A Model for Multi-time Scaled Sequential Decision Making Processes

Hyeong Soo Chang, Pedram Fard, Steven I. Marcus, and Mark A. Shayman
Electrical and Computer Engineering Department and Institute for Systems Research
University of Maryland, College Park, MD 20742
E-mail: {hyeong,pedram,marcus,shayman}@isr.umd.edu

Abstract

We propose a simple analytical model called M time-scale Markov Decision Process (MMDP) for hierarchically structured sequential decision making processes, where decisions in each level in the M-level hierarchy are made in M different time-scales. In this model, the state space and the action space of each level in the hierarchy are non-overlapping with those of the other levels, respectively, and the hierarchy is structured in a "pyramid" sense such that a decision made at level m (slower time-scale) state and/or the state will affect the evolutionary decision making process of the lower level $m+1$ (faster time-scale) until a new decision is made at the higher level but the lower level decisions themselves do not affect the higher level's transition dynamics. The performance produced by the lower level's decisions will affect the higher level's decisions. A hierarchical objective function is defined such that the finite-horizon value of following a (nonstationary) policy at the level $m+1$ over a decision epoch of the level m plus an immediate reward at the level m is the single step reward for the level m decision making process. From this we define "multi-level optimal value function" and derive "multi-level optimality equation". We then give some example control problems that can be modeled as MMDPs.

1 Introduction

Hierarchically structured control problems have been studied extensively in many contexts in various areas over the past years with certain types of models and assumptions. This is because many large-scale control problems that arise in real applications show multi-dimensionally interdependent organized behavior among subsystems that constitute the whole system as decision blocks. Two distinguished hierarchical structures studied in the literature are "multi-level structure", where decision making algorithms in different levels operate in different time scales (see, e.g., [11]) and "multi-layer structure", where algorithms are divided "spatially" and operate at the same time scale (see, e.g., [6]).

This paper focuses on the control problems with a particular multi-level structure — *hierarchically structured sequential decision making processes*, where decisions in each level in the hierarchy are made in *different* time-scales and the hierarchy is structured in a pyramid (bottom-up organization) sense. That is, decisions made in the higher level affect the decision making process of the lower level but the lower level decisions do not affect the higher level (state transition) dynamics even though the performance produced by the lower level decisions will affect the decisions that will be made by the higher level.

A usual approach to the multi-level structured problems is that a slow time-scale subsystem lays aside the details of a fast time-scale dynamics by "average" behavior and then solves its own optimization problem (see, e.g., [7] or chapter 11 in [14]). This approach makes sense especially when the hierarchy in the system is structured in the pyramid sense. This pyramid-like structure was used in the perspective of "performability" and "dependability" in Trivedi et al.'s model [8] even though controls are not involved in the models.

In this paper, we propose a simple analytical model that generalizes Trivedi et al.'s hierarchical model by incorporating controls into the model, which we refer to as Multi-time scale Markov Decision Process (MMDP). The model describes interactions between levels in a hierarchy in the pyramid sense. Each level is associated with distinct state and action spaces. That is, we assume that no two level share any state or control action. The upper level state and/or control induces the lower level MDP dynamics over a finite horizon of length corresponding to the decision epoch of the upper level. In other words, a particular pair of the upper level state and/or control will determine the lower level's state transition function and reward function. Hierarchical objective functions are defined such that the (quasi-steady state) performance measure, the finite horizon value of following a given lower level policy, obtained from the lower level over the decision epoch of the upper level will affect the upper level decision making. From this we define "multi-level value function" and then drive "multi-level optimality equation"

This research was supported in part by the National Science Foundation under Grant DMI-9988867, by the Air Force Office of Scientific Research under Grant F496200110161, and by the Department of Defense under Contract MDA 90499C2521.

for an infinite horizon discounted reward case. Due to the space constraint, we focus on the discounted objective criterion and omit the detailed analysis and study of exact and approximate solution techniques suited for solving MMDPs. Please see [4] for the further detailed studies.

