International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:7, 2008

A State Aggregation Approach to Singularly Perturbed Markov Reward Processes

Dali Zhang, Baoqun Yin, and Hongsheng Xi

***Abstract*—**In this paper, we propose a single sample path based algorithm with state aggregation to optimize the average rewards of singularly perturbed Markov reward processes (SPMRPs) with a large scale state spaces. It is assumed that such a reward process depend on a set of parameters. Differing from the other kinds of Markov chain, SPMRPs have their own hierarchical structure. Based on this special structure, our algorithm can alleviate the load in the optimization for performance. Moreover, our method can be applied on line because of its evolution with the sample path simulated. Compared with the original algorithm applied on these problems of general MRPs, a new gradient formula for average reward performance metric in SPMRPs is brought in, which will be proved in Appendix, and then based on these gradients, the schedule of the iteration algorithm is presented, which is based on a single sample path, and eventually a special case in which parameters only dominate the disturbance matrices will be analyzed, and a precise comparison with be displayed between our algorithm with the old ones which is aim to solve these problems in general Markov reward processes. When applied in SPMRPs, our method will approach a fast pace in these cases. Furthermore, to illustrate the practical value of SPMRPs, a simple example in multiple programming in computer systems will be listed and simulated. Corresponding to some practical model, physical meanings of SPMRPs in networks of queues will be clarified.

***Keywords*—**Singularly perturbed Markov processes, Gradient of average reward, Differential reward, State aggregation, Perturbed close network.

I. Introduction

MARKOV chains are widely applied in modeling many stochastic systems, such as systems in communication networks, finance system, operation research and many other applications. Meanwhile, Markov Reward Processes (MRPs) or Markov Decision Processes (MDPs) are brought to solve problems in these models such as optimization for their performance. The theory of MRPs or MDPs is a mathematical framework for modeling sequential decision tasks, which becomes very popular in the field of intelligent computing and artificial intelligence currently, especially when these Markov models are involving with a large scale state spaces. Today, more and more methods are proposed to solve these problems. One class of these algorithms for optimization of MRPs is based on the simulation of single sample paths. To concentrate our paper on methods based on policy parameterization and gradient improvement, P. Marbach and J. N. Tsitsilis [1] brought the concept of Markov Reward Processes, created and described a popular simulation-based method, displayed the whole schedule of this algorithm, and modified this algorithm into a new way with faster updating.

However, dimensions of state spaces in these models are often too large for our normal algorithms. Hence and forth, the computing of optimization or the searching of the best strategy will waste a lot of time and memory saving. In these models, obtaining a desired optimal control parameters or policies can be quite intensive and the way to solve MRPs or MDPs with a large state space is a challenging issue at present. Till now, the results of effective algorithms for general cases are far from satisfied, but armed with some extra information, such as structures of our models, or certain states with its cost-to-go some effective approaches can be obtained to simplify these problems appropriately. The aim of this paper is to present an algorithm to optimize a class of MRPs with large state spaces, in which underlying Markov chains have hierarchical structures and are called singularly perturbed Markov reward processes (SPMRPs). The asymptotical properties of these singularly perturbed Markov processes (SPMRs) and the properties of the averaging reward of them are throughout studied by G. Yin and Q. Zhang in [2] and [5], and M. Abbad and J. A. Filar in [3] and [4], respectively. Far from difficulties just generated from searching one of best policy for the optimization of performance of SPMRPs with large scale state space, the hierarchical structures in SPMRPs also lead to two-timescale of sample paths generated by these processes, that is to say, the transition among some special subsets of state spaces will take place more slowly than the transition among states in these subsets. This property also makes our recurrent state in the general algorithm in [1] occurs more infrequently and the sparser occurrence of recurrent state will cause the recursion of gradient of performance delayed. While the second algorithm is applied in SPMRPs, the variance will accumulated for the sparser occurrence of recurrent state. Although the third modified method brings a forgetting factor in to reduce the variance accumulated by the second algorithm,

Manuscript received June 15, 2005. This work was sponsored by National Natural Science Foundation of China under Grant (60574065) and Natural Science Foundation of Anhui Province under Grant (050420301).

D. Zhang is with the Department of Automation, University of Science and Technology of China, Hefei 230027, P. R. China (phone: +86-0551-3603047; email: zhangdl@mail.ustc.edu.cn).

B. Yin is with the Department of Automation, University of Science and Technology of China, Hefei 230027, P. R. China (phone: +86-0551-3603047; email: bqyin@ustc.edu.cn).

H. Xi is with the Department of Automation, University of Science and Technology of China, Hefei 230027, P. R. China (phone: +86-0551-3603047; email: xihs@ustc.edu.cn).

but our special sample path is so long that the forgetting factor neglect the sample path transiting in other subsets, and the algorithm only optimize the current part of the single sample path. So the optimization along this sample path will be a local one and results of searching cannot converge to our objectives in a slow velocity. For example, a sample path of the SPMRs may be generated as:

$$1\ 1\ 2\ 2\ \cdots\ 1\ 2\ 3\ 4\ 4\ 3\ \cdots\ 3\ 4\ 1$$

where states 1 and 2 are in the same subset, 3 and 4 in another subset, and transitions among these subsets almost surely take place in a long interval. Name the interval between these two transitions as *segment*. Because almost surely every *segment* lasts a long period enough to lead the forgetting factor $\alpha^k, \alpha \in (0,1) \to 0$ and k is length of a *segment*. According to the example above, the first segment is always transiting between the states 1 and 2, which can be seen as staying in the same state after aggregation.

Here, a practical example in production is introduced to illustrate the practical application of SPMRPs, let's consider a manufacturing system with two subsystems in tandem. Each subsystem has 10 states, so the entire system has 100 states in corresponding:

$$S = \{s_{1,1}, \cdots, s_{1,10}, \cdots, s_{10,1} \cdots s_{10,10}\}$$

where denotation $s_{i,j}, i, j = 1,2 \cdots 10$ means that the subsystem 1 is in state i and subsystem 2 is in state j, and the connection of these subsystems is as in Fig. 1:

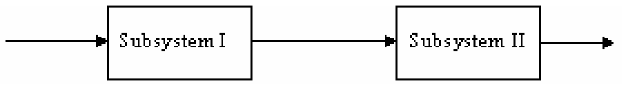

Fig. 1 A two-scale hierarchical manufacturing system

In most of situations, this system can be modeled in a Markov chain as in [14] with 100 states. But when states of the first system change more frequently than those of the second one, this system works as a SPMRP, and states included in our optimizing algorithm is only about 10. And this example is devoted to discrete-time Markov process. In such a problem, the computation effort depends mainly on the number of the states involved in the whole system. Furthermore, in a single sample path of such a system, the interval between the transitions of $s_{*i} \to s_{*,j}, j \neq i$, will be so long that the computation along such a sample path will necessarily waste a lot of saving memory and time. Thus the singular perturbation modeling will lead to significant reduction of complexity, and this point can be throughout described and proved in this paper.

Here, to be more practical, a simulation-based approach is necessary to be brought in to optimize an equivalent averaging reward problem of the original singularly perturbed Markov reward process (SPMRP) before any aggregation, which is proposed in [4] with a policy iteration form. Although, being suitable for many MRP, the approach in [1] can be used to optimize SPMRP in theory, a lot of shortcoming, such as slow paces, large biases accumulated, and unexpected lengths of regenerative cycles, will be brought in because its ignorance of the two-timescale structure of the underlying Markov process. The linear programming method proposed in [3] cannot be used in on-line optimization for its slow pace, either. To overcome these shortcomings and use extra information from its special structure, our method is based on the aggregated state space, which directly cut down the steps of computing, and balancing the frequency of every state taken in computing. Moreover, in our method, the recurrent state is taken place by the recurrent aggregated state, which takes the hierarchical structure and the segmental form of sample paths into account and direct to a simplified algorithm.

