Optimal Cross-layer Wireless Control Policies using TD Learning

Sean Meyn, Wei Chen, and Daniel O'Neill

***Abstract*— We present an on-line crosslayer control technique to obtain policies for wireless networks. Our approach combines network utility maximization and adaptive modulation over an infinite discrete-time horizon using a class of performance measures we call *ime smoothed utility functions*. We model the system as an average-cost Markov decision problem. Model approximations are used to find suitable basis functions for application of least squares TD-learning techniques. The approach yields network control policies that learn the underlying characteristics of the random wireless channel and that approximately optimize network performance.**

***Acknowledgment* Financial support from the National Science Foundation under CCF-0729031 and ITMANET DARPA RK 2006-07284 is gratefully acknowledged.**

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF or DARPA.

I. INTRODUCTION

In wireless systems the characteristics of the RF channel vary randomly over time. Several different probabilistic models are used to represent this variation, but in many practical systems the distribution is generally not well described. In this paper we combine adaptive modulation (AM) and network utility maximization (NUM) to model the network in a manner similar to [1]: AM adapts a transmitter's rate and power as a function of the current channel state. NUM models the upper layer performance of data flows through the network and controls the rate at which packets are injected into the network. NUM models upper layer performance using concave utility functions. Different protocols are generally modeled using different functions. We consider time smoothed utility functions, which measure an exponentially smoothed average data rate for a data flow through the system. The time averaging models the different time scales used by the physical layer and upper layer protocols and also the time sensitivity of the traffic being carried by the network.

To obtain a policy for rate control, we model the system as an average-cost, infinite-horizon Markov decision process (MDP). We use this formulation on pragmatic grounds; without prior knowledge, each packet is of equal value and there is little justification in discounting future packets as inherently having lesser value. The model captures the tradeoff between the demand for average system performance as measured by utility functions, and the cost of supplying this performance as measured by average transmitter power.

The average-cost MDP problem and associated average-cost optimality equation (ACOE) are difficult to solve. In many wireless systems the size of the state space can be very large and the underlying transition probabilities may be unknown, limiting numerical techniques such as value iteration to small problems. We address these issues through the introduction of least squares TD-learning (LSTD) techniques.

The performance of LSTD depends critically on the basis functions chosen to represent the relative value function performance of the system. The main contribution of this paper is to show how a basis can be created so that the LSTD algorithm will provide a good approximation to the relative value function of the system. Our approach is related to the fluid-model approximations for value functions in [2]–[4], previously applied to approximate dynamic programming approaches of [5]. Our approach is more similar to the recent work [6] in which the approximation to the ACOE is obtained through Taylor series approximations. We obtain approximate solutions to the resultant first order ODE, which in turn yield basis functions useful for LSTD-learning. For the case of multiple data flows with identical utility functions, we show that the system experiences a form of state space collapse, with individual flows converging to the same average flow rates. Simple numerical simulations suggest that the basis functions found by our approach yield good approximation to relative value function found using value iteration.

The remainder paper is organized as follows: Sec. II describes the system model. Sections III and IV describe the structure of the optimal control policy and the form of our value function approximations. Sec. IV-D contains an analysis of the optimization problem via state space collapse for multiple flows on a single link. These results are used to create basis functions in LSTD-learning in Sec. V, where numerical results are also surveyed. Sec. VI summarizes our conclusions and ideas for future work.

II. MODEL

For clarity, we consider a single wireless link carrying $m = 1,\dots M$ data flows, under time varying flat fading. Time $t = 0,1,\dots$ is discrete. We model upper layer performance using time-smoothed utility functions [7]. Link performance is the average affine combination of the smoothed utility functions and transmitter power. The objective is to obtain a policy that defines flow rates that is approximately optimal.

The link experiences i.i.d. flat fading, modeled by the channel state process $G(t) > 0$, with unknown marginal distribution. However, in each time period the transmitter is able to sample the channel and so has certain knowledge of the current state of the channel. For notational simplicity, $G(t)$ is normalized by the noise at the receiver. The link

Daniel O'Neill. is with Dept. of Electrical Engineering, Stanford University. The remaining authors are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

transmits at an instantaneous power $\mathscr{E}(t)$, and the link SNR is $G(t)\mathscr{E}(t)$.

The link instantaneous transmission rate is given by

$$\mu(t) = \log\left(1 + \frac{G(t)\mathscr{E}(t)}{\lambda}\right), \tag{1}$$

where $\lambda = -\log(BER)$ [8], and BER is the target bit error rate ceiling.

