Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009

# Approximate Dynamic Programming using Fluid and Diffusion Approximations with Applications to Power Management

Wei Chen, Dayu Huang, Ankur A. Kulkarni, Jayakrishnan Unnikrishnan
Quanyan Zhu, Prashant Mehta, Sean Meyn, and Adam Wierman

*Abstract*— **TD learning and its refinements are powerful tools for approximating the solution to dynamic programming problems. However, the techniques provide the approximate solution only within a prescribed finite-dimensional function class. Thus, the question that always arises is *how should the function class be chosen*? The goal of this paper is to propose an approach for TD learning based on choosing the function class using the solutions to associated fluid and diffusion approximations. In order to illustrate this new approach, the paper focuses on an application to *dynamic speed scaling* for power management.**

## I. Introduction

Stochastic dynamic programming and, specifically, controlled Markov chain models have become central tools for evaluating and designing communication, computer, and network applications. These tools have grown in popularity as computing power has increased; however, even with increasing computing power, it is often impossible to attain exact solutions. This is due to the so-called "curse of dimensionality", which refers to the fact that the complexity of dynamic programming equations often grows exponentially with the dimension of the underlying state space.

However, the "curse of dimensionality" is slowly dissolving in the face of approximation techniques such as Q-learning and TD-learning [3]. These techniques are designed to approximate a solution to a dynamic programming equation within a prescribed finite-dimensional function class. A key determinant of the success of these techniques is the selection of this function class. The question of how to select an appropriate basis has been considered in specific contexts, e.g. [9], [6]. However, determining the appropriate function class for these techniques is still more of an art than a science.

The goal of this paper is to illustrate that a useful function class can be attained by solving the dynamic programming equation for a highly idealized approximate model. Specifically, a useful function class is obtained by first constructing a fluid or diffusion approximation of the model, and solving the corresponding dynamic programming equation for the simpler system.

In the special case of network scheduling and routing, it is known that the dynamic programming equations for the continuous-time model are closely related to the corresponding equations for the discrete-time model [7]. The fluid value function has been used as part of a basis in the approximate dynamic programming approaches of [10], [8]. In this paper we demonstrate that the solution to the dynamic programming equations for the fluid, diffusion, and discrete-time models are closely related in more general classes of models.

In order to provide a concrete illustration of the proposed approximation techniques, the paper considers an example of a stochastic control problem from the area of power management in computer systems. Specifically, an important tradeoff in modern computer system design is between reducing energy usage and maintaining good performance (small delays). To this end, an important technique is *dynamic speed scaling* [2], which dynamically adjusts the processing speed in response to changes in the workload — reducing (increasing) the speed in times when the workload is small (large). Dynamic speed scaling is now common in many chip designs, e.g. [1], and network environments, e.g. wireless communication [12].

For purposes of this paper, dynamic speed scaling is simply a stochastic control problem – a single server queue with a controllable service rate – and the goal is to understand how to control the service rate in order to minimize the total cost, which is a weighted sum of the energy cost and the delay cost. In this context, this paper will illustrate how to use the solutions of the fluid and diffusion models in order to apply TD learning to determine an approximately optimal policy for control. Fluid and diffusion models for the dynamic speed scaling problem are analyzed in Sec. IV. The results of applying TD learning to the speed scaling problem are illustrated in Sec. V. These results highlight the usefulness of the fluid and diffusion solutions for TD learning.

## II. Preliminaries

### A. Markov Decision Processes (MDPs)

In this paper we will consider the following general MDP model. Let $\mathsf{X} = \mathbb{R}^{\ell}_{+}$ denote the state space for the model. The action space is denoted $\mathsf{U}$. In addition there is an i.i.d. process $\boldsymbol{W}$ evolving on $\mathbb{R}^{w}$ that represents a disturbance process. For a given initial condition $X(0) \in \mathsf{X}$, and a sequence $\boldsymbol{U}$ evolving on $\mathsf{U}$, the state process $\boldsymbol{X}$ evolves according to the recursion,

$$X(t+1) = X(t) + f(X(t), U(t), W(t+1)), \qquad t \geq 0. \quad (1)$$

A.W. is with Dept. of CS, California Inst. of Tech.. The remaining authors are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

We restrict to inputs that are defined by a (possibly randomized) stationary policy. This defines a Markov Decision Process (MDP) with controlled transition law

$$P_u(x,A) := \mathsf{P}\{x + f(x,u,W(1)) \in A\}, \ A \in \mathcal{B}(\mathsf{X}).$$

We let $\mathcal{D}_u$ denote the generator in discrete time. For any function $h\colon \mathbb{R} \to \mathbb{R}$,

$$\mathcal{D}_u h\,(x) := \mathsf{E}[h(X(t+1)) - h(X(t)) | X(t) = x, U(t) = u] \tag{2}$$

A cost function $c\colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}_+$ is given, and our goal is to find an optimal control based on this cost function. We focus on the average cost problem, with associated Average Cost Optimality Equation (ACOE):

$$\min_u \bigl(c(x,u) + \mathcal{D}_u h^*\,(x)\bigr) = \eta^* \tag{3}$$

The ACOE is a fixed point equation in the *relative value function* $h^*$, and the optimal cost for the MDP $\eta^*$.

