Les Cahiers du GERAD ISSN: 0711–2440

Self Learning Control of Constrained Markov Decision Processes – A Gradient Approach

Felisa J. Vázquez Abad
Vikram Krishnamurthy

G–2003–51

August 2003

Self Learning Control of Constrained Markov Decision Processes – A Gradient Approach

Felisa J. Vázquez Abad
Département d'informatique et recherche opérationnelle
Université de Montréal, Québec Canada H3C 3J7
and GERAD
vazquez@iro.umontreal.ca

Vikram Krishnamurthy
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada V6T 1Z4
vikramk@ece.ubc.ca

August, 2003

Les Cahiers du GERAD
G–2003–51

Abstract

We present stochastic approximation algorithms for computing the locally optimal policy of a constrained average cost finite state Markov Decision process. Because the optimal control strategy is known to be a randomized policy, we consider here a parameterization of the action probabilities to establish the optimization problem. The stochastic approximation algorithms require computation of the gradient of the cost function with respect to the parameter that characterizes the randomized policy. This is computed by novel simulation based gradient estimation schemes involving weak derivatives. Similar to neuro-dynamic programming algorithms (e.g. Q-learning or Temporal Difference methods), the algorithms proposed in this paper are simulation based and do not require explicit knowledge of the underlying parameters such as transition probabilities. However, unlike neuro-dynamic programming methods, the algorithms proposed here can handle constraints and time varying parameters. Numerical examples are given to illustrate the performance of the algorithms.

Résumé

Nous considérons le problème du contrôle optimale des chaînes de Markov commandées (MDP) avec des contraintes. Sous la randomisation des actions, le problème est parametrisé et peut se ré-écrire en terme d'un problème d'optimisation non-linéaire avec des contraintes non -linéaires, pour lequel on applique l'approximation stochastique. Celle-ci a besoin de calculer certains gradients par rapport aux paramètres de contrôle. Nous proposons une nouvelle méthode pour l'estimationde tels gradients avec des dérivées faibles. Notre méthode est robuste (comme les algorithmes de Q-learning et des differences temporales) et ne suppose pas que les probabilités de transition soient connues. Par ailleurs, notre méthode peut être appliquée lors qu'il y a des contraintes, contrairement aux autres méthodes mentionées. Nous présentons aussi des exemples numériques pour illustrer la performance de nos algorithmes.

Acknowledgments: This work was done while the first author was on leave at the Department of Electrical and Electronic Engineering at the University of Melbourne. The research was supported by the Australian Research Council and research grants from NSERC, Canada and FCAR, Québec.

1 Introduction

This paper presents stochastic approximation based algorithms for computing the optimal policy of a constrained average cost finite-state Markov Decision Process (MDP). Because the optimal control strategy of such problems is known to be a stationary randomized policy (refer to Section 1.2), in this work we focus on randomized policies to pose an optimization problem for the solution of the MDP. The problem can be stated in general in terms of any convenient parameterization of the action probabilities, so that the optimization variable is a vector-valued parameter. The stochastic approximation algorithms require computation of the gradient of the cost function with respect to the parameter that characterizes the class of randomized policies. This is computed by novel simulation based gradient estimation schemes involving weak derivatives. Several methods exist to compute derivative estimators for stochastic processes. Under Lipschitz continuity of the sample performance, it is usually the case that expectations and derivatives can be interchanged (using the dominated convergence theorem) so that the sample derivative is unbiased. This approach is the basis for the "pathwise" estimation techniques such as Infinitesimal Perturbation Analysis (IPA) [11]. In the case of Markov chains, infinitesimal changes in the transition probabilities give rise to jumps in the next state of the process and often implies that discontinuities occur. In this case the pathwise analysis has been extended [9] using the concept of the "realization perturbation factors", which are equivalent to the weak derivative formulation via the Poisson equation in [23]. In this paper we use a robust version of the weak derivative estimators that does not require knowledge of the underlying distribution. Similar to neuro-dynamic programming algorithms [7] (e.g., Q-learning and Temporal Difference methods), the algorithms proposed in this paper are simulation based and do not require explicit knowledge of the underlying parameters of the MDP such as transition probabilities (or equivalently invariant distributions). However, unlike Q-learning or Temporal Difference methods, the algorithms proposed here straightforwardly can handle constraints.