In addition to inherently existing hierarchical and multi-time scale control structure in problems themselves that arise in many different contexts, our model is motivated by in particular the observation made in the networking literature recently. The network traffic shows fluctuations on multiple time-scales — scale invariant burstiness (see, e.g., [19]), and this characteristic in the network traffic has been well-studied by "long-range dependent" or "self-similar" model. However, there are several recent works that investigated the effects of such multi-time scaled behavior by certain relevant Markovian models that approximate the fluctuations in the traffic (see, e.g., [15] [17] [12] and references therein). The usual interests are in calculations of buffer overflow probability distribution but are not concerned with development of analytical multi-time scaled controls that incorporate given traffic models for such behaviors of the network traffic even though some non-Markovian model based approaches are available (see, e.g., [18] and [9]). For example, the slow time-scale ("call-level") relates to the arrival and departure process of video/voice calls and the fast time-scale ("packet-level") relates to the packet arrival process of calls during their "lifetimes". This different time-scaled dynamics causes fluctuations in the traffic at different time-scales and gives rise to a multi-time scaled queueing control problem and we believe that we need to develop an analytical model to approach this kind of control problems.

This paper is organized as follows. We present a formal description of MMDPs and characterize optimal solutions for MMDPs in Section 2 and briefly discuss how to compute the optimal solutions. We then discuss relevant related works of hierarchical models with our model in Section 3. We give three example problems for MMDPs in Section 4 and conclude the present paper in Section 5.

2 Multi-time Scale MDP

We present the two time-scale MDP model for simplicity. The M time-scale model with $M > 2$ can be extended from the two time-scale model without difficulty (please see [4]).

2.1 Model description

The upper level (slow time-scale) MDP has finite state space I and finite action space Λ. At each (discrete) decision time $n \in \{0, 1, 2, ...,\}$ and at state $i_n \in I$, an action $\lambda_n \in \Lambda$ is taken and i_n makes transition to state $i_{n+1} \in I$ according to probability $P^u(i_{n+1}|i_n, \lambda_n)$. Depending on which action has been taken at which state in the upper level MDP, the lower

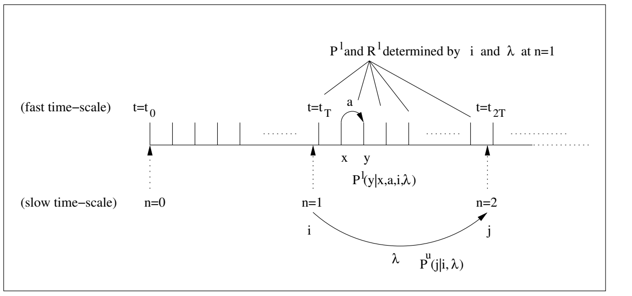

Figure 1: Graphical illustration of time evolution in the two time-scale MDPs

level (fast time-scale) MDP over one-step slow time-scale period is determined accordingly (what we mean by this will be clearer below). Every MDP in the lower level shares the same state and action space. We denote the finite state space and the finite action space by X and A, respectively. We assume that $I \cap X = \emptyset$ and $A \cap \Lambda = \emptyset$, which means that the state and control spaces of the upper level and the lower level are distinct or non-overlapping. We also assume that every control action is admissible at each state in each level for simplicity.

We denote time in the fast time-scale as $t \in \{t_0, t_1, t_2, ...\}$ and $t_{nT} = n$, $n = 0, 1, ...$ and T is a fixed finite scale factor between slow and fast time-scales. We implicitly assume that $t_{nT} = n^+$. That is, there is an infinitesimal gap between t_{nT} and n such that a fast time-scale decision at time t_{nT} is made slightly after a slow time-scale decision at time n has been made.

