Available online at www.sciencedirect.com
SCIENCE DIRECT
automatica
Automatica 42 (2006) 393–403
www.elsevier.com/locate/automatica

The control of a two-level Markov decision process by time aggregation ☆

Yat-wah Wan [a], Xi-Ren Cao [b, *, 1]

[a] Institute of Global Operations Strategy and Logistics Management, National Dong Hwa University, Hualien, Taiwan
[b] Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong

Received 14 September 2003; received in revised form 13 September 2005; accepted 24 November 2005

Abstract

The solution of Markov Decision Processes (MDPs) often relies on special properties of the processes. For two-level MDPs, the difference in the rates of state changes of the upper and lower levels has led to limiting or approximate solutions of such problems. In this paper, we solve a two-level MDP without making any assumption on the rates of state changes of the two levels. We first show that such a two-level MDP is a non-standard one where the optimal actions of different states can be related to each other. Then we give assumptions (conditions) under which such a specially constrained MDP can be solved by policy iteration. We further show that the computational effort can be reduced by decomposing the MDP. A two-level MDP with M upper-level states can be decomposed into one MDP for the upper level and M to $M(M-1)$ MDPs for the lower level, depending on the structure of the two-level MDP. The upper-level MDP is solved by time aggregation, a technique introduced in a recent paper [Cao, X.-R., Ren, Z. Y., Bhatnagar, S., Fu, M., & Marcus, S. (2002). A time aggregation approach to Markov decision processes. *Automatica*, *38*(6), 929–943.], and the lower-level MDPs are solved by embedded Markov chains.

Keywords: Time aggregation; Markov decision processes; Two-level systems; Coupled decisions; Policy iteration; Performance potentials

1. Introduction

In Markov Decision Processes (MDPs) of two-level hierarchical structures, the states are formed by the status of both the upper and the lower levels and state changes can be caused by the status changes at either level. Decisions are also made at each level. Such decisions affect both the state transitions of the two levels and the reward of the MDPs. The existing solutions of such MDPs often rely on the difference in the *time scales* of the two levels, i.e., the rate of state changes in the lower level is faster than that of the upper level by multiple orders of magnitude. Between two upper-level state changes, on average the lower level has already gone through so many state changes that one can make use of the long-run sum or average from the lower level to make decision at an upper-level state change. See Chang, Fard, Marcus, and Shayman (2003) and its references for examples of MDPs with multiple time scales.

In a different context and for a different purpose, singularly perturbed MDPs also make use of two time scales (Abbad, Filar, & Bielecki, 1992; Bielecki & Filar, 1991). Reducible transition probability matrices of policies in the form of disjoint, closed communicating classes are made positive by perturbing with a policy dependent matrix εD, where $\varepsilon > 0$ is a constant. By controlling ε, the inter-class state transitions can be less frequent than the intra-class. One main result of singularly perturbed MDPs is that the optimal control policy for the "limit control MDPs", the cases as $\varepsilon \to 0$, is a good control policy for singularly perturbed MDPs of a sufficiently small ε.

The idea of two time scales also occurs in hybrid stochastic systems (Filar, Gaitsgory, & Haurie, 2001; Filar & Haurie, 2001). The operation modes in the upper level are modeled by Markov jump processes and the characteristics of the lower level are modeled either by deterministic functions (Filar et al., 2001) or by diffusion processes (Filar & Haurie, 2001), with

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Ioannis Paschalidis under the direction of Editor Ian Petersen.

* Corresponding author. Tel.: +852 2358 7048; fax: +852 2358 1485.
E-mail address: eecao@ee.ust.hk (X.-R. Cao).

[1] The research was partially supported by a grant from Hong Kong RGC.

0005-1098/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2005.11.006

both the upper-level operations modes and the lower-level characteristics controllable.

In this paper, we study a class of the two-level MDPs under the long-run average reward criterion. Our two-level MDPs take a structure similar to MDPs of two time scales except that our two-level model is an atypical MDP: an upper-level decision at a state can affect the state transitions of a *group* of states with the same upper-level state. The various decisions, within the lower levels and spanning across both levels, are *coupled*, i.e., we cannot single out and solve the decisions level by level, purpose by purpose. Such coupling effect increases the computational burden to solve the two-level MDP, making it practically infeasible for real-life problems. We then show that the coupling effect disappears if (i) the *sojourn times* of each upper-level state—the duration between entering and leaving an upper-level state—are uncontrollable, and (ii) the set of the initial lower-level state distributions after an upper-level state change is independent of the lower-level states before the upper-level state change. With both of these assumptions, the decisions are decoupled; the computational effort for the optimal policy becomes manageable; and it is possible to implement the centralized control scheme in a decentralized fashion.

To further reduce the computational effort, we show that the whole problem can be decomposed into smaller MDPs, one for the upper level, and a number of MDPs for the lower level, where the number depends on the problem structure. Our solution of the upper-level MDP uses *time aggregation* (Cao, Ren, Bhatnagar, Fu, & Marcus, 2002), and our lower-level of embedded Markov chains. Combining the algorithms of the two levels solves the two-level MDP.

Our model is related to that in Chang et al. (2003), though the two models have two critical differences. First, the number of lower-level state changes between two upper-level state changes is fixed in Chang et al. (2003) but random here. With constant time between two upper-level state changes, it is conceptually straightforward to embed on upper-level state changes and make use of the total rewards in upper-level sojourn times to look for optimal decisions. Nonetheless, the computational effort for all possible nonstationary policies is too large to be materialized. Consequently, Chang et al. (2003) looks for approximations, and bounds the performance of approximations in the same spirit of those in single-level MDPs. As for our model, the calculation of the total rewards of sojourn times is made involved by randomness. We still find computationally simple close-form expressions for such quantities, based on which later we give an exact analysis of the two-level MDP. Second, the upper- and lower-level decisions in Chang et al. (2003) are not coupled as ours. There, any consideration for a state, e.g., its upper-level decision, can be made purely based on the cost and benefit of the state. However, in our model, as shown in (1) below, states with the same upper level must take the same upper-level decisions, a constraint that makes our MDP unconventional.

Our contributions are as follows. First, we tackle a two-level MDP from a perspective that does not rely on multiple time scales. Our analysis is exact, and we allow for random number of lower-level state changes between two upper-level state changes. Second, our two-level control problem is not a standard MDP because each action at the upper level applies to a group of states. We solve such a specially constrained MDP under different assumptions. Third, other than satisfying with an algorithm that solves the two-level MDP, we go further to find algorithms that take less computational effort, which is a continuation of one of the authors' previous work on time aggregation (Cao et al., 2002). Such algorithms can be implemented as the optimal decentralized control. Finally, our approach sheds light on hybrid systems where the lower level is modeled as continuous systems.

2. The two-level MDP

Consider a two-level MDP with M upper-level states and N_i lower-level states for the ith upper-level state, $i = 1, \ldots, M$. In general, N_i can be different from N_m for $1 \leqslant i \neq m \leqslant M$, and in specific applications, lower-level states of different upper levels can bear different physical meanings. For ease of reference, we call an upper-level state a *mode*, and its lower-level states *settings* of the mode. The state of the MDP at period t is denoted by (X_t, Y_t), where X_t is the mode and Y_t is the setting at period t, $t \in \{1, 2, \ldots\}$. (Technically, the state is policy dependent. We omit such dependence for notational simplicity when we refer to a generic case. The same convention is adopted throughout.)

Let $\mathscr{X}_i = \{(i, 1), \ldots, (i, N_i)\}$ be the collection of states in mode i, $i = 1, \ldots, M$. The state space of the MDP is $\mathscr{X} = \mathscr{X}_1 \cup \cdots \cup \mathscr{X}_M$. Arrange the states in the lexicological order with the modes as the primary and the settings as the secondary keys. In ascending lexicological order of states, $\mathscr{X} = \{(1, 1), \ldots, (1, N_1), \ldots, (M, 1), \ldots, (M, N_M)\}$. Define also $\mathscr{X}^u = \{1, \ldots, M\}$ as the collection of mode states.

When the two-level MDP adopts policy $\mathscr{L}$, the policy dictates an action $\alpha_{(i,j)} \in \mathscr{A}_{(i,j)}$ at state (i, j), where $\mathscr{A}_{(i,j)}$ is the action set of the state. Let $\mathscr{A} = \cup_{(i,j)} \mathscr{A}_{(i,j)}$ be the action set of the two-level MDP. We consider only finite action sets and stationary policies.

As in Chang et al. (2003), assume that the evolution of modes in the two-level MDP is not affected by the actions from the lower-level settings. That is, the decision on controlling mode changes can only depend on the upper-level states. Thus, when action $\alpha_{(i,j)}$ is adopted at state (i, j) at time t, the mode change probability $P(X_{t+1} = m | (X_t, Y_t) = (i, j), \alpha_{(i,j)})$ depends only on a decision based on i. We call such a sub-decision (embedded in $\alpha_{(i,j)}$) an upper-level action and denote it as α_i^u. Thus, we have

$$P(X_{t+1} = m | (X_t, Y_t) = (i, j), \alpha_{(i,j)}) = P(X_{t+1} = m | X_t = i, \alpha_i^u) = r_{(i,m)}^{\alpha_i^u}. \quad (1)$$

The mode process itself is an MDP with transition probability matrix $R^{\mathscr{L}^u} = [r_{(i,m)}^{\alpha_i^u}]$, where $\mathscr{L}^u$ denotes the mode policy induced from $\mathscr{L}$. Let $\mathscr{A}_i^u$ be the collection of all upper-level actions $\{\alpha_i^u\}$ at mode i, and $\mathscr{A}^u = \bigcup_i \mathscr{A}_i^u$. $\mathscr{A}^u$ is a finite set.