Data flows arrive at rate $u_m(t) \in \mathbb{R}_+^M$, controlled by the upper layers of the link. This is called the instantaneous source rate. Different flows can correspond to different types of traffic, such as video, data or voice, with different time characteristics. We measure the upper layer performance using time smoothed utility functions. Associated with each flow m is a utility function $\mathscr{U}_m$, which measures the upper layer performance of the averaged flow of u_m. Different flows may have different utility functions, reflecting the use of different protocols. Utility functions are assume to be increasing, and strictly concave. The concavity assumption means that there are diminishing marginal returns with increasing rate.

For a given averaging parameter $\delta \in (0,1)$, the time averaged data flow r evolving on $\mathbb{R}_+^M$ is defined by,

$$r(t+1) = \delta r(t) + (1-\delta)u(t) \tag{2}$$

Averaging the flow rate reflects the demands of different types of traffic. When $\delta = 0$ each period is evaluated independently. This models traffic that is delay sensitive or where packets can't be shifted between time periods. Voice traffic, with the appropriate utility function, can be modeled in this manner. For file transfer, packets can be shifted between periods, with the average rate a more important metric than the instantaneous rate. In this case $\delta \approx 1$ may be appropriate. For video traffic, short term averages may be most appropriate and an intermediate value of δ can be used. For concreteness and simplicity we take $\mathscr{U}_m(r_m) = \log(r_m + 1)$ throughout most of the paper. We denote the total utility by

$$\mathscr{U}(r) = \sum_{m=1}^{M} \mathscr{U}_m(r_m) = \sum_{m=1}^{M} \log(r_m + 1). \tag{3}$$

The system first samples the channel $G(t)$. Based on this and the average data flow rate $r(t)$, it then adjusts its transmitter power $\mathscr{E}(t)$, link rate $\mu(t)$, and the instantaneous source rate $u(t)$. Since no buffering is assumed, the instantaneous traffic rate carried by the link must equal the link instantaneous transmission rate

$$1^T u = \mu, \tag{4}$$

and consequently from (1) the data flow rates and channel state determine the transmitter power. When $G(t) = g$, then

$$\mathscr{E}(t) = \lambda \frac{e^{1^T u} - 1}{g}. \tag{5}$$

In each time period the objective function is the difference between the sum of the utility functions and the scaled cost of the transmitter power used by the system

$$c(r,g,u) = v\mathscr{E}(g,u) - \mathscr{U}(r) = v\lambda \frac{e^{1^T u} - 1}{g} - \mathscr{U}(r), \tag{6}$$

where v is the tradeoff between transmitter power and utility.

III. OPTIMALITY EQUATIONS

Throughout the paper we focus on the average-cost optimality criterion. We describe in this section the ACOE and properties of the associated relative value function. Our development follows [4], [9], [10].

We begin with a single-flow model. It is convenient to define the state process as the joint process $X(t) = (R(t), G(t))$ evolving on $\mathbb{R}^2$. Let P_u denote the controlled Markov transition kernel, interpreted as a linear operator whose domain consists of measurable functions $h\colon \mathbb{R}^2 \to \mathbb{R}$ for which the conditional expectation is finite valued,

$$\begin{aligned} P_u h(r,g) &:= \mathsf{E}[h(R(t+1), G(t+1)) | R(t) = r, G(t) = g] \\ &= \mathsf{E}[h(\delta r + (1-\delta)u, G(1))]. \end{aligned} \tag{7}$$

For a given input sequence U and initial condition $r = R(0)$, the average cost is the limit suprimum

$$\eta^U(r) = \limsup_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n-1} \mathsf{E}[c(R(t), U(t), G(t))]. \tag{8}$$

The infimum over all U is denoted η^*, which is assumed to be independent of r for this model[1]. It is achieved using a state-feedback policy $U^*(t) = \phi^*(R(t), G(t))$. We find ϕ^* by solving the ACOE

$$\min_u \big(P_u h^*(r,g) + c(r,u,g)\big) = h^*(r,g) + \eta^*. \tag{9}$$

The solution h^* is called the relative value function, and the minimizer defines ϕ^*.

We define $L^*(r)$ to be the conditional expectation of h^*, given $R = r$:

$$L^*(r) = \mathsf{E}[h^*(r, G)] \tag{10}$$

so that the ACOE can be expressed

$$\eta^* + h^*(r,g) = \min_{u\geq 0} \big\{ c(r,g,u) + L^*(\delta r + (1-\delta)u) \big\}. \tag{11}$$

The standard technique for computation of the relative value function is through value iteration. In this model the algorithm has a special form: Let h_0 denote an initial guess for h^*, and define inductively for $n \geq 1$,

(i) $h_{n+1}(r,g) = \min_{u\geq 0} \big\{ c(r,u,g) + L_n(\delta r + (1-\delta)u) \big\}$
(ii) $L_{n+1}(r) = \mathsf{E}[h_{n+1}(r, G)]$.

