# Multi-time Scale Markov Decision Processes

Hyeong Soo Chang, *Member, IEEE*, Pedram Fard, *Member, IEEE*, Steven I. Marcus, *Fellow, IEEE*, and Mark Shayman, *Member, IEEE*

***Abstract*— This paper proposes 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 discrete time-scales. In this model, the state space and the control 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 transition dynamics of higher levels. The performance produced by the lower level decisions will affect the higher level decisions. A hierarchical objective function is defined such that the finite-horizon value of following a (nonstationary) policy at level $m+1$ over a decision epoch of level $m$ plus an immediate reward at level $m$ is the single-step reward for the decision making process at level $m$. From this we define "multi-level optimal value function" and derive "multi-level optimality equation". We discuss how to solve MMDPs exactly and study some approximation methods, along with heuristic sampling-based schemes, to solve MMDPs.**

***Index Terms*— Markov decision process, multi-time scale, hierarchical control, rolling horizon**

## I. Introduction

HIERARCHICALLY structured control problems have been studied extensively in many contexts in various areas with many types of models. 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., [21]) and "multi-layer structure", where algorithms are divided "spatially" and operate at the same time-scale (see, e.g., [12]).

This paper focuses on 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 discrete 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. In particular, the approach with a pyramid-like hierarchical structure was used in the perspective of "performability" and "dependability" in Trivedi et al.'s model [14] [23] even though controls are not involved in the model (see also [13] [5] and chapter 11 in [28] for using a similar idea).

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. 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" for infinite horizon discounted reward and average reward, respectively. After discussing the exact methods for computing the optimal multi-level value function, we present approximation methods suited for solving MMDPs and analyze its performance and discuss how to apply some previously published on-line solution schemes in the context of MMDPs.

This paper is organized as follows. We start with some control problem examples to motivate the model proposed in the present paper in Section II and present a formal description of MMDPs and characterize optimal solutions for MMDPs in Section III and discuss solution methodologies in Section IV. We then discuss relevant work of hierarchical models related to our model in Section V. We conclude our paper in Section VI.

## II. Motivating Examples

### A. Production planning

Hierarchical production planning problems have been studied in the operations research literature over many years (see, e.g., [28] for references). We present a simple production planning problem in a manufacturing environment as the first motivating example. We base our discussion on the problem studied in [5].

The production planning problem we consider here is divided into two levels: "marketing management" level and "operational" level. At the marketing management 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 [5]. 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 a finite horizon expected accumulated return at the operational level will be the one-step return for the management level from which the management level makes decisions. We wish to develop a two-level production plan to maximize revenue of the manufacturing system.

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.

H. S. Chang is with the Department of Computer Science and Engineering at Sogang University, Seoul, Korea, and can be reached by e-mail at hschang@ccs.sogang.ac.kr.

P. Fard, S. I. Marcus, and M. Shayman are all with the Department of Electrical and Computer Engineering at the University of Maryland, College Park, and can be reached by e-mail at pedram@eng.umd.edu, marcus@eng.umd.edu and shayman@eng.umd.edu, respectively.

A preliminary version of parts of the paper appears in Proceedings of the 41st IEEE CDC 2002.

### B. Call routing with buffer management or scheduling

In addition to inherently existing hierarchical and multi-time scale control structure in problems themselves that arise in many different contexts, our model is also motivated by the observation made in the networking literature recently. The network traffic shows fluctuations on multiple time-scales — scale invariant burstiness (see, e.g., [34]), 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., [29] [32] [22] and references therein). The usual interests are in calculations of the 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., [33] and [15]). 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.

Consider a simple call-routing problem with buffer management or scheduling. There is a network with $L > 1$ parallel links between a pair of source and destination. At the source, single class (voice) calls arrive with an arrival rate according to Bernoulli process in a slow time-scale. The call's holding time is geometrically distributed in the slow time-scale. The call-level or upper level decision process is to either reject a newly arrived call or route the newly arrived call to one of the $L$ parallel 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 decision making process and the control process in the highest level is to assign video calls among $L$ parallel links or reject).

It is assumed that all voice calls have the same traffic rate (i.e., bandwidth requirement) and this is also true of the video calls. In other words, the model that describes packet (of the same size with the unit time in fast time-scale) arrival process of the voice call is the same. For example, if On/Off model is used, each arriving call has the same On/Off model parameters. This also holds for the video calls. For instance we may use Markov modulated Bernoulli process to model video packet arrival process [11] and it is also assumed that all video calls share the same model parameters.

The upper level control 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 size, e.g., via an admission control or dropping packets at each link or schedule these multiclass packets.

We wish to develop a two-level control policy such that the upper level call admission control effectively balances the loads of each link depending on the performance made by the lower level queueing control of the packets to maintain a desired throughput/delay.

## III. MULTI-TIME SCALE MDP

We first 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 any difficulty and we will remark on this issue later.

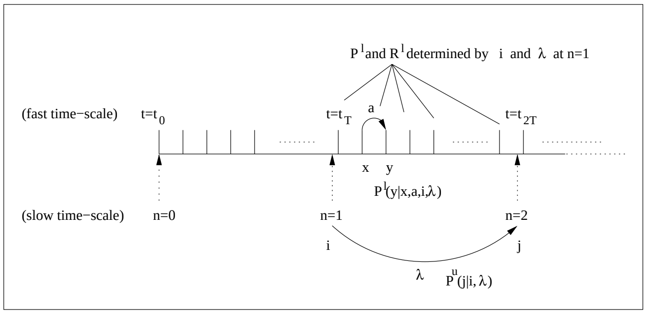

Fig. 1. Graphical illustration of time evolution in the two time-scale MDPs

### A. The model

The upper level (slow time-scale) MDP has a finite state space $I$ and a finite action space $\Lambda$. At each (discrete) decision time $n \in \{0, 1, 2, ..., \}$ and at a state $i_n \in I$, an action $\lambda_n \in \Lambda$ is taken and $i_n$ makes transition to a state $i_{n+1} \in I$ according to the 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 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 $A \cap \Lambda = \emptyset$. 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 $t_{nT} = n + \epsilon$, where $\epsilon$ is a positive number arbitrarily close to zero. 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 be $x \in X$ and the initial state in the upper level MDP be $i \in I$ ($x_{t_0} = x$ and $i_0 = i$ at $n = 0$). Over the time steps of $t_0, t_1, ..., t_{T-1}$, the system follows the lower level MDP evolution. 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 the probability $P^l(y|x, a, i, \lambda)$ and a *nonnegative and 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 $\lambda_1$ 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 $P^l(z|x_{t_{T-1}}, a_{t_{T-1}}, i_0, \lambda_0)$), 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" when referring to infinite horizon and the term "policy" when referring to finite horizon. Define a lower level *decision rule* $d^l = \{\pi^l_n\}$, $n = 0, 1, ...$, as a sequence of $T$-horizon *nonstationary policies* defined such that for all $n$, $\pi^l_n = \{\phi_{t_{nT}}, ..., \phi_{t_{(n+1)T-1}}\}$ is a sequence of functions where for all $k \geq 0$, $\phi_{t_k} : X \times I \times \Lambda \rightarrow A$. We will say that a lower level decision rule is *stationary with respect to the slow time-scale* $n$ if $\pi^l_n = \pi^l_{n'}$ 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 $\pi^l_n$ in this case and use the time $t_0, ..., t_{T-1}$ to refer the sequence of functions of $\pi^l$ if necessary, and denote $\Pi^l$ as the set of all possible such $T$-horizon nonstationary policies $\pi^l$. We will also omit the subscript on $\phi$ if $\pi^l$ is stationary (with respect to the fast time-scale).