The rest of this paper is arranged as follows. In Section 2 a formulation of the problem and a gradient of the equivalent optimization objective are proposed. In Section 3, an example of network of queues in multi-programming will be presented to illustrate these properties. Consequently, an estimator of the gradient, and an exact algorithm of disturbance-controlled case will be proposed with their proof in Appendix. To complete our viewpoints, the shortcoming of a general method and merits of our method will be discussed in the last section through some simulation results.

II. Singularly Perturbed Markov Reward Processes and the Gradient of the Performance Metric

A. Singularly Perturbed Markov Reward Processes

Using the same notation as in [1], consider a discrete time Markov chain $\{i_n\}_{n\geq 0}$ with finite-state space S=$\{1,\cdots,N\}$, and assume its transition probabilities depend on a parameter vector $\theta \in \Re^k$. Here, transition probabilities can be denoted by

$$p_{ij}(\theta) = P(i_n = j \mid i_{n-1} = i; \theta) \tag{1}$$

When the current state is i, the process receives a one-stage reward, which also depends on the parameter θ chosen and can be denoted by $g_i(\theta)$. Transforming the set of transition probabilities $\{p_{ij}(\theta), \forall i, j \in S\}$ into a matrix form $P(\theta) = [\, p_{ij}(\theta)]_{N\times N}$, we set a set $\mathrm{P} = \{P(\theta) \mid \theta \in \Re^k\}$, which includes all of such matrices. Henceforth, such an MRP can be expressed in the form of four-tuple:

$$\Gamma =< S, \{\theta, \theta \in R^k\}, \{g_i(\theta), i \in S\}, \{P(j \mid i; \theta), i, j \in S\} >$$

The performance metric used to compute different parameter θs is average reward criterion $\eta(\theta)$, which is defined as:

$$\eta(\theta) = \lim_{T\to+\infty} \frac{1}{T} E_\theta[\sum_{k=0}^{T} g_{i_k}(\theta)] \tag{2}$$

where i_k is the state at time k. If the transition probabilities matrix $P(\theta)$ is aperiodic, and average reward $\eta(\theta)$ is well defined, and does not depend on the initial state. The average reward can be rewritten as

$$\eta(\theta)=\sum_{i=1}^{N}\pi_i(\theta)g_i(\theta) \tag{3}$$

where steady state distribution vector $\pi(\theta)=(\pi_1(\theta),\cdots,\pi_N(\theta))$ is the unique solution of balance equations:

$$\pi P = \pi \qquad \pi e_N = 1$$

where let $e_N = (1 \quad \cdots \quad 1) \in \Re^N$. From the Lemma 1 in the [1], If the matrix $P(\theta)$ is aperiodic, $p_{ij}(\theta)$ and $g_i(\theta)$ are all bounded for $\forall i, j \in S$, are twice differentiable, and have bounded first and second derivatives, then $\pi(\theta)$ and $\eta(\theta)$ defined above are also twice differentiable, and have bounded first and seconded derivatives.

The main objective of this paper aims to properties of SPMRPs. In fact (see, for example, [2]), any transition probability matrix of a finite-state Markov chain without any transient states can be put into the form:

$$P(\theta)=\begin{pmatrix} P_1(\theta) & & & \\ & P_2(\theta) & & \\ & & \ddots & \\ & & & P_n(\theta) \end{pmatrix}$$

where each $P_\alpha(\theta)$, $\alpha \in \{1,\cdots,n\}$ is a transition probability matrix within the αth recurrent class for $\alpha \in \{1,\cdots,n\}$. Here, denote steady state distribution corresponding to $P_\alpha(\theta)$ by $v_\alpha(\theta)=(v_\alpha^1(\theta),\cdots,v_\alpha^i(\theta),\cdots,v_\alpha^{m_\alpha}(\theta))\in\Re^{1\times m_\alpha}$, where $i \in S_\alpha$. First of all, look into a general Markov chain with its states and the transition probabilities corresponding to any pair of these states satisfies the following assumption:

A1) S= $\bigcup_{i=1}^{n} S_i$, where $S_i \bigcap S_j = \phi$, if $i \neq j$, $n > 1$ $|S_i| = m_i$, $m_1+\cdots+m_n = N$.

A2) $p\{s'|s,\theta\}=0$ whenever $s \in S_i$ and $s' \in S_j$, $i \neq j$.

A3) For every $i = 1,2,\cdots,n$, and for all $\theta \in \Re^k$, the matrix $P_i(\theta)$ is irreducible.

Then consider the situation where the transition probabilities of Γ are perturbed slightly. Define the disturbance law as the set $D=\{d(s'|s,\theta)\,|\,s',s\in S,\theta\in\Re^k\}$, and the elements of the set $D(\theta)$ satisfy: $\sum_{s'\in S} d(s'|s,\theta)=0$, and also transfer these elements into a matrix form as $D(\theta)=(d(s'|s,\theta))_{N\times N}$, where $D(\theta)$ can be seen as a generator. In addition, one more requirement, that there exists some $\varepsilon_0 > 0$ such that $\forall\theta\in\Re^k$, is necessary, and hence:

$$G^\varepsilon(\theta)=G(\theta)+\varepsilon D(\theta) \tag{4}$$

which is a generator of a Markov chain for any $0<\varepsilon<\varepsilon_0$. Shift our aim to the perturbed Markov chain. As in the G. Yin and Q. Zhang [2], suppose that $\{i_n^\varepsilon\}$ is a SPMP influenced by a small disturbance-parameter $\varepsilon > 0$, which can approach zero in any rate, but cannot be equal to zero, and also with a finite state space $S=\{1,2,\cdots,N\}$. The form of the transition probabilities satisfy

$$P^\varepsilon(\theta)=P(\theta)+\varepsilon D(\theta) \tag{5}$$

where $P^\varepsilon(\theta)$ and $P(\theta)$ are both the transition matrices, and $D(\theta)$ is a disturbance matrix. Here, we also assume that $P^\varepsilon(\theta)$ is irreducible for $\forall\varepsilon>0$ as in [2] and [3], that is to say, no matter how close ε approach zero, every subset state of state spaces and states in them will occur eventually, only if the sample path lasting long enough.

One-step rewards of the SPMRP are the same as those in general processes, and still denoted by $g_i(\theta)$, $i \in S$. Under the condition of SPMP, rewritten SPMRP as Γ_ε, where $\varepsilon \in (0,\varepsilon_0]$, in the form of a four-tuple:

$$\Gamma_\varepsilon =< S,\{\theta,\theta\in\Re^k\},\{g_i(\theta),i\in S\},\{P^\varepsilon(j\,|\,i;\theta),i,j\in S\}>$$

Denote steady state distribution vector corresponding to the transition probability matrix $P^\varepsilon(\theta)$ by $\pi_\varepsilon(\theta)$, and one step reward $g(\theta)=(g_1(\theta),\cdots,g_N(\theta))$. Define the average reward as we discussed above as

$$\eta_\varepsilon(\theta)=\pi_\varepsilon(\theta)g^T(\theta) \tag{6}$$

And the optimal value function $\eta_\varepsilon(\theta)$ corresponding to SPMRP is given by

$$\eta^*{}_\varepsilon(\theta)=\max_{\theta\in\Re^k}[\pi_\varepsilon(\theta)g^T(\theta)] \tag{7}$$

Example 2.1: *To illustrate the singularly perturbed processes more clearly, consider an example in hybrid system as:*