Let the initial state in the lower level MDP $x \in X$ and the initial state in the upper level MDP $i \in I$ ($x_{t_0} = x$ and $i_0 = i$ at $n = 0$). An action $\lambda_0 \in \Lambda$ will be taken at i_0 and the next upper level state i_1 will be determined stochastically by P^u. Over the time steps of $t_0, t_1, ..., t_{T-1}$, the system follows the lower level MDP evolution. That is, at the state x at t_0, an action $a \in A$ is taken and x makes transition to the next state $y \in X$, which is the state at time t_1, according to probability $P^l(y|x, a, i, \lambda)$ and the bounded reward of $R^l(x, a, i, \lambda)$ is incurred and this process is repeated at the state y at t_1, and so forth until the time t_{T-1}. That is, the state transition function and the reward function in the lower level MDP (over T-epoch) are *induced* by the upper level state and decision. At time $n = 1$, an upper level action λ_1 will be taken at i_1 (this will trigger a new MDP determination) and starting with a state z at t_T (determined stochastically from $P^l(z|x_{t_{T-1}}, a_{t_{T-1}}, i_0, \lambda_0)$ *for now* — we will consider a distribution over X called δ-initialization function later as a method of determining the states of x_{nT} for all n), the newly determined lower level MDP evolves (over the next T-epoch). See Figure 1 for graphical illustration of time evolution in this process.

Throughout this paper, we will use the term "decision rule" related with infinite horizon and the term "policy" related with finite horizon. Define a lower level *decision rule* $d^l = \{\pi_n^l\}$, $n = 0, 1, ...$, as a sequence of T-horizon *nonstationary policies* defined such that for all n, $\pi_n^l = \{\phi_{t_{nT}}, ..., \phi_{t_{(n+1)T-1}}\}$ is a sequence of functions where for all $k \geq 0$, $\phi_{t_k} : X \times I \times \Lambda \to A$. We will say that a lower level decision rule is *stationary with respect to the slow time-scale* n if $\pi_n^l = \pi_{n'}^l$ for all n, n' and *we will restrict ourselves to only this class of decision rules here*. We will denote the set of all possible such stationary decision rules with respect to the slow time-scale as $\mathcal{D}^l$, and omit the subscript n in π_n^l in this case and use the time $t_0, ..., t_{T-1}$ to refer the sequence of functions of π^l if necessary, and denote Π^l as the set of all possible such T-horizon nonstationary policies π^l. We will also omit the subscript on ϕ if π^l is stationary (with respect to the fast time-scale).

Given a lower level decision rule $d^l \in \mathcal{D}^l$ and a bounded immediate reward function $\mathcal{I}^u$ defined over $I \times \Lambda$ for the upper level, we define a function R^u such that for all $n \geq 0$, for $x \in X$, $i_n \in I$ and $\lambda_n \in \Lambda$,

$$R^u(x, i_n, \lambda_n, \pi^l) = \mathcal{I}^u(i_n, \lambda_n) + E^x_{i_n,\lambda_n}\left\{\sum_{t=t_{nT}}^{t_{(n+1)T-1}} \alpha^{\sigma(t)} R^l(x_t, \phi_t(x_t, i_n, \lambda_n), i_n, \lambda_n)\right\},$$

where $0 < \alpha \leq 1$ and $\sigma(t_{nT+r}) = r$ for all n with $r = 0, 1, ..., T-1$, and the superscript x on E signifies the initial state, $x_{t_{nT}} = x$, and the subscript i_n, λ_n on E signifies that i_n and λ_n for the expectation are fixed. The function R^u is simply the T-horizon total expected (discounted) reward of following the T-horizon nonstationary policy π^l given $i_n \in I$ and $\lambda_n \in \Lambda$ starting with state $x \in X$ with the zero terminal reward function[1] plus an immediate reward of taking an action λ_n at the state i_n at the upper level, and is a bounded function.

The total expected (discounted) reward achieved by the lower level T-horizon nonstationary policy π^l with an immediate reward at the upper level will act as a *single-step reward* for the upper level MDP. Define the upper level *stationary* decision rule d^u as a function $d^u : X \times I \to \Lambda$ and we denote $\mathcal{D}^u$ as the set of all possible such stationary decision rules. Given the initial $x \in X$ and $i \in I$, our goal is to obtain a decision rule pair of $d^l \in \mathcal{D}^l$ and $d^u \in \mathcal{D}^u$ that achieves *the two-level optimal value*, $V^*(x, i)$ defined as:

$$\max_{d^u \in \mathcal{D}^u} \max_{d^l \in \mathcal{D}^l} E^{x,i}\left\{\sum_{n=0}^{\infty} \gamma^n R^u(x_{t_{nT}}, i_n, d^u(x_{t_{nT}}, i_n), \pi^l)\right\},$$

where $0 < \gamma < 1$.