Given a lower level decision rule $d^l \in \mathcal{D}^l$ and a nonnegative and 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 $\lambda_n \in \Lambda$,

$$R^u(x, i_n, \lambda_n, \pi^l) = E^x_{i_n, \lambda_n}\left\{\sum_{t=t_{nT}}^{t_{(n+1)T-1}} \alpha^{\sigma(t)} R^l(x_t, \phi_t(x_t, i_n, \lambda_n), i_n, \lambda_n)\right\} + \mathcal{I}^u(i_n, \lambda_n), 0 < \alpha \leq 1, \tag{1}$$

where $\sigma(t_{nT+r}) = r$ 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 $i_n, \lambda_n$ on $E$ signifies that $i_n$ and $\lambda_n$ for the expectation are fixed. We will use this notational method throughout the paper. The function $R^u$ is simply the $T$-horizon total expected (discounted) reward of following the $T$-horizon nonstationary policy $\pi^l$ 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 $\lambda_n$ at the state $i_n$ at the upper level.

The total expected (discounted) reward achieved by the lower level $T$-horizon nonstationary policy $\pi^l$ with an immediate reward at the upper level will act as a *single-step reward* for the upper level MDP. Define an upper level *stationary* decision rule $d^u$ as a function $d^u : X \times I \rightarrow \Lambda$ and we denote $\mathcal{D}^u$ as the set of all possible such stationary decision rules. Given the initial states $x \in X$ 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 following functional value defined over $X \times I$: with $0 < \gamma < 1$,

$$\begin{aligned} V^*(x, i) &:= \max_{d^u \in \mathcal{D}^u} \max_{d^l \in \mathcal{D}^l} E^{x,i}\left\{\sum_{n=0}^{\infty} \gamma^n R^u(x_{t_{nT}}, i_n, d^u(x_{t_{nT}}, i_n), \pi^l)\right\} \\ &= \max_{d^u \in \mathcal{D}^u} \max_{d^l \in \mathcal{D}^l} E^{x,i}\left\{\sum_{n=0}^{\infty} \gamma^n \left(E^{x_{t_{nT}}}_{i_n, \lambda_n}\left[\sum_{t=t_{nT}}^{t_{(n+1)T-1}} \right.\right.\right. \\ &\left. \alpha^{\sigma(t)} R^l\left(x_t, \phi_t(x_t, i_n, d^u(x_{t_{nT}}, i_n)), i_n, d^u(x_{t_{nT}}, i_n)\right)\right] \\ &\left.\left. + \mathcal{I}^u(i_n, \lambda_n)\right)\right\}, \end{aligned} \tag{2}$$

where we will refer to $V^*$ as *the two-level optimal infinite horizon discounted value function.*

The second functional value defined as our objective function is

$$J^*(x, i) := \max_{d^u \in \mathcal{D}^u} \max_{d^l \in \mathcal{D}^l} \lim_{H \to \infty} \frac{1}{H} E^{x,i}\left\{\sum_{n=0}^{H-1} R^u(x_{t_{nT}}, i_n, d^u(x_{t_{nT}}, i_n), \pi^l)\right\},$$

where we refer to $J^*$ as *the two-level optimal infinite horizon average value function.* 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 $\pi^l$. We will consider the more general case of determining the initial state in a 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 $0 < \alpha < 1$.

### B. *Optimality equations*

For a given pair of $i \in I$ and $\lambda \in \Lambda$, define a set $\Pi^l[i, \lambda]$ of all possible $T$-horizon (lower level) nonstationary policies under the fixed pair of the upper level state $i$ and action $\lambda$:

$$\Pi^l[i, \lambda] := \left\{\pi^l[i, \lambda] \;\middle|\; \pi^l[i, \lambda] := \{\phi^{i,\lambda}_{t_0}, ..., \phi^{i,\lambda}_{t_{T-1}}\}, \right. \\ \left. \phi^{i,\lambda}_{t_k} : X \times \{i\} \times \{\lambda\} \rightarrow A \text{ and } k = 0, ..., T-1\right\}$$

and let $\mathcal{P}^T_{xy}(\pi^l[i, \lambda])$ the probability that a state $y \in X$ is reached by $T$-steps starting with $x$ by following the $T$-horizon nonstationary policy $\pi^l[i, \lambda]$. Note that this probability can be obtained from $P^l$.

We now 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 $\lambda_n$ is taken). Observe that we can view $\pi^l$ 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

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

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 can be derived. In other words, the upper level sequential dynamics is essentially just an MDP with a reward function defined via the lower level MDP dynamics. With a simple adaptation of the standard MDP theory (see, e.g, [1] [16] or [26]), the following results hold for MMDPs. Therefore, we omit proofs.

*Theorem 1:* For all $x \in X$ and $i \in I$,

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

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 $\lambda^*$ and $\pi^*[i, \lambda] = \{\phi^*_{t_k}\}$, and set $d^u(x, i) = \lambda^*$ for $d^u$ and set $\pi^l$ 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 $V^*$.

*Theorem 2:* If there exists a bounded function $\zeta$ defined over $X \times I$ and a constant $g$ such that for all $x \in X$ and $i \in I$,

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

then there exists a decision rule pair of $d^u \in \mathcal{D}^u$ and $d^l \in \mathcal{D}^l$ that achieves $J^*(x, i)$ and $g = J^*(x, i)$ for all $x \in X$ and $i \in I$.

For the conditions that make the "if" part of the above theorem hold, refer to [1] or [16] for a substantial discussion. An optimal decision rule pair can be obtained in a similar way to that stated in Theorem 1.

[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 $R^u$.

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]).

### C. Initialization function

So far we considered the case where $x_{t_{nT}}, n = 1, 2, ...$ is determined by a $T$-horizon nonstationary policy. Considering the more general model, we define an *initialization function* $\delta$ such that we determine or initialize $x_{t_{nT}}, n = 1, 2, ...$ by $\delta$. This is motivated by the specific nature of a given problem or organizing behavior in a hierarchy.

Here are some examples of $\delta$. As before, $\delta$ can be a function defined over $X \times I \times \Lambda$ such that for given $x, i, \lambda$, $\delta(x, i, \lambda)$ 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 $\mathcal{P}^T_{xy}(\pi^l[i, \lambda])$. We will also use the notation $\delta^{\pi^l}$ to explicitly express the dependence on the lower level policy $\pi^l$ if that is the case. Or $\delta$ can be defined such that the determination of $x_{t_{nT}}$ depends on the state $x_{t_{nT-1}}$. For example, for some $x, y \in X$, $i \in I$, $\lambda \in \Lambda$,

$$\delta^{\pi^l}(x, i, \lambda)[y] = \sum_{z \in X} \mathcal{P}^{T-1}_{xz}(\pi^l[i, \lambda])\rho(y|z),$$

where $\rho(y|z)$ denotes the probability that $y$ succeeds $z$.

