Remark 1. If we select a proper $\boldsymbol{\theta}'$ which has $\sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}'}(s|\tau_i)\left[G^{\boldsymbol{\theta}}(s, \theta_i') + f^{\theta_i'}(s)\right] > \sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}'}(s|\tau_i)\left[G^{\boldsymbol{\theta}}(s, \theta_i) + f^{\theta_i}(s)\right]$, then we have $\eta^{\boldsymbol{\theta}'} > \eta^{\boldsymbol{\theta}}$.

In other words, if we can do the above parameter selection efficiently, we can iteratively find better parameters $\boldsymbol{\theta}'$, which is exactly similar to the policy improvement step in the classical policy iteration method. However, although $G^{\boldsymbol{\theta}}(s, \theta_i')$, $G^{\boldsymbol{\theta}}(s, \theta_i)$, $f^{\theta_i'}(s)$, and $f^{\theta_i}(s)$ are known parameters, the value of $\pi^{\boldsymbol{\theta}'}(s|\tau_i)$ is unknown. If we calculate $\pi^{\boldsymbol{\theta}'}(s|\tau_i)$ under every possible $\boldsymbol{\theta}'$, its computation complexity is equivalent to that of exhaustive comparison of $\eta^{\boldsymbol{\theta}'}$ under every possible $\boldsymbol{\theta}'$. Fortunately, we find some conditions under which the policy iteration is valid for this parameterized MDP. The first condition is listed as below.

Condition 1. *If a parameterized MDP satisfies*

$$\pi^{\boldsymbol{\theta}}(s|\tau_i) = \pi^{\boldsymbol{\theta}'}(s|\tau_i), \tag{13}$$

for all $s \in \mathcal{S}_i$, $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{D}$, $i = 1, 2, \ldots, k$, then we can construct a policy iteration algorithm for the parameterized MDP problem (3).

Condition 1 means that the conditional probability $\pi^{\boldsymbol{\theta}}(s|\tau_i)$ remains unvaried when the system parameter $\boldsymbol{\theta}$ changes. If (13) holds, we can rewrite (12) as below.

$$\eta^{\boldsymbol{\theta}'} - \eta^{\boldsymbol{\theta}} = \sum_{i=1}^{k}\pi^{\boldsymbol{\theta}'}(\tau_i)\sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}}(s|\tau_i)\left\{\left[G^{\boldsymbol{\theta}}(s, \theta_i') + f^{\theta_i'}(s)\right] - \left[G^{\boldsymbol{\theta}}(s, \theta_i) + f^{\theta_i}(s)\right]\right\}. \tag{14}$$

Since all the elements in the second summation of (14) are known through calculation or estimation, we can conduct the performance improvement in Remark 1 to repeatedly find a better parameter $\boldsymbol{\theta}'$. Moreover, the *necessary and sufficient condition* of the optimal parameter of this parameterized MDP can be directly derived as follows.

Theorem 1. *The parameter $\boldsymbol{\theta}$ is optimal, if and only if*

$$\sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}}(s|\tau_i)\left[G^{\boldsymbol{\theta}}(s,\theta_i')+f^{\theta_i'}(s)\right] \le \sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}}(s|\tau_i)\left[G^{\boldsymbol{\theta}}(s,\theta_i)+f^{\theta_i}(s)\right], \tag{15}$$

for any $\theta_i'\in\mathbb{D}_i$ and $i=1,2,\ldots,k$.

The proof of this theorem is very straightforward based on (14) and we omit it for simplicity. With the above analysis, we can develop an iterative algorithm to efficiently find the optimal parameter as follows.

Algorithm 1. Policy iteration to find the optimal parameter $\boldsymbol{\theta}^*$ for the parameterized MDP (3).

Initialization

- Choose an arbitrary parameter $\boldsymbol{\theta}^{(0)}\in\mathbb{D}$ as the initial value and set $l=0$.

Evaluation

- For the current system with parameter $\boldsymbol{\theta}^{(l)}$, calculate or estimate the performance potentials $g^{\boldsymbol{\theta}}(s)$'s and the conditional probabilities $\pi^{\boldsymbol{\theta}}(s|\tau_i)$'s, $s\in\mathcal{S}_i$ $i=1,2,\ldots,k$.