## B. The fluid and diffusion models

The fluid model associated with the MDP model is defined by the following mean flow equations,

$$\tfrac{d}{dt} x(t) = \overline{f}(x(t), u(t)), \qquad x(0) \in \mathsf{X},$$

where $\boldsymbol{u}$ evolves on $\mathsf{U}$, and $\overline{f}(x,u) := \mathsf{E}[f(x,u,W(1))]$. The generator for the fluid model is defined similarly. Given $u(0) = u$, $x(0) = x$,

$$\mathcal{D}^{\mathrm{F}}_u h\,(x) = \tfrac{d}{dt} h(x(t))\Big|_{t=0} = \nabla h\,(x) \cdot \overline{f}(x,u). \tag{4}$$

The associated Total Cost Optimality Equation (TCOE) is

$$\min_u \bigl(c(x,u) + \mathcal{D}^{\mathrm{F}}_u J^*\,(x)\bigr) = 0 \tag{5}$$

It is solved with the value function,

$$J^*(x) = \inf_{\boldsymbol{u}} \int_0^\infty c(x(t), u(t))\,dt, \qquad x(0) = x \in \mathsf{X}, \tag{6}$$

provided $J^*$ is finite valued, which requires assumptions on the cost and dynamics. Under these assumptions the optimal policy is any minimizer,

$$\phi^{\mathrm{F}*}(x) \in \arg\min_u \bigl(c(x,u) + \mathcal{D}^{\mathrm{F}}_u J^*\,(x)\bigr) \tag{7}$$

In this paper, motivation for approximate models comes from a Taylor series expansion. In particular, if the fluid value function $J^*$ is smooth then we have the approximation,

$$\begin{aligned} \mathcal{D}_u J^*\,(x) &\approx \mathsf{E}_{x,u}\bigl[\nabla J^*(X(0))(X(1) - X(0))\bigr] \\ &= \nabla J^*\,(x) \overline{f}(x,u) \end{aligned} \tag{8}$$

where the subscript indicates expectation conditional on $X(0) = x,\ U(0) = u$. That is, $\mathcal{D}_u J^* \approx \mathcal{D}^{\mathrm{F}}_u J^*$, where the approximation depends on the smoothness of the function $J^*$.

A diffusion model is obtained similarly. We again choose its dynamics to reflect the behavior of the discrete-time model. To capture the state space constraint we opt for a reflected diffusion, defined by the Ito equation:

$$dX(t) = \overline{f}(X(t), U(t))dt + \sigma(U(t))dN(t) + dI(t), \tag{9}$$

where the process $\boldsymbol{N}$ is a standard Brownian motion on $\mathbb{R}^\ell$ and $\boldsymbol{I}$ is a reflection process. That is, for each $1 \le i \le \ell$, the process $I_i$ is non-decreasing and is minimal subject to the constraint that $X_i(t) \ge 0$ for each $t$ and each $i$. This is captured through the sample path constraint,

$$\int_0^\infty X_i(t)\,dI_i(t) = 0, \quad 1 \le i \le \ell.$$

## C. TD learning

TD learning is a technique for approximating value functions of MDPs within a linearly parameterized class.

Specifically, we define $\{\psi_i : 1 \le i \le d\}$ as real-valued functions on $\mathsf{X}$ and we let $h^r = \sum r_i \psi_i$ or, with $\psi\colon \mathsf{X} \to \mathbb{R}^d$ the vector of basis functions, $h^r = r^{\scriptscriptstyle T} \psi$. Suppose that a stationary policy $\phi$ is applied to the MDP model, and that the resulting Markov chain is ergodic with stationary marginal $\pi$. Let $h$ denote the solution to Poisson's equation $P_\phi h = h - c_\phi + \eta_\phi$ where $P_\phi(x,dy) = P_{\phi(x)}(x,dy)$ is the resulting transition law for the chain, $c_\phi(x) = c(x,\phi(x))$ is the cost as a function of state for this policy, and $\eta_\phi$ is the average cost. TD learning then takes the mean-square error criterion:

$$\tfrac{1}{2}\mathsf{E}_\pi[(h(X(0)) - h^r(X(0)))^2] := \tfrac{1}{2} \int (h(x) - h^r(x))^2\,\pi(dx).$$

Hence the optimal parameter satisfies the fixed point equation,

$$\mathsf{E}_\pi[(h(X(0)) - h^r(X(0)))\psi(X(0))] = 0. \tag{10}$$

In the rest of this section we assume that the control is fixed to be $\phi(x)$. We use $c(x)$ to denote the cost function $c_\phi(x)$ and $\mathsf{E}$ to denote the expectation under this stationary policy.