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, $\delta$ is defined over $X \times \Lambda$ such that $\delta(x, \lambda)$ gives a probability distribution over $X$. The very idea of this $\delta$ is parallel to the transition structure in Markovian slowscale model given in [18]. Finally, the determination of $x_{t_{nT}}$ can be independent of $x_{t_{nT-1}}$ or $x_{t_{(n-1)T}}$, i.e., we can consider the state in the lower level be initialized based on the upper level current state $i$ and the next state $j$. $\delta$ 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 $\delta$, we simply need to rewrite the $V^*$ equation (similarly to the $J^*$ case) by replacing $P^T$ with $\delta$ in each equation in Theorem 1 and 2. In particular, if the $\delta$-function is independent of $\pi^l$ (or we will say that the $\delta$-function is independent of the lower level policies), then we can write the optimality equation for $V^*$ as

$$V^*(x, i) = \max_{\lambda \in \Lambda}\Bigg\{ \max_{\pi^l[i,\lambda]} \Bigg( R^u(x, i, \lambda, \pi_l) \Bigg) + \gamma \sum_{y \in X} \sum_{j \in I} \delta(x, i, \lambda)[y] P^u(j|i, \lambda) V^*(y, j) \Bigg\}.$$

This special case is very interesting because 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 the $T$-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, the case has a flavor of the underlying philosophy of the Stackelberg (leader-follower) game (see, e.g, [2]). This is not true in general because a $\delta$-initialization function may depend on the lower level policy, 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.

We end this section with a brief discussion on how to extend the two time-scale model to $M$ time-scale model with $M > 2$. The transition structure of a particular level $m$ depends on (in general) the states and the actions of the slower time-scale levels $n < m$ and the current state of $m$, and the reward function of the level $m$ is defined with the states and the actions of the levels $n < m$, and an initial state and a policy of the $m + 1$ level. From these transition/reward functions, we can define the multi-level optimal value function and determine the multi-level optimality equation. We note that for the definition of the infinite horizon discounted multi-level optimal value, the slowest level ($m = 1$) always has the discount factor less than 1.

## IV. Solving MMDPs

The methods of obtaining the optimal decision rule for each level in MMDPs are well-established via the well-known MDP theory. We will pay attention to the $\delta$-initialization function that depends on the lower level policies and is defined over $X \times I \times \Lambda$ such that $\delta^{\pi^l}(x, i, \lambda)$ for $x \in X$, $i \in I$, and $\lambda \in \Lambda$ gives a probability distribution over $X$. The discussion here can be easily extended to other $\delta$-functions.

### A. Exact methods

We first discuss the discounted case and then the average case. Define an operator $\Theta$ such that for a (bounded and measurable) function $V$ defined over $X \times I$,

$$\Theta(V)(x, i) = \max_{\lambda \in \Lambda}\Bigg( \max_{\pi^l[i,\lambda] \in \Pi^l[i,\lambda]} \Bigg\{ R^u(x, i, \lambda, \pi^l[i, \lambda]) + \gamma \sum_{y \in X} \sum_{j \in I} \delta^{\pi^l}(x, i, \lambda)[y] P^u(j|i, \lambda) V(y, j) \Bigg\}\Bigg) \quad (3)$$

for all $x$ and $i$. Then, $\Theta$ is a *$\gamma$-contraction-mapping* in sup-norm. For any function $V$ defined over $X \times I$, let $||V|| = \sup_{x,i} |V(x, i)|$. For any bounded and measurable two function $U$ and $V$ defined over $X \times I$, it is true that

$$||\Theta(U) - \Theta(V)|| \leq \gamma ||U - V||.$$

This implies that $V^*$ is unique from the well-known fixed point theorem. Furthermore, for any such $V$,

$$\Theta^n(V) \to V^* \text{ as } n \to \infty,$$

where this method is known as *value iteration*.

For the average reward case, we assume that (appropriately modified) one of the ergodicity conditions in the page 56 of [16] holds. Then, average reward value iteration can be also applied. Let $\Phi$ be an operator that maps a function $V$ defined over $X \times I$ to another function defined over $X \times I$ given by

$$\Phi(V)(x, i) = \max_{\lambda \in \Lambda}\Bigg( \max_{\pi^l[i,\lambda] \in \Pi^l[i,\lambda]} \Bigg\{ R^u(x, i, \lambda, \pi^l[i, \lambda]) + \sum_{y \in X} \sum_{j \in I} \delta^{\pi^l}(x, i, \lambda)[y] P^u(j|i, \lambda) V(y, j) \Bigg\}\Bigg) \quad (4)$$

for all $x$ and $i$. Then, with an arbitrary (bounded and measurable) function $V$ defined over $X \times I$, for all $x \in X, i \in I$,

$$\Phi^n(V)(x, i) - \Phi^{n-1}(V)(x, i) \to g \text{ as } n \to \infty$$

and for any fixed state pair $y \in X$ and $j \in I$,

$$\Phi^n(V)(x, i) - \Phi^n(V)(y, j) \to \zeta(x, i) \text{ as } n \to \infty, x \in X, i \in I.$$

We can also use *“policy iteration”* once $R^u$ is determined. See, e.g., [26]. The running time-complexity of value iteration is in polynomial in $|X||I|$, $|\Lambda| \cdot |A|^{T|X|}$, and $1/(1-\gamma)$ and in particular just one iteration takes $O((|X||I|)^2 \cdot |\Lambda| \cdot |A|^{T|X|})$. For policy iteration, just doing "policy improvement" step takes $O((|X||I|)^2 \cdot |\Lambda| \cdot |A|^{T|X|})$. See [20] for a detailed discussion, including the state and action space dependent time-complexity of the linear programming approach for solving MDPs. Therefore, applying the exact methods for solving MMDPs is very difficult even with relatively small state and action space sizes. In the next two subsections, we study approximation and heuristic methods to solve MMDPs.

### *B. Approximation methods*

There are numerous approximation algorithms to solve MDPs. For details see the books by Puterman [26] or by Bertsekas and Tsitsiklis [4]. In this section, we analyze the performance of an approximation-based scheme for solving MMDPs.

Our first approximation is on the $\delta$-initialization function. One of the main difficulties to obtain an optimal decision rule pair would be the possible dependence of $\delta$ on the lower level nonstationary policies. Suppose that this is the case and consider a $\delta'$-initialization function that is independent of the lower level policies and approximates the given $\delta^{\pi^l}$-initialization function with respect to a given metric. Then there exists a unique function $\hat{V}^*$ defined over $X \times I$ such that for all $x$ and $i$,

$$\hat{V}^*(x,i) = \max_{\lambda \in \Lambda} \Bigg\{ \max_{\pi^l[i,\lambda] \in \Pi^l[i,\lambda]} \Big( R^u(x,i,\lambda,\pi^l[i,\lambda]) \Big) + \gamma \sum_{y \in X} \sum_{j \in I} \delta'(x,i,\lambda)[y] P^u(j|i,\lambda) \hat{V}^*(y,j) \Bigg\}. \tag{5}$$

Note that $\hat{V}^*$ is the optimal value function for a new MDP defined with the reward function of $\max(R^u(\cdot))$ and the transition function defined with $\delta'$ and $P^u$. We can bound then $|\hat{V}^*(x,i) - V^*(x,i)|$ for all $x$ and $i$ by Theorem 4.2 in Müller's work [24] with a metric called the "integral probability metric" on the difference between $\delta'$ and $\delta^{\pi^l}$. Of course, if the MMDP problem to solve is associated with the lower level policy independent $\delta$-function, we wouldn't need this approximation step.