Improvement

- Update the new parameter as

$$\theta_i^{(l+1)}=\arg\max_{\theta_i'\in\mathbb{D}_i}\left\{\sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}}(s|\tau_i)\left[G^{\boldsymbol{\theta}}(s,\theta_i')+f^{\theta_i'}(s)\right]\right\}, \tag{16}$$

  for all $i=1,2,\ldots,N$, where the superscript $\boldsymbol{\theta}$ represents the current parameter $\boldsymbol{\theta}^{(l)}$. In (16), if there exist multiple θ_i''s which attain the maximization and $\theta_i^{(l)}$ is one of such θ_i''s, we choose $\theta_i^{(l+1)}=\theta_i^{(l)}$.

Stopping Rule

- If $\boldsymbol{\theta}^{(l+1)}=\boldsymbol{\theta}^{(l)}$, set $\boldsymbol{\theta}^*=\boldsymbol{\theta}^{(l)}$ and stop;
- Otherwise, set $l:=l+1$ and go to step 2.

Algorithm 1 gives an iterative procedure to find the optimal parameter of the parameterized MDP formulated in Section 2. The optimality of this algorithm requires Condition 1. When the parameter space $\mathbb{D}$ is finite, it is easy to know that the algorithm will stop within finite number of iterations. When the algorithm stops, the output is exactly the optimal parameter, which is guaranteed by Theorem 1.

From the procedure of Algorithm 1, we see that the original complicated optimization problem (3) is decomposed as a series of simple optimization subproblems (16). Each subproblem (16) is easy to solve since $|\mathbb{D}_i|\ll|\mathbb{D}|$. In certain sense, such decomposition avoids the exponential increase of optimization complexity when the dimension of parameters increases. When $\mathbb{D}_i$ is a discrete set, we can even enumerate to solve (16). When $\mathbb{D}_i$ is a continuous interval and $G^{\boldsymbol{\theta}}(s,\theta_i')$ (or $p^{\theta_i'}(s'|s)$ in (11)) and $f^{\theta_i'}(s)$ are linear to θ_i', the subproblem (16) is a *linear programming* with only one parameter θ_i', which is very easy to solve. When $G^{\boldsymbol{\theta}}(s,\theta_i')$ and $f^{\theta_i'}(s)$ are convex w.r.t. θ_i', (16) is a *convex optimization* problem and it is also easy to solve. For other situations, we can analyze the property of (16) to find possible solutions, such as finding the stationary points of Eq. (21) for the example of service rate control problem in Section 4. In summary, the decomposed subproblem (16) is much easier compared with the original parameterized MDP problem (3). This is exactly the reason that the policy iteration type algorithm is very efficient to find the global optimal parameter $\boldsymbol{\theta}^*$.

By analyzing (12) and Remark 1, we can obtain another condition which can also avoid the difficulty of calculating $\pi^{\boldsymbol{\theta}'}(s|\tau_i)$ under every possible $\boldsymbol{\theta}'$. We observe that if $G^{\boldsymbol{\theta}}(s,\theta_i')$ and $f^{\theta_i'}(s)$ can remain unvaried w.r.t. $s\in\mathcal{S}_i$, the calculation of $\pi^{\boldsymbol{\theta}'}(s|\tau_i)$ is unnecessary since $\sum_{s\in\mathcal{S}_i}\pi^{\boldsymbol{\theta}'}(s|\tau_i)=1$. Specifically, we have

Condition 2. *If a parameterized MDP satisfies*

$$p^{\theta_i}(\cdot|s)=p^{\theta_i}(\cdot|s') \quad \text{and} \quad f^{\theta_i}(s)=f^{\theta_i}(s'), \tag{17}$$

for all $s,s'\in\mathcal{S}_i$, $\theta_i\in\mathbb{D}_i$, $i=1,2,\ldots,k$, then we can construct a policy iteration algorithm for the parameterized MDP (3).