1.1 The canonical control problem

To avoid excessive technicalities, our setting is an average cost, unichain, finite state Markov decision process. Consider a unichain Markov Decision Process (MDP) $\{X_n\}$ with finite state space S, as will be defined shortly. Throughout, $n = 0, 1, \ldots$ denotes discrete time. When the system is in state $i \in S$, a finite number of possible actions in the set $\mathcal{A}(i)$ can be taken, where $\mathcal{A} = \cup_{i \in S} \mathcal{A}(i)$ is assumed to be finite. The evolution of the system is Markovian, and the kernel $A(u)$ depends on the action $u \in \mathcal{A}(i)$, that is:

$$\mathbb{P}[X_{n+1} = j | X_n = i, u_n = u] = A_{ij}(u), \quad u \in \mathcal{A}(i). \tag{1}$$

Denote by $\mathfrak{F}_n$ the σ-algebra generated by $(X_1, \ldots, X_n, u_1, \ldots, u_{n-1})$. The filtration of the process is the increasing sequence of σ-algebras $\{\mathfrak{F}_n, n = 1, 2, \ldots\}$. The most general model for control strategies is the case where u_n is a random function whose distribution

depends on the observed trajectory of the process, that is, where u_n is measurable with respect to $\mathfrak{F}_n$ (or "adapted" to the filtration). These are called the **admissible** policies:

$$\mathcal{U} = \{\mathbf{u} = \{u_n\} : u_n \text{ is measurable w.r.t. } \mathfrak{F}_n,\ \forall n \in \mathbb{N}\},$$

which basically means that u_n is a (possibly random) function of $(X_1, \ldots, X_n, u_1, \ldots, u_{n-1})$. By *unichain* [25, pp.348] we mean that every policy where u_n is a deterministic function of X_n consists of a single recurrent class plus possibly an empty set of transient states.

The cost incurred at stage n is given by a known bounded function $c(X_n, u_n) \geq 0$ where $c : S \times \mathcal{A} \to \mathbb{R}$. For any admissible policy $\mathbf{u} \in \mathcal{U}$ define the average cost

$$J_{x_0}(\mathbf{u}) = \lim_{N\to\infty} \sup \frac{1}{N} \mathbb{E}_{\mathbf{u}} \left[\sum_{n=1}^{N} c(X_n, u_n) \mid X_0 = x_0 \right]. \tag{2}$$

The aim is to compute the optimal policy $\mathbf{u}^* \in \mathcal{U}$ that satisfies

$$J_{x_0}(\mathbf{u}^*) = \inf_{\mathbf{u}\in\mathcal{U}} J_{x_0}(\mathbf{u}) \quad \forall x_0 \in S \tag{3}$$

i.e., $\mathbf{u}^*$ has the minimum cost for all initial states $x_0 \in S$.

Denote by $\mathcal{U}_0$ the class of randomized stationary Markovian policies, comprising all elements $\mathbf{u}$ that are stochastic processes (adapted to $\{\mathfrak{F}_n\}$) of the form:

$$\mathcal{U}_0 = \{\mathbf{u} = (u_n, n \in \mathbb{N}) : \forall n, u_n \sim \mu(X_n), \quad \mu \colon x \to \mathcal{M}(\mathcal{A}(x)),\ \forall x \in S\}$$

where for any set G, $\mathcal{M}(G)$ denotes the set of probability measures over G. In other words, a randomized stationary policy is completely determined by a (conditional) distribution $\mu(x)$ on $\mathcal{A}(x), x \in S$. By the unichain assumption, the limit in (2) exists for any $x_0 \in S$ and $\mathbf{u} \in \mathcal{U}_0$, and the limit is independent of x_0.