The TD and LSTD learning algorithms are techniques for computing the optimal parameter. We refer the reader to Chapter 11 of [7] for details of the LSTD learning algorithm used in the numerical results described in this paper and provide only a high-level description of the LSTD algorithm here.

When the parameterization is linear then (10) implies that the optimal parameter can be expressed

$$\begin{aligned} r^* = \Sigma^{-1} z \quad \text{with} \quad \Sigma &= \mathsf{E}_\pi[\psi(X(0))\psi(X(0))^{\scriptscriptstyle T}] \\ z &= \mathsf{E}_\pi[\psi(X(0))h(X(0))]. \end{aligned} \tag{11}$$

To estimate $\Sigma$ and $z$ we define the sequence of *eligibility vectors*,

$$\varphi(t+1) = \varphi(t) + \mathbb{I}\{X(t) \neq x^*\}(\psi(X(t)) - \eta_\psi(t))$$

where $\varphi(0) = \psi(X(0))$, and $\eta_\psi(t)$ the sample mean of $\psi$. We then define,

$$\Sigma_T = \frac{1}{T}\sum_{t=1}^T \psi(X(t))\psi^{\scriptscriptstyle T}(X(t))\,, \quad z_T = \frac{1}{T}\sum_{t=1}^T c(X(t))\varphi(t)$$

The LSTD learning algorithm for average cost defines estimates of $r^*$ in (11) via, $r_T = \Sigma_T^{-1} z_T$. This is consistent provided $\psi$ and $h$ are square integrable.

## III. Power management via speed scaling

Dynamic speed scaling is an increasingly common approach to power management in computer system design. The goal is to control the processing speed so as to optimally balance energy and delay costs – reducing (increasing) the speed in times when the workload is small (large).

We model the dynamic speed scaling problem as a single server queue with controllable service rate. Specifically, we assume that jobs arrive to a single processor and are processed at a rate determined by the current power. The primary model is described in discrete time: For each $t = 0, 1, 2, \dots$ we let $A(t)$ denote the job arrivals in this time slot, $Q(t)$ the number of jobs awaiting service, and $U(t)$ the number of services. It is assumed that $\boldsymbol{A}$ is i.i.d. Hence the MDP model is described as the controlled random walk,

$$Q(t+1) = Q(t) - U(t) + A(t+1), \qquad t \geq 0. \tag{12}$$

This is an MDP model of the form (1) with $\boldsymbol{X} \equiv \boldsymbol{Q}$. The cost function we consider balances the cost of delay with the energy cost associated with the processing speed:

$$c(x,u) = x + \beta\mathcal{P}(u), \tag{13}$$

where $\mathcal{P}$ denotes the power required as a function of the speed $u$, and $\beta > 0$. This form of cost function is common in the literature, e.g., [4], [11].

The remaining piece of the model is to define the form of $\mathcal{P}$. In this paper, we consider two forms of $\mathcal{P}$ based on two different applications. For processor design applications $\mathcal{P}(u) \propto u^{\varrho}$ [11] and for wireless transmission applications $\mathcal{P}(u) \propto e^{\kappa u}$ [12].

## IV. Approximate models

In this section we study the fluid and diffusion approximations of the speed scaling model described in (12). The solutions to these approximate models will later serve as the basis for applying TD learning to determine an approximately optimal control of the speeds.

### A. The fluid model

The fluid model corresponding to the speed scaling model (12) is given by:

$$\tfrac{d}{dt}q(t) = -u(t) + \alpha, \tag{14}$$

where $\alpha$ is the mean of $A(t)$, and the control $u(t)$ and buffer contents $q(t)$ are assumed to be non-negative valued.