We can see that from the definition of the upper level decision rule, the decisions to be made at the upper level must depend on the lower level state, which is the initial state for the lower level MDP evolution over T-horizon in the fast time-scale. The initial state $x_{t_{nT}}, n = 1, 2, ...$ is determined stochastically by following the policy π^l. We will consider more general case of determining the initial state in the later subsection to expand the flexibility of our model. We also remark that even though we added the immediate reward function $\mathcal{I}^u$ in the definition of R^u to make our model description more natural, the function R^l can "absorb" the function $\mathcal{I}^u$ by newly defining the function R^l itself as $R^l(x, i, \lambda, \pi^l) + \frac{1}{T}\mathcal{I}^u(i, \lambda)$ for $\alpha = 1$ and $R^l(x, i, \lambda, \pi^l) + \left(\frac{1-\alpha}{1-\alpha^T}\right)\mathcal{I}^u(i, \lambda)$ for $0 < \alpha < 1$.

2.2 Multi-level optimality equation

Because the upper level sequential dynamics is essentially just an MDP with a reward function defined via the lower level MDP dynamics (by fixing a lower level decision rule), with a simple adaptation of the standard MDP theory (see, e.g, [1]), the following results hold for MMDPs.

We first define a set of all possible T-horizon (lower level) nonstationary policies under a fixed pair of an upper level state and an upper level action. For a given pair of $i \in I$ and $\lambda \in \Lambda$,

$$\Pi^l[i, \lambda] := \left\{\pi^l[i, \lambda] \;\middle|\; \pi^l[i, \lambda] := \{\phi_{t_0}^{i,\lambda}, ..., \phi_{t_{T-1}}^{i,\lambda}\}\right\},$$

where $\phi_{t_k}^{i,\lambda} : X \times \{i\} \times \{\lambda\} \to A, k = 0, ..., T-1$, and let $\mathcal{P}^T_{xy}(\pi^l[i, \lambda])$ the probability that state $y \in X$ is reached by T-steps starting with x by following the T-horizon nonstationary policy $\pi^l[i, \lambda]$, where the probability can be obtained from P^l.

Theorem 2.1 *For all $x \in X$ and $i \in I$,*

$$V^*(x, i) = \max_{\lambda \in \Lambda}\Big(\max_{\pi^l[i,\lambda] \in \Pi^l[i,\lambda]}\Big\{R^u(x, i, \lambda, \pi^l[i, \lambda]) + \gamma \sum_{y \in X}\sum_{j \in I} \mathcal{P}^T_{xy}(\pi^l[i, \lambda]) P^u(j|i, \lambda) V^*(y, j)\Big\}\Big)$$

and V^* is the unique solution to the above equation. Furthermore, for each pair of x and i, let the arguments that achieve the r.h.s of this equation as λ^* and $\pi^*[i, \lambda] = \{\phi^*_{t_k}\}$, and set $d^u(x, i) = \lambda^*$ for d^u and set π^l such that $\phi_{t_k}(x, i, \lambda^*) = \phi^*_{t_k}(x, i, \lambda^*)$ for d^l. The pair of d^u and d^l achieves V^*.