If $h_0 \equiv 0$ then at the nth stage of the algorithm this gives,

$$h_n(r,g) = \min_U \mathsf{E}_{r,g}[\sum_0^{n-1} c(R(t), U(t), G(t))] \tag{12}$$

[1]For sufficient conditions see [2], [4], [9].

where $\mathsf{E}_{r,g}[f(R(t),G(t))] = \mathsf{E}[f(R(t),G(t)) \,|\, R(0)=r, G(0)=g]$. We shall take it for granted that the algorithm is convergent. This means that for a given initial condition (r^*,g^*), the limit

$$h^*(r,g) = \lim_{n\to\infty} (h_n(r,g) - h_n(r^*,g^*)) \tag{13}$$

exists, and is a solution to the ACOE — See [10] for sufficient conditions. This construction is useful for establishing properties of the relative value function in the following result. Similar convexity results can be found for models in the queueing literature (see [4], [6], [11]).

Proposition 1 *If $\mathscr{U}$ is concave and non-decreasing, then $L^*\colon \mathbb{R}_+ \to \mathbb{R}$ is convex and non-increasing.*

Proof: We first establish these properties for the function h_n defined in (12).

We first fix a value of g and the two initial conditions $r_a \geq 0$ and $r_b \geq 0$, and let $\{U_a, U_b\}$ denote inputs that are feasible at their respective initial conditions. For $\theta \in [0,1]$ and $t \geq 0$, we denote $r_\theta = \theta r_a + (1-\theta) r_b$, and $U_c(t) = \theta U_a(t) + (1-\theta) U_b(t)$. This input is feasible from the initial condition r_θ, since linearity of (2) gives $R_c(t) = \theta R_a(t) + (1-\theta) R_b(t)$. From (12) and the convexity of c with respect to r we obtain,

$$\begin{aligned} h_n(r_\theta, g) &\leq \mathsf{E}_{r_\theta,g}[\sum_0^{n-1} (c(R(t), U_c(t), G(t)))] \\ &\leq \mathsf{E}_{r_a,g}[\sum_0^{n-1} \theta (c(R_a(t), U_a(t), G(t)))] \\ &\quad + \mathsf{E}_{r_b,g}[\sum_0^{n-1} (1-\theta)(c(R_b(t), U_b(t), G(t)))]. \end{aligned}$$

Since U_a and U_b are arbitrary feasible inputs we obtain the convexity of h_n.

To see that h_n is nonincreasing we take $r_a \geq r_b$, and note that any input that is feasible for the initial condition r_b is also feasible for the initial condition r_a. Consequently, for any input U_b feasible with respect to the smaller initial condition,

$$\begin{aligned} h_n(r_a, g) &\leq \mathsf{E}_{r_a,g}[\sum_0^{n-1} (c(R(t), U_b(t), G(t)))] \\ &\leq \mathsf{E}_{r_b,g}[\sum_0^{n-1} (c(R(t), U_b(t), G(t)))] \end{aligned}$$

where the second equation again follows by linearity, combined with the form of the cost structure (6) which implies that c is non-increasing in r. Minimizing over all inputs U_b establishes the bound $h_n(r_a,g) \leq h_n(r_b,g)$.

We conclude that $h_n(r,g) - h_n(r^*,g^*)$ is convex and non-increasing for each n, and hence so is the limit $h^*(r,g)$. It is then obvious that L^* shares these properties by applying the definition (10). ∎

In the multi-flow problem we can obtain identical conclusions using similar arguments. We now move on to establish approximations for the value function. These approximations form the architecture for the TD-learning algorithms described in Section V.

IV. VALUE FUNCTION APPROXIMATIONS

We consider several approximations for the function L^*. We begin with Taylor series approximations for the function L^*, similar to those used in [6].

A. Single Flow Single Link Model

If L^* is differentiable, as it will be a.e. under the assumptions of Proposition 1, then we approximate the ACOE (11) via,

$$\eta^* + h^*(r,g) \cong \min_{u\geq 0} \big\{ c(r,g,u) + L^*(r) + \nabla L^*(r)\delta(u-r) \big\}.$$

Substituting G for g and taking expectations then gives,

$$\eta^* + L^*(r) \cong \mathsf{E}[\min_{u\geq 0}(c(r,G,u) + \nabla L^*(r)\delta(u-r))] + L^*(r)$$

and thence,

$$\eta^* \cong \mathsf{E}[\min_{u\geq 0}(c(r,G,u) + \nabla L^*(r)\delta(u-r))]. \tag{14}$$

Justification for this approximation is beyond the scope of this paper. In [6] this is justified by first *defining* $(K^*, \eta^\dagger)$ as the solution to

$$\eta^\dagger = \mathsf{E}[\min_{u\geq 0}(c(r,G,u) + \nabla K^*(r)\delta(u-r))]. \tag{15}$$

It was then shown that $K^* \approx L^*$. Similar bounds can be established for this model.