Condition 2 means that there is no difference in $p^{\theta_i}(\cdot|s)$ or $f^{\theta_i}(s)$ for the states s and s', if they are both in the same subset $\mathcal{S}_i$. Therefore, we can aggregate the states in $\mathcal{S}_i$ as a new state $\tilde{s}_i$, $i=1,2,\ldots,k$. We directly have the following observation.

Remark 2. Condition 2 actually implies that the original parameterized MDP can be aggregated as an equivalent MDP with the state space redefined as $\{\mathcal{S}_0,\tilde{s}_i:i=1,2,\ldots,k\}$.

Thus, with Condition 2, the original MDP problem can be reduced to a new aggregated MDP, which is similar to the state aggregation techniques in MDP theory (Bean, Birge, & Smith, 1987; Jia, 2011). This condition is very special and limited. We just mention it as a brief discussion.

In summary, the decomposition of the parameterized MDP into subproblems (16) is the origin of the advantages of policy iteration. The optimality of such decomposition requires a condition described by (13) or (17). The condition (13) means that the conditional probability $\pi(s|\tau_i)$, $s\in\mathcal{S}_i$, is independent of the value of parameter $\boldsymbol{\theta}$. The effect of changing $\boldsymbol{\theta}$ is proportionally imposed on all $\pi(s)$'s according to their relative ratios. This condition limits the application scope of our approach. Thus, we claim that our approach is only valid for a special category of parameterized MDP. Our study can be viewed as a beneficial attempt to solve such problems from the direction of policy iteration. For other parameterized MDP unsatisfying these conditions, we can still resort to gradient-based methods with the price of suffering aforementioned deficiencies. In practice, there do exist some parameterized MDP problems satisfying these conditions (Cao, 2007; Wang & Cao, 2011; Xia, 2014a). In the next section, we use a service rate control problem to demonstrate this approach, where the condition (13) is satisfied.

4. Application to service rate control

Consider a closed Jackson network with M servers (Gordon & Newell, 1967). The service time of a customer at server j obeys an exponential distribution with rate μ_{j,n_j}, where n_j is the queue length (include the customer being served) of server j, $j=1,2,\ldots,M$, $n_j=0,1,\ldots,N$. That is, the distribution function of the service time x at server j is $F(x)=1-e^{-\mu_{j,n_j}x}$, $x\ge0$. Obviously, $\mu_{j,n_j}=0$ when $n_j=0$. The network is called closed since there is no customer arriving from or exiting to the exterior. The number

of customers in the network is fixed as N. When a customer has been served at server j for a service time x, it means this customer finishes its service and it will transit to server j' with a routing probability $q_{jj'}, j, j' = 1, 2, \ldots, M$. One server can serve only one customer simultaneously. When a server is busy, the newly arriving customer to this server will wait in buffer. The service discipline is first come first serve. The system state is $\boldsymbol{n} := (n_1, n_2, \ldots, n_M)$ and the state space is denoted as $\mathcal{S} := \{\text{all } \boldsymbol{n} : \sum_{j=1}^{M} n_j = N\}$.

The optimization parameter is the service rates of one server, say server m. That is, the parameter in this problem is $\theta_i = \mu_{m,i}$, $i = 1, 2, \ldots, N$, and $\boldsymbol{\theta} = (\mu_{m,1}, \mu_{m,2}, \ldots, \mu_{m,N})$. The value domain of $\mu_{m,i}$ is $\mathbb{D}_i := [a_i, b_i]$, where a_i and b_i are the minimal and maximal service rates, respectively, $i = 1, 2, \ldots, N$. The reward function is defined as $f^{\theta_i}(\boldsymbol{n}) = -\phi(\boldsymbol{n}) - \varphi(\mu_{m,n_m})$, where $\phi(\boldsymbol{n})$ is called the holding cost only related to the system state $\boldsymbol{n}$ and $\varphi(\mu_{m,n_m})$ is called the operating cost related to the service rate μ_{m,n_m}. Example of such reward functions can be $f^{\theta_i}(\boldsymbol{n}) = -(n_1 + n_2) - \mu^2_{m,n_m}$. The optimization goal is to find the optimal parameter $\boldsymbol{\theta}^*$ which maximizes the long-run average performance.