Unconstrained MDP. It is well known [3] that the optimal stationary policy for (3) is a non-randomized Markovian policy, i.e., $u_n^* = \mathbf{u}(X_n) \in \mathcal{A}(X_n)$ with $\mathbf{u}$ being a deterministic mapping (equivalently $\mathcal{M}$ is a degenerate probability measure). Thus the solution of the MDP (3) is obtained by determining the function $\mathbf{u}(i) \in \mathcal{A}(i), i \in S$ which selects the action to be taken at each state. Such policies are called "pure policies".

Constrained MDP. Motivated by several problems in telecommunication network optimization, we consider the MDP (3) subject to L average constraints of the form

$$\lim_{N\to\infty} \sup \frac{1}{N} \mathbb{E}_{\mathbf{u}} \left[\sum_{n=1}^{N} \beta_l(X_n, u_n) \right] \leq \gamma_l, \quad l = 1, \ldots, L \tag{4}$$

where $\beta_l : S \times \mathcal{A} \to \mathbb{R}$ are known bounded functions and γ_l are known constants. It is well known [3] that if $L > 0$ then the optimal policy $\mathbf{u}^* \in \mathcal{U}$. Indeed, the optimal policy is randomized for at most L of the states.

Remark: Since we are considering unichain MDPs, the above average constraints are equivalent to the sample path constraints [27]

$$\mathbb{P}_{\mathbf{u}}\left[\lim\sup_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\beta_l(X_n,u_n)\le\gamma_l\right]=1$$

Such sample path constraints are often used in admission control of telecommunication networks to depict quality of service (QoS) constraints, see [26, 28].

We consider a "simulation" based system. By simulation based we mean that although the transition probabilities $A(u)$, $u\in\mathcal{A}(i)$, $i\in S$ of the underlying system are unknown, the system can be observed under any choice of control actions $\mathbf{u}=\{u_n\}$. In other words, the on-line learning methods that we study must be adapted to the filtration $\{\mathfrak{F}_n; n\ge 0\}$. In this paper we derive and implement a new simulation based method using ideas in stochastic discrete event systems for devising descent algorithms of the average cost function (2). The resulting algorithms are provably convergent to a Kuhn Tucker point, see Section 1.4 for a detailed list of contributions.

1.2 Parameterized Randomized Policies

Randomized policies yield a convenient parameterization of the above model [25]. Define the action probabilities θ parameterized by ψ as:

$$\begin{aligned}\mathbb{P}[u_n=a|X_n=i] &= \theta_{ia}(\psi),\quad a\in\mathcal{A}(i),\ i\in S \\ \text{where}\ \sum_{a\in\mathcal{A}(i)}\theta_{ia}(\psi) &= 1\ \text{for every state}\ i\in S.\end{aligned}\tag{5}$$

Here $\psi\in\Psi$ is a finite dimensional vector which parameterizes the action probabilities θ. Ψ is some suitably defined compact subset of the Euclidean space. The unconstrained problem with pure optimal policy mentioned above is a degenerate case where for each $i\in S$ $\theta_{ia}=1$ for some $a\in\mathcal{A}(i)$.

In this paper we focus on two possible parameterizations ψ:

1. Canonical Coordinates $\psi=\theta$**.** The most obvious parameterization of the action probabilities θ is to choose $\psi=\theta$. Thus $\psi=\{\psi_i\}=\{\theta_i\}$, $i\in S$ is the set of action probability vectors $(\psi_{ia}; a\in\mathcal{A}(i))$ satisfying (5). Let $\theta\in\Theta$ where Θ denotes the compact set

$$\Theta=\left\{\theta_i\in[0,1]^{\mathcal{A}(i)}:\sum_{a\in\mathcal{A}(i)}\theta_{ia}=1; i\in S\right\}.\tag{6}$$

A disadvantage of the canonical parameterization is that the gradient algorithms we will use to compute estimates of θ need to cope with both equality and inequality constraints in (6).