The first order condition for the optimality of u is

$$v\lambda \frac{e^{u^*}}{g} + \nabla L^*(r)\delta = 0, \qquad G = g.$$

This can be solved to give an approximation for ϕ^*,

$$\phi^*(r,g) \cong \big\{ \log(\frac{\delta g}{v\lambda} |\nabla L^*(r)|) \big\}_+. \tag{16}$$

Substitution into (14) then gives a nonlinear fixed point equation for ∇L^*.

B. Multi-Flow Single Link Model

Similar to (14), the average-cost optimality equation is approximated as

$$\eta^* \cong \mathsf{E}[\min_{u\geq 0}(c(r,G,u) + \delta \nabla L^{*T}(r)(u-r))], \tag{17}$$

where $\nabla L^*(r) = (\partial L^*/\partial r_1, \ldots, \partial L^*/\partial r_M)$. The first order condition for optimality gives u^* as a function of $g = G$,

$$\begin{aligned} v\lambda \frac{e^{1^T u^*}}{g} + \nabla_i L^* \delta = 0 \quad &\text{if} \quad u_i^* > 0, \\ v\lambda \frac{e^{1^T u^*}}{g} + \nabla_i L^* \delta \geq 0 \quad &\text{if} \quad u_i^* = 0, \end{aligned}$$

where $\nabla_i L^* = \partial L^*(r)/\partial r_i$. Solving for u^* then gives the approximation,

$$\phi^*(r,g) \cong \begin{cases} \big\{ \log(\frac{\delta g}{v\lambda} |\nabla_i L^*|) \big\}_+ & \text{if } i = \arg\min_j \nabla_j L^* \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

In the remainder of this section we obtain approximations for the relative value function using these results. The simplest approximations are obtained by considering a large initial condition.

C. Approximations for large r

The functions h^* and L^* are convex in $r \in \mathbb{R}^M$ and are bounded on bounded subsets of $\mathbb{R}^M$, since $\mathscr{U}$ is continuous. To obtain further structure we consider large values of $R(0)$ and apply the approximations from above.

For this we apply the dynamic programming equation, which in the case of average cost may be interpreted as a martingale representation of the relative value function: For any time $T \geq 1$, and $(R(0), G(0)) = (r, g)$,

$$h^*(r,g) = \min_U \mathsf{E}\Big[\sum_0^{T-1} \big(c(R(t),U(t),G(t)) - \eta^*\big) + L^*(R(T))\Big] \tag{19}$$

The minimum is achieved using the policy ϕ^*. Note that we have used the identity $\mathsf{E}[h^*(R(T),G(T))] = \mathsf{E}[L^*(R(T))]$.

When $\mathscr{U}_i(r_i) = \log(r_i + 1)$ as assumed here, we obviously have $\frac{d}{dr}\mathscr{U}_i(r_i) \to 0$ as $r_i \to \infty$, and consequently $\lim_{r_i \to \infty} \nabla_i L^*(r) = 0$. Based on the approximation (18) we conclude that $U_i^*(t) \cong 0$ when $R_i(t)$ is sufficiently large. The evolution of R is thus approximated by $R_i(t+1) = \delta R_i(t)$ when $R_i(t)$ is large.

Consider the single flow case, with $r = R(0) \gg 1$. Assume that $r_\bullet \geq 1$ is a constant for which $\phi^*(r,g) = 0$ whenever $r \geq r_\bullet$. We then have, $R(t+1) = \delta R(t),\ 0 \leq t \leq T$, where T is the first time that $R(t) < r_\bullet$. Applying (6), this gives the approximation,

$$h^*(r,g) \approx \mathsf{E}_{r,g}\Big[\sum_0^{T-1} \big(-\mathscr{U}(R(t)) - \eta^*\big) + L^*(R(T))\Big] \tag{20}$$