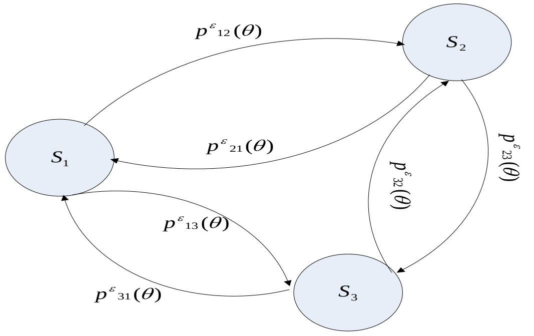
$p^\varepsilon{}_{12}(\theta)$
S_2
S_1
$p^\varepsilon{}_{21}(\theta)$
$p^\varepsilon{}_{32}(\theta)$
$p^\varepsilon{}_{23}(\theta)$
$p^\varepsilon{}_{13}(\theta)$
$p^\varepsilon{}_{31}(\theta)$
S_3

Fig. 2 An Example Hybrid System

where every $S_i, i=1,2,3$, *is a subspace of* S *and with the transition probabilities* $p^\varepsilon{}_{ij}(\theta)=P(i_n\in S_j\,|\,i_{n-1}\in S_i;\theta)$ *being equal to a* $O(\varepsilon)$. *In another level, each* $S_i, i=1,2,3$, *also have its own structure as a Markov chain as:*

$$p_{kl}^\varepsilon(\theta;S_i)=P(i_n=s_{i,k}\,|\,i_{n-1}=s_{i,k};\theta)$$

where $s_{i,k}, s_{i,l}\in S_i$ *and* $p^\varepsilon{}_{kl}(\theta;S_i) >> p^\varepsilon{}_{ij}(\theta)$.

To deal with SPMRPs, a new Markov reward process will be constructed through the algorithm of state aggregation, and aggregated states in S_i as a new state in original sample path. The problem created by aggregation still has an optimization objective asymptotically converging to the original one. Here, Denote the aggregated state space by $\Omega = \{S_1, S_2, \cdots, S_n\}$, and for a simple denotation as $\Omega = \{1, 2, \cdots, \alpha, \cdots, n\}$. To proceed, we define a matrix $\tilde{1}$ as

$$\tilde{1} = \begin{pmatrix} 1_{m_1} & & & \\ & 1_{m_2} & & \\ & & \ddots & \\ & & & 1_{m_n} \end{pmatrix}$$

where entry vectors $1_{m_\alpha} = (1, 1, \cdots, 1)^T$, which are of m_α elements, respectively. From [2] and [3], we can construct a new Markov chain as $\{\alpha_k\}$, generated by the generator

$$\overline{Q}(\theta) = diag\{v_1(\theta), \cdots, v_\alpha(\theta) \cdots, v_{|\Omega|}(\theta)\} D(\theta)\tilde{1} \qquad (8)$$

where $v_\alpha(\theta)$ is steady state distribution vector of transition probability $P_\alpha(\theta)$, and let $\overline{\pi}(\theta)$ be steady state distribution vector of transition probability matrix $\overline{P}(\theta) = \overline{Q}(\theta) + I_{n\times n}$. Directly from Lemma 2.1 in [4], there is another optimization problem named *Aggregated Limiting Problem*

$$\overline{\Gamma}_\varepsilon =< \Omega, \{\theta, \theta \in \Re^k\}, \{\overline{g}_\alpha(\theta), \alpha \in \Omega\}, \{\overline{P}(\alpha \mid \beta; \theta), \alpha, \beta \in \Omega\} > \qquad (9)$$

where vectors $\hat{g}_\alpha(\theta) = (g_1(\theta), \cdots, g_{\alpha_i}(\theta), \cdots g_{m_\alpha}(\theta)), \forall \alpha_i \in S_\alpha$ $\overline{g}_\alpha(\theta) = v_\alpha(\theta)\hat{g}_\alpha(\theta), \forall \alpha \in \Omega$, $\overline{g}(\theta) = (\overline{g}_1(\theta), \overline{g}_2(\theta) \cdots, \overline{g}_n(\theta))$. And its average reward function is defined as

$$\overline{\eta}(\theta) = \overline{\pi}(\theta)\overline{g}^T(\theta) \qquad (10)$$

Lemma 2.1: *Assume assumption A1)-A3), we have*

$$\lim_{\varepsilon \to 0 \varepsilon > 0} \left| \max_{\theta \in \Re^k} \left[\overline{\pi}(\theta)\overline{g}^T(\theta) \right] - \max_{\theta \in \Re^k} \left[\pi^\varepsilon(\theta) g^T(\theta) \right] \right| = 0 \qquad (11)$$

and in an equal form as:

$$\left| \max_{\theta \in \Re^k} \left[\overline{\pi}(\theta)\overline{g}^T(\theta) \right] - \max_{\theta \in \Re^k} \left[\pi^\varepsilon(\theta) g^T(\theta) \right] \right| = o(\varepsilon) \qquad (12)$$

which can be directly proved from the [2].

So any maximizing parameter θ for *Aggregated Problem* is also a maximizing parameter θ for the original problem and vice-versa. In the next subsection, the equation of gradient of aggregated performance metric takes place of the original one. From the above lemma, we can easily prove that $|\nabla \overline{\eta}(\theta) - \nabla \eta(\theta)| = o(\varepsilon)$. When optimizing the sample path generated by original perturbed process along the gradient of $\overline{\eta}(\theta)$, the optimized value of $\eta(\theta)$ also can be achieved.

B. Properties of the Gradient of the Performance Metric

For any $\theta \in \Re^k$ and $i \in S$, denote the differential reward by $\upsilon_i(\theta)$ of state i by (as in [1]):

$$\upsilon_i(\theta) = E_\theta \left[\sum_{k=0}^{T-1} (g_{i_k}(\theta) - \eta(\theta)) \mid i_0 = i \right] \qquad (13)$$

where i_k is the state at time k, and $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. With this definition, it is easy to find out that $\upsilon_{i^*}(\theta) = 0$ and that the vector $\upsilon(\theta) = (\upsilon_1(\theta), \cdots, \upsilon_N(\theta))$ is a unique solution to Poisson equation:

$$g(\theta) = \upsilon(\theta) + \eta(\theta)e_N - P^\varepsilon(\theta)\upsilon(\theta)$$

All these denotations defined above is the same as in [1]. Corresponding to the *aggregated limiting problem*, define a new differential reward $D_\alpha(\theta)$, of an aggregated state $\alpha \in \Omega$ by

$$\tilde{D}_\alpha(\theta) = E_\theta \left[\sum_{k=0}^{T'-1} (\overline{g}_{\alpha_k}(\theta) - \overline{\eta}(\theta)) \mid \alpha_0 = \alpha \right] \qquad (14)$$

where α_k is the aggregated state at time t_k, the kth epoch with the transition between two different subsets $S_i, S_j, i, j \in \Omega, i \neq j$, and the aggregated state i can be seen as an index for a certain segment covering the sample path dominated by states in some subspace S_i, and we let $T' = \min\{k > 0 \mid \alpha_0 = \alpha\}$, where there is a trivial assumption that the recurrent state of original problem is still in the recurrent state of aggregated problem, i.e. $i^* \in S_*$, and generally speaking, we can set the recurrent state of the aggregated process as any subset S_*, if it contains the recurrent state in original problem. There are still some similar properties as the original problem as $\tilde{D}_{\alpha^*}(\theta) = 0$ and we let vector $\tilde{D}(\theta) = (\tilde{D}_1(\theta), \cdots \tilde{D}_n(\theta))$, which is also a unique solution to Poisson equation:

$$\overline{g}(\theta) = \tilde{D}(\theta) + \overline{\eta}(\theta)e_n - \overline{P}(\theta)\tilde{D}(\theta) \qquad (15)$$

A theorem for the equation of the gradient of the average reward $\overline{\eta}(\theta)$, with respect to θ, will be displayed. Before the beginning of our theorem, there are some results derived from the A1)-A3) which is necessary, and here still named them as A4) -A5):

A4) The Markov chain corresponding to $\overline{P}(\theta)$ is aperiodic and irreducible. That is to say, there is a state α^* that is recurrent for the chain. This point can be directly derived from the aperiodicity and irreducibility of $P^\varepsilon(\theta)$, and at least S can be divided into some S_α with $i^* \in S_{\alpha^*}$, which α^* is an index of some aggregated state.

