Asynchronous Control for Coupled Markov Decision Systems

Michael J. Neely
University of Southern California

***Abstract*— This paper considers optimal control for a collection of separate Markov decision systems that operate asynchronously over their own state spaces. Decisions at each system affect: (i) the time spent in the current state, (ii) a vector of penalties incurred, and (iii) the next-state transition probabilities. An example is a network of smart devices that perform separate tasks but share a common wireless channel. The model can also be applied to data center scheduling and to various types of cyber-physical networks. The combined state space grows exponentially with the number of systems. However, a simple strategy is developed where each system makes separate decisions. Total complexity grows only linearly in the number of systems, and the resulting performance can be pushed arbitrarily close to optimal.**

I. Introduction

This paper considers control for a collection of coupled systems. Each system is a semi-Markov decision process that operates in continuous time over its own state space. Decisions at each system affect the time spent in each state, the transition probabilities to the next state, and a vector of penalties or rewards. The systems are coupled through constraints on the sum of time averages of their penalties and rewards.

An example is a collection of smart devices that repeatedly perform complex tasks such as image or video processing, compression, or other types of computation. These tasks may also generate or request data for wireless transmission. Each device has a state space that corresponds to different task functions and/or different energy saving modes of operation. Decisions in each state affect energy expenditure, computation time, and the amount of data generated or requested for wireless communication. The state transition times are not synchronized across devices. Further, the devices are coupled through the multi-access constraints of the wireless network. This presents a challenging and important problem of asynchronous control of coupled Markov decision systems. Such problems also arise in data center scheduling and in control of cyber-physical networks.

This paper demonstrates that optimality can be achieved by separate controllers at each system. While the size of the combined state space vector grows exponentially in the number of systems, the solution complexity grows only linearly. Indeed, the complexity of the controller at each system depends on the size of its own state space. Thus, the solution can be used even when the number of systems is large, say, 100 or 1000, provided that the state space of each system is small.

In Section IV a nonlinear program for the optimal control policy is derived. The problem is non-convex and has fractional terms with different denominators. This is more complex than a linear program or a linear fractional program. General problems of this type are intractable. However, the problem under study has special structure that allows an optimal solution. It is shown to be equivalent to a linear program via a nonlinear change of variables. This change of variables is inspired by techniques used in [1][2] to solve linear fractional programs associated with (single) unconstrained semi-Markov decision systems. The current work can be viewed as a generalization of [1][2] to the case of multiple asynchronous systems with multiple coupled constraints.

The linear programming formulation assumes all underlying probabilities of the system are known. Section V treats a more complex scenario where each system can observe a vector of random events with possibly unknown probability distribution (such as a vector of wireless channel states used for opportunistic transmission). Learning-based approaches to discrete time Markov decision problems are considered in [3] using a 2-timescale analysis and in [4] using policy gradients. The current paper takes a different approach that utilizes Lyapunov optimization theory. It builds on the Lyapunov method for optimizing renewal systems in [5] and semi-Markov decision systems in [6]. The result in [6] treats a single Markov system and uses a more complex bisection routine to evaluate a drift-plus-penalty ratio expression. The current paper uses a change of variables that results in a drift-plus-penalty expression without a ratio, and hence does not require a bisection step. The current paper is also related to recent work in [7] that treats asynchronous scheduling at a data center. The work in [7] develops an online policy for asynchronous control, but treats a simpler class of systems that do not have an embedded Markov structure.

II. System Model

Consider a collection of S separate Markovian systems, where S is a positive integer. Define $\mathcal{S} \triangleq \{1, \ldots, S\}$. Each system $s \in \mathcal{S}$ has a finite state space $\mathcal{K}^{(s)}$ and operates in continuous time. The timeline for each system is segmented into back-to-back intervals called *frames*. Each frame represents the time spent in one state. The size of each frame can vary depending on random events and control actions. Let $\{T^{(s)}[r]\}_{r=0}^{\infty}$ be the sequence of frame sizes for system s, where r is a frame index in the set $\{0, 1, 2, \ldots\}$. Frame boundaries are not necessarily synchronized across systems.

Let $k^{(s)}[r]$ be the state of system s during frame r. At the beginning of each frame r, the system observes a *random event* $\omega^{(s)}[r]$ that takes values in some abstract event space $\Omega^{(s)}$. It then chooses a *control action* $\alpha^{(s)}[r] \in \mathcal{A}^{(s)}$, where $\mathcal{A}^{(s)}$ is

This material is supported in part by one or more of: the NSF Career grant CCF-0747525, the Network Science Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory W911NF-09-2-0053.

an abstract set of possible actions for system s. The 3-tuple $(k^{(s)}[r], \omega^{(s)}[r], \alpha^{(s)}[r])$ determines:

- The frame size $T^{(s)}[r]$.
- A vector of $L+1$ penalties for frame r, for some non-negative integer L. This penalty vector has the form:

$$\boldsymbol{y}^{(s)}[r] = (y_0^{(s)}[r], y_1^{(s)}[r], \ldots, y_L^{(s)}[r])$$

- The next-state transition probabilities $P_{ij}^{(s)}[r]$ (assuming that $i = k^{(s)}[r]$ is the current state for system s).