It is simplest to approximate $T(r)$ and the right hand side of (20) by first approximating $\{R(t) : 0 \leq t \leq T\}$ via the differential equation (or fluid model)

$$\tfrac{d}{dt} r(t) = -(1-\delta) r(t).$$

This has the solution $r(t) = r(0)e^{-(1-\delta)t}$, $t \leq T$, where $r(0) = R(0) = r$. On writing $r(T) = r_\bullet$ we can solve $re^{-(1-\delta)T} = r_\bullet$ to obtain,

$$T = \frac{1}{1-\delta}\big(\log(r) - \log(r_\bullet)\big).$$

Based on this, we approximate (20) by,

$$h^*(r,g) \approx \int_0^T \big(-\mathscr{U}(r(t)) - \eta^*\big)dt + L^*(r_\bullet),$$

valid for $r \gg r_\bullet$. Moreover, for $t \leq T$ we have $\mathscr{U}(r(t)) \approx \log(r(t)) = \log(r) - (1-\delta)t$. On combining these approximations we finally approximate $h^*(r,g)$ by a function that is quadratic in $\log(r)$,

$$h^*(r,g) \approx \theta_0 + \theta_1 \log(r) + \theta_2 (\log(r))^2, \qquad r \gg r_\bullet,$$

with $\{\theta_i\}$ constants.

For application to TD-learning we prefer to express this approximation for L^* instead of h^*, and modify the approximation slightly so that it will be meaningful for small values of r: For parameters $\theta_1, \theta_2 \in \mathbb{R}$, and $r \gg r_\bullet$,

$$L^*(r) \approx \theta_1 \psi_1(r) + \theta_2 \psi_2(r) \tag{21}$$

$$\psi_1(r) = \log(r+1), \quad \psi_2(r) = \big(\log(r+e)\big)^2 \tag{22}$$

The shift by 1 and by e is to firstly ensure that the right hand side of (21) is finite at the origin, and secondly to enable an approximation that is convex over $r \in \mathbb{R}_+$. The right hand side of (21) is convex whenever θ_1 and θ_2 are non-positive.

Generalization of this bound to the multi-flow case is straightforward. However, we can obtain a far simpler description by exploiting a form of *state space collapse* observed in this model.

D. State space collapse

The notion of state space collapse comes from the heavy-traffic theory of stochastic networks [4]. In this context, a reduction in dimension is obtained through a separation of time-scales, much like in singular perturbation analysis in dynamical systems and Markov chains. Here state space collapse follows from the special structure of the system.

The set onto which the state is "collapsing" is the ray denoted $S := \{r | r_i = r_j, 0 \leq i, j \leq M\}$. There are various reasons to suspect that the process $R(t)$ will favor this region:

(i) *S is absorbing*. Suppose that $R(0) \in S$. Then by symmetry of the model we conclude that $U_i^*(0) = U_j^*(0)$ for each i, j. It follows from (2) that $R(1) \in S$, and thus $R(t) \in S$ for each t.

(ii) *ϕ^* favors S*. It can be shown from the approximation (18): If $R(0)$ is far from S, then $U_i(0)$ will be large only for i for which $R_i(0) \ll \overline{R}(0)$, where the bar denotes average.

A third way of understanding this collapse is through a relaxation. We consider the multi-flow model in which $R(t)$ is constrained to $\mathbb{R}_+^M$ and $1^{\intercal}U(t)$ is constrained to be non-negative, but no constraints are imposed on the individual values $\{U_i(t) : 1 \leq i \leq M\}$. We call this the relaxed problem. We prove in this subsection,

Proposition 2 *For each t and initial condition $R(0) \in \mathbb{R}_+^M$, the optimal solution for the relaxation satisfies $R^*(t) \in S$.*

The proof follows easily from the following lemma. Let $\hat{h}^*$ denote the relative value function for the relaxation, and define $\hat{h}^+(r,g) := \hat{h}^*(r,g) + \mathscr{U}(r)$.

Lemma 1 *The function $\hat{h}^+(r,g)$ depends on r only through $\overline{r} := M^{-1}\sum r_i$.*

Proof: Denote by $\{\hat{h}_n\}$ the sequence of solutions to the value iteration algorithm, initialized with $\hat{h}_0 \equiv 0$, and define $\hat{h}_n^+(r,g) = \hat{h}_n(r,g) + \mathscr{U}(r)$ for each n, r, g. To prove the lemma we establish by induction that $\hat{h}_n^+$ is a function of $(\overline{r}, g)$ for each n.

For $n = 0$ it is trivial. If it is true for a given $n \geq 0$, then we apply the definition,

$$\hat{h}_{n+1}(r,g) = \min_{1^{\intercal}u \geq 0} \big\{c(r,u,g) + \hat{L}_n(\delta r + (1-\delta)u)\big\}$$

where $\hat{L}_n(r) = \mathsf{E}[\hat{h}_n(r,G)]$. Applying (6) that defines $c(r,g,u) = \frac{\lambda \nu}{g}(e^{1^{\intercal}u} - 1) - \mathscr{U}(r)$, we obtain

$$\hat{h}_{n+1}^+(r,g) = \min_{1^{\intercal}u \geq 0} \Big\{\frac{\lambda \nu}{g}(e^{1^{\intercal}u} - 1) + \hat{L}_n(\delta r + (1-\delta)u)\Big\} \tag{23}$$

To prove the lemma we must show that the right hand side is determined by $(\overline{r},g)$.