It is assumed here that the cost function vanishes at the equilibrium $q(t) = 0$, $u(t) = \alpha$. In this case the total cost $J^*$ defined in (6) is finite for each $x$. The infimum in (6) is over all feasible $\boldsymbol{u}$. Feasibility means that $u(t) \geq 0$ for each $t$, and the resulting state trajectory $\boldsymbol{q}$ is also non-negative valued. In this section we consider two classes of normalized cost functions,

$$\begin{aligned} &\textit{Polynomial cost} \quad c(x,u) = x + \beta([u-\alpha]_+)^{\varrho} \\ &\textit{Exponential cost} \quad c(x,u) = x + \beta[e^{\kappa u} - e^{\kappa\alpha}]_+ \end{aligned} \tag{15}$$

where $[\,\cdot\,]_+ = \max(0, \cdot\,)$, and the parameters $\beta, \kappa, \varrho$ are positive. The normalization is used to ensure that $c(0,\alpha) = 0$. Observe that the cost is also zero for $u < \alpha$ when $x = 0$. However, it can be shown that the $u$ that achieves the infimum in (6) is never less than $\alpha$.

We now return to (8) to show that the fluid value function provides a useful approximation to the solution to the average cost optimality equations. We construct a cost function $c^\circ$ that approximates $c$, along with a constant $\eta^\circ > 0$ such that $J^*$ satisfies the ACOE for this cost function:

$$\min_{0\leq u\leq x}\{c^\circ(x,u) + P_u J^*(x)\} = J^*(x) + \eta^\circ. \tag{16}$$

This construction is based on the two error functions,

$$\begin{aligned} \mathcal{E}(x,u) &= c(x,u) - J^*(x) + P_u J^*(x) \\ \underline{\mathcal{E}}(x) &= \min_{0\leq u\leq x} \mathcal{E}(x,u) \end{aligned} \tag{17}$$

The constant $\eta^\circ \in \mathbb{R}_+$ is arbitrary, and the perturbation of the cost function is defined as

$$c^\circ(x,u) = c(x,u) - \underline{\mathcal{E}}(x) + \eta^\circ$$

Based on the definition of $\underline{\mathcal{E}}$, we conclude that (16) is satisfied. To demonstrate the utility of this construction it remains to obtain bounds on the difference between $c$ and $c^\circ$.

We begin with some structural results for the fluid value function. Proofs are omitted due to lack of space. Note that part (ii) is obtained from bounds on the "Lambert $W$ function" [5].

**Proposition 1.** *For any of the cost functions defined in* (15)*, the fluid value function $J^*$ is increasing, convex, and its second derivative $\nabla^2 J^*$ is non-increasing. Moreover,*

(i) *For polynomial cost the value function and optimal policy are given by, respectively,*

$$J^*(x) \;=\; x^{\frac{2\varrho-1}{\varrho}} \frac{\varrho}{2\varrho-1}\Big(\frac{1}{\beta(\varrho-1)}\Big)^{\frac{\varrho-1}{\varrho}} \tag{18}$$

$$\phi^{\text{F}*}(x) \;=\; \Big(\frac{x}{\beta(\varrho-1)}\Big)^{1/\varrho} + \alpha, \qquad x \in \mathbb{R}_+. \tag{19}$$

(ii) *For exponential cost the value function satisfies the following upper and lower bounds: On setting $\tilde\beta = \beta e^{\kappa\alpha}$ and $\tilde x = x - \tilde\beta$, there are constants $C_-, C_+$ such that, whenever $x \geq \tilde\beta(e^2+1)$,*

$$C_- + \frac{\kappa}{2}\frac{\tilde x^2}{\log(\tilde x) - \log(\beta) - (\kappa\alpha+1)} \leq J^*(x) \leq C_+ + \frac{\kappa}{2}\tilde x^2$$

Part (i) of the above proposition exposes a connection between the fluid control policy and prior results about speed scaling obtained in the literature on worst-case algorithms [2]. In particular, the optimal fluid control corresponds to a speed scaling scheme that is known to have a small competitive ratio.

Next, we can derive a lower bound on the difference $c - c^\circ$ relatively easily.

**Lemma 2.** $\mathcal{E}(x,u) \geq 0$ *everywhere, giving* $c \geq c^\circ - \eta^\circ$.

*Proof:* Convexity of $J^*$ gives the bound,

$$J^*(Q(t+1)) - J^*(Q(t)) \geq \nabla J^*(Q(t)) \cdot (Q(t+1) - Q(t))$$

Consequently, for each $x \in \mathbb{R}_+, u \in \mathbb{R}_+$ we have the lower bound,

$$\begin{aligned} P_u J^*(x) &= J^*(x) + \mathsf{E}_{x,u}[J^*(Q(1)) - J^*(Q(0))] \\ &\geq J^*(x) + \mathsf{E}_{x,u}[\nabla J^*(Q(0)) \cdot ((Q(1)) - Q(0))] \\ &= J^*(x) + \nabla J^*(x) \cdot (-u + \alpha) \end{aligned}$$