Proof: We define an MDP that operates in the slow time-scale n as follows. The state at time n is a pair of the lower level state and the upper level state, $(x_{t_{nT}}, i_n)$. An action at state $(x_{t_{nT}}, i_n)$ is a composite control of $\lambda_n \in \Lambda$ and $\pi^l[i_n, \lambda_n] \in \Pi^l[i_n, \lambda_n]$ (from our assumption that $t_{nT} = n^+$, $\pi^l[i_n, \lambda_n]$ will be taken slightly after λ_n is taken). Observe that we can view π^l as one-step action at the slow time-scale. More precisely, the admissible action set for state $(x_{t_{nT}}, i_n)$ is defined as the set given by

$$\{(\lambda, \tau) | \lambda \in \Lambda, \tau \in \Pi^l[i_n, \lambda]\}$$

[1] It is our assumption that the initial state for the next epoch in the slow time-scale does not contribute the reward for the previous epoch. However, a terminal reward can be defined by a function over X, in which case we need to add the terminal reward term in R^u.

The transition probability from $(x_{t_{nT}}, i_n)$ to $(x_{t_{(n+1)T}}, i_{n+1})$ is determined directly from $\mathcal{P}^T$ and P^u. Then, from the standard MDP theory, for this MDP, we can write Bellman's optimality equation and an optimal decision rule that achieves the unique optimal value at each state is derived, from which we can conclude our result. ■

Even though we assumed finite state spaces with finite action spaces, the issue of infinite/finite state/action space and bounded/unbounded reward function can be discussed from the well-known MDP theory (see, e.g., [1]).

2.3 Initialization function

So far we considered the case where $x_{t_{nT}}, n = 1, 2, ...$ is determined by the T-horizon nonstationary policy. In the model we described before, $x_{t_{nT}}, n = 1, 2, ...$ is (stochastically) determined by following the policy π^l with given x_{t_0}. Considering more general model, we first define an *initialization function* δ. We then determine $x_{t_{nT}}, n = 1, 2, ...$ by the function δ. That is, the lower level MDP initial state for each new T-period (in the fast time-scale) is initialized by the function δ. This is motivated by problem specific nature — organizing behavior in a hierarchy.

As in the previous description of the model, δ can be a function defined over $X \times I \times \Lambda$ such that for given x, i, λ, $\delta(x, i, \lambda)$ is a probability distribution over X. Given $x \in X$, $i \in I$ and $\lambda \in \Lambda$, we will use the notation of $\delta(x, i, \lambda)[y]$ to denote the probability defined on $y \in X$ by $\delta(x, i, \lambda)$. In the previous model description, $\delta(x, i, \lambda)[y]$ corresponds to $\mathcal{P}^T_{xy}(\pi^l[i, \lambda])$.

For some cases, the slow time-scale decisions (e.g., "reset" control, etc.) only will affect the new initial lower level state. In this case, δ is defined over $X \times \Lambda$ such that $\delta(x, \lambda)$ gives a probability distribution over X. The very idea of this δ is parallel to the transition structure in Markovian slowscale model given in [10]. Furthermore, determination of $x_{t_{nT}}$ can be independent of $x_{t_{nT}-1}$ or $x_{t_{(n-1)T}}$. For example, we can consider the state in the lower level is initialized depending on the upper level current state i and the next state j. δ is defined over $I \times I$ such that $\delta(i, j)$ for some $i, j \in I$ gives a probability distribution over X.

With the introduction of δ, we simply need to rewrite V^* equation with δ in transition probabilities. In particular, if the δ-function is independent of $\pi^l[i, \lambda]$, the *optimal* finite T-horizon value at the lower level will act as a single step reward for the upper level (along with an immediate reward). The upper level decision maker in this case directs/determines a problem at each time that the lower level decision maker needs to solve and the lower level decision maker seeks a "*local*" optimal solution for T-horizon and follows one of the optimal nonstationary policies that achieve the solution. The decision process of how to direct a problem at each time for the upper decision maker will depend on the local optimal performance made by the lower decision maker. In this sense, this has the similar underlying philosophy of the Stackelberg (leader-follower) game (see, e.g, [2]). However, this is not true in general because δ-initialization function may depend on the lower level policy $\pi^l[i, \lambda]$, where in this case the lower level decision maker needs to choose a policy not only concerned with the local performance of the policy but also effects of the policy in the future performance.