Assumption 2.1. $R^{\mathscr{L}^u}$ is an irreducible transition probability matrix in $\mathscr{X}^u$ for any $\mathscr{L}^u$.

Assumption 2.1 ensures that all modes are non-trivial. The MDP is decomposable into smaller independent MDPs if Assumption 2.1 is violated.

Assumption 2.2. At any period, the setting action is made infinitesimally later than the corresponding realization of the mode action.

This is a purely technical assumption often made in literature (e.g., Chang et al., 2003). It matches with dynamics in real-life applications; e.g., usually strategic decisions (upper-level actions) have already been implemented when operations decisions (lower-level actions) are considered.

Consider now the setting changes. Suppose that action $\alpha_{(i,j)}$ is adopted at state (i, j) at period t. If the mode remains unchanged at the next period, the conditional setting probability is $P(Y_{t+1}=n|Y_t=j, \alpha_{(i,j)}, X_{t+1}=i, X_t=i)$. Denote the decision that determines this transition probability as action $\alpha^c_{(i,j)}$, i.e.,

$$P(Y_{t+1}=n|Y_t=j, \alpha_{(i,j)}, X_{t+1}=i, X_t=i) =: s^{\alpha_{(i,j)}}_{(j,n)} = s^{\alpha^c_{(i,j)}}_{(j,n)}.$$

When the mode changes from i to $m \neq i$ at the next period,

$$P(Y_{t+1}=n|Y_t=j, \alpha_{(i,j)}, X_{t+1}=m, X_t=i) =: q^{\alpha_{(i,j)}}_{(i,j),(m,n)} = q^{\alpha^p_{(i,j)}}_{(i,j),(m,n)},$$

where $\alpha^p_{(i,j)}$ denotes the sub-action that determines the initial setting after the mode change.

Define the transition probability matrix $S^{(i),\mathscr{L}} = [s^{\alpha_{(i,j)}}_{(j,n)}]$ for mode i when the mode remains unchanged after a transition, $i = 1, \ldots, M$, and $Q^{(i,m),\mathscr{L}} = [q^{\alpha_{(i,j)}}_{(i,j),(m,n)}]$ when the mode changes from i to m, $1 \leqslant i \neq m \leqslant M$.

Assumption 2.3. $S^{(i),\mathscr{L}}$ is irreducible for any $\mathscr{L}$, $1 \leqslant i \leqslant M$.

By Assumption 2.3, the settings in mode i are non-trivial. When combined together, Assumptions 2.1 and 2.3 ensure that states of the MDP communicate for any policy.

The above discussion reveals that action $\alpha_{(i,j)}$ can be represented as the triple $(\alpha^u_i, \alpha^c_{(i,j)}, \alpha^p_{(i,j)})$, where the mode action α^u_i determines mode changes, the initial setting distribution action $\alpha^c_{(i,j)}$ determines the setting when there is a mode change, and the setting transition action $\alpha^p_{(i,j)}$ that determines the setting when a mode preserves. Because the latter two components are related to the lower level, we also use the notation lower-level action $\alpha^l_{(i,j)} = (\alpha^c_{(i,j)}, \alpha^p_{(i,j)})$. Such a convention is also adopted by policy. Thus, the policy $\mathscr{L} = (\mathscr{L}^u, \mathscr{L}^c, \mathscr{L}^p)$ has three components, and the lower-level policy $\mathscr{L}^l = (\mathscr{L}^c, \mathscr{L}^p)$ has two.

While actions and policies are in component forms, without further justification, it is impossible to determine the optimal actions component by component. For example, it is not clear whether the determination of the lower-level component $\alpha^c_{(i,j)}$ affected by that of $\alpha^p_{(i,j)}$ and α^u_i. Even for the determination of α^u_i, the upper-level action to which the lower-level actions play no direct effect, it is not clear what effect the difference in settings (of the same mode) will assert on the decisions of the optimal policies at the upper level.

Contrarily to the possible interaction of the optimal decision, there is no interaction of the action sets for different types of components, at any state. Specifically, the action set of $\alpha^c_{(i,j)}$ does not depend on the choice of $\alpha^p_{(i,j)}$; nor does the action set of $\alpha^p_{(i,j)}$ depending on the choice of $\alpha^p_{(m,n)}$ for $(i, j) \neq (m, n)$.

When we consider all the actions simultaneously, the transition probability becomes

$$P((X_{t+1}, Y_{t+1}) = (m, n)|(X_t, Y_t) = (i, j), \alpha_{(i,j)}) = \begin{cases} r^{\alpha^u_i}_{(i,i)} s^{\alpha_{(i,j)}}_{(j,n)} & \text{if } m = i; \\ r^{\alpha^u_i}_{(i,m)} q^{\alpha_{(i,j)}}_{(i,j),(m,n)} & \text{if } m \neq i. \end{cases} \quad (2)$$

For a given policy, $\{(X_t, Y_t)\}$ is a discrete-time Markov chain (DTMC) with the transition probability matrix

$$P = \begin{bmatrix} r_{(1,1)} S^{(1)} & r_{(1,2)} Q^{(1,2)} & \cdots & r_{(1,M)} Q^{(1,M)} \\ r_{(2,1)} Q^{(2,1)} & r_{(2,2)} S^{(2)} & \cdots & r_{(2,M)} Q^{(2,M)} \\ \vdots & \vdots & \vdots & \vdots \\ r_{(M,1)} Q^{(M,1)} & r_{(M,2)} Q^{(M,2)} & \cdots & r_{(M,M)} S^{(M)} \end{bmatrix}. \quad (3)$$

By Assumptions 2.1 and 2.3, P is irreducible and hence positive. Whether P is aperiodic or not depends also on $S^{(i)}$, $i = 1, \ldots, M$, and $Q^{(i,m)}$, $1 \leqslant i \neq m \leqslant M$.

Let $(\cdot)^{\mathrm{T}}$ be the transpose of a vector or a matrix $(\cdot)$. The performance function of mode i, $f_i = (f_{(i,1)}, \ldots, f_{(i,N_i)})^{\mathrm{T}}$, is the column vector of reward per period when the MDP is at settings 1 to N_i of mode i; the performance function of the MDP is then

$$f = (f_{(1,1)}, \ldots, f_{(M,N_M)})^{\mathrm{T}}. \quad (4)$$

With a positive chain structure and a finite decision set for any policy, the optimal policy for the long-run average reward is well-defined. The objective is to find an optimal policy $\mathscr{L}^*$ that attains

$$\max_{\mathscr{L}} \left(\lim_{\tau \to \infty} \frac{\sum_{t=1}^{\tau} E[f_{(X_t^{\mathscr{L}}, Y_t^{\mathscr{L}})}]}{\tau} \right), \quad (5)$$

subject to the transition probabilities specified in (2) as dictated by $\mathscr{L}$.

We will explain in the next section that our two-level MDP is a non-standard one in which some actions of groups of states are coupled. Such coupling eludes standard solution techniques. Fortunately, if the class of MDP possesses certain structural properties that weaken the interaction of states, policy iteration algorithms can be derived to find optimal policies. These structural properties, defined in Assumptions 2.4 and 2.5 below, are also necessary to guarantee the existence of such a solution (i.e., if they are not satisfied, policy iteration does not work except for very special cases). In addition, when such properties hold, the computational effort can be reduced by decomposing the MDP with a large state space into a number of MDPs with small state spaces.

Assumption 2.4. For every mode i, $r_{(i,i)}$ is not controllable, i.e., for any mode action α_i^u, $r_{(i,i)}^{\alpha_i^u} = \zeta_i$, $0 < \zeta_i < 1$, a constant for mode i, $1 \leqslant i \leqslant M$.

Assumption 2.4 says that the time staying in a mode is not affected by any policy. This often reflects the intrinsic randomness of the system. For example, a company cannot change the global economic conditions, nor the built-in lifetime of a bought machine.

Assumption 2.5. Consider any mode m, $1 \leqslant m \leqslant M$. Suppose that it is possible to change to mode m from any two states (i_1, j_1) and (i_2, j_2), where $j_1 \in \mathscr{X}_{i_1}$, $j_2 \in \mathscr{X}_{i_2}$, $i_1, i_2 \neq m$, and i_1 and i_2 may or may not be the same. Then for any action $\alpha_{(i_1,j_1)}$ at state (i_1, j_1), there exists action $\alpha_{(i_2,j_2)}$ at state (i_2, j_2) such that the initial setting distributions of mode m are the same, i.e., $q_{(i_1,j_1),(m,n)}^{\alpha_{(i_1,j_1)}} = q_{(i_2,j_2),(m,n)}^{\alpha_{(i_2,j_2)}}$ for all $1 \leqslant n \leqslant N_m$.

Assumption 2.5 says that the set of all possible initial setting distributions in a new mode m does not depend on the previous state. For example, if there are, say, 20 different initial setting distributions for a change to mode m from state (i_1, j_1), then there are the same 20 initial setting distributions from any state $(i, j) \in \mathscr{X}$ that change to mode m. States can have more than 20 actions to mode m; two or more actions from a state can lead to the same initial setting distribution. The assumption is to model abrupt changes that make settings in the previous mode irrelevant.