From the definition (17) this gives,

$$\mathcal{E}(x,u) \geq c(x,u) + \nabla J^*(x) \cdot (-u+\alpha)$$

Non-negativity follows from the TCOE (5). □

Further, we can derive an upper bound on $c - c^\circ$ in two simple steps. We first write,

$$\underline{\mathcal{E}}(x) \leq \mathcal{E}(x, \phi^{\text{F}*}(x)) \tag{20}$$

where $\phi^{\text{F}*}(x)$ is the optimal policy for the fluid model given in (7). Next we apply the second order Mean Value Thoerem to bound $\mathcal{E}$. Given $Q(0) = x$ and $U(0) = u$ we have $Q(1) = x - u + A(1)$. For some random variable $\overline{Q}$ between $x$ and $x - u + A(1)$ we have

$$\begin{aligned} \mathcal{D}_u J^*(x) &:= \mathsf{E}_{x,u}[J^*(Q(1)) - J^*(Q(0))] \\ &= \nabla J^*(x) \cdot (-u+\alpha) \\ &\quad + \tfrac{1}{2}\mathsf{E}\left[\nabla^2 J^*\left(\overline{Q}\right) \cdot (-u + A(1))^2\right] \end{aligned} \tag{21}$$

Proposition 1 states that the second derivative of $J^*$ is non-increasing. Hence we can combine (21) with (20) to obtain,

$$\underline{\mathcal{E}}(x) \leq \tfrac{1}{2}\mathsf{E}\left[\nabla^2 J^*(x - \phi^{\text{F}*}(x)) \cdot (-\phi^{\text{F}*}(x) + A(1))^2\right]. \tag{22}$$

Lemma 3 provides an implication of this bound in the special case of quadratic cost.

**Lemma 3.** *For polynomial cost* (15) *with* $\varrho = 2$, $\beta = \frac{1}{2}$, *we have* $\underline{\mathcal{E}}(x) = \mathcal{O}(\sqrt{x})$*, and hence* $c(x,u) \leq c^\circ(x,u) + \mathcal{O}(\sqrt{x})$*.*

*Proof:* The optimal policy is given in (19), giving $\phi^{\text{F}*}(x) = \mathcal{O}(\sqrt{x})$ in this special case. The formula (18) gives $\nabla^2 J^*(x) = \mathcal{O}(1/\sqrt{x})$. The bound (22) then gives $\underline{\mathcal{E}}(x) = \mathcal{O}(\sqrt{x})$. □

Lemma 4 is an extension to the case of exponential cost. There is no space here for a proof.

**Lemma 4.** *For exponential cost* (15)*, with* $\beta = 1$*, we have* $\underline{\mathcal{E}}(x) \leq \kappa \log(x)^2$ *for all* $x$ *sufficiently large. For such* $x$ *we have* $c(x,u) \leq c^\circ(x,u) - \eta^\circ + \kappa \log(x)^2$*.* □

Hence, for quadratic or exponential cost, the fluid value function $J^*$ can be interpreted as the relative value function for a cost function that approximates $c(x,u)$.

## *B. The diffusion model*

We next consider the diffusion model introduced in (9). We motivate the model using the second order Taylor series approximation (21). This continuous-time model will be used to obtain additional insight regarding the structure of $h^*$.

The ACOE for the diffusion model is similar to the total cost DP equation for the fluid model:

$$\min_{u \geq 0}\{c(x,u) + \mathcal{D}_u h^*(x)\} = \eta^* \tag{23}$$

where $\eta^*$ is the average cost, $h^*$ is called the relative value function, and $\mathcal{D}_u$ denotes the usual differential generator. This is defined for $C^2$ functions $g\colon \mathbb{R}_+ \to \mathbb{R}_+$ via,

$$\mathcal{D}_u g(x) = \frac{d}{dx} g(x)(-u+\alpha) + \tfrac{1}{2}\sigma^2(u)\frac{d^2}{dx^2} g(x)$$

However, for a *reflected* diffusion the domain of the differential generator is restricted to those $C^2$ functions satisfying the boundary condition,

$$\frac{d}{dx} g(x)\Big|_{x=0} = 0 \tag{24}$$

This is imposed so that the reflection term vanishes in the Ito formula:

$$dg(Q(t)) = f_g(Q(t), U(t))\, dt + \sigma(U(t))\frac{d}{dx} g(Q(t)) dN(t)$$

with $f_g(x,u) = \mathcal{D}_u g(x)$.

The variance term is selected so that the action of the differential generator on a smooth function will be similar to that of the discrete generator. The second order Taylor series expansion (21) suggests the value:

$$\sigma^2(u) = \mathsf{E}[(u - A(1))^2] = u^2 - 2\alpha u + m_A^2,$$

where $m_A^2$ is the second moment of $A(1)$. We adopt this form in the remainder of this section.