2. Spherical Coordinates $\psi = \alpha$**.** Suppose without loss of generality that $\mathcal{A}(i) = \{0, \ldots, d(i)\}$. A novel solution was introduced in [16] for Hidden Markov Model parameter estimation through the use of spherical coordinates. Adapted to our MDP problem, it reads as follows: Fix the control agent $i \in S$. To each value $\theta_{ia}, a \in \mathcal{A}(i)$ associate the values $\lambda_{ia} = \sqrt{\theta_{ia}}$. Then (5) yields $\sum_{a\in\mathcal{A}(i)} \lambda_{ia}^2 = 1$ and λ_{ia} can be interpreted as the coordinates of a vector that lies on the surface of the unit sphere in $\mathbb{R}^{d(i)+1}$, where $d(i)+1$ is the size of $\mathcal{A}(i)$. In spherical coordinates, the angles are $\alpha_{ip}, p = 1, \ldots d(i)$, and the radius is always of size unity. If, for instance, $d(i) = 2$ (i.e., there are 3 actions), then

$$\lambda_{i0} = \cos(\alpha_{i1}), \lambda_{i1} = \sin(\alpha_{i1})\cos(\alpha_{i2}), \text{ and } \lambda_{i2} = \sin(\alpha_{i1})\sin(\alpha_{i2}), \tag{7}$$

(because the radius is one). For the general case where $d(i) \geq 2$ in $\mathcal{A}(i) = \{0, \ldots, d(i)\}$, the spherical coordinates parameterization α reads as follows: $\lambda_{ia} = \sqrt{\theta_{ia}}$ and

$$\lambda_{ia} = \begin{cases} \cos(\alpha_{i,1}) & \text{if } a = 0 \\ \cos(\alpha_{i,(a+1)}) \prod_{k=1}^{a} \sin(\alpha_{i,k}) & 1 \leq a \leq d(i)-1 \\ \sin(\alpha_{i,d(i))}) \prod_{k=1}^{d(i)-1} \sin(\alpha_{i,k}) & a = d(i) \end{cases} \tag{8}$$

Thus $\alpha_{ip} \in \boldsymbol{\alpha}$ where $\boldsymbol{\alpha}$ denotes the compact set

$$\boldsymbol{\alpha} = \{\alpha_{ip} \bmod 2\pi \in [0, 2\pi];\ i \in S, p \in \mathcal{P}(i)\} \tag{9}$$

It is clear that under this α parameterization, the control variables $\alpha_{ip}, p \in \mathcal{P}(i) = \{1, \ldots d(i)\}$ do not need to satisfy any physical constraints, since $\sin(\alpha_{i,p}) = \sin(\alpha_{i,p} \bmod 2\pi)$, etc.

1.3 Parameterized Constrained MDP

For convenience, we now formulate the above MDP problem as a stochastic optimization problem, where the instantaneous random cost is independent of ψ but the expectation is with respect to a measure parameterized by ψ. Such a "parameterized integrator" formulation is common in gradient estimation, see [23]. Consider the augmented (obviously homogeneous) Markov chain $Z_n \stackrel{\triangle}{=} (X_n, u_n)$ with state space in $\mathcal{Z} = S \times \mathcal{A}$ and transition probabilities parameterized by ψ given by

$$\begin{aligned} P_{i,a,j,a'}(\psi) \stackrel{\triangle}{=}\ & \mathbb{P}(X_{n+1} = j, u_{n+1} = a' \mid X_n = i, u_n = a) = \theta_{j,a'}(\psi)\, A_{ij}(a), \\ & i, j \in S, a \in \mathcal{A}(i), a' \in \mathcal{A}(j) \end{aligned} \tag{10}$$

From the unichain assumption it follows that for any $\psi \in \Psi$ the chain $\{Z_n\}$ is ergodic, and it possesses a unique invariant probability measure $\pi_{i,a}(\psi); i \in S,\ a \in \mathcal{A}(i)$.