A5) For every $i, j \in S$, the function $p^\varepsilon_{ij}(\theta)$ and $g_i(\theta)$ are bounded, twice differentiable, and have bounded first and second derivatives. Hence, for every $\alpha, \beta \in \Omega$, the functions $\overline{P}_{\alpha\beta}(\theta)$ and $\overline{g}_\alpha(\theta)$, as the linear function of $p^\varepsilon_{ij}(\theta)$ and $g_i(\theta)$, are also bounded, twice differentiable, and have bounded first and second derivatives.

To be more generality, here, we use A4) and A5) to instead that of A1)-A3).

Theorem 2.1 *Let assumption A4), A5) hold. Then the gradient of the aggregated limiting average reward is*

$$\overline{\eta}(\theta)=\sum_{\alpha\in\Omega}\overline{\pi}_{\alpha}(\theta)(\nabla g_{\alpha}(\theta)v_{\alpha}^{T}(\theta)+g_{\alpha}(\theta)\nabla v_{\alpha}^{T}(\theta)+ \sum_{\beta\in\Omega}[\sum_{i\in S_{\alpha}}(\sum_{j\in S_{\beta}}\nabla d_{ij}(\theta))v_{\alpha}^{i}(\theta)+\sum_{i\in S_{\alpha}}(\sum_{j\in S_{\beta}}d_{ij}(\theta))\nabla v_{\alpha}^{i}(\theta)]\tilde{D}_{\beta}(\theta)) \tag{16}$$

The proof of this theorem is in *Appendix*. Specially, when the disturbance matrix $D(\theta)$ is irrelative with the parameter θ, we have

Corollary 2.1 *Let assumption A4), A5) hold, and disturbance factors are independent of the parameter θ. Then*

$$\nabla\overline{\eta}(\theta)=\sum_{\alpha\in\Omega}\overline{\pi}_{\alpha}(\theta)(\nabla g_{\alpha}(\theta)v_{\alpha}^{T}(\theta)+g_{\alpha}(\theta)\nabla v_{\alpha}^{T}(\theta)+ \sum_{\beta\in\Omega}[\sum_{i\in S_{\alpha}}(\sum_{j\in S_{\beta}}d_{ij})\nabla v_{\alpha}^{i}(\theta)]\tilde{D}_{\beta}(\theta)) \tag{17}$$

The equations given by Theorem 2.1 and Corollary 2.1 involve no terms of the steady state distribution of $P^{\varepsilon}(\theta)$, but involve with those of $P_{\alpha}(\theta)$, $\alpha\in\Omega$. Hence, differing from the counterpart in [1], an algorithm to estimate terms as $\nabla v_{\alpha}^{T}(\theta)$ is necessary, which will be introduced in Section 4, but we will still find that our schedule of computing the gradient is much better than that in [1] for a singularly perturbed Markov reward chain, and it will reduce the iteration steps.

III. The Model in Network of Queue

These theoretic properties of singularly perturbed Markov chains can applied to model a great deal of stochastic systems, such as network of queue, information processing of operation systems, market network. To illustrate the application of singularly perturbed Markov chain in these practical systems, and to explain the physical meaning of the concepts in SPMRP, a hierarchical network of queues is introduced.

Let us apply some concepts to a simple model of a network, which was first analyzed by J. R. Jackson [14]. Consider a simple model of a multiprogramming paging system:

1. N active user terminals originate with Poisson rate λ requests for program processing, and name these requests as jobs. There may exist at most one job per terminal at a system epoch.
2. Jobs are processed in main memory on a multi-programmed basis. Let J be the current number of jobs being programmed in the system.
3. Pages that cannot be contained in main memory are located in an auxiliary memory level from which they are loaded on a page on demand strategy.

Therefore, at any epoch, multi-programmed jobs are in one of three states: ready state, requesting but not receiving process signals form the processor; running state, receiving the control of processor; suspended state, waiting for a page transfer from auxiliary to main memory. Here, use a figure to clarify our system:

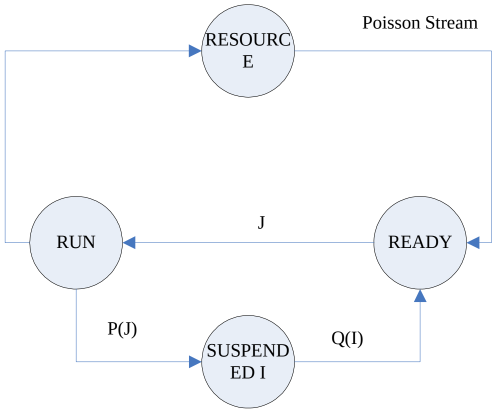

Fig. 3 A Multi-Programming System

Corresponding to this figure, there are some additional assumptions: first of all, at any time, each multi-programmed job is allocated an equal number of quantities of page frames, and denote such a mount by J available in main memory. The probability of a page transfer from running state to suspended state is assumed to be $P(J)$, $J=1,2,\cdots N$. On the other hand, assume that the requests for page transfer from auxiliary to main memory are not necessarily served on a FIFO basis, but in some order which depends upon current state of the auxiliary memory so as to optimize the overall page transfer rate. The rate can be seen as a function of the number of suspended jobs, and denoted by $Q(I)$. Furthermore, the probability that a page transfer from the processor back to the resource can be neglected.

From a practical view in computer systems, the rates of transition among state running, suspended, and ready will usually be much higher than the rates at which jobs are generated and completed. So the whole system can be decomposed into two parts, one with some resource terminals that can be seen as a resource of Poisson stream; on the other hand, the part of processing can be seen as an aggregated state with state running, suspended, and ready.

Take the jobs in states running, suspended and ready, respectively as a three-tuple (s_R, s_s, s_r), and use the three-tuple to index the state of processor. If the numbers of jobs in states： running, suspended, and ready are constant, the system can be seen as a close network of queue, and hence the transition among these states can be seen as Markov chain. While in some practical sense, these Markov chains are not stable, and they are perturbed by Poisson stream, which leads to the change of the mount of jobs. Divide the original state space into subsets which we need as:

$$S_J=\{(s_R,s_s,s_r)\mid s_R+s_s+s_r=J\}, J=0,1,2,\cdots\overline{J}$$

where use $\overline{J}$ denote the super of the number of jobs in processing. And subset series $\{S_J, J=1,\cdots\overline{J}\}$ satisfies the assumption A1)-A3), and can be taken as aggregated states. For a more simple and direct way to our aggregation, use the symbols $\{1,2,\cdots,\bar{J}\}$ to index these aggregated states. Till now, the physical meaning of aggregation of such a perturbed close Jackson network can be clarified. Before the aggregation, it should be taken into consideration that not only whether a new job is arrived, but also the distribution of current jobs that are in three different states. But, based on the fact that the event that a new job arrive takes place less frequently than that of transition of these three states in the sense of a completely close Jackson network. So, before the arrive of the $J+1$th, the distribution of J jobs have already entered the stationary steady state, and it means that whenever a new jobs arrive, the current state can seen as the same in the stochastic sense, i.e. the difference among current states $(1,1,2),(1,2,1),\cdots(0,0,4)$ can be neglect before the 5th job arrive. Obviously, the aggregated chain is still a Markov chain, for it describes a Poisson stream with a constant arrive rate. The transition probability matrix is just a dimension of $\bar{J}\times\bar{J}$, which is much less the original matrix with its dimension $O(\bar{J}^3)\times O(\bar{J}^3)$.