First, we verify whether this parameterized MDP satisfies the aforementioned assumptions in Section 2. In this problem, $\mathcal{S}_i$ is equivalent to the set of states where the queue length of server m equals i, i.e., $\mathcal{S}_i := \{\text{all } \boldsymbol{n} : n_m = i\}$. Obviously, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ when $i \neq j$ and $\mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_N = \mathcal{S}$. Therefore, Assumption 1 is valid for this problem. Moreover, we can verify that when the values of parameter $\mu_{m,i}$'s, $i = 1, 2, \ldots, N$, are changed, the structure of $\mathcal{S}_i$'s keeps unvaried. That is, Assumption 2 is also valid.

In order to apply the approach proposed in Section 3, we have to verify whether this problem satisfies Conditions 1 or 2. Fortunately, we find that Condition 1 is satisfied. Eq. (13) in Condition 1 means that if we change the service rate $\mu_{m,i}$, the conditional probability $\pi(\boldsymbol{n}|\tau_i)$, $\boldsymbol{n} \in \mathcal{S}_i$, remains unvaried. This conclusion can be derived from the property of product-form solution of Jackson networks and the detailed proof can be referred to the author's recent paper (Xia & Shihada, 2013).

Below, we apply Algorithm 1 to solve this problem. First, we derive the specific form of the difference equation (14) for this problem. For clarity, we use the notation $\boldsymbol{\mu}$ instead of the parameter $\boldsymbol{\theta}$ and $\boldsymbol{\mu} := (\mu_{m,1}, \mu_{m,2}, \ldots, \mu_{m,N})$. When the service rates are changed from $\boldsymbol{\mu}$ to $\boldsymbol{\mu}'$, with the structure of $p(\cdot|\boldsymbol{n})$ in closed Jackson networks, we can rewrite (14) as follows.

$$\eta^{\boldsymbol{\mu}'} - \eta^{\boldsymbol{\mu}} = \sum_{i=1}^{N} \pi^{\boldsymbol{\mu}'}(\tau_i) \Big\{ G^{\boldsymbol{\mu}}(i)[\mu'_{m,i} - \mu_{m,i}] - \Big[\varphi(\mu'_{m,i}) - \varphi(\mu_{m,i})\Big] \Big\} \tag{18}$$

where $G^{\boldsymbol{\mu}}(i)$ is an aggregated performance potential defined as below.

$$G^{\boldsymbol{\mu}}(i) := \sum_{\boldsymbol{n} \in \mathcal{S}_i} \pi^{\boldsymbol{\mu}}(\boldsymbol{n}|\tau_i) \sum_{j=1}^{M} q_{mj} \Big[g^{\boldsymbol{\mu}}(\boldsymbol{n}_{-m+j}) - g^{\boldsymbol{\mu}}(\boldsymbol{n}) \Big], \tag{19}$$

where $\boldsymbol{n}_{-m+j} := (n_1, \ldots, n_m - 1, \ldots, n_j + 1, \ldots, n_M)$ is called the neighboring state of $\boldsymbol{n}$ with $n_m \geq 1$.

Based on (18), Algorithm 1 needs to do some modifications to comply with the specific formulation of this problem. Comparing (14) with (18), we correspondingly modify (16) of Algorithm 1 as below.

$$\mu^{(l+1)}_{m,i} = \arg\max_{\mu'_{m,i} \in \mathbb{D}_i} \left\{ \mu'_{m,i} G^{\boldsymbol{\mu}}(i) - \varphi(\mu'_{m,i}) \right\}, \tag{20}$$

for all $i = 1, 2, \ldots, N$.