Introducing a Lagrange multiplier $\alpha \ge 0$, the Lagrangian relaxation for this optimization problem is expressed

$$\hat{h}^{+}_{n+1}(r,g) = \min_{u\in\mathbb{R}^M} \Big\{ \frac{\lambda v}{g}(e^{1^T u}-1) + \hat{L}_n(\delta r + (1-\delta)u) - \alpha \overline{u} \Big\}$$

with $\overline{u} := 1^T u$. On taking derivatives with respect to u we obtain for the optimizing value u^*,

$$(1-\delta)\nabla_m \hat{L}_n(r^*) = \alpha - \frac{\lambda v}{g} e^{1^T u^*}, \qquad 1 \le m \le M,$$

where $r^* := \delta r + (1-\delta)u^*$. Observe that the derivative is independent of $m = 1,\dots,M$. To complete the proof we apply the induction hypothesis, which implies that the function $\hat{L}_n$ can be expressed as $\hat{L}_n(r) = -\mathscr{U}(r) + \ell_n(\overline{r})$ for a function $\ell_n \colon \mathbb{R} \to \mathbb{R}$. From (3) this then gives,

$$(1-\delta)\mathscr{U}_m'(r_m^*) = (1-\delta)\ell_n'(\overline{r}^*) - \alpha + \frac{\lambda v}{g} e^{M\overline{u}^*} \tag{24}$$

with $\overline{u}^* = M^{-1}1^T u^*$, and $\overline{r}^* = \delta\overline{r} + (1-\delta)\overline{u}^*$. We can invert (24) (recall $\mathscr{U}_m(r_m) = \log(r_m+1)$ is independent of m) to obtain,

$$r_m^* = \mathscr{U}_1'^{-1}\Big(\ell_n'(\overline{r}^*) + \frac{1}{(1-\delta)}\big(-\alpha + \frac{\lambda v}{g} e^{M\overline{u}^*}\big)\Big). \tag{25}$$

We now consider two cases: If $\alpha > 0$ then complementary slackness gives $M\overline{u}^* = 0$. Hence $\overline{r}^* = \delta\overline{r}$, and (25) becomes

$$r_m^* = \mathscr{U}_1'^{-1}\Big(\ell_n'(\delta\overline{r}) + \frac{1}{(1-\delta)}\big(-\alpha + \frac{\lambda v}{g}\big)\Big) \tag{26}$$

On summing over m,

$$\delta\overline{r} = M^{-1}\sum_m r_m^* = \mathscr{U}_1'^{-1}\Big(\ell_n'(\delta\overline{r}) + \frac{1}{(1-\delta)}\big(-\alpha + \frac{\lambda v}{g}\big)\Big)$$

The Lagrange multiplier α is determined as the solution to this equation, and is hence a function of $(\overline{r},g)$. It follows from (26) that r^* is a function of $(\overline{r},g)$, and thence from (23) that $\hat{h}_{n+1}(r,g)$ is a function of $(\overline{r},g)$, provided $\alpha > 0$.

If $\alpha = 0$ we again sum each side of (25) over m to obtain

$$\overline{r}^* = \mathscr{U}_1'^{-1}\Big(\ell_n'(\overline{r}^*) + \frac{1}{(1-\delta)}\frac{\lambda v}{g} e^{M\overline{u}^*}\Big)$$

Once again, from (25) and $\overline{r}^* = \delta\overline{r} + (1-\delta)\overline{u}^*$ we conclude that $\overline{u}^*$ and r^* are each determined as functions of $(\overline{r},g)$, and once again (23) then implies that $\hat{h}_{n+1}(r,g)$ is a function of $(\overline{r},g)$. ■

The optimal policy is obtained as the minimum,

$$u^* = \hat{\phi}^*(r,g) = \operatorname*{arg\,min}_{1^T u \ge 0} \big\{ c(r,u,g) + \hat{L}^*(\delta r + (1-\delta)u) \big\}$$

where $\hat{L}^*(r) = \mathsf{E}[\hat{h}^+(r,G)] - \mathscr{U}(r)$. Lemma 1 implies that $\mathsf{E}[\hat{h}^+(r,G)]$ is a function only of $\overline{r}$. Familiar arguments then show that given any value of $r = R(0)$, the next state $r^* = R(1)$ lies in the set S, which proves Proposition 2.