These are given by functions $\hat{T}^{(s)}(\cdot)$, $\hat{y}_l^{(s)}(\cdot)$, $\hat{P}_{ij}^{(s)}(\cdot)$:

$$\begin{aligned} T^{(s)}[r] &= \hat{T}^{(s)}(k^{(s)}[r], \omega^{(s)}[r], \alpha^{(s)}[r]) \\ y_l^{(s)}[r] &= \hat{y}_l^{(s)}(k^{(s)}[r], \omega^{(s)}[r], \alpha^{(s)}[r]) \ \forall l \in \{0, 1, \ldots, L\} \\ P_{i,j}^{(s)}[r] &= \hat{P}_{i,j}^{(s)}(\omega^{(s)}[r], \alpha^{(s)}[r]) \ \forall i, j \in \mathcal{K}^{(s)} \end{aligned}$$

A. Assumptions

For simplicity of exposition, assume that for each $s \in \mathcal{S}$, the sets $\mathcal{A}^{(s)}$ and $\Omega^{(s)}$ are finite. Assume that the $\omega^{(s)}[r]$ processes are independent across systems. Further, for each system $s \in \mathcal{S}$, the processes $\{\omega^{(s)}[r]\}_{r=0}^{\infty}$ are independent and identically distributed (i.i.d.) across frames $r \in \{0, 1, 2, \ldots\}$. For each $\omega \in \Omega^{(s)}$, define $\pi^{(s)}(\omega) \triangleq Pr[\omega^{(s)}[r] = \omega]$.

The transition probabilities are non-negative and satisfy the following for all $(k^{(s)}[r], \omega^{(s)}[r], \alpha^{(s)}[r])$:

$$\textstyle\sum_{j \in \mathcal{K}^{(s)}} \hat{P}_{ij}^{(s)}(\cdot) = 1 \quad \forall s \in \mathcal{S}, \forall i \in \mathcal{K}^{(s)}$$

The frame sizes are assumed to be bounded by some positive minimum and maximum values $T_{min}^{(s)}$ and $T_{max}^{(s)}$ for all $(k^{(s)}[r], \omega^{(s)}[r], \alpha^{(s)}[r])$:

$$T_{min}^{(s)} \leq \hat{T}^{(s)}(\cdot) \leq T_{max}^{(s)}$$

The penalties can be positive, negative, or zero (negative penalties can be used to represent rewards), and are bounded by some finite minimum and maximum values $y_{l,min}^{(s)}, y_{l,max}^{(s)}$ for all $(k^{(s)}[r], \omega^{(s)}[r], \alpha^{(s)}[r])$:

$$y_{l,min}^{(s)} \leq \hat{y}_l^{(s)}(\cdot) \leq y_{l,max}^{(s)}$$

B. Optimization Objective

The time average penalty of type $l \in \{0, 1, \ldots, L\}$ incurred by system s up to frame R is given by:

$$\frac{\sum_{r=0}^{R-1} y_l^{(s)}[r]}{\sum_{r=0}^{R-1} T^{(s)}[r]}$$

Multiplying the numerator and denominator of the above expression by $1/R$ and taking a limit as $R \to \infty$ gives an expression for the *time average penalty* of type l in system s:

$$\frac{\overline{y}_l^{(s)}}{\overline{T}^{(s)}} = \lim_{R \to \infty} \frac{\frac{1}{R}\sum_{r=0}^{R-1} y_l^{(s)}[r]}{\frac{1}{R}\sum_{r=0}^{R-1} T^{(s)}[r]}$$

where $\overline{y}_l^{(s)}$ is a *frame average* that is defined:

$$\overline{y}_l^{(s)} \triangleq \lim_{R \to \infty} \tfrac{1}{R} \textstyle\sum_{r=0}^{R-1} y_l^{(s)}[r]$$

and $\overline{T}_l^{(s)}$ is defined similarly.

At the beginning of the rth frame for system s, the controller observes the random event $\omega^{(s)}[r]$ and chooses an action $\alpha^{(s)}[r] \in \mathcal{A}^{(s)}$. The goal is to design decision-making policies for each system so that the resulting time averages solve the following optimization problem:

$$\text{Minimize:} \quad \textstyle\sum_{s \in \mathcal{S}} \frac{\overline{y}_0^{(s)}}{\overline{T}^{(s)}} \tag{1}$$

$$\text{Subject to:} \quad \textstyle\sum_{s \in \mathcal{S}} c_l^{(s)} \frac{\overline{y}_l^{(s)}}{\overline{T}^{(s)}} \leq d_l \ \ \forall l \in \{1, \ldots, L\} \tag{2}$$

$$\alpha^{(s)}[r] \in \mathcal{A}^{(s)} \ \ \forall s \in \mathcal{S}, \forall r \in \{0, 1, 2, \ldots\} \tag{3}$$

where $c_l^{(s)}$, d_l are given real numbers for $l \in \{1, \ldots, L\}$ and $s \in \mathcal{S}$. It is assumed throughout that the constraints of problem (1)-(3) are feasible.