In (20), $G^{\boldsymbol{\mu}}(i)$ is a known parameter calculable or estimable and $\varphi(\cdot)$ is a given function. Therefore, solving (20) is very easy. In general, when $\varphi(\cdot)$ is differentiable, we only have to solve $\frac{\partial \varphi}{\partial \mu_{m,i}} = G^{\boldsymbol{\mu}}(i)$ to find the feasible solutions in $\mathbb{D}_i$. We denote the feasible solutions as $\hat{\mu}_{m,i}$'s which have

$$\frac{\partial \varphi}{\partial \mu_{m,i}}\Big|_{\hat{\mu}_{m,i}} = G^{\boldsymbol{\mu}}(i), \quad \hat{\mu}_{m,i} \in \mathbb{D}_i. \tag{21}$$

$\hat{\mu}_{m,i}$ is also called a *stationary point* of the function $\mu'_{m,i} G^{\boldsymbol{\mu}}(i) - \varphi(\mu'_{m,i})$. Therefore, the search space of (20) is reduced from a continuous space $\mathbb{D}_i$ to a *discrete set* $\hat{\mathbb{D}}_i$ which is defined as

$$\hat{\mathbb{D}}_i = \{a_i,\ b_i,\ \hat{\mu}'_{m,i}\text{s}\}. \tag{22}$$

The subproblem (20) is simplified as

$$\mu^{(l+1)}_{m,i} = \arg\max_{\mu'_{m,i} \in \hat{\mathbb{D}}_i} \left\{ \mu'_{m,i} G^{\boldsymbol{\mu}}(i) - \varphi(\mu'_{m,i}) \right\}, \tag{23}$$

for all $i = 1, 2, \ldots, N$. This is a series of very simple optimization problems compared with the original optimization problem. Moreover, based on (23), we have the following remarks when φ satisfies some conditions.

Remark 3. When φ is a linear function of $\mu_{m,i}$, for example, $\varphi(\mu_{m,i}) = c\mu_{m,i}$ where c is a coefficient, (20) will become even simpler and the optimum can be achieved at the boundary a_i or b_i. The subproblem (20) becomes: If $G^{\boldsymbol{\mu}}(i) > c$, choose $\mu^{(l+1)}_{m,i} = b_i$; If $G^{\boldsymbol{\mu}}(i) < c$, choose $\mu^{(l+1)}_{m,i} = a_i$.

Remark 4. When φ is a concave function of $\mu_{m,i}$, for example, $\varphi(\mu_{m,i}) = -c\mu^2_{m,i}$, (20) will also become simpler. We observe that $\mu'_{m,i} G^{\boldsymbol{\mu}}(i) - \varphi(\mu'_{m,i})$ in (20) is a convex function w.r.t. $\mu'_{m,i}$. Obviously, the maximum of this convex function can be achieved at the boundary a_i or b_i and we replace $\hat{\mathbb{D}}_i$ with a two-element set $\{a_i, b_i\}$ in Algorithm 1.

Below, we discuss a numerical experiment of this problem. Consider a closed Jackson network with $M = 3$ and $N = 6$. The routing probabilities are $q_{11} = 0, q_{12} = 0.3, q_{13} = 0.7, q_{21} = 0.6, q_{22} = 0, q_{23} = 0.4, q_{31} = 0.5, q_{32} = 0.5, q_{33} = 0$. The service rates of server 1 and 2 are $\mu_1 = 30$ and $\mu_2 = 40$, respectively. The service rates of server 3 are the adjustable parameters of this problem. The value domain of $\mu_{3,i}$ is $\mathbb{D}_i = [10, 50]$, for all $i = 1, 2, \ldots, 6$. The reward function is $f^{\boldsymbol{\mu}}(\boldsymbol{n}) = n_1 + n_2 - \cos(\mu_{3,n_3}/4)$, $\boldsymbol{n} \in \mathcal{S}$. That is, $\varphi(\mu_{3,i}) = \cos(\mu_{3,i}/4)$ and the condition in Remark 3 or 4 is not satisfied. We solve (21) to obtain the stationary points $\hat{\mu}_{3,i}$'s. It is easy to verify that

$$\hat{\mu}_{3,i} = \begin{cases} 4[2k_1\pi + \arcsin(-4G^{\boldsymbol{\mu}}(i))]; \\ 4[2k_2\pi + \pi - \arcsin(-4G^{\boldsymbol{\mu}}(i))]; \end{cases}$$

where π is the constant number pi (a little abuse of notation and π in other parts of the paper denotes the steady state distribution of Markov systems), k_1 and k_2 are proper integers which guarantee $10 \leq \hat{\mu}_{3,i} \leq 50$. The values of $G^{\boldsymbol{\mu}}(i)$'s in Algorithm 1 are obtained with numerical calculation for the purpose of illustration. Thus, we obtain the reduced search space $\hat{\mathbb{D}}_i$ in (22), $i = 1, 2, \ldots, 6$. The initial values of $\mu_{3,i}$ are all chosen as 10, $i = 1, 2, \ldots, 6$. We implement Algorithm 1 and obtain the experiment results.