When a judgment before the job enter the processor is added, and assume that in different state $S_{Rsr}:(s_R,s_s,s_r)$, the cost of the computing can be denoted by $R(s_{Rsr})$. Then, model the optimization of this multi-programmed system as a singularly perturbed Markov reward process, or more precise, model it as a singularly perturbed close network. Set a parameter $\theta\in[0,1]$ to control the acceptation and rejection of a new job: the new job will be accepted by probability θ and rejected otherwise. Our goal is to choose a reasonable θ that is a function of the jobs in the close network currently, and hence this parameter can be seen as a policy $\theta((s_R,s_r,s_s))$, corresponding to every state. While from discussion above, we knew that the action of judging a new job can be accept or not is based on the event a new job is arriving at the front of judgment, so current state in close network have already enter a stationary steady state distribution. Hence, corresponding to aggregated states, our policy can be simplified as $\theta(s_R+s_r+s_s)$. And the whole model is described as in Fig. 4:

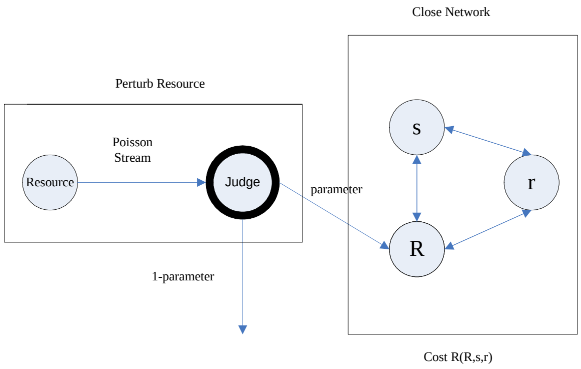

Fig. 4 A Multi-Programming System with Parameter and Cost

In this section, our definitions of SPMRPs are applied into a practical issue, a multi-programmed system. From the models set above, we acquire some useful physical meaning of the singularly perturbed Markov chain and its aggregated chains. We also applied the reward or cost process into such a system, and create a denotation of perturbed close network. From the analysis of policy and aggregation in these systems, we prove that our models of singularly perturbed Markov reward process and aggregation idea are all with practical value. There still a lot of some other widely studied models which can be use to illustrate a SPMRP, and after modeling a practical system in a SPMRP, the next step of us is to propose an effective algorithm to estimate the gradient of performance metric and search an optimized value following it.

IV. The Simulation-based Optimization

In this section, a simulation-based algorithm to computing the gradient of $\eta(\theta)$ is proposed, which is asymptotically replaced by $\nabla\bar{\eta}(\theta)$, and both optimizations of performance are evolving with the same sample path. Moreover, in this section, it will also be clarified that why steps of updating are less than the general algorithm in [1], and display the whole schedule of this algorithm. As proposed in Section 2, the extra estimators such as $\nabla v_\alpha(\theta)$ and $v_\alpha(\theta)$ are necessarily created before estimating $\nabla\bar{\eta}(\theta)$ from $\tilde{D}_\beta(\theta)$. In the first subsection we will show that only with the information in a single sample path long enough, we can get these terms in a recursive way approaching theoretical ones.

A. Estimators of $\nabla v_\alpha(\theta)$

Compared with the algorithm without states aggregation (such as in [1]), $\nabla v_\alpha(\theta)$ can not be neglect here, so a method invented to estimate these terms is necessary in our algorithm. However, these terms are not directly acquired through sample paths simulated by transition probability matrix $P^\varepsilon(\theta)$. To obtain them, we should first look into sample paths generated by matrices $P_\alpha(\theta)$ corresponding to estimate $\nabla v_\alpha(\theta)$. $v_\alpha(\theta)$ is steady state distribution vector of $P_\alpha(\theta)$, which can be easily estimated from counting the occurrence of each states along the sample path , and we also have balance equations:

$$v_\alpha(\theta)P_\alpha(\theta)=v_\alpha(\theta) \qquad\qquad v_\alpha(\theta)\tilde{1}_{m_\alpha}=1 \qquad (18)$$

Here, we will propose a method based on a single sample path generated by $P^\varepsilon(\theta)$ to approach results of theoretical ones of $P_\alpha(\theta)$. Now we take steady state probability of the *ith* state $v_\alpha^i(\theta)$, $i\in S_\alpha$ as a new performance with matrix $P_\alpha(\theta)$. To compute this performance and its gradient, we set one-step reward vector $\phi_\alpha^i=\{\phi_\alpha^{i1},\cdots,\phi_\alpha^{ij},\cdots,\phi_\alpha^{im_\alpha}\}\in\Re^{1\times m_\alpha}$ as:

$$\phi_\alpha^{ii}=1 \qquad \phi_\alpha^{ij}=0 \qquad \forall j\neq i \qquad i,j\in S_\alpha \qquad (19)$$

As the performance metric introduced in Section 2 as equation (2), to compute all parameters for the whole singularly perturbed problem, we first set the average reward criterion defined by

$$\mu(\theta)=\lim_{T\to+\infty}\frac{1}{T}E_\theta[\sum_{k=0}^{T}\phi_\alpha^{ii_k}(\theta)]$$

where the process can be seen as being generated by some $P_\alpha(\theta)$,so $i_k\in S_\alpha$, and we can easily find that $\mu(\theta)=v_\alpha^i(\theta)$, when we reform the average performance metric as

$$\mu(\theta)=v_\alpha(\theta)\phi_\alpha^i=v_\alpha^i(\theta)$$

And we rewrite the gradient of this average performance metric as

$$\nabla v_\alpha^i(\theta)=\sum_{i\in S_\alpha}v_\alpha^i(\theta)\sum_{j\in S_\alpha}\nabla p_{ij}(\theta)d_j^i(\theta) \tag{20}$$

where $p_{ij}(\theta)$ is an entry in $P(\theta)$ in (5), and $d_j^i(\theta)$ is defined as a differential reward corresponding to this partial problem to compute $\nabla v_j^i(\theta)=\nabla\mu(\theta)$, defined by

$$d_j^i(\theta)=E_\theta[\sum_{k=0}^{T-1}(\phi_\alpha^{ii_k}-v_\alpha^i(\theta))\,|\,i_0=j] \tag{21}$$

where $i,i_k,j\in S_\alpha$ and $T=\min\{k>0\,|\,i_k=i*\}$ is the first future epoch state $i*$ is visited. Through an approximation we can connect this theoretical result with our simulation sample paths as follows:

Lemma 4.1: *For any* $\alpha_i,\alpha_j\in S_\alpha$*, we have*

$$\left|\nabla\tilde{v}_\alpha^i(\theta)-\nabla v_\alpha^i(\theta)\right|=o(\varepsilon)$$

where

$$\nabla\tilde{v}_\alpha^i(\theta)=\sum_{i\in S_\alpha}v_\alpha^i(\theta)\sum_{j\in S_\alpha}\nabla p_{\alpha_i\alpha_j}^{\varepsilon}(\theta)d_j^i(\theta) \tag{22}$$

and α_i *is the* ith *state in the set* S_α.

It can be proof this problem in a short way as

$$\begin{aligned}\nabla v_\alpha^i(\theta)&=\sum_{i\in S_\alpha}v_\alpha^i(\theta)\sum_{j\in S_\alpha}\nabla p_{ij}(\theta)d_j^i(\theta)\\&=\sum_{i\in S_\alpha}v_\alpha^i(\theta)\sum_{j\in S_\alpha}\nabla[p_{\alpha_i\alpha_j}^{\varepsilon}(\theta)-\varepsilon q_{\alpha_i\alpha_j}(\theta)]d_j^i(\theta)\end{aligned}$$

So the single sample path can be used to approach theoretical results, and as $\varepsilon\to0$ have $\nabla\tilde{v}_\alpha^i(\theta)=\nabla v_\alpha^i(\theta)$.