Further, for the remainder of the section, we restrict to the case of quadratic cost:

$$c(x,u) = x + \tfrac{1}{2}u^2, \tag{25}$$

In this case the minimizer in (23) is given by,

$$\phi^*(x) := \frac{\nabla h^*(x) + \alpha \nabla^2 h^*(x)}{1 + \nabla^2 h^*(x)} \tag{26}$$

It can be shown that $h^*$ is convex. Consequently, subject to the boundary condition (24), it follows that $\phi^*(x) \geq 0$ for each $x$. Substituting (26) into (23) gives the fixed point equation,

$$x + \alpha \nabla h^* + \tfrac{1}{2} m_A^2 \nabla^2 h^* - \frac{(\alpha \nabla^2 h^* + \nabla h^*)^2}{2(1 + \nabla^2 h^*)} = \eta^*. \tag{27}$$

Although the cost function (25) does not satisfy $c(0,\alpha) = 0$, the TCOE (5) for the fluid model admits the solution,

$$J^*(x) = \alpha x + \tfrac{1}{3}[(2x + \alpha^2)^{3/2} - \alpha^3] \tag{28}$$

Furthermore, the function $h^\circ(x) = J^*(x) + \frac{1}{2}x$ approximately solves the dynamic programming equation for the diffusion. In fact, it is straightforward to show that $h^\circ(x)$ solves the ACOE for the diffusion exactly under a modified cost function:

$$c^\circ(x,u) = c(x,u) + \frac{1}{8}\Big(\frac{y}{y+1} - 4\frac{\sigma_A^2}{y}\Big) + \eta^\circ,$$

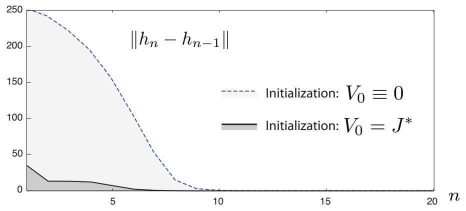

Fig. 1: The convergence of value iteration for the quadratic cost function (25). The error $\|h_{n+1} - h_n\|$ converges to zero *much faster* when the algorithm is initialized using the fluid value function.

where $\sigma_A^2 = m_A^2 - \alpha^2$, and $y := (2x + \alpha^2)^{\frac{1}{2}}$. The constant $\eta^\circ$ is again arbitrary. Regardless of its value, the optimal average cost of $c^\circ$ is equal to $\eta^\circ$. It is also easy to see that $|c^\circ(x,u) - c(x,u)|$ is uniformly bounded over $x$ and $u$.

The only issue that remains is the fact that $h^\circ(x)$ does not satisfy the boundary condition (24) since

$$\nabla h^\circ(x)\Big|_{x=0} = 2\alpha + \tfrac{1}{2}.$$

This gap is resolved through an additional perturbation. Specifically, fix $\vartheta > 0$, and introduce the decaying exponential,

$$h^{\circ\circ}(x) = h^\circ(x) - (2\alpha + \tfrac{1}{2})\vartheta e^{-x/\vartheta}$$

The gradient vanishes at the origin following this perturbation. This function solves the ACOE for the diffusion for a function $c^{\circ\circ}$ which retains the property that $c^{\circ\circ}(x,u) - c(x,u)$ is uniformly bounded.

Based on this form we are motivated to enlarge the basis to approximate the relative value function with $\psi^1 = J^*$ and $\psi^2(x) \equiv x$.

## V. Experimental results

In this section we present results from experiments conducted for the speed scaling model described in Section III. Each of the value function approximations used in these experiments were based on insights obtained from the fluid and diffusion models.

In all of the numerical experiments described here the arrival process $\boldsymbol{A}$ is a scaled geometric distribution,

$$A(t) = \Delta_A G(t), \qquad t \geq 1, \tag{29}$$

where $\Delta_A > 0$ and $\boldsymbol{G}$ is geometrically distributed on $\{0, 1, \dots\}$ with parameter $p_A$. The mean and variance of $A(t)$ are given by, respectively,

$$m_A = \Delta_A \frac{p_A}{1 - p_A}, \qquad \sigma_A^2 = \frac{p_A}{(1 - p_A)^2}\Delta_A^2. \tag{30}$$