The second approximation is on the value of $R^*$ defined as

$$R^*(x,i,\lambda) = \max_{\pi^l[i,\lambda] \in \Pi^l[i,\lambda]} \Big( R^u(x,i,\lambda,\pi^l[i,\lambda]) \Big)$$

and on $\hat{V}^*(x,i)$. It will be often impossible to get the true $R^*$ due to a large state space size of the lower level and a relatively large $T$ even though theoretically we can use "backward induction". Obtaining the true value of the function $\hat{V}^*$ is also almost infeasible in many cases with the similar reasons. Suppose that we approximate $R^*$ by $\hat{R}$ such that

$$\sup_{x,i,\lambda} |R^*(x,i,\lambda) - \hat{R}(x,i,\lambda)| \le \kappa$$

and $\hat{V}^*$ by some bounded and measurable function $U$ defined over $X \times I$ such that

$$\sup_{x,i} |\hat{V}^*(x,i) - U(x,i)| \le \epsilon.$$

We will discuss an example of such $\hat{R}$ and $U$ later in this subsection. Now define a stationary (upper level) decision rule $d^u$ such that for all $x \in X$ and $i \in I$,

$$d^u(x,i) \in \operatorname*{arg\,max}_{\lambda \in \Lambda} \Bigg( \hat{R}(x,i,\lambda) + \gamma \sum_{y \in X} \sum_{j \in I} \delta'(x,i,\lambda)[y] P^u(j|i,\lambda) U(y,j) \Bigg).$$

Our goal is to bound the performance of the decision rule $d^u$ from $\hat{V}^*$. We define the value of following the decision rule $d^u$ given an initialization function $\delta'$ as follows:

$$\hat{V}(x,i) = E^{x,i}_{\delta'} \left\{ \sum_{n=0}^{\infty} \gamma^n \hat{R}(x_{t_{nT}}, i_n, d^u(x_{t_{nT}}, i_n)) \right\},$$

where we used (by abusing the notation) $E_{\delta'}$ to indicate that $x_{t_{nT}}, n = 1, 2, ...$ is a random variable denoting (lower level) state at time $t_{nT}$ determined stochastically by $\delta'$. We now state a performance bound as a theorem below.

*Theorem 3:* If $\sup_{x,i,\lambda} |R^*(x,i,\lambda) - \hat{R}(x,i,\lambda)| \le \kappa$ and $\sup_{x,i} |\hat{V}^*(x,i) - U(x,i)| \le \epsilon$,

$$|\hat{V}^*(x,i) - \hat{V}(x,i)| \le \frac{2\gamma\epsilon + \kappa}{1-\gamma} \text{ for all } x \in X \text{ and } i \in I.$$

*Proof:* Let the argument that achieves the maximum in the r.h.s of Equation (3) with replacing $\delta^{\pi^l}$ by $\delta'$ be $\lambda_U$ for a function $U$. We will use the notation $\Theta'$ for this replacement. From the contraction mapping property of the $\Theta'$ operator, for all $x \in X$ and $i \in I$,

$$|\Theta'(\hat{V}^*)(x,i) - \Theta'(U)(x,i)| \le \gamma \cdot \sup_{x,i} |\hat{V}^*(x,i) - U(x,i)| \le \gamma\epsilon. \tag{6}$$

We show that $|\Theta'(U)(x,i) - \hat{V}(x,i)| \le \frac{\gamma\epsilon(1+\gamma)+\kappa}{1-\gamma}$ for all $x \in X$ and $i \in I$. It then follows that from $\Theta'(\hat{V}^*) = \hat{V}^*$,

$$\begin{aligned} |\hat{V}^*(x,i) - \hat{V}(x,i)| &\le |\Theta'(\hat{V}^*)(x,i) - \Theta'(U)(x,i)| \\ &\quad + |\Theta'(U)(x,i) - \hat{V}(x,i)| \\ &\le \gamma\epsilon + \frac{\gamma\epsilon(1+\gamma)+\kappa}{1-\gamma} = \frac{2\gamma\epsilon+\kappa}{1-\gamma}, \end{aligned}$$

which gives the desired result.

Now, extending the proof idea of Theorem 3.1 in [17] with the given bound assumptions and the bound of Equation (6), it can be shown that for all $l = 0, 1, ...,$ and $x \in X, i \in I$,

$$\begin{aligned} \Theta'(U)(x,i) &\le E^{x,i}_{\delta'} \left[ \sum_{n=0}^{l} \gamma^n \hat{R}(x_{t_{nT}}, i_n, d^u(x_{t_{nT}}, i_n)) \right] \\ &\quad + \gamma^{l+1} E_{\delta'}[\Theta'(U)(x_{t_{(l+1)T}}, i_{l+1})] \\ &\quad + \gamma\epsilon(1+\gamma) + \cdots + \gamma^{l+1}\epsilon(1+\gamma) \\ &\quad + \kappa(1 + \gamma + \cdots + \gamma^l). \end{aligned} \tag{7}$$

Since $\Theta'(U)$ is bounded, the second term on the r.h.s. of Equation (7) converges to zero as $l \to \infty$ and the first term becomes $\hat{V}(x,i)$ by the definition and the last two terms sum to $\frac{\gamma\epsilon(1+\gamma)+\kappa}{1-\gamma}$ as $l \to \infty$. It follows that $\Theta'(U)(x,i) - \hat{V}(x,i) \le \frac{\gamma\epsilon(1+\gamma)+\kappa}{1-\gamma}$. This proves the upper bound case.

As for the lower bound case, by the similar arguments for the upper bound case, we can then show that $\Theta'(U)(x,i) - \hat{V}(x,i) \ge -\frac{\gamma\epsilon(1+\gamma)+\kappa}{1-\gamma}$. This concludes our proof. ■

We remark that a related work for this theorem can be found in Corollary 1 in [30] with the assumption of the finite state space and the result of the work only gives an upper bound. Our analysis takes a totally different approach and can be applied to *Borel* state space even though our proof shows for the countable case. Furthermore, the result gives not only a lower bound but also a tighter bound (the upper bound given in [30] is $\frac{2\gamma\epsilon+2\kappa}{1-\gamma}$). Once we bound $V^*(x,i)$ from $\hat{V}^*(x,i)$ by Müller's work, we have a bound for the optimal value function value of the original MMDP at $x$ and $i$ from $\hat{V}(x,i)$. Now we give an example of $\hat{R}$. *From now on, we assume that the lower level reward function $R^l$ is defined such that it absorbs the upper level immediate reward function $\mathcal{I}^u$* as we discussed in the subsection III-A. Our approximation uses a lower level policy $\pi^l$ that guarantees the $T$-horizon total expected discounted reward of following the policy $\pi^l$ is within an error bound from the optimal finite-horizon value.

The methodology of the example is the *rolling horizon* approach [17] where we choose a horizon $h \ll T$ and solve for the optimal $h$-horizon total expected discounted reward and we define a (greedy) stationary policy with respect to the value function. We begin by defining $h$-horizon total expected discounted reward with $h = 1, ..., T$ for every given $i \in I$ and $\lambda \in \Lambda$:

$$R^*_h(x, i, \lambda) = \max_{\pi^l[i,\lambda]} E^x_{i,\lambda} \left\{ \sum_{t=0}^{h-1} \alpha^t R^l(x_t, \phi^{i,\lambda}_t(x_t, i, \lambda), i, \lambda) \right\}, \tag{8}$$

where $0 < \alpha < 1$ and $R^*_0(x, i, \lambda) = 0$ for all $x \in X$. We also let $R^{\pi^l}(x, i, \lambda) = R^u(x, i, \lambda, \pi^l[i, \lambda])$ defined in Equation (1) for every $i$ and $\lambda$ with $0 < \alpha < 1$ and $R_{\max} = \max_{x,a,i,\lambda} R^l(x, a, i, \lambda)$.