From Theorem 3.2 in [5], for any $i_t\in S$, $\alpha_t\in\Omega$, $\alpha\in S_\alpha$, $\alpha\in\Omega$ we have

$$\sup_{t\in[0,T]}E_\theta\left|\varepsilon\sum_{t=0}^{k-1}(1_{\{i_t=S_{\alpha_j}\}}-v_\alpha^{\alpha_j}1_{\{\alpha_t=\alpha\}})\right|^2=o(\varepsilon) \tag{23}$$

So we can use the states series generated from the sample path to approximate the $v_\alpha^i(\theta)$ by

$$v_\alpha^i(\theta)=\frac{v_\varepsilon^i(\theta)}{\sum_{j\in S_\alpha}v_\varepsilon^j(\theta)}+o(\varepsilon)=\frac{\sum_{t=0}^{k-1}1_{\{i_t=S_{\alpha_j}\}}}{\sum\limits_{t=0}^{k-1}1_{\{\alpha_t=\alpha\}}}+o(\varepsilon) \tag{24}$$

where $v_\varepsilon^i(\theta)$ is the ith elements of $v_\varepsilon(\theta)$ which is steady state distribution vector of $P^\varepsilon(\theta)$.

Remark 4.2: *Although the whole process of the gradient of every steady state distribution is the same as that in Section 2, not only by equation (24) steady state distribution can be easily achieved, but also from the iteration in equation (22) we can also update it. So the whole process of computing* $\nabla v_\alpha(\theta)$ *will be more direct and simple than the algorithm proposed in Section 2.*

Remark 4.3: *From equation (24), it can be found out that all* $v_\alpha(\theta)$ *and* $\nabla v_\alpha(\theta)$ *are irrelative with any other aggregated state* $\beta\in\Omega$. *These terms can be generated from a successive series dominated by states in a constant subset of the original state space S, even when this series is interrupt by some other states, if and only if this series contains enough such states,* $\{i_k\}_{\{k<\infty,i_k\in S_k\}}$ *simulated in this single sample path.*

B. The Optimization for Disturbance-Controlled Model

In this subsection, a special but useful case will be taken into count. Transition matrix $P(\theta)$ is irrelative with parameter θ, and only the behavior of perturbation factor $D(\theta)$ is controlled by parameters. So equation (5) can be rewritten as $P^\varepsilon(\theta)=P+\varepsilon D(\theta)$. This case can be name as a *Disturbance Controlled Model*. Corresponding to the network mentioned in Section 3, it means that the resource of jobs, and Poisson rate of jobs as $\lambda(\theta)$, which is controlled by parameter, and the transition in three states: running, ready, and suspended is out of the influence of these parameters. In this case, we have $\nabla v_\alpha(\theta)=0$, $\forall\theta\in\Omega$, and derived from Theorem 2.1, there is:

Corollary 4.4: *Let assumption A4), A5) hold, and only disturbance factors are dependent on parameter* θ. *Then*

$$\nabla\overline{\eta}(\theta)=\sum_{\alpha\in\Omega}\overline{\pi}_\alpha(\theta)(\nabla g_\alpha(\theta)v_\alpha^T+\sum_{\beta\in\Omega}[\sum_{i\in S_\alpha}(\sum_{j\in S_\beta}\nabla d_{ij}(\theta))v_\alpha^i]\tilde{D}_\beta(\theta)) \tag{25}$$

Lemma 4.5: *The aggregated chain associated with the generator* $\overline{Q}(\theta)$ *is a Markov chain on a finite or countable state space* Ω *defined by* $\alpha_k=i_{T_k}$, *where* T_k s *are the successive epochs at which the chain enter another subset, i.e. aggregated state.*

This can be directly obtained from Lemma 2 in [6], and the proof is omitted here. From (25), we can find out that the gradient of the performance is irrelative with $\nabla v_\alpha(\theta)$, so that only some counters are required to record steady state distribution vectors. Meanwhile, a long-term decision series for $\nabla v_\alpha(\theta)$ are not necessary. Here, the whole schedule of our algorithm for this case will be listed in Algorithm 1, and from the expression of this algorithm, it can be easily found out that no matter how many states in the original state space S, the algorithm only evolves with the aggregated state space Ω.

```
Algorithm 1: Optimization Algorithm Based on Aggregation
Step 0  Compute $\bar{g}_\alpha(\theta)$, and $D^i(\theta)=\sum_{j\in\beta} d_{ij}(\theta)$ for all $\alpha,\beta\in\Omega$ and $i\in S$.
Step 1  Estimate the steady state distributions of $P_\alpha(\theta)$ as:
            $\hat{x}_{\alpha_i}(T)=\sum_{k=0}^{T} 1_{\{i_k=\alpha_i\}}$        $\hat{x}_\alpha(T)=\sum_{k=0}^{T} 1_{\{i_k\in S_\alpha\}}$
            $\hat{v}_\alpha^{\alpha_i}(T)=\frac{\hat{x}_{\alpha_i}(T)}{\hat{x}_\alpha(T)}$
        until for some $\delta$, $|\hat{v}_\alpha^{\alpha_l}(T+1)-\hat{v}_\alpha^{\alpha_l}(T)|<\delta$. By the way, this step can be completed in just one segment of the sample path, in which the states of subset $S_\alpha$ take place far more frequently than others.
Step 2  Compute some factors
            $\hat{F}_1^\alpha(\theta)=\hat{v}_\alpha \bar{g}_\alpha^T(\theta)$        $\hat{F}_2^{\alpha\beta}(\theta)=\hat{v}_\alpha[D^\alpha(\theta)]^T+1$
        where $\hat{v}_\alpha=(\hat{v}_\alpha^{i_1},\cdots,\hat{v}_\alpha^{i_{m_\alpha}})$,
              $D^\alpha(\theta)=(D^{i_1}(\theta),\cdots,D^{i_{m_\alpha}}(\theta))$
        and $i_{1,\cdots,}i_{m_\alpha}\in S_\alpha$.
Step 3  Recursive in every epoch $k$, where $\alpha_k=i_{T_k}\neq i_{T_k-1}$ and compute following equation iteratively until the recurrent aggregated state $S_*$ or $\alpha_*$ is first revisited in future:
            $\hat{D}(\theta,\hat{\eta}(\theta_m))=\sum_{k=n}^{t_{m+1}-1}(\hat{F}_1^{\alpha_k}(\theta_m)-\hat{\eta}(\theta_m))$
            $F_m(\theta_m,\hat{\eta}(\theta_m))=\sum_{k=t_m}^{t_{m+1}-1}(\nabla\hat{F}_1^{\alpha_n}(\theta_m)+\hat{D}_{i_n}(\theta_m,\hat{\eta}(\theta_m))\frac{\nabla\hat{F}_2^{\alpha_{n-1}\alpha_n}(\theta_m)}{\hat{F}_2^{\alpha_{n-1}\alpha_n}(\theta_m)})$
            $\theta_{m+1}=\theta_m+\gamma_m F_m(\theta_m,\hat{\eta}(\theta_m))$
            $\hat{\eta}_{m+1}(\theta)=\hat{\eta}_m(\theta)+\lambda\gamma_m\sum_{n=t_m}^{t_{m+1}-1}(\bar{g}_{\alpha_n}(\theta_m)-\hat{\eta}_m(\theta_m))$
        where $\gamma_m$ is a positive step size sequence, $\lambda>0$ allows to scale the step size for updating $\hat{\eta}_m(\theta)$ by a positive constant, and $\alpha_k$ is the aggregated state.
Step 4  When $\alpha_*$ is revisited, return to Step 1, unless $|\bar{\eta}(\theta_{t_{m+l}})-\bar{\eta}(\theta_{t_m})|<\delta$, where $\delta>0$ is a number small enough. Sometimes, to need a more exact result, we should require a performance series $\{\bar{\eta}(\theta_{t_{m+1}}),k=1,2,\cdots l\}$ enter some stable domain, where $l$ is large enough.
```

From the expression of this algorithm, it is clear that no matter how many states in the original state space S, the algorithm only evolve with the aggregated state in the aggregated space Ω. For example, even when there is a SPMRP problem with ten thousand states, if the whole state space can be aggregated as a problem with ten states, the complexity and updating steps of this algorithm is the same as a small-dimension problem with ten states. Henceforth, our methods can simplify a class of large-scale states problem in Markov Decision Processes and Markov Reward Processes.