### A. Value iteration

We begin by computing the actual solution to the average cost optimality equation using value iteration. This provides a reference for evaluating the proposed approach for TD learning. We restrict to the special case of the quadratic cost function given in (25) due to limited space. The arrival process is taken of the form (29), with $p_A = 0.96$ and $\Delta_A$ chosen so that the mean $m_A$ is equal to unity:

$$1 = m_A = \Delta_A \frac{p_A}{1 - p_A} \quad \textit{and} \quad \Delta_A = 1/24 \tag{31}$$

The state space is truncated for practical implementation of value iteration. In the experiments that follow we take $\mathsf{X} = \{\Delta_A m : m = 0, \dots, N_\ell\}$ with $N_\ell = 480$. The model becomes,

$$Q(t+1) = [Q(t) - U(t) + A(t+1)], \qquad t \geq 0,$$

where $[\,\cdot\,]$ represents projection to the interval $[0, 20]$, and $U(t)$ is restricted to non-negative integer multiples of $\Delta_A$.

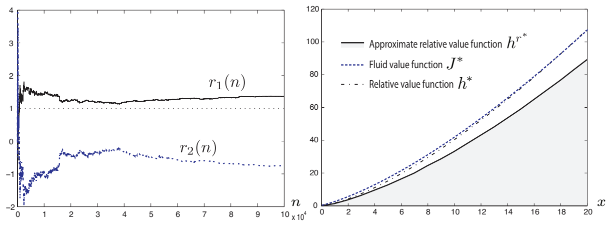

Fig. 2: Simulation results for the dynamic speed scale model with quadratic cost. The plot on the left shows estimates of the coefficients in the optimal approximation of $h^*$ using the basis obtained from the fluid and diffusion models (see (33)). In the plot on the right the final approximation $h^{r^*}$ is compared to the fluid value function and the relative value function.

Let $V_n$ denote the $n$th value function obtained. The approximate solution to the ACOE at stage $n$ is taken to be the normalized value function $h_n(x) = V_n(x) - V_n(0)$, $x \in \mathsf{X}$. The convergence of $\{h_n\}$ to $h^*$ is illustrated in Fig. 1. The comparison of $J^*$ and $h^*$ shown in Fig. 2 was computed using this algorithm.

Shown in Fig. 3 is the optimal policy and the $(c, J^*)$-myopic policy, $\phi^J(x) = \arg\min_{0\leq u\leq x}\{c(x,u) + P_u J^*(x)\}$.

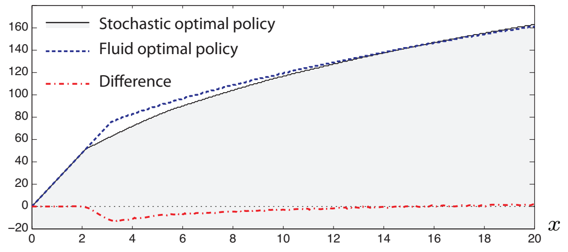

Fig. 3: The optimal policy compared to the $(c, J^*)$-myopic policy for the quadratic cost function (25).

### B. TD learning

We are now ready to apply TD learning to approximate the relative value function in the case of a specific policy. The policies considered here are taken to be the following translation of the optimal policy for the fluid model,

$$\phi_\diamond^{F*}(x) = \lfloor \min(x, \phi^{F*}(x)) \rfloor, \qquad x \in \mathbb{R}_+ \tag{32}$$

where here $x$ is restricted to the lattice on which $\boldsymbol{Q}$ evolves, and $\lfloor a \rfloor$ indicates the nearest point on this lattice for $a \in \mathbb{R}_+$. In the next section we show how to combine TD learning and policy improvement in order to determine an approximately optimal solution.

We consider only polynomial costs due to space constraints. Additionally, we maintain the arrival distribution defined by (29), and the specification $p_A = 0.96$ used in the previous subsection. We consider several values of $\Delta_A$ to investigate the impact of variance on the estimation algorithm.

We take the following as the basis for TD learning

$$\psi_1(x) = J^*(x), \quad \psi_2(x) = x, \qquad x \geq 0. \tag{33}$$

In the special case of quadratic cost, with $c(x,u) = x + \frac{1}{2}u^2$, this choice is motivated by the diffusion approximations presented in Sec. IV-B. We begin with results in this special case. Recall that the case of quadratic costs models the scenario of speed scaling in microprocessors.

The fluid value function $J^*$ associated with the quadratic cost function (25) is given in (28). Fig. 2 shows a result obtained after 100,000 iterations of the LSTD algorithm. The initial condition was taken to be $r(0) = (0,0)^{\intercal}$. The value of the coefficient $r_1^*$ corresponding to $\psi_1 = J^*$ was found to be close to unity. Hence the approximate relative value function $h^{r^*}$ is approximated by $J^*$, where $r^*$ is the final value obtained from the LSTD algorithm. This conclusion is plainly illustrated in Fig. 2 where a plot of the function $h^{r^*}$ is compared to the fluid value function $J^*$ and the solution to the ACOE $h^*$.