For simplicity, it is assumed that each system $s \in \mathcal{S}$ has a state $0 \in \mathcal{K}^{(s)}$ that is positive recurrent under any stationary policy for choosing $\alpha^{(s)}[r]$. This occurs, for example, when each state has a positive probability of returning to state 0 under any $(\omega^{(s)}[r], \alpha^{(s)}[r])$. This assumption is not crucial, but simplifies some technical details. In particular, it can be shown that it ensures the initial states of the system do not affect optimality. Such a state 0 often naturally exists when systems have an *idle state* that is returned to infinitely often.

III. An Example Network of Smart Devices

Consider a network of M wireless smart devices. Each device contains two embedded chips: a *processing chip* and a *communication chip*. The processing chip operates over variable length frames and is used for computation and task processing. The communication chip operates over fixed frame sizes and is used for wireless transmission and reception over one of L possible transmission links.

The processing chip at each device $m \in \{1, \ldots, M\}$ is assumed to have three states:

$$\mathcal{K}^{(m)} = \{\text{idle, processing mode 1, processing mode 2}\}$$

The different states can represent different functionalities or tasks that the chip performs, and/or different energy-saving modes that affect computation time and energy expenditure.

Let $\mathcal{A}^{(m)}$ be an abstract space of processing actions for each device $m \in \{1, \ldots, M\}$. For simplicity, assume there is no random event process $\omega^{(m)}[r]$ for these chips. The action $\alpha^{(m)}[r]$ at device m affects the energy expenditure $e^{(m)}[r]$, the frame duration $T^{(m)}[r]$, transition probabilities to the next state, and generates $b_l^{(m)}[r]$ bits for transmission over link l:

$$\begin{aligned} e^{(m)}[r] &= \hat{e}^{(m)}(k^{(m)}[r], \alpha^{(m)}[r]) \\ T^{(m)}[r] &= \hat{T}^{(m)}(k^{(m)}[r], \alpha^{(m)}[r]) \\ b_l^{(m)}[r] &= \hat{b}_l^{(m)}(k^{(m)}[r], \alpha^{(m)}[r]) \ \ \forall l \in \{1, \ldots, L\} \end{aligned}$$

Finally, define an $(M+1)th$ system that represents all of the L wireless links. This system operates in discrete time with fixed frame sizes $T^{(M+1)}[r] = 1$ for all $r \in \{0, 1, 2, ...\}$, and has only one Markov state $k^{(M+1)}[r] = 0$ for all $r \in \{0, 1, 2, \ldots\}$ (so that system $M+1$ has no Markov dynamics). However, this system has a time-varying *channel state process* $\omega^{(M+1)}[r] = (\eta_1[r], \ldots, \eta_L[r])$, where $\eta_l[r]$ represents the state of wireless channel l on frame r. Let $\mathcal{A}^{(M+1)}$ represent

the set of transmission/reception control actions on each frame (for example, this set might restrict the network to transmit over only one link per frame). Let $e^{(M+1)}[r]$ and $\mu_l[r]$ be the energy expended and bits transmitted over link l on frame r:

$$\begin{aligned} e^{(M+1)}[r] &= \hat{e}^{(M+1)}(\omega^{(M+1)}[r], \alpha^{(M+1)}[r]) \\ \mu_l[r] &= \hat{\mu}_l(\omega^{(M+1)}[r], \alpha^{(M+1)}[r]) \ \ \forall l \in \{1, \ldots, L\} \end{aligned}$$

The goal is to operate each system to minimize total average power expenditure subject to transmission rate constraints:

$$\text{Minimize:} \quad \overline{e}^{(M+1)} + \sum_{m=1}^{M} \frac{\overline{e}^{(m)}}{\overline{T}^{(m)}} \tag{4}$$

$$\text{Subject to:} \quad \sum_{m=1}^{M} \frac{\overline{b}_l^{(m)}}{\overline{T}^{(m)}} \leq \overline{\mu}_l \ \ \forall l \in \{1, \ldots, L\} \tag{5}$$

$$\alpha^{(m)}[r] \in \mathcal{A}^{(m)} \tag{6}$$

where the final constraint $\alpha^{(m)}[r] \in \mathcal{A}^{(m)}$ holds for all $m \in \{1, \ldots, M+1\}$ and all $r \in \{0, 1, 2, \ldots\}$.