V. Simulation And Results

In this section, some examples will be given to illustrate our algorithm, which will update the performance and parameters along the sample path generated by a singularly perturbed Markov processes. Here, name every epoch of the data updating as iteration, and iteration steps is an important measurement to value a method. The sample path generated by the transition probability matrix with the form:

$$P(\theta)=\begin{pmatrix} P_1 & \Theta & \Theta \\ \Theta & P_2 & \Theta \\ \Theta & \Theta & P_3 \end{pmatrix}+\varepsilon\times\begin{pmatrix} D_{1,1}(\theta) & D_{1,2}(\theta) & D_{1,3}(\theta) \\ D_{2,1}(\theta) & D_{2,2}(\theta) & D_{2,3}(\theta) \\ D_{3,1}(\theta) & D_{3,2}(\theta) & D_{3,3}(\theta) \end{pmatrix}$$

where all entries in matrix Θ are zeros, and the disturbance matrix is controlled by parameters. All entries in matrix $D_{\alpha,\beta}(\theta)$ are larger than 0, $\forall\theta\in\Re^k$, for any $\alpha\neq\beta$, and here chose $\varepsilon=0.001$, and here $P(\theta)$ is a transition matrix. All blocks of transition matrix P_α, $\alpha=1,2,3$ are with a large-scale state space. The current state i^* is assumed in the subset S_1 corresponding to the transition matrix P_1, so aggregated state 1 is the recurrent state in our simulation.

In our algorithm, view the sample path as the evolution of aggregated states. So the recurrent state should be redefined in the sense of aggregation. Denote this new state by α^*. The other details in our simulation are omitted here, such as the value of every entry in matrix P_α, and the initial state in our simulation. Obviously, from our results displayed in those figures below, it is clear that algorithm with state aggregation will lead to a smoother optimization and faster convergence.

The results in Fig.5-(a) and Fig.5-(b) show that the convergence in algorithm with state aggregation will be better than the one without aggregation. It is just because of the sample path of the singularly perturbed Markov chain, which always transits among some long segments, each of which are dominated by a certain subsets $S_\alpha,\alpha=1,2,3$, so iterations in an algorithm without aggregation are constrained as a local optimization within some subset until the single sample path reach a new segment dominated by another subset. Here, results of the algorithm without aggregation are also given to compare the metrics our methods.

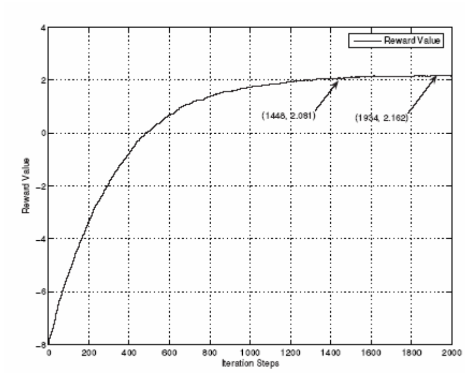

Fig. 5 (a) Iteration for performance with state aggregation

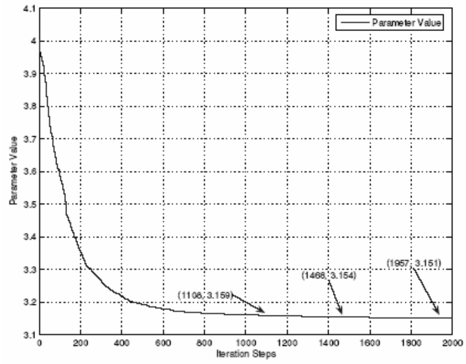

Fig. 5 (b) Iteration for parameter with state aggregation

To illustrate new algorithm more clearly, we give another example, a singularly perturbed Markov reward process whose states can be divided into two subsets. Hence, we can clarify the transitions between these subsets from simulation more precisely. Here, we give our results of simulation for this example in Fig. 6 and Fig. 7:

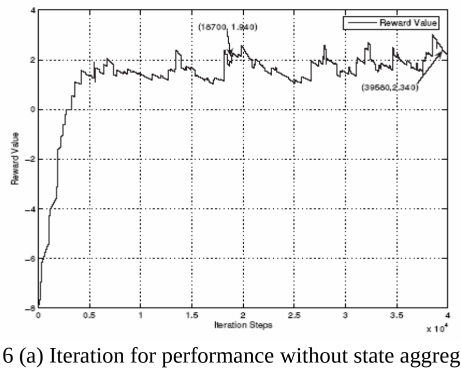

Fig. 6 (a) Iteration for performance without state aggregation

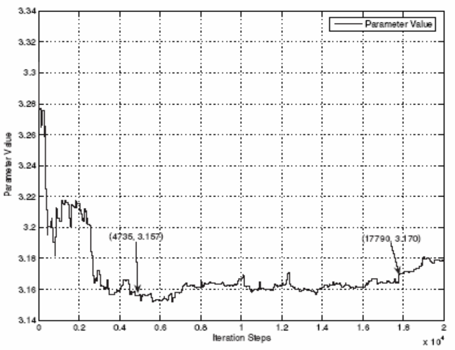

Fig. 6 (b) Iteration for parameter without state aggregation

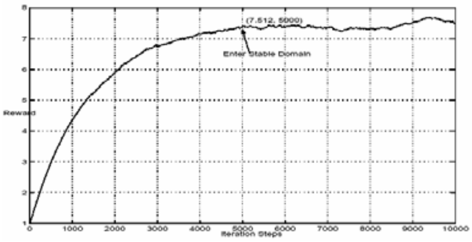

Fig. 7 (a) Iteration for performance with state aggregation

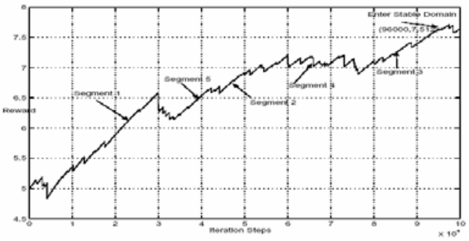

Fig. 7 (b) Iteration for performance with state aggregation

From this figure, there is some undulation caused by the transition between two different subsets. In this simulation, let the recurrent state in the first subset, and Segment 1, Segment 5 and Segment 3 are dominated by the first subset, so the change in these parts of the sample path contribute a lot to the whole optimization, while Segment 4 is dominated by the other subset, and it evolves with variance accumulation. Fig.8 is a curve of theoretical performances around the optimize parameter:

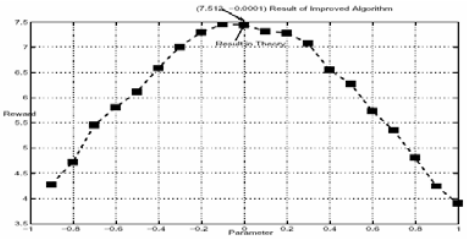

Fig. 8 Performances of a set of parameters around the optimized value

VI. Conclusion

In this paper, the gradient of performance of singularly perturbed Markov reward processes with aggregated states is obtained, and an algorithm is designed to save iteration steps and the mount of computing during the optimization. With the character that the new algorithm is based on the single sample path, it also can optimize the objective on-line with the evolution of practical processes. Furthermore, the special case in Section 4 is also a widely applied model in practice, so that the schedule set in this section will be helpful in future studies of singularly perturbed Markov reward processes, which has been partly revealed by these simulations.

Appendix

Proof of **Theorem 2.1**:

Carry out the proof using vector notation, and using the superscript T to denote the transpose. All gradient are taken with respect to parameter vector θ.