With Assumptions 2.1–2.5, we show that the two-level MDP is solvable by policy iteration, and the computation can further be reduced. Our main result is:

Theorem 2.6. *Suppose that Assumptions* 2.1–2.5 *hold for a two-level MDP. Then its optimal policy can be found by solving* $(M+1)$ *MDPs, the upper-level MDP of M states, and M lower-level MDPs such that the ith one is of* N_i *states*, $i = 1, \ldots, M$.

There is a trade off of model generality and computational effort. The following assumption is less restrictive on the initial setting distributions of new modes.

Assumption 2.7. Consider any mode m, $1 \leqslant m \leqslant M$. Suppose that it is possible to change to mode m from any two states (i, j_1) and (i, j_2) of the same mode, where $j_1, j_2 \in \mathscr{X}_i$, $i \neq m$. Then for any action $\alpha_{(i,j_1)}$ at state (i, j_1), there exists action $\alpha_{(i,j_2)}$ at state (i, j_2) such that the initial setting distributions of mode m are the same, i.e., $q_{(i,j_1),(m,n)}^{\alpha_{(i,j_1)}} = q_{(i,j_2),(m,n)}^{\alpha_{(i,j_2)}}$ for all $1 \leqslant n \leqslant N_m$.

Assumption 2.7 says that the set of all possible initial setting distributions in a new mode m after a jump from mode i is the same for all states in mode i. This is less restrictive than Assumption 2.5, which requires that the set of initial distributions is also independent of i. With this more flexible problem structure, a two-level MDP is still solvable by policy iteration, though it takes a bit more computational effort.

Theorem 2.8. *Suppose that Assumptions* 2.1–2.4, *and* 2.7 *hold for a two-level MDP. Then its optimal policy can be found by solving at most* $M(M-1)+1$ *MDPs: the upper-level MDP of M states, and one lower-level MDP of size* N_m *for each possible mode transition from mode i to mode m*, $1 \leqslant i \neq m \leqslant M$. *Whenever Assumption* 2.5 *holds for modes* i_1 *and* i_2 *for a mode change to m*, $1 \leqslant i_1 \neq i_2 \leqslant M$, *the same lower-level MDP works for both modes* i_1 *and* i_2, *reducing the computational effort of one lower-level MDP of size* N_m.

3. The coupling and decoupling of the two-level MDP

We will use policy iteration (cf. Cao, 1998, 1999; Cao & Chen, 1997; Cao, Yuan, & Qiu, 1996) to explain the coupling effect of actions and the way to decouple them. Other methods face the same difficulty as policy iteration for the coupled two-level MDP, and they are not as efficient to decouple actions.

3.1. The coupling of the two-level MDP

Let P be the transition probability matrix of a given policy, say, the *base* policy, upon which we want to improve for a general two-level MDP $\{(X_t, Y_t)\}$. Assume that P is positive but otherwise the assumptions specified in Theorems 2.6 and 2.8 may not hold. Let the row vector $\pi = (\pi_{(1,1)}, \ldots, \pi_{(M,N_M)})$ be the stationary probability distribution of P. As shown in Cao (1998), the potential g of the policy is found from the Poisson equation

$$(I - P + e\pi)g = f \tag{6}$$

and the policy improvement of the MDP is equivalent to looking for a policy $\mathscr{L}$ that gives the (componentwise) maximum of

$$\{P^{\mathscr{L}} g + f^{\mathscr{L}}\}. \tag{7}$$

Without loss of generality, we let $f^{\mathscr{L}} = f$ and drop the term in later discussion.

Consider a typical policy improvement step for $\{(X_t, Y_t)\}$. Let $g_i = (g_i(1), \ldots, g_i(N_i))^{\mathrm{T}} = (g_{(i,1)}, \ldots, g_{(i,N_i)})^{\mathrm{T}}$ be the potential vector of the base policy for settings in mode i, and $g = (g_1^{\mathrm{T}}, \ldots, g_M^{\mathrm{T}})^{\mathrm{T}} = (g_{(1,1)}, \ldots, g_{(M,N_M)})^{\mathrm{T}}$ be the (full) potential vector of the base policy. From (3) and (7), the policy improvement step of $\max\{P^{\mathscr{L}} g\}$ at state (i, j) gives

$$\max_{\alpha_{(i,j)} = (\alpha_i^u, \alpha_{(i,j)}^c, \alpha_{(i,j)}^p)} \left[\left(\sum_{m \neq i} r_{(i,m)}^{\alpha_i^u} \left(\sum_n q_{(i,j),(m,n)}^{\alpha_{(i,j)}^c} g_{(m,n)} \right) \right) + r_{(i,i)}^{\alpha_i^u} \left(\sum_n s_{(j,n)}^{\alpha_{(i,j)}^p} g_{(i,n)} \right) \right]. \tag{8}$$

Irrespective of the upper-level action, the optimal setting transition action of state (i, j) is

$$\alpha_{(i,j)}^{p,*} = \arg\max_{\alpha_{(i,j)}^p} \left(\sum_n s_{(j,n)}^{\alpha_{(i,j)}^p} g_{(i,n)} \right) \tag{9}$$

and the maximal action on the initial setting for a mode change is

$$\alpha^{c,*}_{(i,j)} = \arg\max_{\alpha^c_{(i,j)}} \left(\sum_n q^{\alpha^c_{(i,j)}}_{(i,j),(m,n)} g_{(m,n)} \right). \tag{10}$$

Let $\tilde{g}^c_{(i,m)}(j) = \sum_n q^{\alpha^{c,*}_{(i,j)}}_{(i,j),(m,n)} g_{(m,n)}$ and $\tilde{g}^p_i(j) = \sum_n s^{\alpha^{p,*}_{(i,j)}}_{(j,n)} g_{(i,n)}$, where the functional index j (between the pair of parentheses) indicates the policy improvement for setting j. Substituting $\tilde{g}^c_{(i,m)}(j)$ and $\tilde{g}^p_i(j)$ into (8), the policy iteration for the mode action becomes

$$\max_{\alpha^u_i} \left[\sum_{m \neq i} r^{\alpha^u_i}_{(i,m)} \tilde{g}^c_{(i,m)}(j) + r^{\alpha^u_i}_{(i,i)} \tilde{g}^p_i(j) \right]$$
for every $(i, j) \in \mathcal{X}_i$. (11)

From (11), a mode action α^u_i determines a mode change distribution $(r^{\alpha^u_i}_{(i,1)}, \ldots, r^{\alpha^u_i}_{(i,M)})$ that affects all states $(i, j) \in \mathcal{X}_i$, not only a single state (i, j). That is, the effect of the actions taken at different states $(i, j) \in \mathcal{X}_i$ are coupled. To be an executable policy, settings in the same mode need to adopt the same mode policy. However, in policy iteration of a standard MDP, the actions have to be chosen for each state (i, j) independently. In general, actions that maximize (8) for all states $(i, j) \in \mathcal{X}_i$ may not exist, i.e., the policy iteration approach does not work for general two-level MDPs. The same restriction is also applicable to other MDP solution methods.

3.2. The decoupling of the two-level MDP

To decouple the effect of actions α^u_i, $\alpha^c_{(i,j)}$, and $\alpha^p_{(i,j)}$ at different states $(i, j) \in \mathcal{X}_i$, we need the action $\alpha^{u,*}_i$ as found in (11) to be independent of j for all $(i, j) \in \mathcal{X}_i$. We will show that this is the case if Assumptions 2.4 and either of 2.5 or 2.7 hold. First, observe that if

$$\tilde{g}^c_{(i,m)}(j) = \tilde{g}^c_{(i,m)} \quad \text{for all } (i, j) \in \mathcal{X}_i \tag{12}$$

and

$$\tilde{g}^p_i(j) = \tilde{g}^p_i \quad \text{for all } (i, j) \in \mathcal{X}_i, \tag{13}$$

i.e., both $\tilde{g}^c_{(i,m)}(j)$ and $\tilde{g}^p_i(j)$ are independent of j, then every term in (11) is independent of the setting j. Therefore, there is an action $\alpha^{u,*}_i$ that maximizes (11) for all $(i, j) \in \mathcal{X}_i$.

However, (13) is a very strong assumption. It holds, for example, when settings in each mode are stochastically equivalent. Instead of having (13), suppose that

$$r^{\alpha^u_i}_{(i,i)} = \zeta_i \quad \text{for all } \alpha^u_i, \quad 1 \leqslant i \leqslant M. \tag{14}$$

When (12) and (14) hold, (11) becomes

$$\arg\max_{\alpha^u_i} \left[\sum_{m \neq i} r^{\alpha^u_i}_{(i,m)} \tilde{g}^c_{(i,m)} + \zeta_i \tilde{g}^p_i(j) \right]$$
$$= \arg\max_{\alpha^u_i} \left[\sum_{m \neq i} r^{\alpha^u_i}_{(i,m)} \tilde{g}^c_{(i,m)} \right] + \zeta_i \tilde{g}^p_i(j) \tag{15}$$

for every $(i, j) \in \mathcal{X}_i$. Again, $\alpha^{u,*}_i$ is independent of j.