From the experiment results, we find that Algorithm 1 iterates only 4 times to find the optimal service rates. The update process of service rates $\mu_{3,i}$, $i = 1, 2, \ldots, 6$, and the corresponding system performance η are listed in Table 1. The optimal service rates of server 3 are the last column of Table 1. The system performance η is improved repeatedly during the iteration. The optimal system performance is $\eta^* = 4.8939$. From the iteration process, we notice that the performance improvement is significant at the

Table 1
The update process of $\mu_{3,i}$, $i = 1, 2, \ldots, 6$, and the corresponding η.

i	$\mu_{3,i}^{(0)}$	$\mu_{3,i}^{(1)}$	$\mu_{3,i}^{(2)}$	$\mu_{3,i}^{(3)}$	$\mu_{3,i}^{(4)}$
1	10	40.2460	38.3906	38.2892	38.2889
2	10	50.0000	39.2638	39.0747	39.0743
3	10	40.9219	39.4477	39.6188	39.6186
4	10	40.2306	39.7063	39.8717	39.8716
5	10	39.5271	39.6587	39.7866	39.7866
6	10	38.8719	39.2827	39.3641	39.3642
η	1.5237	4.7050	4.8934	4.8939	4.8939

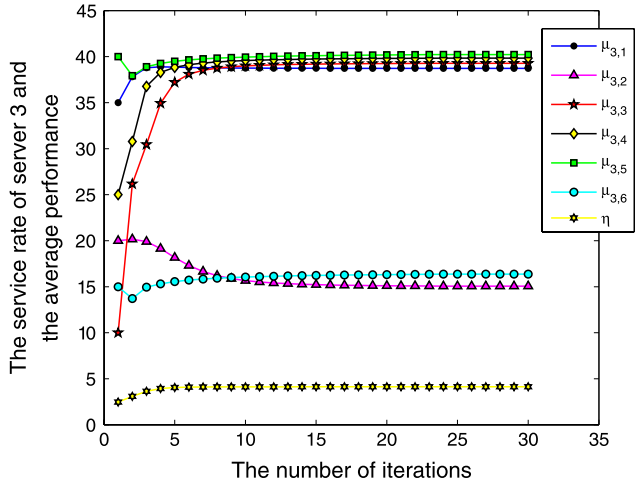

Fig. 1. The iteration procedure of service rates and performance in a gradient-based approach.

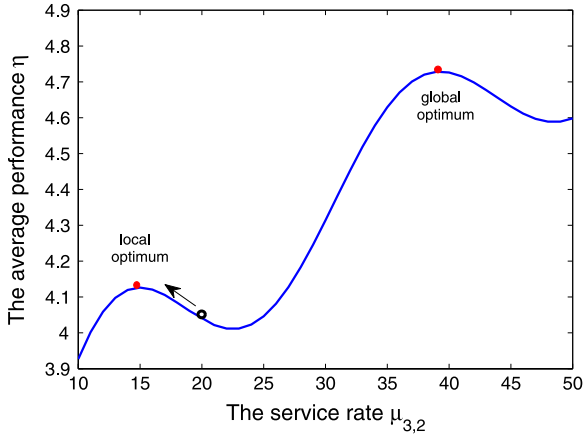

Fig. 2. The curve of the system average performance w.r.t. the service rate $\mu_{3,2}$.

first iteration. That is, even with only a few iterations, the policy iteration algorithm can approach to the global optimum rapidly.