IV. The Nonlinear Program Transformed

To begin, first assume there are no random event processes $\omega^{(s)}[r]$. It can be shown that the problem (1)-(3) can be solved by *stationary and randomized* algorithms (see related results in [8][9][2]). Specifically, each system $s \in \mathcal{S}$ observes its current state $k^{(s)}[r]$ and independently chooses a control action $\alpha^{(s)}[r]$ according to a probability distribution $p^{(s)}(\alpha)$:

$$Pr\left[\alpha^{(s)}[r] = \alpha | k^{(s)}[r] = k\right] = p^{(s)}(\alpha|k)$$

The $p^{(s)}(\alpha|k)$ probabilities are non-negative and sum to 1:

$$\sum_{\alpha \in \mathcal{A}^{(s)}} p^{(s)}(\alpha|k) = 1 \ \ \forall k \in \mathcal{K}^{(s)}$$

The fraction of frames that system s spends in each state under this policy can be viewed as a "steady state" distribution that satisfies a global balance equation. A standard trick is to define variables $\phi^{(s)}(k, \alpha)$ that intuitively represent the steady state probability that system s is in state k and chooses action α. They should satisfy (see, for example, [10][8][9][2]):

$$\sum_{\alpha \in \mathcal{A}^{(s)}} \phi^{(s)}(k, \alpha) = \sum_{i \in \mathcal{K}^{(s)}, \alpha \in \mathcal{A}^{(s)}} \phi^{(s)}(i, \alpha) \hat{P}_{ik}^{(s)}(\alpha) \tag{7}$$

$$\phi^{(s)}(k, \alpha) \geq 0 \tag{8}$$

$$\sum_{k \in \mathcal{K}^{(s)}, \alpha \in \mathcal{A}^{(s)}} \phi^{(s)}(k, \alpha) = 1 \tag{9}$$

where (7) is for all $k \in \mathcal{K}^{(s)}$, and (8) is for all $k \in \mathcal{K}^{(s)}$, $\alpha \in \mathcal{A}^{(s)}$. Constraint (7) can be interpreted as a balance equation. Its left-hand-side represents the steady state probability that system s is in state k. Its right-hand-side represents the probability of transitioning into state k in the next frame. It should be noted that this "steady state" is with respect to *frame averages* (corresponding to the steady state of the embedded Markov chain), and is not the same as the *time average* steady state (which would also include the time spent in each state).

Given values $\phi^{(s)}(k, \alpha)$ that satisfy (7)-(9), one can define a stationary randomized policy by:

$$p^{(s)}(\alpha|k) = \frac{\phi^{(s)}(k, \alpha)}{\sum_{\beta \in \mathcal{A}^{(s)}} \phi^{(s)}(k, \beta)}$$

This gives rise to the following nonlinear program for computing the optimal stationary policy for problem (1)-(3):

$$\text{Minimize:} \quad \sum_{s \in \mathcal{S}} \left[\frac{\sum_{k,\alpha} \phi^{(s)}(k,\alpha) \hat{y}_0^{(s)}(k,\alpha)}{\sum_{k,\alpha} \phi^{(s)}(k,\alpha) \hat{T}^{(s)}(k,\alpha)} \right] \tag{10}$$

$$\text{Subject to:} \quad \sum_{s \in \mathcal{S}} c_l^{(s)} \left[\frac{\sum_{k,\alpha} \phi^{(s)}(k,\alpha) \hat{y}_l^{(s)}(k,\alpha)}{\sum_{k,\alpha} \phi^{(s)}(k,\alpha) \hat{T}^{(s)}(k,\alpha)} \right] \leq d_l \ \ \forall l \in \{1, \ldots, L\} \tag{11}$$

$$\phi^{(s)}(k, \alpha) \text{ satisfies (7)-(9)} \tag{12}$$

where the summations $\sum_{k,\alpha}$ above are understood to be over $k \in \mathcal{K}^{(s)}$, $\alpha \in \mathcal{A}^{(s)}$. The above problem has variables $\phi^{(s)}(k, \alpha)$ and constants $c_l^{(s)}$, d_l, $\hat{y}_l^{(s)}(k, \alpha)$, $\hat{T}^{(s)}(k, \alpha)$. The constraints (7)-(9) are linear in the variables $\phi^{(s)}(k, \alpha)$. The problem also involves fractional terms where the numerators and denominators are linear functions of the variables $\phi^{(s)}(k, \alpha)$. Problems with fractional terms with different denominators are non-convex and are generally intractable. However, all fractional terms in the problem above have the same denominator for each system $s \in \mathcal{S}$. This property is exploited in the first result below, which transforms the problem via a nonlinear change of variables. This change of variables is inspired by similar techniques in [1][2] which treat (single) unconstrained semi-Markov systems.