Note that (14) is indeed Assumption 2.4. Next, Assumption 2.7 implies that for any m the set of all possible initial distributions $q^{\alpha^c_{(i,j)}}_{(i,j),(m,n)}$, $n = 1, \ldots, N_m$, corresponding to all different $\alpha^c_{(i,j)}$, is the same for all states $(i, j) \in \mathcal{X}_i$. Thus, the maximal value of (10), $\tilde{g}^c_{(i,m)}(j) = \sum_n q^{\alpha^{c,*}_{(i,j)}}_{(i,j),(m,n)} g_{(m,n)}$, is the same for all $(i, j) \in \mathcal{X}_i$. Therefore, Assumption 2.7 leads to (12). Of course, so does the more restrictive Assumption 2.5.

Indeed we have established Proposition 3.1 and from here onwards we only consider two-level MDPs with decoupled actions.

Proposition 3.1. *Suppose that Assumptions* 2.1–2.4 *and either Assumption* 2.5 *or* 2.7 *holds for the two-level MDP. Then the effect of mode and setting actions at different states* $(i, j) \in \mathcal{X}_i$ *are decoupled, and the problem can be solved by policy iteration. When the mode is preserved, the maximal setting action* $\alpha^{p,*}_{(i,j)}$ *is defined by* (9). *When there is a mode change, the maximal initial setting action* $\alpha^{c,*}_{(i,j)}$ *is defined by* (10), *no matter what the original mode and setting are. Settings in the same mode take the same maximal mode action* $\alpha^{u,*}_i$ *as defined in* (15).

Our set of assumptions is sufficient for the results. While the set is not necessary for specially designed two-level MDPs, in general, no policy-iteration type of solution exists for the two-level MDPs that violate some of the assumptions. For example, when $r^{\alpha^u_i}_{(i,i)}$ and hence the sojourn times depend on upper-level actions, (11) shows that settings may not have a common optimal upper-level action, and policy iteration is not applicable.

Finally, observe from (10) that for the optimal policy (or the improved policy in each iteration), the initial distribution after the mode changes to m, $q^{\alpha^c_{(i,j)}}_{(i,j),(m,n)}$, $m = 1, \ldots, N_m$, does not depend on j under Assumption 2.7, and not on both i and j under Assumption 2.5. Therefore, we consider only policies with this property. Denote $(q^{\alpha^c_{(i,j)}}_{(i,j),(m,1)}), \ldots, q^{\alpha^c_{(i,j)}}_{(i,j),(m,N_m)})$ as $\theta^{(m)}$ for Assumption 2.5 and $\theta^{(i,m)}$ for Assumption 2.7. Then in (3), under the two assumptions, $Q^{(i,m)} = Q^{(m)} = e^{\mathrm{T}}\theta^{(m)}$ and $Q^{(i,m)} = e^{\mathrm{T}}\theta^{(i,m)}$, respectively.

4. Policy iteration for two-level MDPs

In this section, we provide a policy iteration algorithm for a two-level MDP with decoupled actions. In the next section, we show that the computational effort can be reduced by decomposing the problem into the upper-level and a number of lower-level problems, all of them are of smaller sizes.

As the preparation for subsequent discussion, we introduce phase-type distributions discussed in Chapter 2 of Neuts (1981). Let N be a positive integer; B be an $N \times N$ non-negative matrix; B_0 be an N-dimensional non-negative column vector where the row sums of $[B, B_0]$ are equal to 1; $\beta = (\beta_1, \ldots, \beta_N)$ be an N-dimensional non-negative row vector; β_{N+1} be a non-negative number such that $\sum_{m=1}^{N+1} \beta_m = 1$; e be an N-dimensional column vector with all elements equal to 1.

Lemma 4.1. *Let L follow the discrete phase-type distribution defined by the absorption time of the $(N+1)$-state DTMC*

$$\begin{bmatrix} B & B_0 \\ 0 & 1 \end{bmatrix}$$

with state space $\{1,\ldots,N+1\}$ and the initial distribution (β, β_{N+1}); L_q be the number of visits to state q before absorption, $q=1,\ldots,N$. Then $E(L)=\beta(I-B)^{-1}e$ and $E(L_q)=\beta(I-B)^{-1}e_q$, with e_q being the qth column of the identity matrix, $q=1,\ldots,N$.

The *sojourn time* ξ of a mode of the two-level MDP (for a given policy) is the duration of a visit to the mode (for the given policy). Denote the length of a sojourn time ξ by $\lambda(\xi)$. Let $\xi_k^{(m)}$ be the kth sojourn time of mode m (in a generic sample path of the two-level MDP). Its length is $\lambda(\xi_k^{(m)})$. Refer to the sequence of sojourn times of mode m as the m-sojourn time. Let $t_k^{(m)}$ be the instant at which the kth m-sojourn time starts; i.e.,

$$t_1^{(m)} = \min\{t : t>0, X_t = m\},$$
$$t_{k+1}^{(m)} = \min\{t : t > t_k^{(m)} + \lambda(\xi_k^{(m)}), X_t = m\}.$$

The kth sojourn time of mode m is formed by the set of states

$$\{(X_{t_k^{(m)}}, Y_{t_k^{(m)}}), \ldots, (X_{t_k^{(m)}+\lambda(\xi_k^{(m)})-1}, Y_{t_k^{(m)}+\lambda(\xi_k^{(m)})-1})\},$$

and the total reward collected on $\xi_k^{(m)}$ is

$$\sum_{t=t_k^{(m)}}^{t_k^{(m)}+\lambda(\xi_k^{(m)})-1} f_{(X_t,Y_t)}.$$

By Assumption 2.4, each m-sojourn time $\xi^{(m)}$ follows the geometric distribution with the probability of success $1-\zeta_m$, no matter what the (mode) policy is. By construction, $\xi^{(m)}$ can also be expressed as the phase-type distribution with $B_0 = (1-r_{(m,m)})e$, $B = r_{(m,m)}S^{(m)}$, and $(\beta, \beta_{N+1}) = (\theta, 0)$, where θ is an N_m-dimensional probability row vector that denotes the initial setting distribution at mode m. From Lemma 4.1, $E(\xi^{(m)}) = \theta(I - r_{(m,m)}S^{(m)})^{-1}e$, with the expected total reward $\theta(I-r_{(m,m)}S^{(m)})^{-1}f_m$.

The form of θ depends on the assumption taken. Consider the case that mode i changes to mode m. As discussed in Section 3, we have $\theta=\theta^{(m)}$, $Q^{(i,m)}=Q^{(m)}=e^{\mathrm{T}}\theta^{(m)}$ if Assumption 2.5 holds; and $\theta=\theta^{(i,m)}$, $Q^{(i,m)}=e^{\mathrm{T}}\theta^{(i,m)}$ if Assumption 2.7 holds. Let $\rho=(\rho_1,\ldots,\rho_M)$ be the stationary distribution of R, i.e.,

$$\sum_{m=1}^{M}\rho_m = 1 \quad \text{and} \quad \rho = \rho R. \tag{16}$$

By Lemma 4.1 and Little's formula, we have the following results for later usage.

Proposition 4.2. *Let $\theta=\theta^{(m)}$ if Assumption 2.5 holds, $1\leqslant m\leqslant M$; and $\theta=\theta^{(i,m)}$ if Assumption 2.7 holds, $1\leqslant i\neq m\leqslant M$. For any policy, the stationary distribution of the transition probability matrix P of the policy is given by*

$$\pi_{(m,n)} = \sum_{i\neq m}\rho_i r_{(i,m)}\theta(I - r_{(m,m)}S^{(m)})^{-1}e_n,$$
$$1\leqslant n\leqslant N_m, \quad m=1,\ldots,M. \tag{17}$$

The long-run average reward from mode m,

$$v_m = \sum_{i\neq m}\rho_i r_{(i,m)}\theta(I - r_{(m,m)}S^{(m)})^{-1}f_m, \quad m=1,\ldots,M. \tag{18}$$

The long-run average reward of the system is

$$\eta = \sum_{m=1}^{M} v_m = \sum_{m=1}^{M}\sum_{i\neq m}\rho_i r_{(i,m)}\theta(I - r_{(m,m)}S^{(m)})^{-1}f_m \tag{19}$$

in general, and when $\theta=\theta^{(m)}$,

$$\eta = \sum_{m=1}^{M}\rho_m\theta^{(m)}(I - r_{(m,m)}S^{(m)})^{-1}f_m. \tag{20}$$

Algorithm 1 (*Policy iterations for the two-level MDP*).

1. Set $k=1$. Arbitrarily choose an initial policy $\mathscr{L}_1$, where the initial setting distribution takes the form $\theta^{(m)}$ for Assumption 2.5, and the form $\theta^{(i,m)}$ for Assumption 2.7.
2. At the kth iteration:
   (a) Find the transition probability matrix $P^{\mathscr{L}_k}$ of policy $\mathscr{L}_k$, and find its stationary distribution $\pi^{\mathscr{L}_k}$ from (17), where $\theta=\theta^{(m)}$ if Assumption 2.5 holds, and $\theta=\theta^{(i,m)}$ if Assumption 2.7 holds.
   (b) Solve the Poisson equation $(I-P^{\mathscr{L}_k}+e\pi^{\mathscr{L}_k})g^{\mathscr{L}_k}=f$ for the potential vector $g^{\mathscr{L}_k}$ of policy $\mathscr{L}_k$.
   (c) Take the (componentwise) argument maximum of $\max_{\mathscr{L}}\{P^{\mathscr{L}}g^{\mathscr{L}_k}\}$ to determine an improved policy $\mathscr{L}_{k+1}$, where the improved actions of each state are found from (9), (10), and (15). Whenever applicable, the action of a state in $\mathscr{L}_k$ should be kept for $\mathscr{L}_{k+1}$ if the action is maximal for the state in both $\mathscr{L}_k$ and $\mathscr{L}_{k+1}$.
3. Stop if $\mathscr{L}_k=\mathscr{L}_{k+1}$; otherwise set $k=k+1$ and return to step 2.