The methods of obtaining the optimal decision rules for each level in MMDPs are well-established via the well-known solution algorithms: value iteration and policy iteration. We can define a contraction mapping (in sub-norm) operator defined over $X \times I$ and show that iterative application of the operator with an arbitrary bounded function defined over $X \times I$ converges to the fixed point, implying V^* is unique. Application of policy iteration in the context of MMDPs is also straightforward.

3 Related Works

For many interesting decision problems (e.g., queueing problems), the state spaces in different levels, (X and I), are non-overlapping. Sutton et al. [16] considers a multi-time MDP model in the dimension of the action space only (action hierarchy) by defining "*options*" or "temporally extended" actions, where the state spaces in different time-scales are the same in the model. In our model, the upper level action $\lambda \in \Lambda$ is not temporally extended action from the action space of the lower level MDPs but is a control at its own right. It can be roughly said that the lower level policies defined over different upper level state and controls are semi-Markov options [16] that depend on the upper level state and action. A similar hierarchical structure in the dimension of only action space was studied in the Markov slowscale model and the delayed slowscale Model by Jacobson et al. [10]. They consider two level action hierarchy, where the upper level control is not necessarily an option. However, the upper level control does not change the transition and reward structure of the whole T-horizon evolutionary process.

The recent work by Ren and Krogh on multi-mode MDPs [13] studies a nonstationary MDP, where a variable called the system operating mode determines the evolution of the lower level MDP. However, the transition of the modes operates with the same time scale with the lower level MDP, making the whole transition dynamics of the system operates in the *one* time-scale and the reward structure is defined over the one time scale. It is our fundamental assumption that the upper level decision making process operates in a different and slower time scale than the lower level.

In Trivedi et al.'s work [8], the performance models (fast time-scale model) are *solved* to obtain performance measures (corresponding to R^u roughly under the lower level policy independent δ-function). These measures are used as reward rates which are assigned to states of the dependability model (slow time-scale). The dependability model is then solved to obtain performability measures. The lower level is modeled by a continuous-time Markov chain and the upper level is modeled by a Markov reward process (alternatively, generalized stochastic petri net. can be used). If we fix the upper level and the lower level decision rules in our model with the lower level policy independent δ-function, an MMDP becomes (roughly) the model described by Trivedi et al.'s — in our model, the lower level model is also a Markov reward/decision process.

4 Example Problems

There are many interesting problems that can be modeled by MMDPs. Due to the space constraint, in this section we illustrate this by considering only three examples in different contexts (see [4] for more examples, including multi-armed bandit problems, asset allocation, etc.). The description for each problem is rather abstract but detailed enough to convey the idea of the MMDP modeling.

4.1 Call routing with buffer management

We consider a simple call-routing problem with buffer management motivated from a "load-balancing" control problem.

Consider a network with M parallel links between two points of source and destination. At the source, (voice) calls of each class arrive with different Bernoulli processes in a slow time-scale, where classes are set up based on quality constraints. Each call's holding time is geometrically distributed in the slow time-scale. The call-level or upper level decision process is either to reject a newly arrived call or to decide where to dispatch or route the newly arrived call to one of the M links if accepted. We assume that for each link, there are (possibly zero) cross traffic (video) calls. For simplicity, the video calls are initially set up and do not depart (if we incorporate the dynamics of video call arrival and departure process, we would have a three-level MMDP and the control process in the highest level is to assign video calls among M parallel links or reject).

A Markov modulated Bernoulli process is used to model video packet arrival process [5] and it is assumed that each video call shares the same model parameters and a Markov On/Off process is associated with each voice. We then can induce a model for voice/video traffic that describes packet arrival dynamics at the fast time-scale given the number of currently pending voice/video calls with an appropriate choice of the δ-initialization function at each link.

The upper level state consists of the number of currently pending voice and video calls at each link. The lower level state consists of the traffic states for voice traffic and video traffic and the number of packets in the (finite FIFO) buffer for voice and video at each link. The control action at the lower level is to control the queue length of voice and video packets at each link. The lower level reward function is given by the sum of the reward function at the individual link plus reward of routing/rejecting a voice call.