First of all, let us take Poisson equation into consideration:

$$\bar{g}(\theta)=\tilde{D}(\theta)+\bar{\eta}(\theta)e_n-\bar{P}(\theta)\tilde{D}(\theta)$$

and left-multiply both sides with $\nabla\bar{\pi}^T(\theta)$, which is steady state distribution after aggregation, to obtain:

$$\nabla\bar{\pi}^T(\theta)\bar{g}(\theta)=\nabla\bar{\pi}^T(\theta)\tilde{D}(\theta)+\bar{\eta}(\theta)\nabla\bar{\pi}^T(\theta)e_n-\nabla\bar{\pi}^T(\theta)\bar{P}(\theta)\tilde{D}(\theta) \quad (26)$$

Note that $\bar{\pi}^T(\theta)e_n=1$, so we have $\nabla\bar{\pi}^T(\theta)e_n=0$. Using the balance equation $\bar{\pi}^T(\theta)\bar{P}(\theta)=\bar{\pi}^T(\theta)$, we have:

$$\nabla\bar{\pi}^T(\theta)=\nabla(\bar{\pi}^T(\theta)\bar{P}(\theta))=(\nabla\bar{\pi}^T(\theta))\bar{P}(\theta)+\bar{\pi}^T(\theta)(\nabla\bar{P}(\theta))$$

Right-multiply the equation above both sides by $R(\theta)$, have:

$$\nabla\bar{\pi}^T(\theta)\tilde{D}(\theta)=(\nabla\bar{\pi}^T(\theta))\bar{P}(\theta)\tilde{D}(\theta)+\bar{\pi}^T(\theta)(\nabla\bar{P}(\theta))\tilde{D}(\theta)$$

and using the result in (26), we have:

$$\begin{aligned}\nabla\bar{\pi}^T(\theta)\bar{g}(\theta)&=(\nabla\bar{\pi}^T(\theta))\bar{P}(\theta)\tilde{D}(\theta)+\bar{\pi}^T(\theta)(\nabla\bar{P}(\theta))\tilde{D}(\theta)\\&\quad+\bar{\eta}(\theta)\nabla\bar{\pi}^T(\theta)e_n-\nabla\bar{\pi}^T(\theta)\bar{P}(\theta)\tilde{D}(\theta)\\&=\bar{\pi}^T(\theta)[\nabla\bar{P}(\theta)]\tilde{D}(\theta)\end{aligned}$$

Thus:

$$\bar{\eta}(\theta)=\nabla[\bar{\pi}^T(\theta)\bar{g}(\theta)]=\bar{\pi}^T(\theta)\nabla\bar{g}(\theta)+\bar{\pi}^T(\theta)[\nabla\bar{P}(\theta)]\tilde{D}(\theta) \quad (27)$$

so, take the definition of the items in aggregated problem into equation (27), then:

$$\begin{aligned}\nabla\bar{\eta}(\theta)&=\bar{\pi}_\alpha(\theta)\sum_{\alpha\in\Omega}\nabla[g_\alpha(\theta)v_\alpha^T(\theta)]+\\&\qquad\bar{\pi}_\alpha(\theta)\nabla[diag\{v_1(\theta),\cdots,v_\alpha(\theta)\cdots,v_{|\Omega|}(\theta)\}D(\theta)\tilde{1}+I]\tilde{D}(\theta)\end{aligned}$$

$$\begin{aligned}&=\sum_{\alpha\in\Omega}\bar{\pi}_\alpha(\theta)(\nabla g_\alpha(\theta)v_\alpha^T(\theta)+g_\alpha(\theta)\nabla v_\alpha^T(\theta)+\\&\sum_{\beta\in\Omega}[\sum_{i\in S_\alpha}(\sum_{j\in S_\beta}\nabla d_{ij}(\theta))v_\alpha^i(\theta)+\sum_{i\in S_\alpha}(\sum_{j\in S_\beta}d_{ij}(\theta))\nabla v_\alpha^i(\theta)]\tilde{D}_\beta(\theta))\end{aligned}$$

where the $d_{ij}(\theta)$ is the $(i,j)th$ element of matrix $D(\theta)$. If the assumption that the transition matrix $P(\theta)$ is irrelative of the parameter θ, then comes the result:

$$\nabla\bar{\eta}(\theta)=\sum_{\alpha\in\Omega}\bar{\pi}_\alpha(\theta)(\nabla g_\alpha(\theta)v_\alpha^T+\sum_{\beta\in\Omega}[\sum_{i\in S_\alpha}(\sum_{j\in S_\beta}\nabla d_{ij}(\theta))v_\alpha^i]\tilde{D}_\beta(\theta))$$

This completes our proof.

References

[1] P. Marbach and J.N. Tsitsiklis "*Simulation-based optimization of Markov reward processes,* " IEEE Transactions on Automatic Control, Vol. 46, No.2, February, 2001, pp.191-209.

[2] G. Yin and Q. Zhang "*Singularly perturbed discrete-Time Markov chains,*" SIAM Journal Appl. Math. Vol.61, No.3, 2 000, pp 834-854.

[3] M. Abbad and J.A. Filar "*Perturbation and Stability Theory for Markov Control Problems,*" IEEE Transactions on Automatic Control, Vol.37, No.9, September, 1992, pp 1415-1420.

[4] M. Abbad and J. A. Filar "*Algorithms for singularly perturbed limiting average Markov Control Problem,*" *IEEE Transactions on Automatic Control*, Vol.37, No.9, September, 1992, pp 1421-1425.

[5] G. Yin Q. Zhang and G. Badowski "*Discrete-time singularly perturbed Markov chain: Aggregation, Occupation measures, and switching diffusion limit,*" Adv. Appl. Prob.Vol.35, 2003, pp 449-476.

[6] F. Delebecque "*A reduction process for perturbed Markov chains,*" SIAM Journal Appl. Math. Vol. 43, No.2, April 1983.

[7] R.H. Liu, Q. Zhang and G. Yin "*Singularly perturbed Markov decision processes in discrete time,*" Decision and Control, 2001. Proceedings of the 40th IEEE Conference on. Vol. 3, 4-7 December. 2001, pp 2119 – 2124.

[8] M. Abbad, J.A. Filar, and T.R. Bielecki, "*Singularly perturbed Markov control problem: limiting average cost,*" Decision and Control, 1989. Proceedings of the 28th IEEE Conference on. Vol. 2, 4-7 December. 1989- pp. 1263 - 1266.

[9] Xi-ren Cao, "*The potential structure of sample paths and performance sensitivities of Markov systems The potential structure of sample paths and performance sensitivities of Markov systems,*" IEEE Transactions on Automatic Control, Vol. 49, Issue,12 , December. 2004 pp. 2129 - 2142.

[10] H. Fang and Xi-ren Cao "*Potential-based online policy iteration algorithms for Markov decision processes,*" IEEE Transactions on Automatic Control, Vol. 49 , Issue, 4 , April. 2004, pp. 493 - 505.

[11] H. Tang, H. Xi and B. Yin. "*Performance optimization of continuous time Markov control processes based on performance potentials,*". Int. Journal of Systems Science, 2003, Vol. 34(1), pp, 63-71.

[12] D. Zhang, H. Xi and B. Yin "*Simulation-based optimization for singularly perturbed Markov reward process with aggregated states,*" Int. Conference of Intelligence Computing 2005. Accepted.

[13] J. Ledoux "*On weak lump ability of denumerable Markov chains,*" Statistics & Probability Letters, Vol. 25, 1995, pp, 329-339.

[14] J.R. Jackson. "*Job shop-like queuing systems,*" Man. Sci. Vol. 9 October, 1963, pp, 131-142.

[15] P.J. Courtois. "*On the near-complete-decomposability of networks of queues and of stochastic models of multiprogramming computing system*". Scientific Rep. CMU-CS-72-11, Carnegie-Mellon U, November, 1971