Algorithm 1 improves on every iteration. Given that the action set is finite, Algorithm 1 finds the optimal policy in finite number of iterations. Note that Algorithm 1 is computationally intensive. At each iteration, the solution of $g^{\mathscr{L}_k}$ in Step 2(b) takes the inverse of an $(\sum_{m=1}^{M}N_m)\times(\sum_{m=1}^{M}N_m)$ matrix. The computational effort required, to the first-order approximation, is about $O(\sum_{m=1}^{M}N_m)^3$.

5. The decomposition of two-level MDPs

5.1. The lower-level problem

Similar to $\xi^{(m)}$, we define an (i,m)-sojourn time $\xi^{(i,m)}$ as an m-sojourn time that is changed from mode i. Let $h_f(\xi^{(m)})$ and

$h_f(\xi^{(i,m)})$ be, respectively, the total reward from an m-sojourn time $\xi^{(m)}$ and an (i,m)-sojourn time $\xi^{(i,m)}$ for a performance function f. Setting $t=1$ as the beginning of the sojourn time ξ, we have $h_f(\xi)=\sum_{t=1}^{\lambda(\xi)} f_{(X_t,Y_t)}$. The expected total reward are

$$H_f(m)=E[h_f(\xi^{(m)})]=\theta^{(m)}(I-\zeta_m S^{(m)})^{-1}f_m \tag{21}$$

and

$$H_f(i,m)=E[h_f(\xi^{(i,m)})]=\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}f_m \tag{22}$$

for the m- and (i,m)-sojourn times, respectively, $1\leqslant i\neq m\leqslant M$. By Assumption 2.4, $H_f(m)<\infty$, $H_f(i,m)<\infty$, and the expected length of an m- or (i,m)-sojourn time is

$$H_1(m)=\frac{1}{1-\zeta_m},\quad m=1,\ldots,M. \tag{23}$$

From (19), (20) and (22), the long-run average reward for a given policy is

$$\eta=\sum_{m=1}^{M}(1-r_{(m,m)})\rho_m H_f(m) \text{ under Assumption 2.5} \tag{24}$$

and

$$\eta=\sum_{m=1}^{M}\sum_{i\neq m}\rho_i r_{(i,m)}H_f(i,m) \text{ under Assumption 2.7.} \tag{25}$$

In both cases, the lower-level policy is to maximize the total reward within a mode sojourn time, which can be done mode by mode.

In Section 5.1.1 below, we will illustrate the procedure to determine $\alpha^{c,*}_{(m,n)}$ and its corresponding $\alpha^{p,*}_{(m,n)}$ under Assumption 2.7. We consider (i,m)-sojourn time with $\theta=\theta^{(i,m)}$. There can be as many as $M(M-1)$ different mode transitions, potentially one for mode change from i to m, $1\leqslant i\neq m\leqslant M$.

When Assumption 2.5 holds instead, there will only be M lower-level MDPs, one for each mode. The procedure to determine $\alpha^{c,*}_{(m,n)}$ and $\alpha^{p,*}_{(m,n)}$ for an m-sojourn time follows a similar procedure by taking $\theta^{(i,m)}=\theta^{(m)}$ for all $1\leqslant i\leqslant M$. We skip the details for conciseness.

5.1.1. The lower-level chain

As discussed before, the optimal lower-level policy $\mathscr{L}^{l,*}=(\mathscr{L}^{c,*},\mathscr{L}^{p,*})$ is found from

$$\max_{\mathscr{L}^l}\{H_f(i,m)\}, \tag{26}$$

maximizing the expected total reward of an (i,m)-sojourn time. The problem can be transformed into an average-reward MDP.

Consider a fixed lower-level policy $\mathscr{L}^l$. Let $t_k^{(i,m)}$ be the period that the kth (i,m)-sojourn time occurs, and let $\{\tilde{Y}_\psi\}$ be the DTMC embedded on the (i,m)-sojourn times, i.e., in $\{\tilde{Y}_\psi\}$, only the reward and time contributed from (i,m)-sojourn times are recorded. $\{\tilde{Y}_\psi\}$ can be constructed in the following way:

$$\tilde{Y}_1=Y_{t_1^{(i,m)}},\quad \tilde{Y}_\psi=Y_{t_1^{(i,m)}+\psi-1}\quad \text{for } 1\leqslant\psi\leqslant\lambda(\xi_1^{(i,m)})$$

and in general,

$$\tilde{Y}_{(\sum_{q=1}^{k}\lambda(\xi_q^{(i,m)})+\psi)}=Y_{t_{k+1}^{(i,m)}+\psi-1}\quad \text{for } 1\leqslant\psi\leqslant\lambda(\xi_{k+1}^{(i,m)}). \tag{27}$$

$\{\tilde{Y}_\psi\}$ is the embedded chain formed by "cutting and pasting" together all the (i,m)-sojourn times on a sample path of the original chain. By construction, $\{\tilde{Y}_\psi\}$ is positive. For any given policy, the long-run average reward of $\{\tilde{Y}_\psi\}$ is,

$$\begin{aligned}\tilde{\eta}(i,m)&=\lim_{K\to\infty}\frac{\sum_{k=1}^{K}h_f(\xi_k^{(i,m)})}{\sum_{k=1}^{K}h_1(\xi_k^{(i,m)})}\\&=\lim_{K\to\infty}\frac{(1/K)\sum_{k=1}^{K}h_f(\xi_k^{(i,m)})}{(1/K)\sum_{k=1}^{K}h_1(\xi_k^{(i,m)})}=\frac{H_f(i,m)}{H_1(i,m)}\\&=(1-\zeta_m)H_f(i,m),\end{aligned} \tag{28}$$

where we use (23) to get the last equality. Comparing (26) and (28), the total-reward problem on (i,m)-sojourn times and the average-reward problem of $\{\tilde{Y}_\psi\}$ have the same optimal policy.

Let $\tilde{S}^{(i,m)}=[s^{(i,m)}_{(j,n)}]$ be the transition probability matrix of $\{\tilde{Y}_\psi\}$ for a fixed pair of policies $\mathscr{L}^c$ and $\mathscr{L}^p$. $\tilde{S}^{(i,m)}$ can be found from conditioning: Consider any period that the state is (m,j). If there is no mode change, the chain is in state (m,n) in the next period with probability $s^{\alpha_{(i,j)}}_{(j,n)}$; if there is a mode change, when the chain next visits mode m through a mode change from i, the initial distribution is determined by $q^{\alpha_{(i,j)}}_{(i,j),(m,n)}$ (cf. (2) and (3)). Then

$$\tilde{S}^{(i,m)}=\zeta_m S^{(m)}+(1-\zeta_m)Q^{(i,m)}. \tag{29}$$

The stationary distribution $\tilde{\pi}^{(i,m)}=\{\tilde{\pi}_1^{(i,m)},\ldots,\tilde{\pi}_{N_m}^{(i,m)}\}$ of $\tilde{S}^{(i,m)}$ can easily be found from the given $\theta^{(i,m)}$ and $S^{(m)}$. Define an alternate renewal process such that the system is "on" when the two-level MDP is in an (i,m)-sojourn time, and is "off" otherwise. A transition from mode i into mode m defines a renewal. Within the "on" time, the expected number of visits to state (m,n) is $\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}e_n$. From (23),

$$\begin{aligned}&\sum_{n=1}^{N_m}\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}e_n\\&\quad=\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}e=\frac{1}{1-\zeta_m}.\end{aligned}$$

By the Renewal Reward Theorem,

$$\begin{aligned}\tilde{\pi}_n^{(i,m)}&=\frac{\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}e_n}{\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}e}\\&=(1-\zeta_m)\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}e_n,\end{aligned}$$

i.e.,

$$\tilde{\pi}^{(i,m)} = (1-\zeta_m)\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1}. \tag{30}$$

Check that $\tilde{\pi}^{(i,m)}$ satisfies the balanced equation $\tilde{\pi}^{(i,m)}\tilde{S}^{(i,m)} = \tilde{\pi}^{(i,m)}$. The long-run average reward of $\{\tilde{Y}_\psi\}$ is

$$\tilde{\eta}^{(i,m)} = \tilde{\pi}^{(i,m)} f_m = (1-\zeta_m)\theta^{(i,m)}(I-\zeta_m S^{(m)})^{-1} f_m, \tag{31}$$

which is consistent with (28).

5.1.2. The lower-level algorithm

The lower-level problem for an (i,m)-sojourn time becomes the optimal control of $\{\tilde{Y}_\psi\}$ such that the policies $\mathscr{L}^c$ and $\mathscr{L}^p$ exert their effect together through $\tilde{S}^{(i,m)}$. Supposedly, rows of $Q^{(i,m)}$ should be the same, which means that states under the same mode take the same initial setting decision. Such coupling violates the procedure for policy iteration. Instead of taking $Q^{(i,m)} = e^{\mathrm{T}}\theta^{(i,m)}$, allow $Q^{(i,m)}$ in (29) to have different rows. By the nature of the problem, the optimal solution will come out naturally of the form of $Q^{(i,m)} = e^{\mathrm{T}}\theta^{(i,m)}$.