As a comparison, we conduct another experiment adopting the gradient-based approach, which is often used in the literature. Suppose the current service rate is denoted as $\mu_{3,i}^{(l)}$, where $i = 1, 2, \ldots, 6$ and l is the index of iterations. The system performance gradients w.r.t. the parameter $\mu_{3,i}$'s are numerically calculated or estimated from the current sample path using PA, LR, SP, etc. After we obtain the performance gradient $\nabla\eta$ at the current service rate $\mu_{3,i}^{(l)}$, we update the service rate as $\mu_{3,i}^{(l+1)} = \mu_{3,i}^{(l)} + \gamma_l \times \nabla\eta$, where γ_l is the step-size at the lth iteration and it is chosen as $\gamma_l = 200/l$. The stopping criterion is that the maximal gap between two successive solutions is smaller than 0.1%. As we know, the gradient-based approach is usually trapped into a local optimum and the final output heavily depends on the initial value of parameters. We arbitrarily choose the initial value of parameters as $\boldsymbol{\mu} = (35, 20, 10, 25, 40, 15)$.

The update process of the service rates and the associated system performance are illustrated in Fig. 1. From this figure, we see that the final output of service rates of server 3 is $\hat{\boldsymbol{\mu}}^* = (38.7175, 15.0678, 39.2719, 39.8325, 40.2260, 16.3665)$ and the associated system performance is $\hat{\eta}^* = 4.1260$. Compared with the previous results obtained from Algorithm 1 in Table 1, we see that the solution obtained from the gradient-based approach is not the true optimal solution. Actually, $\hat{\boldsymbol{\mu}}^*$ is a local optimal solution. The gap between $\hat{\eta}^*$ and η^* is around 15%, which cannot be ignored. Fig. 2 is the curve of the system performance w.r.t. the service rate $\mu_{3,2}$. This figure is obtained with a brute-force enumeration. When the gradient-based approach starts at the initial point denoted as the black circle in Fig. 2, it will move on the direction to the local optimum. This figure also partly illustrates that $\hat{\boldsymbol{\mu}}^*$ is a local optimum, which corresponds to the red point on the left hand side in Fig. 2.

From the above experiment results, we see that our approach is effective for this service rate control problem, which is an example of the special parameterized MDP defined in this paper. Please note, we only control the service rates of one server. This approach cannot be directly extended to control the service rates of more than one server, since such problem does not satisfy Assumption 1. How to simultaneously optimize all the servers' service rates is a future research topic. Some preliminary work of this problem from a game theoretic perspective can be referred to the recent paper of Xia (2014b).

5. Discussion and conclusion

In this paper, we discuss the optimization theory for the parameterized MDP. With respect to a special category of parameterized MDP satisfying some assumptions, we propose a policy iteration type algorithm to find the optimal parameters. The sufficient conditions for the optimality of this approach are also derived. We use a service rate control problem as an example to demonstrate the efficiency of our approach. Compared with the traditional gradient-based approaches, such as PA, LR, SP, and policy gradient based method, our approach is more efficient since it has a fast convergence speed and the global optimization capability.

Our approach is valid under appropriate assumptions and conditions, which limit the application scope of our results. Although Condition 1 or 2 is valid for some problems, it is valuable to study the effect on the final result when these conditions are unsatisfied. Moreover, the estimation of aggregated potentials is important for our approach. How to combine with some computing budget allocation methods, such as OCBA (Chen, Lin, Yucesan, & Chick, 2000), to improve the computing efficiency is another research topic.

Acknowledgments

The authors would like to thank the three anonymous referees and the associate editor for their valuable comments which help improve this paper.

This work was supported in part by the National Natural Science Foundation of China (60736027, 61174072, 61203039, 61222302, 91224008, and U1301254), the National 111 International Collaboration Project (B06002), the Specialized Research Fund for the Doctoral Program of Higher Education (20120002120009), the Program for New Star of Science and Technology in Beijing (No. xx2014B056), the TNList Funding, the TNList Cross-Discipline Foundation, and the TNList Funding for Excellent Young Scholars.

References

Baxter, J., & Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 319–350.

Bean, J. C., Birge, J. R., & Smith, R. L. (1987). Aggregation in dynamic programming. *Operations Research*, *35*, 215–220.

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., & Lee, M. (2009). Natural actor-critic algorithms. *Automatica*, *45*, 2471–2482.