Consider the following linear program defined over new variables $\gamma^{(s)}(k, \alpha)$ for $s \in \mathcal{S}$, $k \in \mathcal{K}^{(s)}$, $\alpha \in \mathcal{A}^{(s)}$:

$$\text{Minimize:} \quad \sum_{s \in \mathcal{S}} \sum_{k,\alpha} \gamma^{(s)}(k, \alpha) \hat{y}_0^{(s)}(k, \alpha) \tag{13}$$

$$\text{Subject to:} \quad \sum_{s \in \mathcal{S}} \sum_{k,\alpha} \gamma^{(s)}(k, \alpha) c_l^{(s)} \hat{y}_l^{(s)}(k, \alpha) \leq d_l \ \ \forall l \in \{1, \ldots, L\} \tag{14}$$

$$\sum_{\alpha} \gamma^{(s)}(k, \alpha) = \sum_{i, \alpha} \gamma^{(s)}(i, \alpha) \hat{P}_{ik}^{(s)}(\alpha) \tag{15}$$

$$\gamma^{(s)}(k, \alpha) \geq 0 \tag{16}$$

$$\sum_{k,\alpha} \gamma^{(s)}(k, \alpha) \hat{T}^{(s)}(k, \alpha) = 1 \tag{17}$$

where summations $\sum_{\alpha}$ and $\sum_{k,\alpha}$ are understood to be over $\alpha \in \mathcal{A}^{(s)}$ and $k \in \mathcal{K}^{(s)}$. The constraints (15) are for all $s \in \mathcal{S}$, $k \in \mathcal{K}^{(s)}$, the constraints (16) are for all $s \in \mathcal{S}$, $k \in \mathcal{K}^{(s)}$, $\alpha \in \mathcal{A}^{(s)}$, and the constraints (17) are for all $s \in \mathcal{S}$.

Theorem 1: The optimal objective function value is the same for the original problem (10)-(12) and the new problem (13)-(17). Further, if $\gamma^{(s)}(k, \alpha)$ are variables that solve the new problem, then the following variables $\phi^{(s)}(k, \alpha)$ solve the original problem:

$$\phi^{(s)}(k, \alpha) = \frac{\gamma^{(s)}(k, \alpha)}{\sum_{i \in \mathcal{K}^{(s)}, \beta \in \mathcal{K}^{(s)}} \gamma^{(s)}(i, \beta)} \tag{18}$$

Proof: Let $\phi^{(s)}(k, \alpha)$ be values that solve the original problem (10)-(12), and let $V_{original}$ be the value of the optimal objective function:

$$V_{original} = \sum_{s \in \mathcal{S}} \left[\frac{\sum_{k,\alpha} \phi^{(s)}(k,\alpha) \hat{y}_0^{(s)}(k,\alpha)}{\sum_{k,\alpha} \phi^{(s)}(k,\alpha) \hat{T}^{(s)}(k,\alpha)} \right] \tag{19}$$

Define:

$$\gamma^{(s)}(k, \alpha) = \frac{\phi^{(s)}(k, \alpha)}{\sum_{i \in \mathcal{K}^{(s)}, \beta \in \mathcal{A}^{(s)}} \phi^{(s)}(i, \beta) \hat{T}^{(s)}(i, \beta)} \tag{20}$$

and note that because the $\hat{T}^{(s)}(k,\alpha)$ values are strictly positive and the $\phi^{(s)}(k,\alpha)$ values are non-negative and sum to 1, the denominator in (20) must be positive. Because the $\phi^{(s)}(k,\alpha)$ values satisfy the constraints (11)-(12), it can be shown that the $\gamma^{(s)}(k,\alpha)$ values defined by (20) satisfy the constraints (14)-(17). Indeed, the definition of $\gamma^{(s)}(k,\alpha)$ in (20) immediately implies constraint (17), non-negativity of $\phi^{(s)}(k,\alpha)$ immediately implies (16), and dividing the constraint (7) by $\sum_{i\in\mathcal{K}^{(s)},\beta\in\mathcal{A}^{(s)}}\phi^{(s)}(i,\beta)\hat{T}^{(s)}(i,\beta)$ implies (15). Finally, substituting (20) into (11) and using (17) implies constraint (14). Further, by substituting (20) into (19) it is easy to see that the objective function associated with these $\gamma^{(s)}(k,\alpha)$ variables is equal to $V_{original}$. It follows that the optimal objective function value of the new problem is less than or equal to $V_{original}$, that is, $V_{new}\leq V_{original}$, where V_{new} is defined as the minimum objective function value (13) for the new problem.