Let $\mathbb{E}_{\pi(\psi)}$ denote expectation w.r.t measure $\pi(\psi)$ parameterized by ψ. From (2) we have

$$J_{x_0}(\mathbf{u}) = \mathbb{E}_{\pi(\psi)}[c(Z)] \stackrel{\triangle}{=} \sum_{i\in S} \sum_{a\in\mathcal{A}(i)} \pi_{i,a}(\psi) c(i,a)$$

Similarly, the L constraints (4) can be expressed as

$$B_l(\psi) \stackrel{\triangle}{=} \mathbb{E}_{\pi(\psi)}[\beta_l(Z)] - \gamma_l = \sum_{i \in S} \sum_{a \in \mathcal{A}(i)} \pi_{i,a}(\psi)\beta_l(i,a) - \gamma_l \leq 0 \quad l = 1, \ldots, L \tag{11}$$

Thus the optimization problem (3) with constraints (4) can be written as

$$\min_{\psi \in \Psi} C(\psi) \quad \text{where } C(\psi) \stackrel{\triangle}{=} \mathbb{E}_{\pi(\psi)}[c(Z)], \tag{12}$$

$$\text{subject to:} \quad B_l(\psi) \leq 0, \quad l = 1, \ldots, L, \text{ and} \tag{13}$$

$$\psi \in \Psi \tag{14}$$

In the sequel we refer to the constraints $\psi \in \Psi$ in (14) as **parameter constraints**. The constraints $B_l(\psi)$ (13) will be called **MDP constraints**. It is clear that the parameter constraints depend on the parameterization ψ used whereas the MDP constraints are part of the MDP problem specification. The problem (12), (14) without the MDP constraints (13) will be called an unconstrained MDP. As should be evident from this formulation, the optimal control problem depends uniquely on the *invariant* distribution $\pi(\psi)$ of the chain, rather than the (unknown) transition probabilities $A_{ij}(u)$ themselves.

We make the following assumptions in the rest of the paper:

Assumption 1 *For $\psi \in \Psi$, $C(\psi), B(\psi) \in C^2$ (twice continuously differentiable).*

Indeed for $\psi = \alpha$ or θ, $C(\cdot)$ and $B(\cdot)$ are analytic functions (and hence infinitely differentiable) in our signal model.

Assumption 2 *The minima ψ^* of (12), (13), (14) are regular, i.e., $\nabla_\psi B_l(\psi^*)$, $l = 1, \ldots, L$ are linearly independent. Then ψ^* belongs to the set of Kuhn Tucker points*

$$KT = \Big\{ \psi^* \in \Psi : \exists\, \mu_l \geq 0,\, l = 1, \ldots, L \text{ such that} \\ \nabla_\psi C + \nabla_\psi B \mu = 0, \quad B'\mu = 0 \Big\} \tag{15}$$

where $\mu = (\mu_1 \ldots, \mu_L)'$. Moreover ψ^* satisfies the second order sufficiency condition $\nabla_\psi{}^2 C(\psi^*) + \nabla_\psi{}^2 B(\psi^*)\mu > 0$ (positive definite) on the subspace $\{y \in \mathbb{R}^L : \nabla_\psi B_l(\psi^*) y = 0\}$ for all $l : B_l(\psi^*) = 0, \mu_l > 0$.