Busoniu, L., De Schutter, B., Babuska, R., & Ernst, D. 2010. Using prior knowledge to accelerate online least-squares policy iteration. In *Proceedings 2010 IEEE international conference on automation, quality and testing, robotics (AQTR-10)*, 28 May, Cluj-Napoca, Romania.

Cao, X. R. (2007). *Stochastic learning and optimization—a sensitivity-based approach*. New York: Springer.

Cao, X. R., & Chen, H. F. (1997). Perturbation realization, potentials, and sensitivity analysis of Markov processes. *IEEE Transactions on Automatic Control*, *42*, 1382–1393.

Cao, X. R., & Zhang, J. (2004). Performance sensitivities for parameterized Markov systems. *Journal of Control Theory and Applications*, *2*, 65–68.

Chen, C. H., Lin, J., Yucesan, E., & Chick, S. E. (2000). Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic System: Theory Applications*, *10*, 251–270.

Feinberg, E. A., & Shwartz, A. (2002). *Handbook of Markov decision processes: methods and applications*. Kluwer.

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, *33*, 75–84.

Gordon, W. J., & Newell, G. F. (1967). Closed queueing systems with exponential servers. *Operations Research*, *15*, 252–265.

Guo, X., & Hernandez-Lerma, O. (2009). *Continuous-time Markov decision processes: theory and applications*. Springer.

Heidergott, B., Hordijk, A., & Leder, N. (2010). Series expansions for continuous-time Markov processes. *Operations Research*, *58*(3), 756–767.

Heidergott, B., Hordijk, A., & Uitert, M. V. (2007). Series expansions for finite-state Markov chains. *Probability in the Engineering and Informational Sciences*, *21*(3), 381–400.

Ho, Y. C., & Cao, X. R. (1983). Optimization and perturbation analysis of queueing networks. *Journal of Optimization Theory and Applications*, *40*, 559–582.

Ho, Y. C., & Cao, X. R. (1991). *Perturbation analysis of discrete event systems*. Norwell: Kluwer Academic Publishers.

Hong, L. J., & Nelson, B. L. (2006). Discrete optimization via simulation using COMPASS. *Operations Research*, *54*, 115–129.

Jia, Q. S. (2011). On state aggregation to approximate complex value functions in large-scale Markov decision processes. *IEEE Transactions on Automatic Control*, *56*, 333–344.

Lagoudakis, M. G., & Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*, *4*, 1107–1149.

Marbach, P., & Tsitsiklis, J. N. (2001). Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control*, *46*, 191–209.

Puterman, M. L. (1994). *Markov decision processes: discrete stochastic dynamic programming*. New York: John Wiley & Sons.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, *37*, 332–341.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, *12*, 1057–1063.

Wang, D.X., & Cao, X.R. (2011). Event-based optimization for POMDP and its application in portfolio management. In *Proceedings of the 18th IFAC world congress* (pp. 3228–3233). Milano, Italy.

Xia, L. (2014a). Event-based optimization of admission control in open queueing networks. *Discrete Event Dynamic Systems: Theory and Applications*, *24*, 133–151.

Xia, L. (2014b). Service rate control of closed Jackson networks from game theoretic perspective. *European Journal of Operational Research*, *237*, 546–554.

Xia, L., & Jia, Q.S. (2013). Policy iteration for parameterized Markov decision processes and its application. In *Proceedings of the 9th Asian Control Conference (ASCC2013)* (pp. 1–6). June 23–26, 2013, Istanbul, Turkey.

Xia, L., & Shihada, B. (2013). Max–min optimality of service rate control in closed queueing networks. *IEEE Transactions on Automatic Control*, *58*, 1051–1056.

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.

Qing-Shan Jia received the B.E. degree in Automation in July 2002 and the Ph.D. degree in Control Science and Engineering in July 2006, both from Tsinghua University, Beijing, China. He is an Associate Professor in the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua University. He was a visiting scholar at Harvard University in 2006, at the Hong Kong University of Science and Technology in 2010, and at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology in 2013. His research interests include theories and applications of discrete event dynamic systems and simulation-based performance evaluation and optimization of complex systems.