Let $\mathscr{L}^l_k = (\mathscr{L}^c_k, \mathscr{L}^p_k)$ be the lower-level policy found in the kth iteration of the lower-level algorithm, where $\mathscr{L}^c_k$ and $\mathscr{L}^p_k$ are $\mathscr{L}^c$ and $\mathscr{L}^p$ in the kth iteration, respectively.

Algorithm 2 (*Policy iterations for the lower level for an* (i,m)*-sojourn time*).

1. Set $k=1$. Arbitrarily choose an initial lower-level policy $\mathscr{L}^l_1 = (\mathscr{L}^c_1, \mathscr{L}^p_1)$, which specifies $\theta^{(i,m),\mathscr{L}^c_1}$ and $S^{(m),\mathscr{L}^p_1}$.
2. At the kth iteration:
   (a) Find $\tilde{S}^{(i,m),\mathscr{L}^l_k}$ from (29) and $\tilde{\pi}^{(i,m),\mathscr{L}^l_k}$ from (30).
   (b) Solve the Poisson equation $(I - \tilde{S}^{(i,m),\mathscr{L}^l_k} + e\tilde{\pi}^{(i,m),\mathscr{L}^l_k})\tilde{g}_m^{\mathscr{L}^l_k} = f_m$ for the potential vector $\tilde{g}_m^{\mathscr{L}^l_k} = (\tilde{g}_m^{\mathscr{L}^l_k}(1), \ldots, \tilde{g}_m^{\mathscr{L}^l_k}(N_i))^{\mathrm{T}}$ of the lower-level policy $\mathscr{L}^l_k$ for (i,m)-sojourn times.
   (c) Take the (componentwise) argument maximum of $\max_{\mathscr{L}^l}\{\tilde{S}^{(i,m),\mathscr{L}^l}\tilde{g}_i^{\mathscr{L}^l_k}\}$ to determine an improved lower-level policy $\mathscr{L}^l_{k+1}$. Whenever applicable, the action of a setting in $\mathscr{L}^l_k$ should be kept for $\mathscr{L}^l_{k+1}$ if the action is maximal for the setting in both $\mathscr{L}^l_k$ and $\mathscr{L}^l_{k+1}$.
3. Stop if $\mathscr{L}^l_k = \mathscr{L}^l_{k+1}$; otherwise set $k = k+1$ and return to Step 2.

Algorithm 2 improves on every iteration. With the finite action space, the algorithm will stop at a maximum lower-level policy after a finite number of iterations. Given that $\tilde{S}^{(i,m),\mathscr{L}}\tilde{g}_m^{\mathscr{L}^l_k} = \zeta_m S^{(m),\mathscr{L}}\tilde{g}_m^{\mathscr{L}^l_k} + (1-\zeta_m)Q^{\mathscr{L}}\tilde{g}_m^{\mathscr{L}^l_k}$, the actions $\alpha^c_{(m,j)}$ and $\alpha^p_{(m,j)}$ can be determined separately, and the improved policy does satisfy $Q^{(i,m)} = e^{\mathrm{T}}\theta^{(i,m)}$ (i.e., the initial distribution for different rows are the same). In this sense, allowing Q in (29) to have different rows only has conceptual meaning.

5.2. The upper-level problem

We apply the *time aggregation* approach to solve the upper-level problem. In time aggregation, a subset of states is chosen as the embedded states. The reward between two visits of the embedded states is "aggregated" together and is assigned to the leading embedded states. The approach was first proposed for performance gradients in Zhang and Ho (1991) and then extended to MDPs in Cao et al. (2002). For Cao et al. (2002), the embedded Markov chain for a given mode policy is determined by a fixed set of states, not by the *fixed type of transitions* corresponding to the changes of modes as ours. The expected total reward in a sojourn time of a mode is aggregated to a visit of the mode. The aggregated rewards of the sojourn times depend on the assumptions adopted, and their determination is described in the previous section for the lower level.

5.2.1. The upper-level time-aggregated chain

$\{X_t\}$ stays at the same mode during a sojourn time of the mode. The beginning of the sojourn times marks the transitions that the time aggregated chain bases on. Define

$$t_1 = 1 \quad \text{and} \quad t_{\varphi+1} = \min\{t : t_\varphi < t, X_t \neq X_{t_\varphi}\}.$$

Basically, $\{t_\varphi\}$ is the sequence of periods of new modes formed by sorting $\{t_k^{(m)}, m=1,\ldots,M; k=1,2,\ldots\}$ in ascending order. Define an embedded DTMC $\{\tilde{X}_\varphi\}$ of $\{X_t\}$ by

$$\tilde{X}_\varphi = X_{t_\varphi} \quad \text{for all } \varphi = 1,2,\ldots \tag{32}$$

$\{\tilde{X}_\varphi\}$ gives the successive new modes of the machine. Its transition probability matrix is

$$\tilde{R} = [\tilde{r}_{(i,m)}] \quad \text{where } \tilde{r}_{(i,m)} = \begin{cases} 0 & \text{if } i=m, \\ \dfrac{r_{(i,m)}}{1-\zeta_i}, & i \neq m. \end{cases} \tag{33}$$

Let σ be the rate of the occurrence of sojourn times.

$$\sigma = \sum_{m=1}^{M}(1-\zeta_m)\rho_m, \tag{34}$$

where $\rho = (\rho_1,\ldots,\rho_M)$, the stationary distribution of R, is found from (16). It is straightforward to show that the stationary distribution of $\tilde{R}$ is

$$\tilde{\rho} = (\tilde{\rho}_1,\ldots,\tilde{\rho}_M) \quad \text{where } \tilde{\rho}_i = \frac{(1-\zeta_i)}{\sigma}\rho_i, \quad i = 1,\ldots,M. \tag{35}$$

The subsequent development depends on the assumption taken. First consider Assumption 2.5. From (24), (33), and (35), the long-run average reward of $\{\tilde{X}_\varphi\}$ is

$$\sum_{i=1}^{M}\tilde{\rho}_i H_f(i) = \frac{\eta}{\sigma}. \tag{36}$$

From (23) and (35), the average length of a sojourn time is

$$\sum_{i=1}^{M}\tilde{\rho}_i H_1(i) = \frac{1}{\sigma}. \tag{37}$$

For $\varepsilon > 0$, re-define the reward of a visit of state m in the chain $\{\tilde{X}_\varphi\}$ (cf. Cao et al., 2002) to

$$\gamma_\varepsilon(m) = H_f(m) - \varepsilon H_1(m), \quad 1 \leqslant m \leqslant M,$$
$$\text{i.e., } \gamma_\varepsilon = H_f - \varepsilon H_1. \tag{38}$$

Let $\mathscr{L}^u$ be any upper-level policy. From (35)–(38), the long-run average reward of $\{\tilde{X}_\varphi\}$ is

$$\eta_\varepsilon^{\mathscr{L}^u} := \tilde{\rho}^{\mathscr{L}^u}\gamma_\varepsilon = \frac{\eta^{\mathscr{L}^u} - \varepsilon}{\sigma^{\mathscr{L}_u}}.$$

Note that ε is arbitrary. Suppose that $\varepsilon = \eta$, the long-run average reward of a given upper-level policy, the *base policy* $\mathscr{L}$ from which we want to improve upon. The performance function of $\{\tilde{X}_\varphi^{\mathscr{L}^u}\}$ for this particular case becomes

$$\gamma_\eta = H_f - \eta H_1 \tag{39}$$

under policy $\mathscr{L}^u$. Its long-run average reward is $\eta_\eta^{\mathscr{L}_u} = (\eta^{\mathscr{L}^u} - \eta)/\sigma^{\mathscr{L}^u}$, whose value reflects the difference in the performance of $\mathscr{L}_u$ and the based policy $\mathscr{L}$. Indeed, we have established

Proposition 5.1. *For any lower-level policy, let η be the long-run average reward of $\{(X_t, Y_t)\}$ of the (upper-level) base policy. Re-define the reward function for the embedded chain $\{\tilde{X}_\varphi\}$ to γ_η with $\varepsilon = \eta$ in* (39)*, where η is the long-run average reward of the base policy. Then for any upper-level policy $\mathscr{L}^u$, we have $\eta^{\mathscr{L}^u} > \eta$, or $\eta^{\mathscr{L}^u} = \eta$, or $\eta^{\mathscr{L}^u} < \eta$, if $\eta_\eta^{\mathscr{L}^u} > 0$, or $\eta_\eta^{\mathscr{L}^u} = 0$, or $\eta_\eta^{\mathscr{L}^u} < 0$, respectively. Furthermore, if we have $\eta_\eta^{\mathscr{L}^u} \leqslant 0$ for all $\mathscr{L}^u$, then $\mathscr{L}$ is the optimal upper-level policy for the given lower-level policy.*

When Assumption 2.7 holds, recall from (22) that $H_f(i, m)$ is the expected total reward for the (i, m) sojourn time. It follows from (25), (33), and (35) that $\sum_{i=1}^M \tilde{\rho}_i \sum_{m\neq i} \tilde{r}_{(i,m)} H_f(i, m) = \eta/\sigma$. Note that $\tilde{r}$ is policy dependent. Define a policy dependent performance function $\mathscr{H}_f$ such that $\mathscr{H}_f(i) = \sum_{m\neq i} \tilde{r}_{(i,m)} H_f(i, m)$, $1 \leqslant i \leqslant M$. Then $\sum_{i=1}^M \tilde{\rho}_i \mathscr{H}_f(i) = \eta/\sigma$. As an analogy of (38), re-define γ_ε under Assumption 2.7 as

$$\gamma_\varepsilon = \mathscr{H}_f - \varepsilon H_1. \tag{40}$$

Proposition 5.1 applies with the policy dependent performance function $\mathscr{H}_f$.