Now let $\gamma^{(s)}(k,\alpha)$ represent optimal variables that solve the new problem (13)-(17), and define $\phi^{(s)}(k,\alpha)$ according to (18). By similar substitutions, it can be seen that these $\phi^{(s)}(k,\alpha)$ values satisfy the constraints (11)-(12) of the original problem and produce an objective function value in (10) that is equal to V_{new}. Hence, $V_{new}=V_{original}$, and these $\phi^{(s)}(k,\alpha)$ values are optimal for the original problem. □

Theorem 1 transforms the original nonlinear problem into a linear program with variables $\gamma^{(s)}(k,\alpha)$. Recall that there are S systems. Suppose each system has at most K_{max} states and an action space size of at most A_{max}, for some positive numbers K_{max} and A_{max}. Thus, the total number of variables $\gamma^{(s)}(k,\alpha)$ is at most $SK_{max}A_{max}$, which grows *linearly* in the number of systems. It is easy to see that the number of constraints of the linear program (13)-(17) also grows linearly in the number of systems. The total complexity is essentially the same as the complexity associated with each system separately solving its own Markov decision problem on its own state space.

V. Lyapunov Optimization

The previous section solves for the optimal conditional probabilities $p^{(s)}(\alpha|k)$, but does not treat cases when there are observed random events $\omega^{(s)}[r]$. For such cases, one needs conditional probabilities $p^{(s)}(\alpha|\omega,k)$. The number of ω vectors can be enormous, in which case it is not practical to consider estimating the probabilities of each and computing the optimal $p^{(s)}(\alpha|\omega,k)$ probabilities. However, Lyapunov optimization can treat related problems of optimizing time averages in systems with random events, without knowing the probabilities of these events and regardless of the cardinality of the event space [5][11][12][13]. Rather than attempting to compute the optimal probabilities for every possible event, the Lyapunov policies make online decisions based on greedily minimizing a drift-plus-penalty expression. Recent work in [6] extends this by developing an online policy for a (single) semi-Markov decision system, provided that certain target information is given.

Specifically, suppose that for each system $s\in\mathcal{S}$, one is given values $P_{ij}^{*(s)}$, $y_{l,k}^{*(s)}$, $T_k^{*(s)}$ that respectively represent desired targets for the fraction of time the embedded Markov chain transitions from i to j, the average type l penalties incurred while in state k, and the average time spent in state k. Then one can use the online policy of Section IV in [6] to control the system and meet these targets, without requiring the probability distribution for the random events $\omega^{(s)}[r]$. In the following, a Lyapunov-based algorithm for computing the optimal targets corresponding to the asynchronous control problem (1)-(3) is developed.

A. The Time Average Problem

As in [6], consider a modified collection of systems with no Markov dynamics, where "state variables" $k^{(s)}[r]$ for system s can be chosen as decision variables every frame r. Define the following attributes $q_{ij}^{(s)}[r]$ for all $s\in\mathcal{S}$ and $i,j\in\mathcal{K}^{(s)}$:

$$q_{ij}^{(s)}[r]=1_i^{(s)}[r]\hat{P}_{ij}^{(s)}(\omega^{(s)}[r],\alpha^{(s)}[r]) \tag{21}$$

where $1_i^{(s)}[r]$ is an indicator function that is 1 if $k^{(s)}[r]=i$, and 0 else. Let $\overline{1}_i^{(s)}$ be its frame average. The problem (1)-(3) can be transformed as (compare with Section III in [6]):

$$\text{Minimize:}\quad \textstyle\sum_{s\in\mathcal{S}}\frac{\overline{y}_0^{(s)}}{\overline{T}^{(s)}} \tag{22}$$

$$\text{Subject to:}\quad \textstyle\sum_{s\in\mathcal{S}}c_l^{(s)}\frac{\overline{y}_l^{(s)}}{\overline{T}^{(s)}}\leq d_l\ \forall l\in\{1,\ldots,L\} \tag{23}$$

$$\overline{1}_k^{(s)}=\textstyle\sum_{i\in\mathcal{K}^{(s)}}\overline{q}_{ik}^{(s)} \tag{24}$$

$$k^{(s)}[r]\in\mathcal{K}^{(s)} \tag{25}$$

$$\alpha^{(s)}[r]\in\mathcal{A}^{(s)} \tag{26}$$

$$k^{(s)}[r]\text{ is chosen before }\omega^{(s)}[r]\text{ is known} \tag{27}$$

where (24) holds for all $s\in\mathcal{S}$, $k\in\mathcal{K}^{(s)}$, and (25)-(27) hold for all $s\in\mathcal{S}$, $r\in\{0,1,2,\ldots\}$. The objective function (22) is identical to (1), and the constraints (23) and (26) are the same as (2)-(3). Constraint (24) is a balance equation similar to (7) and, together with (25) and (27), ensures the resulting time averages can actually be achieved on the Markov decision system. Constraint (27) is subtle, and ensures the decisions $k^{(s)}[r]$ are independent of $\omega^{(s)}[r]$.