*Proposition 1:* For every given $i \in I$ and $\lambda \in \Lambda$ and a selected $h$ in $\{1, ..., T\}$, define a lower level stationary policy $\pi^l[i, \lambda]$ as

$$\phi^{i,\lambda}(x, i, \lambda) \in \arg\max_{a \in A} \Bigg( R^l(x, a, i, \lambda) + \alpha \sum_{y \in X} P^l(y|x, a, i, \lambda) R^*_{h-1}(y, i, \lambda) \Bigg) \text{for all } x \in X.$$

Then, for all $x, i, \lambda$,

$$0 \leq R^*(x, i, \lambda) - R^{\pi^l}(x, i, \lambda) \leq \frac{R_{\max}\alpha^h(1 - \alpha^T)}{1 - \alpha}.$$

*Proof:* The lower bound is from the definition of $R^*$. Fix arbitrary $i \in I$ and $\lambda \in \Lambda$. Define an operator $\Omega$ that maps a (bounded) function $V$ defined over $X$ to another function defined over $X$ given by

$$\Omega(V)(x) = \max_{a \in A} \left( R^l(x, a, i, \lambda) + \alpha \sum_{y \in X} P^l(y|x, a, i, \lambda) V(y) \right). \tag{9}$$

It is well-known that $R^*_h = \Omega^h(R^*_0)$, where $\Omega^h$ denotes the successive application of the $\Omega$ operator by $h$ times (see, e.g., [26] [16], etc.). By the contraction mapping property of $\Omega$, (with $\|f\| = \sup_{x,i,\lambda} |f(x, i, \lambda)|$),

$$\begin{aligned} \|R^*_T - R^*_h\| &\leq \alpha \|R^*_{T-1} - R^*_{h-1}\| \\ &\leq \cdots \leq \alpha^h \|R^*_{T-h} - R^*_0\| \\ &\leq \alpha^h(1 + \alpha + \cdots + \alpha^{T-h-1})R_{\max} \\ &\leq \frac{R_{\max}(\alpha^h - \alpha^T)}{1 - \alpha}. \end{aligned} \tag{10}$$

Following the proof idea of Theorem 3.1 in [17], we can show that for all $w = 0, 1, ..., T - 1$ and for all $x \in X$,

$$\begin{aligned} R^*_h(x, i, \lambda) \leq\; & E^x_{i,\lambda} \left[ \sum_{t=0}^{w} \alpha^t R^l(x_t, \phi^{i,\lambda}(x_t, i, \lambda), i, \lambda) \right] \\ & + \alpha^{w+1} E_{i,\lambda}[R^*_h(x_{w+1}, i, \lambda)]. \end{aligned} \tag{11}$$

We let $w = T - 1$. It follows then that from the previous inequality (11), for all $x, i$ and $\lambda$,

$$R^*_h(x, i, \lambda) \leq R^{\pi^l}(x, i, \lambda) + \frac{R_{\max}\alpha^T(1 - \alpha^h)}{1 - \alpha}.$$

Therefore, we have that for all $x, i$ and $\lambda$,

$$\begin{aligned} R^*(x, i, \lambda) - R^{\pi^l}(x, i, \lambda) \leq\; & R^*(x, i, \lambda) - R^*_h(x, i, \lambda) \\ & + \frac{R_{\max}\alpha^T(1 - \alpha^h)}{1 - \alpha}. \end{aligned}$$

Combining the result in Equation (10) with the previous inequality, we finally have that

$$R^*(x, i, \lambda) - R^{\pi^l}(x, i, \lambda) \leq \frac{R_{\max}\alpha^h(1 - \alpha^T)}{1 - \alpha}.$$

∎

For every given $\kappa > 0$, letting $\kappa \geq \frac{R_{\max}\alpha^h(1-\alpha^T)}{1-\alpha}$ gives the rolling horizon size for a desired error bound for $R^*$. We remark that by letting $T \to \infty$, the above result precisely gives the result of Theorem 3.1 in [17]. A similar approach can be taken for the upper level MDP. We can choose a fixed rolling horizon for the upper level. The value function defined by the horizon approximates $\hat{V}^*$ in Equation (5), i.e., an example of $U$. If both levels use the rolling horizon approach, we have a two-level approximation. We can easily draw an error bound of the two-level rolling horizon approach from the results obtained in this subsection. In practice, getting the true value of $R^*_h$ will be also difficult even though $h$ is small due to the curse of dimensionality. A way of getting away with a large state space is to use a *sampling* method to approximate $R^*_h$ (see [19] [6]).

For the average reward case, we consider the case where one of the ergodicity conditions in the page 56 of [16] holds. Furthermore, we assume that the similar approximation to the first approximation for the discounted case is done by a $\delta'$-initialization function that is independent of the lower level policies. Then there exists a constant $\hat{g}$ and a function $\hat{\zeta}$ such that for all $x$ and $i$,

$$\hat{g} + \hat{\zeta}(x, i) = \max_{\lambda \in \Lambda} \Bigg\{ \max_{\pi^l[i,\lambda] \in \Pi^l[i,\lambda]} \Big( R^u(x, i, \lambda, \pi^l[i, \lambda]) \Big) + \sum_{y \in X} \sum_{j \in I} \delta'(x, i, \lambda)[y] P^u(j|i, \lambda) \hat{\zeta}(y, j) \Bigg\}$$

and that $|\hat{g} - g|$ is bounded with respect to the degree of the approximation by $\delta'$ for $\delta^{\pi^l}$.

We focus on the second approximation for the average case. We will denote $R^*_h$ defined in Equation (8) with $\alpha = 1$ as $\bar{R}^*_h$ and $\bar{R}^{\pi^l} = R^u(x, i, \lambda, \pi^l[i, \lambda])$ defined in Equation (1) with $\alpha = 1$, and the operator $\Omega$ in Equation (9) with $\alpha = 1$ as $\bar{\Omega}$. Suppose that we approximate $\bar{R}^*(= \bar{R}^*_T)$ by $\hat{R}$ as before such that

$$\sup_{x,i,\lambda} |\bar{R}^*(x, i, \lambda) - \hat{R}(x, i, \lambda)| \leq \kappa$$

and that $\hat{\zeta}$ is approximated by some function $U$ defined over $X \times I$ such that

$$\sup_{x,i} |\hat{\zeta}(x, i) - U(x, i)| \leq \epsilon.$$

Define a stationary (upper level) decision rule $d^u$ such that for all $x \in X$ and $i \in I$,

$$d^u(x, i) \in \arg\max_{\lambda \in \Lambda} \Bigg( \hat{R}(x, i, \lambda) + \sum_{y \in X} \sum_{j \in I} \delta'(x, i, \lambda)[y] P^u(j|i, \lambda) U(y, j) \Bigg).$$

The value of following the decision rule $d^u$ given an initialization function $\delta'$ is defined as follows:

$$\hat{J}(x, i) = \lim_{H \to \infty} \frac{1}{H} E^{x,i}_{\delta'} \left\{ \sum_{n=0}^{H-1} \hat{R}(x_{t_nT}, i_n, d^u(x_{t_nT}, i_n)) \right\}.$$

We now state a performance bound as a theorem below.