1.4 Summary of Main results

Our approach is to use a gradient based stochastic approximation algorithm to optimize $C(\psi)$ in (12) with respect to the parameterized action probabilities $\theta(\psi)$ defined in (5). The stochastic approximation algorithm is of the form

$$\psi(n+1) = \Pi\left[\psi(n) - \epsilon(n)\, \widehat{\nabla_\psi C}(\psi(n))\right]. \tag{16}$$

(suitably modified with Lagrange multipliers to handle the constraints, see (19), (20)). Here $\widehat{\nabla_\psi C}$ denotes an estimate of the gradient $\nabla_\psi C$, $\Pi[\cdot]$ denotes an appropriate projection to deal with the constraints (13), and (14) and $\epsilon(n) > 0$ denotes the step size. If the parameters of the MDP are time-invariant, then there exists a unique optimum ψ^* for the constrained MDP (12), (13), (14), see (18) below.

1. Parameter Constraints: For the parameter constraints (14), Section 2.1 and Section 2.2 present algorithms in terms of the canonical coordinates θ (involving generalized gradients) and spherical coordinates α. We show that the parameterization used in [4] is equivalent to the canonical coordinates. The advantage of using spherical coordinates in the parameter estimation of Hidden Markov Models has been extensively documented, see [16] and references therein. By studying the ordinary differential equation (ODE) associated with the stochastic gradient algorithm (16), Section 2.3 shows that the MDP algorithms using spherical coordinates have much better convergence properties than those using the generalized gradient.

2. MDP Constraints: For the MDP constraints (13), because $B(\psi)$ is is not *a priori* known (since the invariant measure $\pi(\psi)$ is not known), it is not possible to use the usual gradient projection methods in [18] for example. The first order primal dual method using Lagrangians does not work either due to lack of local convexity. In Section 3, ODE versions of the primal dual algorithms with a penalty function and augmented Lagrangian (multiplier) algorithms are presented for solving the constrained MDP together with a convergence analysis. We show in numerical examples that these algorithms have better performance than gradient-based hard projection algorithms and are computationally more efficient.

3. Measured Valued Derivatives: One important issue in implementing the stochastic gradient algorithm (16) is to construct consistent estimates $\widehat{\nabla_\psi C}(\psi)$ of the gradient $\nabla_\psi C(\psi)$ for any $\psi \in \Psi$. We propose gradient estimators based on measured valued differentiation (weak derivatives) and prove the consistency of these estimators. Such algorithms have recently been proposed in the DES literature - see [23], [9]. There are three widely used methodologies for simulation based gradient estimation – score function method, sample path derivatives and weak derivatives. The Score Function method usually suffers from large variance. In addition, as we will explain later on, the method may yield unbounded variance estimation in some important cases. Section 4 considers the measured valued formulation of [12] that uses finite horizon estimation. In the limit for infinite horizon it recovers the formulas of weak derivatives and sample path estimation via perturbation realization factors.

4. Parameter Free Gradient Estimation: The main idea behind the simulation based learning algorithm in this paper is to compute estimates of the gradient $\widehat{\nabla_\psi C}(\psi)$ without explicit knowledge of the parameters of the MDP. The parameter free gradient estimation algorithms we present in Section 5 use the concept of a "phantom" system.

A phantom system starting at time k is a sequence of states and actions that result by choosing a different action to the nominal system (MDP) at time k, and subsequently propagates with the same transition probability laws. Because of Markov chain coupling, eventually after a finite number of time points the phantom system achieves the same state as the nominal system. After this time point their evolution is identical in distribution and the phantom is said to be dead. The difference between the costs of the nominal trajectory and the phantom trajectory (which is non zero until the phantom dies) can then be used to determine information about the gradient of the cost function with respect to the action probabilities. Conventionally in [10, 9], off-line simulations are used to generate the evolution of a phantom – we will call such phantoms the *Doeblin* phantoms. However, we are interested in solving a MDP with unknown and possibly time varying parameters. In Section 5 we present a novel method based on "cut and paste" techniques which look at the past observation history to create "frozen phantoms". By filtering these frozen phantoms we derive a parameter free consistent algorithm for estimating gradients of the cost without explicit knowledge of the parameters of the MDP and without requiring off-line simulations of the system. Section 5 also analyses the statistical properties (variance versus bias trade-off) of the frozen phantoms. Finally, in numerical examples the performance of the resulting phantom gradient estimation algorithm is compared with the score function method of [4].