### C. TD learning with policy improvement

So far, the TD learning algorithm was used to compute an approximation of the relative value function for the specific policy given in (32). In this section, we construct a policy using TD learning and policy improvement.

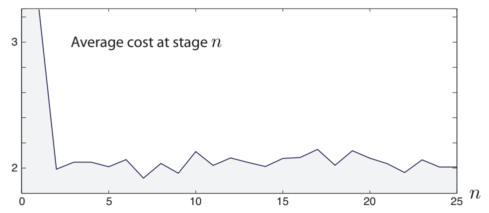

Fig. 4: Simulation result for TDPIA with the quadratic cost function (25), and basis $\{\psi_1, \psi_2\} \equiv \{J^*, x\}$.

The policy iteration algorithm (PIA) is a method to construct an optimal policy through the following steps. The algorithm is initialized with a policy $\phi^0$ and then the following operations are performed in the $k$th stage of the algorithm:

(i) Given the policy $\phi^k$, find the solution $h^k$ to Poisson's equation $P_{\phi^k}h^k = h^k - c_k + \eta_k$, where $c_k(x) = c(x, \phi^k(x))$, and $\eta_k$ is the average cost.

(ii) Update the policy via $\phi^{k+1}(x) \in \arg\min_u\{c(x,u) + P_u h^k(x)\}$.

In order to combine TD learning with PIA, the TDPIA algorithm considered replaces the first step with an application of the LSTD algorithm, resulting in an approximation $h^k_{\text{TD}}$ to the function $h^k$. The policy in (ii) is then taken to be $\phi^{k+1}(x) \in \arg\min_u\{c(x,u) + P_u h^k_{\text{TD}}(x)\}$.

We illustrate this approach in the case of the quadratic cost function (25), using the basis given in (33). The initial policy was taken to be $\phi^0(x) = \min(x,1)$, $x \geq 0$. Fig. 4 shows the estimated average cost in each of the twenty iterations of the algorithm. The algorithm results in a policy that is nearly optimal after just a few iterations.

## VI. Concluding remarks

The main message of this paper is that idealized models (fluid and diffusion approximations) are useful for determining the function class for TD learning. This approach is applicable for control synthesis and performance approximation of Markov models in a wide range of applications. The motivation for this approach is a simple Taylor series argument that can be used to bound the difference between the relative value function $h^*$ and the fluid value function $J^*$.

To illustrate the application of this approach for TD learning, this paper focuses on a power management problem: dynamic speed scaling. This application reveals that this approach to approximation yields remarkably accurate results. In particular, numerical experiments revealed that (i) value iteration initialized using the fluid approximation results in much faster convergence, and (ii) policy iteration coupled with TD learning quickly converges to an approximately optimal policy when the fluid and diffusion models are considered in the construction of a basis.

## Acknowledgment

Financial support from the National Science Foundation (ECS-0523620 and CCF-0830511), ITMANET DARPA RK 2006-07284, and Microsoft Research is gratefully acknowledged.[1]

## References

[1] Intel Xscale.
[2] Nikhil Bansal, Tracy Kimbrel, and Kirk Pruhs. Speed scaling to manage energy and temperature. *J. ACM*, 54(1):1–39, March 2007.
[3] D.P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Atena Scientific, Cambridge, Mass, 1996.
[4] Jennifer M. George and J. Michael Harrison. Dynamic control of a queue with adjustable service rate. *Operations Research*, 49(5):720–731, September 2001.
[5] A. Hoorfar and M. Hassani. Inequalities on the lambert $w$ function and hyperpower function. *Journal of Inequalities in Pure and Applied Mathematics (JIPAM)*, 9(2), 2008.
[6] S. Mannor, I. Menache, and N. Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Oper. Res.*, 134(2):215–238, 2005.
[7] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, Cambridge, 2007.
[8] C.C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queueing networks. Submitted for publication., 2006.
[9] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Automat. Control*, 42(5):674–690, 1997.
[10] M. H. Veatch. Approximate dynamic programming for networks: Fluid models and constraint reduction, 2004. Submitted for publication.
[11] A. Wierman, L. Andrew, and A. Tang. Power-aware speed scaling in processor sharing systems. In *Proc. of INFOCOM*, pages 2007–15, 2009.
[12] L. Xie and P. R. Kumar. A network information theory for wireless communication: scaling laws and optimal operation. *IEEE Trans. on Info. Theory*, 50(5):748–767, 2004.

[1]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, DARPA, or Microsoft.