*Theorem 4:* Assume that one of the ergodicity conditions in the page 56 in [16] holds. If $\sup_{x,i,\lambda} |\bar{R}^*(x, i, \lambda) - \hat{R}(x, i, \lambda)| \leq \kappa$ and $\sup_{x,i} |\hat{\zeta}(x, i) - U(x, i)| \leq \epsilon$,

$$|\hat{g} - \hat{J}(x, i)| \leq 2\epsilon + \kappa \text{ for all } x \in X \text{and } i \in I.$$

*Proof:* Let the argument that achieves the r.h.s of Equation (4) with replacing $\delta^{\pi^l}$ by $\delta'$ be $\lambda_U$ for a function $U$. We will use the notation $\Phi'$ for this replacement.

Now, for all $x \in X$ and $i \in I$,

$$\begin{aligned}
\Phi'(U)(x,i) &= \bar{R}^*(x,i,\lambda_U) \\
&\quad + \sum_{y \in X}\sum_{j \in I} \delta'(x,i,\lambda_U)[y] P^u(j|i,\lambda_U) U(y,j) \\
&\quad \text{by the definition of } \Phi' \\
&\le \hat{R}(x,i,\lambda_U) + \kappa \\
&\quad + \sum_{y \in X}\sum_{j \in I} \delta'(x,i,\lambda_U)[y] P^u(j|i,\lambda_U) U(y,j) \\
&\quad \text{by the given assumption} \\
&\le \hat{R}(x,i,d^u(x,i)) + \kappa \\
&+ \sum_{y \in X}\sum_{j \in I} \delta'(x,i,d^u(x,i))[y] P^u(j|i,d^u(x,i)) U(y,j) \\
&\quad \text{by the definition of } d^u.
\end{aligned}$$

Under the ergodicity assumption, there exists a stationary probability distribution $\mathcal{P}$ over $X \times I$ for the induced Markov chain by $d^u$. Summing both sides with respect to $\mathcal{P}$ at the last inequality of the above equations, we have that

$$\begin{aligned}
\sum_{x,i} \mathcal{P}(x,i)\Phi'(U)(x,i) &\le \sum_{x,i} \mathcal{P}(x,i)\hat{R}(x,i,d^u(x,i)) \\
&+ \kappa + \sum_{x,i} \mathcal{P}(x,i)\Big(\sum_{y \in X}\sum_{j \in I} \delta'(x,i,d^u(x,i))[y] \\
&\times P^u(j|i,d^u(x,i))U(y,j)\Big). \qquad (12)
\end{aligned}$$

The first term on the right side is equal to $\hat{J}(x,i)$ by Lemma 3.3 (b.ii) in [17], and the third term on the right side is equal to $\sum_{x,i} \mathcal{P}(x,i)U(x,i)$ from the invariance property of $\mathcal{P}$.

Observe that if $|\hat{\zeta}(x,i) - U(x,i)| \le \epsilon$ for all $x \in X$ and $i \in I$, then $|\Phi'(\hat{\zeta})(x,i) - \Phi'(U)(x,i)| \le \epsilon$ for all $x \in X$ and $i \in I$. This implies that for all $x \in X$ and $i \in I$,

$$\begin{aligned}
\Phi'(\hat{\zeta})(x,i) - \hat{\zeta}(x,i) - 2\epsilon &\le \Phi'(U)(x,i) - U(x,i) \\
&\le \Phi'(\hat{\zeta})(x,i) - \hat{\zeta}(x,i) + 2\epsilon.
\end{aligned}$$

Therefore, rearranging the terms in Equation (12) and from the previous observation,

$$\begin{aligned}
\hat{J}(x,i) + \kappa &\ge \sum_{x,i} \mathcal{P}(x,i)[\Phi'(U)(x,i) - U(x,i)] \\
&\ge \sum_{x,i} \mathcal{P}(x,i)[\Phi'(\hat{\zeta})(x,i) - \hat{\zeta}(x,i)] - 2\epsilon \\
&= \hat{g} - 2\epsilon.
\end{aligned}$$

With the similar arguments, we can also show that $\hat{J}(x,i) - \kappa \le \hat{g} + 2\epsilon$. ■

We now provide a counterpart result to Proposition 1 for the undiscounted case ($\alpha = 1$) under an ergodicity assumption.

Define $C := \{(x,a)|x \in X, a \in A\}$. For every given $i \in I$ and $\lambda \in \Lambda$, we define $R^l(c,i,\lambda) := R^l(x,a,i,\lambda)$ and $P^l(y|c,i,\lambda) := P^l(y|x,a,i,\lambda)$ for all $c \in C$.

*Assumption 1:* There exists a positive number $\nu < 1$ such that for every given $i$ and $\lambda$,

$$\sup_{c,c' \in C} \sum_{y \in X} |P^l(y|c,i,\lambda) - P^l(y|c',i,\lambda)| \le 2\nu,$$

We give a performance bound of the rolling horizon policy in terms of span semi-norm; for a bounded function $V$ defined over $X \times I \times \Lambda$ and fixed $i \in I$ and $\lambda \in \Lambda$ (with abusing the notations), $\mathrm{sp}(V) = \sup_x V(x,i,\lambda) - \inf_x V(x,i,\lambda)$.

*Proposition 2:* Assume that the ergodicity condition 1 holds. For every given $i \in I$ and $\lambda \in \Lambda$ and a selected $h$ in $\{1,...,T\}$, define a lower level stationary policy $\pi^l$ as

$$\begin{aligned}
\phi^{i,\lambda}(x,i,\lambda) \in \arg\max_{a \in A}\Big(& R^l(x,a,i,\lambda) \\
&+ \sum_{y \in X} P^l(y|x,a,i,\lambda)\bar{R}^*_{h-1}(y,i,\lambda)\Big) \text{for all } x \in X.
\end{aligned}$$

Then, for all $i$ and $\lambda$,

$$\mathrm{sp}(\bar{R}^* - \bar{R}^{\pi^l}) \le T \cdot \frac{2\nu^{h-1}R_{\max}}{1-\nu} + \frac{2(\nu^h - \nu^T)R_{\max}}{(1-\nu)^2}$$

*Proof:* We begin with a slightly modified version of Theorem 4.8(a) [16] by Lemma below. See the proof there.

*Lemma 1:* Assume that the ergodicity condition 1 holds. For every given $i \in I$ and $\lambda \in \Lambda$ and $h = 1,...,T$, there exists a constant $j^*$ such that for all $x \in X$,

(a) $\quad \bar{R}^*_h(x,i,\lambda) - \bar{R}^*_{h-1}(x,i,\lambda) \ge \frac{-\nu^{h-1}R_{\max}}{1-\nu} + j^*$

(b) $\quad \bar{R}^*_h(x,i,\lambda) - \bar{R}^*_{h-1}(x,i,\lambda) \le \frac{\nu^{h-1}R_{\max}}{1-\nu} + j^*$

Fix $i$ and $\lambda$. Let $\rho_1 = \frac{-\nu^{h-1}R_{\max}}{1-\nu} + j^*$ and $\rho_2 = \frac{\nu^{h-1}R_{\max}}{1-\nu} + j^*$. With a similar reasoning in the proof of Proposition 1 and with the inequality in Lemma 1(a), we can deduce that for all $w = 0,1,...,T-1$ and for all $x \in X$,

$$\begin{aligned}
\bar{R}^*_h(x,i,\lambda) \quad \le \quad & E^x_{i,\lambda}\left[\sum_{t=0}^{w} R^l(x_t, \phi^{i,\lambda}(x_t,i,\lambda), i, \lambda)\right] \\
&+ E_{i,\lambda}[\bar{R}^*_h(x_{w+1},i,\lambda)] - (w+1)\rho_1.
\end{aligned}$$