5. Adaptive Control of Constrained MDP with time varying parameters: Putting the parameter free gradient estimation algorithm into a stochastic approximation algorithm (16) results in a new simulation based algorithm for solving the constrained MDP. As explained below (Section 1.5), just like neuro-dynamic programming methods, explicit knowledge of the parameters of the constrained MDP are not required. Choosing a decreasing step size for (16) of the form $\epsilon(n) = n^{-\gamma}$, where $0.5 < \gamma \leq 1$, results in estimates $\psi(n) \to \psi^*$ almost surely (under suitable regularity and consistency assumptions on the gradient estimator). However, in this paper we are concerned with the case where the MDP parameters (e.g., transition probabilities A) can slowly change with time. In such cases, it is appropriate to chose $\epsilon(n)$ as a constant (non-decreasing) step size to track the time varying parameter ψ. In Section 6, details of constant step size stochastic approximation algorithms based on the ODEs of Section 3 for solving the constrained MDP are presented. Weak convergence of the estimates $\psi(n)$ to ψ^* is established for the constant step size algorithm (16). Section 7 demonstrates the performance of the algorithms on a constrained MDP with time varying transition probabilities.

1.5 Other approaches

The above problem (12) is a constrained stochastic adaptive control problem involving a finite state Markov decision process – adaptive because the underlying parameter $A(u)$ is unknown [17]. There are two methodologies that are widely used for solving such problems: ***direct methods***, where the unknown parameter is estimated *simultaneously* while

updating the control policy, and ***implicit methods*** – such as simulation based methods, where $A(u)$ are not estimated in order to compute the control policy.

Neuro-dynamic programming methods [7] have widely been used to solve MDPs. They broadly include two classes of algorithms: (i) Q-Learning and (ii) Temporal difference methods. Both these methods use Robbins Munro stochastic approximation algorithms to solve for the optimal policy. However, a key problem with these neuro-dynamic programming methods is that they cannot deal with constraints of the form (4). There seems to be no obvious way of modifying these algorithms to deal with randomized policies.

Another approach that also uses stochastic approximation is that presented in the book [24]. The book [24] considers a different parameterization to us – instead of the parameter θ, they consider the parameter to be the invariant measure $\pi_{i,a}$ of Z_n described above. It is well known [17] that (12) formulated in terms of the invariant measure π is the following linear program:

$$\begin{aligned} \min_{\pi} & \sum_{i\in S}\sum_{a\in\mathcal{A}(i)} \pi_{ia}c(i,a) \qquad (17)\\ \text{subject to } & \sum_{i\in S}\sum_{a\in\mathcal{A}(i)} \pi_{ia}\beta_l(i,a) < \gamma_l \\ & \sum_{a\in\mathcal{A}(j)} \pi_{ja} = \sum_{i\in S}\sum_{a\in\mathcal{A}(i)} \pi_{ia}A_{ij}(a) \\ & \sum_{i\in S, a\in\mathcal{A}(j)} \pi_{ia} = 1, \quad 0\le\pi_{ia}\le 1, \quad i\in S, a\in\mathcal{A}(j). \end{aligned}$$

With θ^* and π^* denoting the optimal solutions of (12) and (17), respectively, it is straightforward to show that

$$\theta^*_{ia} = \frac{\pi^*_{ia}}{\sum_{u\in\mathcal{A}(i)}\pi^*_{iu}} \qquad (18)$$

The main idea in [24] is to use stochastic approximations for each component of π_{ia} to optimize the above cost function (17). Because the above cost function is linear in π_{ia} the gradient is merely the observed cost $c(i,a)$. On the other hand the constraints in (17) are difficult to handle via stochastic approximation. The authors deal with the parameter constraints using a normalization procedure common in the Learning Automata Theory. In [31] it was shown that this normalization approach can be adapted to represent the projection method that we will introduce in Section 2.1. The MDP constraints are dealt with via Lagrange multipliers, a penalty function approach and a gradient projection method. The last two algorithms are direct methods and therefore require estimation of the transition probabilities $A_{ij}(u)$. Although the Lagrange method does not require explicit estimation of $A_{ij}(u)$, a closer analysis reveals that maximum likelihood estimation is implicitly carried out in the estimation of the coefficients (gradients) of the Lagrangian function.