Fig. 1 (a) shows $\hat{L}^*$ for a model with two flows — The value function L^* is shown in Fig. 2 (b) below. Fig. 1 (b) plots the error $|\hat{L}^* - L^*|$ as a function of r. The relative error is extremely small in this example.

V. EXAMPLES AND SIMULATION

The results of the preceding section motivate an approximation of the form,

$$L^*(r) \approx -\mathscr{U}(r) + \theta_1\psi_1(\overline{r}) + \theta_2\psi_2(\overline{r}) \tag{27}$$

where the basis functions $\{\psi_i\}$ are defined in (22). In this section we apply LSTD-learning to compute parameters $\{\theta_1,\theta_2\}$ that give rise to the best approximation.

We note that in many wireless systems the size of the state space can be very large and the underlying transition probabilities may be unknown, limiting value iteration to small problems and necessitating an on line learning approach such as LSTD-learning. For the simple cases considered here, the difference between the relative value function using value iteration and the approximation by the proposed bases was found to be extremely small. This suggests that the basis (27) accurately approximates the relative value function and the performance characteristics of the system. In more complex problems we suggest the application of policy improvement combined with LSTD-learning as in [6].

In our numerical experiments we restrict to a model in which the i.i.d. channel state takes on only three values $\{1,2,3\}$, with probability $\{0.25,0.5,0.25\}$ respectively.

A. Value Iteration

In this subsection we compute the relative value function for a discrete version of the problem using the value iteration algorithm (VIA). Unfortunately the complexity of the value iteration technique grows exponentially with the dimension of the state space, limiting its use to smaller problems. Here we consider for a single link with either a single flow $M = 1$ or two flows $M = 2$. The state space is 100 for single flow case and 10^4 for the two-flow model.

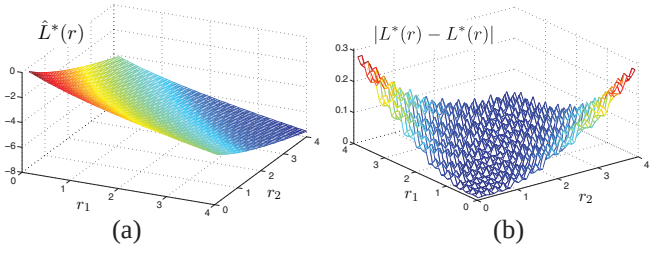


Fig. 1. (a) Relaxed relative value function $\hat{L}^*$ computed using value iteration. (b) Difference between relaxed and unrelaxed value function

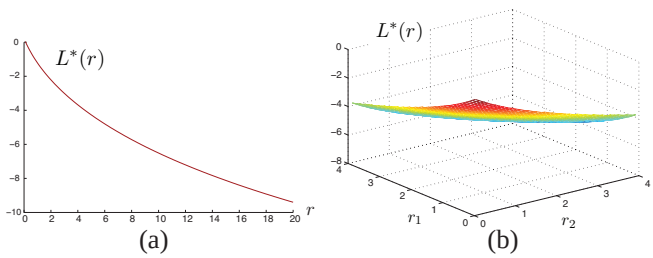
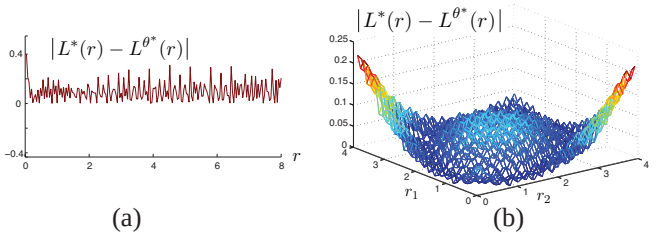


Fig. 2. (a) VIA result for single flow relative value function. (b) VIA result for two flows relative value function.

We compute the relative value function using value iteration, as described in [4], [12]. The relative value function is normalized relative to $L^*(0)$ as $L^*(r) = L^*(r) - L^*(0)$. The results are shown in Fig. 2. The plots illustrate the convexity, symmetry and non-increasing properties of L^* for both the single and multiple flows cases.

B. Basis Approximation to Value Function

In this subsection we find the best least squares approximation to the value function using the basis (22). In particular we minimize $\|L^*(r) + \mathscr{U}(r) - \theta\psi(\overline{r})\|^2$, where ψ is defined in (22). The distance between the relative value function from VIA and the approximation by the proposed bases is shown in Fig. 3. The small error demonstrates the effectiveness of the bases.