We let $w = T-1$. It follows then that from the previous inequality,

$$\bar{R}^*_h(x,i,\lambda) \le \bar{R}^{\pi^l}(x,i,\lambda) + E_{i,\lambda}[\bar{R}^*_h(x_T,i,\lambda)] - T\rho_1.$$

By the same arguments, we have that

$$\bar{R}^*_h(x,i,\lambda) \ge \bar{R}^{\pi^l}(x,i,\lambda) + E_{i,\lambda}[\bar{R}^*_h(x_T,i,\lambda)] - T\rho_2.$$

Combining the above two inequalities, it follows that

$$\mathrm{sp}(\bar{R}^*_h - R^{\pi^l}) \le T(\rho_2 - \rho_1) = T \cdot \frac{2\nu^{h-1}R_{\max}}{1-\nu}. \qquad (13)$$

Now, from the span semi-norm contraction property of $\bar{\Omega}$ [16], we have that

$$\mathrm{sp}(\bar{R}^*_T - \bar{R}^*_h) \le \nu\, \mathrm{sp}(\bar{R}^*_{T-1} - \bar{R}^*_{h-1}) \le \cdots \le \nu^h\, \mathrm{sp}(\bar{R}^*_{T-h}). \quad (14)$$

From Lemma 1, we can also deduce that for all $x \in X$,

$$-\frac{R_{\max}(1-\nu^h)}{(1-\nu)^2} + hj^* \le \bar{R}^*_h(x,i,\lambda) \le \frac{R_{\max}(1-\nu^h)}{(1-\nu)^2} + hj^*.$$

Therefore, $\mathrm{sp}(\bar{R}^*_{T-h}) \le \frac{2R_{\max}(1-\nu^{T-h})}{(1-\nu)^2}$. Combining Equation (13) and (14) with the previous inequality, we have the desired result:

$$\mathrm{sp}(\bar{R}^* - \bar{R}^{\pi^l}) \le T \cdot \frac{2\nu^{h-1}R_{\max}}{1-\nu} + \frac{2(\nu^h - \nu^T)R_{\max}}{(1-\nu)^2}. \qquad (15)$$

■

We remark that the above result also gives a bound on the finite horizon *average* reward by dividing the both hand sides of Equation (15) by the horizon $T$. In particular, the result by letting $T \to \infty$ in this case does not coincide exactly with the result obtained in Theorem 5.1 in [17] — our result is loose by a factor of 2 in terms of span semi-norm even though the upper bound part in Theorem 5.1 would be the same. This is because the lower bound on the result of Theorem 5.1 is 0 incorporating the fact that the infinite horizon average reward of any stationary decision rule is no bigger than the optimal infinite horizon average reward, where we couldn't take advantage of the fact in our proof steps.

Suppose that we have a lower level policy dependent initialization function and we now know that the set of local optimal lower level policies that solve the lower level MDP problem for given $i \in I$ and $\lambda \in \Lambda$. As we can observe, a lower level decision rule determined from these policies does not necessarily achieve the optimal multi-level value because it is a locally optimal or greedy choice. However, solving the optimality equation given in Theorem 1, for example, is difficult because the size of the set $\Pi^l[i, \lambda]$ is often huge. We should somehow utilize the fact that we know the local optimal lower level policies. To illustrate this, we study the discounted case only. For this purpose, let $\Pi^*[i, \lambda]$ be the set of $\pi^l[i, \lambda]$'s that solve the lower level MDP problem for given $i \in I$ and $\lambda \in \Lambda$, i.e., achieving $R^*$. We then define a pair of upper and lower level decision rules, $\tilde{d}^u$ and $\tilde{d}^l$, from the arguments that achieve the following equation:

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

such that we set $\tilde{d}^u(x, i) = \tilde{\lambda}$ and set $d^l = \{\tilde{\pi}^l\}$, where $\tilde{\lambda}$ and $\tilde{\pi}^l[i, \tilde{\lambda}]$ are the arguments that achieve the above equation. We let the two-level value of following the pair of $\tilde{d}^u$ and $\tilde{d}^l$ be $\tilde{V}(x, i)$. It is left for the reader to check that for all $x$ and $i$ with $0 < \alpha < 1$,

$$0 \leq V^*(x, i) - \tilde{V}(x, i) \leq \frac{\gamma \mu R_{\max}(1 - \alpha^T)}{(1 - \gamma)(1 - \alpha)},$$

where $\mu$ is an ergodicity coefficient such that for any $x, x'$ and $i, i'$ and for any $\lambda, \lambda'$ and any $\pi, \pi' \in \Pi^l$,

$$\sum_{y \in X} \sum_{j \in I} \Bigg| \delta^{\pi}(x, i, \lambda)[y] P^u(j|i, \lambda) \\ - \delta^{\pi'}(x', i', \lambda')[y] P^u(j|i', \lambda') \Bigg| \leq 2\mu$$

with $0 < \mu < 1$. Note that we can define $\tilde{d}^u$ and $\tilde{d}^l$ with respect to a bounded value function $U$ that approximates $V^*$ and draw an error bound from $V^*$ by using the above result we just have drawn.

### C. Heuristic on-line methods

The discussion so far dealt with "off-line" methods for solving MMDPs. Even though various approximation/exact algorithms can be applied for some control problems, it will often require analyzing and utilizing certain structural properties on the problems, which might be very cumbersome in many interesting problems. In this section, we briefly discuss how to apply previously published two on-line (sampling-based) heuristic techniques in the context of MMDPs.

The first example approach called "(parallel) rollout" is based on the decision rule/policy improvement principle in the "policy iteration" algorithm (see, e.g., [3] [8] [9]). We simulate or rollout heuristic decision rule(s) available in on-line manner via Monte-Carlo simulation at each decision time and use the estimated value of following the heuristic decision rule(s) to create an (approximate) improved decision rule with respect to the heuristic decision rule(s). In particular, parallel rollout is useful if sample paths can be divided in a way that a particular heuristic decision rule is near-optimal for particular system trajectories. The parallel rollout method yields a decision rule that dynamically combines the multiple decision rules automatically adapting to different system trajectories and improves the performances of all of the heuristic decision rules.

We briefly discuss how to apply the rollout. Suppose that we have a heuristic decision rule pair of $d^l$ for the lower level and $d^u$ for the upper level. At each decision time $n$ (in the slow time-scale), we measure the utility of taking each candidate action $\lambda \in \Lambda$ as follows. We take a candidate action (in an imaginary sense) and then from the next step, we simulate $d^l$ and $d^u$ over a finite sampling horizon over many randomly simulated traces, giving the approximate value of following the decision rule pair. The single-step reward of taking action $\lambda$ associated with the lower level quasi-steady state performance is also estimated by simulation by following the decision rule $d^l$. The sum of the estimated single-step reward (plus the immediate reward of taking $\lambda$) plus the estimated value of following the decision rule pair $d^l$ and $d^u$ gives the utility measure of the candidate action $\lambda$. At each time $n$, we take the action with the highest utility measure. At the fast time-scale, we just follow the decision rule $d^l$.