4.2 Composite (controlled) performance and dependability model

We give an extension of the example discussed by Goševa-Popstojanova and Trivedi [8] to show how the MMDP model can be used to extend their hierarchical model of performance and dependability (or availability) into an interesting model that incorporates controls. We don't base our discussion on exactly the same example given in [8]. Our example is appropriately modified from the example into our contexts.

There are N multiprocessors, where each processor has a different service characteristic and a different failure rate. The concept of "dependability" is related with failure, reconfiguration, and repair of these processors with respect to system behavior, and closely related with dependability, availability measures a potential service capacity that the system can deliver [8]. For our extension, we naturally introduce a "maintenance" operation in a slow time-scale. We wish to maintain the system such that a better availability/dependability is achieved. The measure of availability will depend on the performance made by each processor in a fast time-scale.

Each processor k's upper level state has two parts. The first part indicates whether the processor is up or down. The second part is the processor k's currently effective exponential service rate S in a finite set of $\{S_{k,1}, ..., S_{k,m}\}$ if the first part is up, where we assume that there is a Markov process that describes transition of service rates for each processor k in the slow time-scale. If the first part is down, the second part indicates a current "symptom" or reasons of failure from a finite set of $\{F_{k,1}, ..., F_{k,m}\}$. The processor k is subject to failure $F_{k,n}$ with an exponential rate $f_{k,n}$. An upper level action is to select one processor to be repaired among currently failed processors. The immediate repair cost at the upper level is, for example, a time to repair associated with symptom F.

At the lower level, jobs or customers of type k (associated with weight w_k) arrive at processor k according to Poisson process with an arrival rate of a_k. Each job requires a processing time that is exponentially distributed with a rate τ_k and it is assumed that on a job's arrival the processing time is known to the processor. The lower level state is the queue size at each processor k in terms of the total processing time for the jobs in the queue. The lower level action for each processor is to do admission control to newly arriving jobs based on the current processor's service rate (and its future stochastic variation if we use the lower level dependent δ-function) to effectively control the throughput and the waiting time of the processed jobs based on a given tradeoff parameter. The performance made by each processor k times w_k will contribute to the upper level single step reward.

Our overall goal in this example is to maintain the system (at both levels) to make overall system's availability optimal over infinite horizon. In addition to our example here, many interesting variants of Trivedi et al.'s composite performance and availability model can be considered via the MMDP modeling.

4.3 Production planning

The production planning problem we consider here is based on [3] and is divided into two levels: "marketing management" level and "operational" level. At the marketing manageme nt level, we need to control which *family* to produce over each (slow time-scale) decision epoch, where a family is a set of items consuming the same amount of resources and sharing the same setup [3]. The upper level state consists of (stochastically) available resources for each family and (stochastic) setup costs for each family, and some market-dependent factors. The upper level action is to choose which family to produce.

At the operational level, we need to determine actual quantities of the items in the family (the lower level actions) given stochastic (Markovian) demands for the items, production capacity, holding cost, material cost, etc., which will constitute a state of the lower level.

The return at the operational level will be a function of the unit selling price of the items, the inventory holding costs, the setup costs of the (current) family, the production quantity of the items, etc. and the T-epoch expected accumulated return at the operational level will be the one-step return for the management level.

5 Concluding Remarks

In the evolutionary process of MDPs, the outcome of taking an action at a state is the next state. Usually, the matter of *when* this outcome is known to the system is not critical as long as the system comes to know the next state before the next decision time. However, this might be an issue on the MMDP model. In our model, we assumed that the next state at the upper level is known at the near boundary of the next time step (refer Figure 1), which is quite reasonable we believe. If the effect of taking an action $\lambda \in \Lambda$ at a state $i \in I$ is immediate, which is the next state $j \in I$, $P^l(y|x,i,\lambda)$ will be possibly given as $P^l(y|x,j)$. This issue is the problem specific matter and needs to be resolved by the system design.