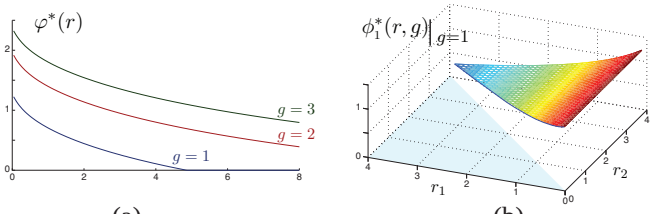


(a) (b)

Fig. 3. (a) $|L^*(r) - L^{\theta^*}(r)|$ for the single flow model. (b) $|L^*(r) - L^{\theta^*}(r)|$ for the two-flow model.

C. LSTD-Learning

LSTD-learning is an online learning method for finding approximations to value functions. The technique seeks the best least squares approximation to an unknown value function over a given class of basis functions using sampled data [4]. The technique can be extended to a form of generalized policy iteration to jointly estimate the relative value function and find the associated control policies (see recent examples in [6]).



(a) (b)

Fig. 4. (a) Control policy for the single flow model. (b) Control values $u_1^* = \phi_1^*(r,g)$ when $g = 1$ for the two-flow model; $u_1^* = 0$ when $r_1 > r_2$.

We use LSTD-learning to obtain the optimal coefficients for the set of bases (27). The policy used in LSTD is taken from (18). The relative value function $L^*(r)$ used in the policy is approximated by (27). The corresponding control policies are shown in Fig. 4 for both (a) the single flow case, and (b) for two-flow model. As anticipated, the policy is zero when the state is large. Fig. 5 shows results obtained after 100,000 iterations of the LSTD algorithm. Convergence was rapid for both the single-flow and two-flow models.

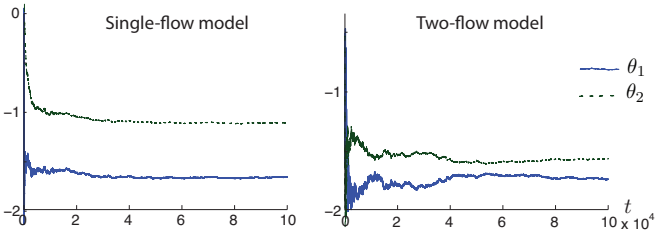


Fig. 5. Coefficients from LSTD for the single-flow and two-flow models.

VI. CONCLUSIONS & FUTURE WORK

We have seen that a combination of MDP modeling and approximate dynamic programming techniques allow tight approximations of optimal policies for crosslayer network control design. The application of fluid model approximations and state space collapse emerge as a general purpose tool for many other stochastic control problems.

A focus of current research is the extension of these techniques to multiple interfering links. The greatest challenge is addressing the complex rate region in these models. Our hope is that relaxation techniques can result in tractable solutions that adequately approximate real-world systems.

REFERENCES

[1] Daniel O'Neill and Stephen Boyd Andrea J.Goldsmith. Optimizing adaptive modulation in wireless networks via utility maximization. In *Proc. IEEE International Conference on Communications*, May 2008.

[2] S. P. Meyn. The policy iteration algorithm for average reward Markov decision processes with general state space. *IEEE Trans. Automat. Control*, 42(12):1663–1680, 1997.

[3] S. G. Henderson, S. P. Meyn, and V. B. Tadić. Performance evaluation and policy selection in multiclass networks. *Discrete Event Dynamic Systems: Theory and Applications*, 13(1-2):149–189, 2003. Special issue on learning, optimization and decision making (invited).

[4] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, Cambridge, 2007.

[5] C.C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queueing networks. Submitted for publication., 2006.

[6] Wei Chen, Dayu Huang, Ankur A. Kulkarni, Jayakrishnan Unnikrishnan, Quanyan Zhu, Prashant Mehta, Sean Meyn, and Adam Wierman. Approximate dynamic programming using fluid and diffusion approximations with applications to power management. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3575–3580, Dec. 2009.

[7] D. ONeill, E. Akuiyibo, S.P. Boyd, and A.J. Goldsmith. Optimizing adaptive modulation in wireless networks via multi-period network utility maximization. *IEEE Wireless Communications and Networking Conference, 2010*, 2010.

[8] G. J. Foschini and J. Salz. Digital communications over fading radio channels. *Bell Syst. Tech. J.*, pages 429–456, February 1983.

[9] V. S. Borkar. Convex analytic methods in Markov decision processes. In *Handbook of Markov decision processes*, volume 40 of *Internat. Ser. Oper. Res. Management Sci.*, pages 347–375. Kluwer Acad. Publ., Boston, MA, 2002.

[10] R-R. Chen and S. P. Meyn. Value iteration and optimization of multiclass queueing networks. *Queueing Syst. Theory Appl.*, 32(1-3):65–97, 1999.

[11] B. Hajek. Optimal control of two interacting service stations. *IEEE Trans. Automat. Control*, AC-29:491–499, 1984.

[12] D.P. Bertsekas and S.E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 1996.