The (parallel) rollout approach can be referred as a lower bound approach as the value of following any decision rule pair is a lower bound to the optimal value. On the other hand, the next example called "hindsight optimization" [10] is based on an upper bound. Hindsight optimization can be viewed as a heuristic method of adapting the (deterministic) optimal sample-path based solutions into an on-line solution. Instead of evaluating a decision rule pair by simulation as in the rollout, for each random trace of the system, the optimal *action sequence* that maximizes the reward sum is obtained. The average over many random traces will give an upper bound on the optimal value. We use the upper bound in the action utility measure. The hindsight optimization approach turns out to be effective in some problems (see, e.g., [7] [35]) even though the question of when this approach is useful is still open. However, we note that as long as the *ranking* of the utility measures of candidate actions reflects well the true ranking (especially the highest one), these heuristic methods can be expected to work well.

## V. Related Work

In this section, we compare several key papers that can be related with our work in hierarchical modeling. We first discuss a key paper by Sutton et al. [31] because the paper cites almost all of the hierarchical MDP works in (at least) artificial intelligence literature and some in the control literature and generalizes the previous works by one framework. For many interesting decision problems (e.g., queueing problems), the state spaces in different levels, ($X$ and $I$), are non-overlapping. Sutton's work considers a multi-time MDP model in the dimension of the action space only (action hierarchy) by defining "*options*" or "temporally extended" actions. The state spaces in different time-scales are the same in Sutton's model and the option *does not* determine or change the underlying reward or state transition structure. On the other hand, 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. We can roughly say that the lower level policies defined over different upper level state and controls are semi-Markov options [31] 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. [18]. 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 [27] studies a nonstationary MDP, where a variable called the system operating mode determines an evolution of the 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.

Even though the situation being considered is totally different, Pan and Basar's work [25] considers a class of differential games that exhibit possible multi-time scale separation. Given a problem defined in terms of a singularly perturbed differential equation, differently time-scaled games are identified and each game is solved *independently* and from this a composite solution is developed, which is an approximate solution for the original problem. In our model, the upper level MDP solution must depend on the solution for the lower level MDPs. Finally, as we mentioned before, we can view our model as an MDP-based extension or a generalization of Trivedi's hierarchical performability and dependability model. In Trivedi's work, the performance models (fast time-scale model) are *solved* to obtain performance measures (in our model, this measure corresponds to the function value of $R^*$ with the lower level policy independent $\delta$-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 network can be used). We can see that if we fix the upper level and the lower level decision rules in our model with the lower level policy independent $\delta$-function, an MMDP becomes (roughly) the model described by Trivedi — in our model, the lower level model is also a Markov reward/decision process.

## VI. 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 model for the case where a lower level action affects the higher level transitions (in a different time-scale) is still an open problem.

An extension of our model into a *partially observable* MMDP is straightforward because a partially observable MDP can be transformed into an MDP with information state space (see, e.g., [1]). We restricted the MMDP formulation to discrete-time domain in the present paper. Extending the model into a continuous-time domain in parallel to semi-MDP would not be difficult, where in particular, in this case the decision epoch $T$ at the upper level would be a bounded random variable.

Finally, it would be interesting to extend our model into the Markov game settings making multi-time scaled Markov games. The "optimal equilibrium value of game" over a finite horizon at the lower level game will be used as one-step cost/reward for the upper level game.

## 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. Control and Optimization*, 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] D. P. Bertsekas and D. A. Castanon, "Rollout algorithms for stochastic scheduling problems," *J. of Heuristics*, vol. 5, pp. 89–108, 1999.

[4] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming.* Athena Scientific, 1996.

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

[6] H. S. Chang, M. Fu, and S. I. Marcus, "An adaptive sampling algorithm for solving Markov decision processes," TR 2002-19, ISR, Univ. of Maryland, 2002.

[7] H. S. Chang, R. Givan, and E. K. P. Chong, "On-line scheduling via sampling," in *Proc. 5th Int. Conf. on Artificial Intelligence Planning and Scheduling*, 2000, pp. 62–71.

[8] H. S. Chang, R. Givan, and E. K. P. Chong, "Parallel rollout for online solution of partially observable Markov decision processes," submitted to *Discrete Event Dynamic Systems: Theory and Application*, 2002.

[9] H. S. Chang and S. I. Marcus, "Approximate receding horizon control for Markov decision processes: average reward case," TR 2001-46, ISR, Univ. of Maryland, 2001.

[10] E. K. P. Chong, R. Givan, and H. S. Chang, "A framework for simulation-based network control via hindsight optimization," in *Proc. 39th IEEE CDC*, 2000, pp. 1433–1438.

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

[12] J. Forestier and P. Varaiya, "Multilayer control of large Markov chains," *IEEE Trans. on Automatic Control*, vol. AC-23, No. 2, pp. 298–304, 1978.

[13] 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.

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

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

[16] O. Hernández-Lerma, *Adaptive Markov Control Processes.* Springer-Verlag, 1989.

[17] O. Hernández-Lerma and J. B. Lasserre, "Error bounds for rolling horizon policies in discrete-time Markov control processes," *IEEE Trans. on Automatic Control*, vol. 35, no. 10, pp. 1118–1124, 1990.

[18] M. Jacobson, N. Shimkin and A. Shwartz, "Piecewise stationary Markov Decision Processes, I: constant gain," submitted to *Mathematics of Operations Research*.

[19] M. Kearns, Y. Mansour, and A. Y. Ng, "A sparse sampling algorithm for near-optimal planning in large Markov decision processes," in *Proc. 16th International Joint Conf. on Artificial Intelligence*, 1999, pp. 1224–1231.

[20] M. Littman, T. Dean, and L. Kaelbling, "On the complexity of solving Markov decision problems," in *Proc. 11th Annual Conf. on Uncertainty in Artificial Intelligence*, 1995, pp. 394–402.

[21] 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.

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

[23] J. Muppala, M. Malhotra, and K. Trivedi, "Markov dependability models of complex systems: analysis techniques," *Reliability and Maintenance of Complex Systems*, S. Ozekici (ed.), pp. 442–486, Springer-Verlag, Berlin, 1996.

[24] A. Müller, "How does the value function of a Markov decision process depend on the transition probabilities?," *Math. of Operations Research*, vol. 22, no. 4, pp. 872–885, 1997.

[25] Z. Pan and T. Basar, “Multi-time scale zero-sum differential games with perfect state measurements,” *Dynamics and Control*, vol. 5, pp. 7–30, 1995.

[26] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.

[27] 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.

[28] 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.

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

[30] S. Singh and R. Yee, “An upper bound on the loss from approximate optimal-value functions,” *Machine Learning*, vol. 16, pp. 227–233, 1994.

[31] 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.

[32] D. Tse, R.G. Gallager and J.N. Tsitsiklis, “Statistical multiplexing of multiple time-scale Markov streams,” *IEEE J. on Selected Areas in Communications*, vol. 13, no. 6, pp. 1028–1039, 1995.

[33] T. Tuan and K. Park, “Multiple time scale congestion control for self-similar network traffic,” *Performance Evaluation*, vol. 36-37, pp. 359–386, 1999.

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

[35] G. Wu, E. K. P. Chong, and R. Givan, “Congestion control via online sampling,” in *Proc. of INFOCOM*, 2001, pp. 1271–1280.

PLACE
PHOTO
HERE

**Hyeong Soo Chang**

PLACE
PHOTO
HERE

**Pedram Fard**

PLACE
PHOTO
HERE

**Steven I. Marcus**

PLACE
PHOTO
HERE

**Mark Shayman**