5.2.2. The upper-level algorithm given a lower-level policy

We can develop policy iteration algorithms based on Proposition 5.1. In the following algorithm, $\mathscr{L}_k^u$, the kth upper-level policy found by the algorithm, is considered as the base policy $\mathscr{L}$ in Proposition 5.1. For notational simplicity, let

$$\gamma_{\eta^k} = \begin{cases} H_f - \eta^{\mathscr{L}_k^u} H_1 & \text{under Assumption 2.5,} \\ \mathscr{H}_f - \eta^{\mathscr{L}_k^u} H_1 & \text{under Assumption 2.7,} \end{cases} \tag{41}$$

be the value of γ_ε when $\varepsilon = \eta^{\mathscr{L}_k^u}$, where γ_{η^k} is policy independent under Assumption 2.5 and is policy dependent under Assumption 2.7.

In the policy improvement step under Assumption 2.7, for the new policy $\mathscr{L}_{k+1}^u$, the performance function becomes $\gamma_{\eta^k}^{\mathscr{L}_{k+1}^u} = \mathscr{H}_f^{\mathscr{L}_{k+1}^u} - \eta^{\mathscr{L}_k^u} H_1$, where $\mathscr{H}_f^{\mathscr{L}_{k+1}^u}(i) = \sum_{m\neq i} \tilde{r}_{(i,m)}^{\mathscr{L}_{k+1}^u} H_f(i, m)$. Recall that $H_1(i)$, $H_f(i)$ under Assumption 2.5, and $H_f(i, m)$ under Assumption 2.7 have already been found from (23), (21), and (22), respectively.

Algorithm 3 (*Policy iterations for the upper level*).

1. Set $k = 1$. Arbitrarily choose an initial upper-level policy $\mathscr{L}_1^u$.
2. At the kth iteration:
   (a) Find $\rho^{\mathscr{L}_k^u}$ from (16), $\eta^{\mathscr{L}_k^u}$ from (24) if Assumption 2.5 holds, or from (25) if Assumption 2.7 holds, $\sigma^{\mathscr{L}_k^u}$ from (34), $\tilde{R}^{\mathscr{L}_k^u}$ from (33) and its stationary distribution $\tilde{\rho}^{\mathscr{L}_k^u}$ from (35), and γ_{η^k} from (41).
   (b) Solve the Poisson equation
   $$(I - \tilde{R}^{\mathscr{L}_k^u} + e\tilde{\rho}^{\mathscr{L}_k^u})\tilde{g}^{\mathscr{L}_k^u} = \gamma_{\eta^k} \tag{42}$$
   for the potential vector $\tilde{g}^{\mathscr{L}_k^u} = (\tilde{g}^{\mathscr{L}_k^u}(1), \ldots, \tilde{g}^{\mathscr{L}_k^u}(M))^{\mathrm{T}}$ of upper-level policy $\mathscr{L}_k^u$.
   (c) Take the (componentwise) argument maximum of
   $$\max_{\mathscr{L}^u}\{\tilde{R}^{\mathscr{L}^u}\tilde{g}^{\mathscr{L}_k^u}\} \quad \text{under Assumption 2.5,}$$
   $$\max_{\mathscr{L}^u}\{\tilde{R}^{\mathscr{L}^u}\tilde{g}^{\mathscr{L}_k^u} + \gamma_{\eta^k}^{\mathscr{L}_k^u}\} \quad \text{under Assumption 2.7,}$$
   to determine an improved upper-level policy $\mathscr{L}_{k+1}^u$. Whenever applicable, the action of a mode in $\mathscr{L}_k^u$ should be kept in $\mathscr{L}_{k+1}^u$ if the action is maximal for the mode in both $\mathscr{L}_k^u$ and $\mathscr{L}_{k+1}^u$.
3. Stop if $\mathscr{L}_k^u = \mathscr{L}_{k+1}^u$; otherwise set $k = k + 1$ and return to step 2.

By Proposition 5.1, each iteration of Algorithm 3 leads to an increase in the objective function. Given that the action space is finite, Algorithm 3 will stop at a maximal upper-level policy after a finite number of iterations.

5.3. The combined algorithm for the two-level model

Using the optimal lower-level policy $(\mathscr{L}^{c,*}, \mathscr{L}^{p,*})$ of Algorithm 2 as the input of Algorithm 3 to deduce the optimal upper-level policy $\mathscr{L}^{u,*}$, we have

Algorithm 4 (*The time-aggregated algorithm for the two-level model*).

1. Use Algorithm 2 to find the optimal lower-level policy $\mathscr{L}^{l,*} = (\mathscr{L}^{c,*}, \mathscr{L}^{p,*})$.
2. Given the optimal lower-level policy $\mathscr{L}^{l,*}$, use Algorithm 3 to find the optimal upper-level policy $\mathscr{L}^{u,*}$ and stop.

Given Assumption 2.4, the upper-level policies $\mathscr{L}^u$ play no effect on the expected total reward of any m-sojourn time.

Consider policy $\mathscr{L} = (\mathscr{L}^u, \mathscr{L}^l)$. From (24) and (25),

$$\eta^{\mathscr{L}} = \sum_{i=1}^{M} (1 - r^{\mathscr{L}^u_i}_{(i,i)}) \rho_i^{\mathscr{L}^u} H_f^{\mathscr{L}^l}(i)$$

or

$$\eta^{\mathscr{L}} = \sum_{i=1}^{M} \sum_{i \neq m} \rho_i^{\mathscr{L}^u} r^{\mathscr{L}^u_i}_{(i,m)} H_f^{\mathscr{L}^l}(i,m).$$

The optimal lower-level policy as found in Section 5.1 is in fact optimal for the original problem. Hence, Algorithm 4 finds the optimal policy of the two-level problem in a finite number of iterations. As we specialize to different sets of assumptions for the lower level, Assumptions 2.5 and 2.7, we get Theorems 2.6 and 2.8, respectively.

In all algorithms, step 2(b) takes most computational effort. For Algorithm 1, it is of order $O((\sum_{m=1}^{M} N_m)^3)$. For Algorithm 4, under Assumption 2.5, the computation is of order $O(M^3) + \sum_{m=1}^{M} O(N_m^3)$, and under Assumption 2.7, the computation is of order $O(M^3) + \sum_{m=1}^{M} (M-1) O(N_m^3)$. In typical real-life applications, $M \ll \sum_{m=1}^{M} N_m$, which makes Algorithm 4 attractive. For example, when $M = 10$ and $N_m = 1000$ for $1 \leqslant m \leqslant M$, the computational effort of Algorithm 4 is around one-tenth of Algorithm 1 even under Assumption 2.7.

Policy iteration can still be applied when R, $Q^{(m)}$, and $S^{(m)}$ are unknown. See Cao (1999, 1998); Cao and Chen (1997); Cao et al. (1996), and Cao and Wan (1998) for the single-sample path-based approach, particularly Cao et al. (2002) for the approach on a time-aggregated chain.

5.4. A numerical example

Consider a two-level MDP with $M = 3$, $N_1 = 3$, $N_2 = 4$, $N_3 = 2$, $\mathscr{X} = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (2,4), (3,1), (3,2)\}$ and performance function $f = (10, 5, 6, 4, 8, 7, 3, 10, 2)^{\mathrm{T}}$. Using Roman numerals I, II, etc., to label actions, the mode transition probabilities in (1) are

$$[r^{\mathrm{I}}_{(i,m)}] = \begin{bmatrix} 0.99 & 0.01 & 0 \\ 0.002 & 0.99 & 0.008 \\ 0.007 & 0.003 & 0.99 \end{bmatrix},$$

$$[r^{\mathrm{II}}_{(i,m)}] = \begin{bmatrix} 0.99 & 0.005 & 0.005 \\ 0.005 & 0.99 & 0.005 \\ 0.005 & 0.005 & 0.99 \end{bmatrix},$$

$$[r^{\mathrm{III}}_{(i,m)}] = \begin{bmatrix} 0.99 & 0 & 0.01 \\ 0.007 & 0.99 & 0.003 \\ 0.004 & 0.006 & 0.99 \end{bmatrix},$$

where modes can take actions combined from rows of $[r^{\mathrm{I}}_{(i,m)}]$, $[r^{\mathrm{II}}_{(i,m)}]$, and $[r^{\mathrm{III}}_{(i,m)}]$, an interpretation that holds for all action matrices listed below.