Now consider the following transformed problem, similar in spirit to the transformation of the previous section: For each system $s\in\mathcal{S}$, define new variables $\theta^{(s)}[r]$ that are chosen every frame $r\in\{0,1,2,\ldots\}$ over the interval $[1/T_{max}^{(s)},1/T_{min}^{(s)}]$. Consider the problem:

$$\text{Minimize:}\quad \textstyle\sum_{s\in\mathcal{S}}\overline{\theta^{(s)}y_0^{(s)}} \tag{28}$$

$$\text{Subject to:}\quad \textstyle\sum_{s\in\mathcal{S}}c_l^{(s)}\overline{\theta^{(s)}y_l^{(s)}}\leq d_l\ \forall l\in\{1,\ldots,L\} \tag{29}$$

$$\overline{\theta^{(s)}1_k^{(s)}}=\textstyle\sum_{i\in\mathcal{K}^{(s)}}\overline{\theta^{(s)}q_{ik}^{(s)}} \tag{30}$$

$$\overline{\theta^{(s)}T^{(s)}}=1 \tag{31}$$

$$k^{(s)}[r]\in\mathcal{K}^{(s)} \tag{32}$$

$$\alpha^{(s)}[r]\ \in\mathcal{A}^{(s)} \tag{33}$$

$$1/T_{max}^{(s)}\leq\theta^{(s)}[r]\leq 1/T_{min}^{(s)} \tag{34}$$

$$k^{(s)}[r]\text{ is chosen before }\omega^{(s)}[r]\text{ is known} \tag{35}$$

where frame averages $\overline{\theta^{(s)}y_l^{(s)}}$ are defined:

$$\overline{\theta^{(s)}y_l^{(s)}}\triangleq\lim_{R\to\infty}\tfrac{1}{R}\textstyle\sum_{r=0}^{R-1}\theta^{(s)}[r]y_l^{(s)}[r]$$

and frame averages $\overline{\theta^{(s)}T^{(s)}}$, $\overline{\theta^{(s)}q_{ik}^{(s)}}$, $\overline{\theta^{(s)}1_k^{(s)}}$ are defined similarly. It can be shown that the original problem (22)-(27) and the new problem (28)-(35) have the same optimal objective function value (proof omitted for brevity). Further, the solution to the new problem can be used to construct optimal targets $P_{ij}^{*(s)}$, $y_{l,k}^{*(s)}$, $T_k^{*(s)}$ for the original problem as follows:

$$y_{l,k}^{*(s)} = \frac{\overline{\theta^{(s)}1_k^{(s)}y_l^{(s)}}}{\overline{\theta^{(s)}1_k^{(s)}}} \ , \ T_k^{*(s)} = \frac{\overline{\theta^{(s)}1_k^{(s)}T^{(s)}}}{\overline{\theta^{(s)}1_k^{(s)}}} \ , \ P_{ij}^{*(s)} = \frac{\overline{\theta^{(s)}q_{ij}^{(s)}}}{\overline{\theta^{(s)}1_i^{(s)}}}$$

B. Virtual Queues

Using the drift-plus-penalty technique of [5], the constraints (29)-(31) are treated with virtual queues $Z_l[r]$, $H_k^{(s)}[r]$, $J^{(s)}[r]$ for $l \in \{1, \ldots, L\}$, $s \in \mathcal{S}$, $k \in \mathcal{K}^{(s)}$:

$$\begin{aligned} Z_l[r+1] &= \max\left[Z_l[r] + \sum_{s\in\mathcal{S}} c_l^{(s)}\theta^{(s)}[r]y_l^{(s)}[r] - d_l, 0\right] \\ H_k^{(s)}[r+1] &= H_k^{(s)}[r] + \theta^{(s)}[r]1_k^{(s)}[r] - \sum_{i\in\mathcal{K}^{(s)}} \theta^{(s)}[r]q_{ik}^{(s)}[r] \\ J^{(s)}[r+1] &= J^{(s)}[r] + \theta^{(s)}[r]T^{(s)}[r] - 1 \end{aligned}$$

C. The Drift-Plus-Penalty Algorithm

For a given parameter $V \geq 0$, define $f^{(s)}(k, \omega, \alpha)$ by:

$$\begin{aligned} f^{(s)}(k,\omega,\alpha) \triangleq V\hat{y}_0^{(s)}(k,\omega,\alpha) + \textstyle\sum_{l=1}^{L} Z_l[r]c_l^{(s)}\hat{y}_l^{(s)}(k,\omega,\alpha) \\ + H_k^{(s)}[r] - \textstyle\sum_{j\in\mathcal{K}^{(s)}} H_j^{(s)}[r]\hat{P}_{kj}^{(s)}(\omega,\alpha) \\ + J^{(s)}[r]\hat{T}^{(s)}(k,\omega,\alpha) \end{aligned}$$

Define $g^{(s)}(\theta, k, \omega, \alpha) \triangleq \theta f^{(s)}(k, \omega, \alpha)$. Define $\mathcal{B}^{(s)}$ as the set of all (θ, α) values that satisfy $1/T_{max}^{(s)} \leq \theta \leq 1/T_{min}^{(s)}$, $\alpha \in \mathcal{A}^{(s)}$. At the beginning of each frame r and for each $s \in \mathcal{S}$, observe virtual queues and perform the following:

- ($k^{(s)}[r]$ selection) Choose $k^{(s)}[r]$ as the index $k \in \mathcal{K}^{(s)}$ that minimizes the following (breaking ties arbitrarily):

$$\mathbb{E}\left\{\min_{(\theta,\alpha)\in\mathcal{B}^{(s)}} g^{(s)}(\theta, k, \omega^{(s)}[r], \alpha)\right\} \qquad (36)$$

  where the expectation above is with respect to the randomness of $\omega^{(s)}[r]$.
- ($\alpha^{(s)}[r]$, $\theta^{(s)}[r]$ selection) Once the $k^{(s)}[r]$ decision is made, observe the actual $\omega^{(s)}[r]$ and choose $\alpha^{(s)}[r]$ as the minimizer of $f^{(s)}(k^{(s)}[r], \omega^{(s)}[r], \alpha)$ over all $\alpha \in \mathcal{A}^{(s)}$, breaking ties arbitrarily. Then chose $\theta^{(s)}[r]$ by:

$$\theta^{(s)}[r] = \begin{cases} \frac{1}{T_{min}^{(s)}} & \text{if } f^{(s)}(k^{(s)}[r], \omega^{(s)}[r], \alpha^{(s)}[r]) \leq 0 \\ \frac{1}{T_{max}^{(s)}} & \text{otherwise} \end{cases}$$

- (Virtual Queue Update) Update the virtual queues according to the update equations in Section V-B.

The resulting algorithm satisfies all constraints whenever it is possible to do so, and yields an objective function that differs by $O(1/V)$ from optimal, with a corresponding polynomial convergence time tradeoff with V [5].

D. Discussion

The above algorithm selects $\alpha^{(s)}[r]$ and $\theta^{(s)}[r]$ without knowledge of the distribution of $\omega^{(s)}[r]$. Selection of $k^{(s)}[r]$ requires evaluation of the expectation in (36). This decision is trivial in special cases such as that given in Section III, where the systems $s \in \mathcal{S}$ that have random event processes $\omega^{(s)}[r]$ are 1-state systems (without Markov dynamics) for which one always selects $k^{(s)}[r] = 0$, and the systems that have Markov dynamics do not have $\omega^{(s)}[r]$ processes (so that the expectation in (36) reduces to the deterministic minimum). In the general case, the expectation (36) can be efficiently estimated based on a collection of past samples of $\omega^{(s)}[r]$, as justified by the max-weight learning framework of [14].

The algorithm above can be viewed as an offline algorithm for computing desired targets and finding the optimal time average quantities given a sample sequence of observed $\{\omega^{(s)}[r]\}_{r=0}^{\infty}$ values for each system. In an online implementation where such a sample sequence is gradually observed, the algorithm acts over *virtual frames* that run slower than the actual system. Specifically, the operations required on the rth virtual frame cannot be performed until the $\omega^{(s)}[r]$ value for each system s is observed. Each observed value is stored in memory as needed. The resulting weighted averages achieved in this virtual system act as progressively updated targets that are passed into an online algorithm such as [6] that runs separately on each actual system.

REFERENCES

[1] B. Fox. Markov renewal programming by linear fractional programming. *Siam J. Appl. Math*, vol. 14, no. 6, Nov. 1966.

[2] H. Mine and S. Osaki. *Markovian Decision Processes*. American Elsevier, New York, 1970.

[3] V. S. Borkar. An actor-critic algorithm for constrained Markov decision processes. *Systems and Control Letters (Elsevier)*, vol. 54, pp. 207-213, 2005.

[4] F. J. Vázquez Abad and V. Krishnamurthy. Policy gradient stochastic approximation algorithms for adaptive control of constrained time varying markov decision processes. *Proc. IEEE Conf. on Decision and Control*, Dec. 2003.

[5] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[6] M. J. Neely. Online fractional programming for Markov decision systems. *Proc. Allerton Conf. on Communication, Control, and Computing*, Sept. 2011.

[7] M. J. Neely. Asynchronous scheduling for energy optimality in systems with multiple servers. *Proc. 46th Conf. on Information Sciences and Systems (CISS)*, March 2012.

[8] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2005.

[9] E. Altman. *Constrained Markov Decision Processes*. Boca Raton, FL, Chapman and Hall/CRC Press, 1999.

[10] S. Ross. *Introduction to Probability Models*. Academic Press, 8th edition, Dec. 2002.

[11] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.

[12] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396-409, April 2008.

[13] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466-478, March 1993.

[14] M. J. Neely, S. T. Rager, and T. F. La Porta. Max weight learning algorithms for scheduling in unknown environments. *IEEE Transactions on Automatic Control*, to appear.