We made the assumption that action spaces at all levels in the hierarchy are distinct. Even though we believe that this is a natural assumption, we speculate that for some applications, some actions might be shared by different levels. Our assumption can be relaxed (with added complexity to the model) so that some actions are shared by different levels as long as any action taken at a state in a level does not affect the higher level state transitions. Developing a multi-time scale model for the case where a lower level action affects the higher level transitions (in a different time-scale) is still open problem.

References

[1] A. Arapostathis, V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus, "Discrete-time controlled Markov processes with average cost criterion: a survey," *SIAM J. Cont. and Opt.*, vol. 31, no. 2, pp. 282–344, 1993.

[2] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, Academic Press, London/New York, 1995.

[3] G. Bitran, E. A. Hass, and K. Matsuo, "Production planning of style goods with high set-up costs and forecast revisions," *Oper. Res.*, vol. 34, pp. 226–236, 1986.

[4] H. S. Chang, P. Fard, S. Marcus, and M. Shayman, "Multi-time scale Markov decision processes," Tech. Rep. 2002-6, ISR, Univ. of Maryland, College Park, 2002.

[5] W. Fischer and K. Meier-Hellstern, "The Markov-modulated Poisson process (MMPP) cookbook," *Perf. Eval.*, vol. 18, pp. 149–171, 1992.

[6] J. Forestier and P. Varaiya, "Multilayer control of large Markov chains," *IEEE Trans. on Auto. Contr.,* vol. AC-23, no. 2, pp. 298–304, 1978.

[7] S. B. Gershwin, "Hierarchical flow control: a framework for scheduling and planning discrete events in manufacturing systems," *Proc. of the IEEE*, vol. 77, no. 1, pp. 195–208, 1989.

[8] K. Goseva-Popstojanova and K. S. Trivedi, "Stochastic modeling formalisms for dependability, performance and performability," *Perf. Eval. - Origins and Directions, Lecture Notes in Computer Science*, G. Haring, C. Lindemann, M. Reiser (eds.), pp. 385–404, Springer Verlag, 2000.

[9] M. Grossglauser and D. Tse, "A time-scale decomposition approach to measurement-based admission control," submitted to *IEEE/ACM Trans. on Net.*

[10] M. Jacobson, N. Shimkin and A. Shwartz, "Piecewise stationary Markov Decision Processes, I: constant gain," submitted to *Math. of Oper. Res.*

[11] M. Mahmoud, "Multilevel systems control and applications: a survey," *IEEE Trans. on Systems, Man, and Cybernetics,* vol. SMC-7, no. 3, pp. 125–143, 1977.

[12] M. Mandjes and A. Weiss, "Sample path large deviations of a multiple time-scale queueing model," submitted, 1999.

[13] Z. Ren and B. H. Krogh, "Mode-matching control policies for multi-mode Markov decision processes," in *Proc. of ACC*, vol. 1, 2001, pp. 95–100.

[14] S. P. Sethi and Q. Zhang, *Hierarchical Decision Making in Stochastic Manufacturing Systems*, in series Systems and Control: Foundations and Applications, Birkhäuser Boston, Cambridge, MA, 1994.

[15] A. Shwartz and A. Weiss, "Multiple time scales in Markovian ATM models I. Formal calculations," CC PUB 267, Electrical Engineering, Technion, 1999.

[16] R. Sutton, D. Precup, and S. Singh, "Between MDPs and Semi-MDPs: a framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, pp. 181–211, 1999.

[17] D. Tse, R.G. Gallager and J.N. Tsitsiklis, "Statistical multiplexing of multiple time-scale Markov streams," *IEEE JSAC*, vol. 13, no. 6, pp. 1028–1039, 1995.

[18] T. Tuan and K. Park, "Multiple time scale congestion control for self-similar network traffic," *Perf. Eval.*, vol. 36-37, pp. 359–386, 1999.

[19] W. Willinger, M. Taqqu, W. Leland and D. Wilson, "Self-similarity in high-speed packet traffic: analysis and modeling of Ethernet traffic measurements," *Stat. Sci.* vol. 10, pp. 67–85, 1995.