When a mode is preserved, there are 2, 3, and 4 actions to determine the setting for states in modes 1, 2, and 3, respectively. For mode 1, the setting transition probability matrices for the two actions are

$$[s^{\mathrm{I}}_{(1,j),(1,n)}] = \begin{bmatrix} 0 & 0.6 & 0.4 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

$$[s^{\mathrm{II}}_{(1,j),(1,n)}] = \begin{bmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 0.3 & 0.7 & 0 \end{bmatrix}.$$

For mode 2, the setting transition probability matrices for the three actions are

$$[s^{\mathrm{I}}_{(2,j),(2,n)}] = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 \end{bmatrix},$$

$$[s^{\mathrm{II}}_{(2,j),(2,n)}] = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix},$$

$$[s^{\mathrm{III}}_{(2,j),(2,n)}] = \begin{bmatrix} 0 & 0.4 & 0.3 & 0.3 \\ 0.3 & 0 & 0.2 & 0.5 \\ 0.1 & 0 & 0.2 & 0.7 \\ 0 & 0.7 & 0.3 & 0 \end{bmatrix}.$$

For mode 3, the setting transition probability matrices for the four actions are

$$[s^{\mathrm{I}}_{(3,j),(3,n)}] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad [s^{\mathrm{II}}_{(3,j),(3,n)}] = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix},$$

$$[s^{\mathrm{III}}_{(3,j),(3,n)}] = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}, \quad [s^{\mathrm{IV}}_{(3,j),(3,n)}] = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}.$$

The initial distribution of settings of new mode 1 are

$$[\theta^{(1),\mathrm{I}}] = [0.7 \;\; 0.2 \;\; 0.1], \quad [\theta^{(1),\mathrm{II}}] = [0.25 \;\; 0.5 \;\; 0.25],$$
$$[\theta^{(1),\mathrm{III}}] = [0.25 \;\; 0.25 \;\; 0.5].$$

The initial distribution of settings of new mode 2 are

$$[\theta^{(2),\mathrm{I}}] = [0.25 \;\; 0.25 \;\; 0.25 \;\; 0.25],$$
$$[\theta^{(2),\mathrm{II}}] = [0.4 \;\; 0.2 \;\; 0.2 \;\; 0.2],$$
$$[\theta^{(2),\mathrm{III}}] = [0.2 \;\; 0.2 \;\; 0.2 \;\; 0.4].$$

The initial distribution of settings of new mode 3 are

$$[\theta^{(3),\mathrm{I}}] = [0.5 \;\; 0.5], \quad [\theta^{(3),\mathrm{II}}] = [0.8 \;\; 0.2],$$
$$[\theta^{(3),\mathrm{III}}] = [0.2 \;\; 0.8].$$

For the two-level MDP, the given set of transition probabilities satisfies Assumptions 2.1 and 2.3. With $\zeta_i = 0.99$ for all modes, its actions are decoupled. As innocent as the prior numbers, they can form $(3^3)(3^3)(2^3)(3^4)(4^2) = 7,558,272$ different policies.

Applying the lower-level algorithm to the three modes, the optimal initial setting distribution action $\alpha^{c,*}_{(i,j)} = \theta^{(i),*}$, the optimal setting transition actions $\alpha^{p,*}_{(i,j)}$ of states, and their corresponding total expected return in a sojourn time $H_f^*(i)$ are shown in Table 1. The optimal lower-level transition probabilities $S^{\mathscr{L}^{p,*}(i)}$ can be constructed from $\alpha^{p,*}_{(i,j)}$ of mode i. As for the upper level, the optimal actions of modes 1, 2, and 3 are actions III, I, and I, respectively.

Table 1
The optimal initial setting distributions, optimal actions, and $H_f^*(i)$

Mode i	$\theta^{(i),*}$	$(\alpha_{(i,j)}^{p,*})$	$H_f^*(i)$
1	(0.7, 0.2, 0.1)	(I, II, I)	748.3274
2	(0.25, 0.25, 0.25, 0.25)	(I, II, II, III)	619.5318
3	(0.8, 0.2)	(IV, I)	926.4786

When Assumption 2.7 holds instead, initial setting distributions may depend on original as well as new modes. See Wan and Cao (2005) for an expanded example with this feature.

6. Conclusion

In this paper, we show that for a two-level MDP, if the sojourn time of each mode is uncontrollable and the sets of the initial setting distributions after a mode change are independent of the settings before the mode change, the effect of the actions at different states can be decoupled and the problem can be solved with policy iteration accordingly. Furthermore, the upper-level MDP is solved by the time-aggregated approach, and the lower-level MDP for each mode is solved as a total-cost MDP with an embedded chain. The approach allows distributive implementation of the centralized control, and it saves computational effort compared with the standard policy iteration.

When the assumptions hold, our solution approach works so long as the distribution on settings of a new mode is independent of the settings of the old mode. Thus, our approach can be extended to cases with set up costs for new modes or with action dependent costs that are independent of settings before any mode change. See the extension in the conclusion of Wan and Cao (2005) for examples on each of these two cases.

Acknowledgements

We thank three anonymous referees and the Associate Editor for their constructive comments that help improve the content of the paper.

References

Abbad, M., Filar, J. A., & Bielecki, T. R. (1992). Algorithms for singularly perturbed limiting average control problem. *IEEE Transactions on Automatic Control*, *37*(9), 1421–1425.

Bielecki, T. R., & Filar, J. A. (1991). Singularly perturbed Markov control problems. *Annals of Operations Research*, *29*, 153–168.

Cao, X.-R. (1998). The relation among potentials, perturbation analysis, and Markov decision processes. *Journal of Discrete Event Dynamic Systems*, *8*(1), 71–87.

Cao, X.-R. (1999). Single sample path based optimization of Markov chains. *Journal of Optimization: Theory and Application*, *100*(3), 527–548.

Cao, X.-R., & Chen, H.-F. (1997). Perturbation realization, potentials, and sensitivity analysis of Markov processes. *IEEE Transactions on Automatic Control*, *42*(10), 1382–1393.

Cao, X.-R., Ren, Z. Y., Bhatnagar, S., Fu, M., & Marcus, S. (2002). A time aggregation approach to Markov decision processes. *Automatica*, *38*(6), 929–943.

Cao, X.-R., & Wan, Y.-w. (1998). Algorithms for sensitivity analysis of Markov systems through potentials and perturbation realization. *IEEE Transactions on Control Systems Technology*, *6*(4), 482–494.

Cao, X.-R., Yuan, X. M., & Qiu, L. (1996). A single sample path-based performance sensitivity formula for Markov chains. *IEEE Transactions on Automatic Control*, *41*(12), 1814–1817.

Chang, H. S., Fard, P. J., Marcus, S. I., & Shayman, M. (2003). Multitime scale Markov decision processes. *IEEE Transactions on Automatic Control*, *48*(6), 976–987.

Filar, J. A., Gaitsgory, V., & Haurie, A. B. (2001). Control of singularly perturbed hybrid stochastic systems. *IEEE Transactions on Automatic Control*, *46*(2), 179–190.

Filar, J. A., & Haurie, A. (2001). A two-factor stochastic production model with two time scales. *Automatica*, *37*(10), 1505–1513.

Neuts, M. F. (1981). *Matrix-geometric solution in stochastic models—an algorithmic approach*. Baltimore: Johns Hopkins University Press.

Wan, Y.-w., & Cao, X.-R. (2005). The control of a two-level Markov decision process by time aggegration. *Technical Report*, Institute of Global Operations Strategy and Logistics Management, National Dong Hwa University, Taiwan.

Zhang, B., & Ho, Y. C. (1991). Performance gradient estimation for very large finite Markov chains. *IEEE Transactions on Automatic Control*, *36*(10), 1218–1227.

Yat-wah Wan received the B.S. degree in Mechanical Engineering from the University of Hong Kong, M.S. degree in Industrial Engineering from the Texas A & M University, and the Ph.D. degree in Operations Research from the University of California, Berkeley. From August 1991 to December 1993, he served in the Department of Manufacturing Engineering, City Polytechnics of Hong Kong, and from December 1993 to July 2004 in the Department of Industrial Engineering and Engineering Management, Hong Kong University of Science and Technology. He is currently an Associate Professor in the Institute of Global Operations Strategy and Logistics Management, National Dong Hwa University, where he joined in August 2004. His research interests include the control and optimization of stochastic systems, transportation, and logistics.

Xi-Ren Cao received the M.S. and Ph.D. degrees from Harvard University, in 1981 and 1984, respectively, where he was a research fellow from 1984 to 1986. He then worked as a principal and consultant engineer/engineering manager at Digital Equipment Corporation, U.S.A, until October 1993. Since then, he is a Professor of the Hong Kong University of Science and Technology (HKUST), Hong Kong, China. He is the director of the Center for Networking at HKUST. He held visiting positions at Harvard University, University of Massachusetts at Amherst, AT&T Labs, University of Maryland at College Park, University of Notre Dame, Tsinghua University, University of Science and Technology of China, and other universities.

Dr. Cao owns three patents in data- and tele-communications and published two books in the area of discrete event dynamic systems. He received the Outstanding Transactions Paper Award from the IEEE Control System Society in 1987 and the Outstanding Publication Award from the Institution of Management Science in 1990. He is a Fellow of IEEE, Chairman of IEEE Fellow Evaluation Committee of IEEE Control System Society, Associate Editor at Large of *IEEE Transactions of Automatic Control,* Editor-in-Chief of *Discrete Event Dynamic Systems: Theory and Applications,* and he is/was on Board of Governors of IEEE Control Systems Society, associate editor of a number of international journals and chairman of a few technical committees of international professional societies. His current research areas include discrete event dynamic systems, stochastic learning and optimisation, performance analysis of communication systems, and signal processing.