The closest approach to that considered in this paper is the one presented in [4]. The MDP in [4] is identical to that considered here, but without the MDP constraints (13). [4] give an example of a parameterized class of policies which is identical to the canonical coordinates θ. We will show in numerical examples (Section 2.3) that the parameterization involving spherical coordinates has superior convergence properties. In addition, the approach for derivative estimation in [4] is via the Score Function method, which usually suffers from unbounded variance for infinite horizon costs. To alleviate this problem the authors use a forgetting factor that introduces a bias in the derivative estimation. Our derivative estimators are more efficient and consistent, with provably bounded variance. Typically, the score function method has a variance that is several orders of magnitude larger than the measured valued derivative estimator.

The above methods all use a "simulation optimization" approach, where almost sure convergence to the true optimal value can be shown under an appropriate choice of the parameters of the algorithms. In particular all stochastic approximations involved in the above mentioned methodologies use decreasing step size. One of the motivations of the present work is to implement a stochastic approximation procedure with constant step size in order for the controlled Markov chain to be able to deal with tracking slowly varying external conditions which result in slowly varying $A(u)$.

Remark: Our setting assumes perfect observation of the process $\{X_n\}$ and we call this a MDP. The paper [4] considers a partially observed MDP (POMDP), but assume that the observations Y_n of the process X_n belong to a finite set. In that work they consider suboptimal strategies of the form $\theta_{ia} = \mathbb{P}\{u_n = a \,|\, Y_n = i\}$. Such a policy is clearly not optimal for a POMDP since the optimal policy is a measurable function of the history $(Y_1, \ldots, Y_k, u_1, \ldots, u_{k-1})$ which is summarized by a continuous-valued information state [15]. Such suboptimal POMDP models are a special case of the problem considered here and our method can be applied in a straightforward manner.

1.6 User's Guide

A summary of the key equations for implementing the learning based algorithm proposed in this paper in spherical coordinates for constrained MDP (12), (13), (14) is as follows:

Input Parameters: Cost matrix $(c(i,a))$, constraint matrix $(\beta(i,a))$, batch size N (we chose $N = 1000$ in our numerical examples).

Step 0. Initialize: Set $n = 0$ and initialize $\alpha(n) \in \boldsymbol{\alpha}$ and vector $\lambda(n) \in \mathbb{R}^L_+$

Step 1. MDP Simulation: At time n, simulate MDP over batch $k \in [nN, (n+1)N - 1]$ using randomized policy $\theta(\alpha(n))$.

Step 2. "Parameter free" Gradient Estimation: Compute $\widehat{\nabla_\alpha C}(\alpha(n))$, $\widehat{\nabla_\alpha B}(\alpha(n))$ over the batch $k \in [nN, (n+1)N - 1]$ using the filtered frozen phantom algorithm given in (60) of Section 5.3.

Step 3. Update Policy $\theta(\alpha)$: Use a penalty function primal dual based stochastic approximation algorithm to update α as

$$\alpha(n+1) = \left[\alpha(n) - \epsilon\left(\widehat{\nabla_\alpha C}(\alpha(n)) + \widehat{\nabla_\alpha B}(\alpha(n))\left(\lambda(n) + \rho\widehat{B}(\alpha(n))\right)\right)\right] \bmod 2\pi \quad (19)$$

$$\lambda(n+1) = \max\left[0, \lambda(n) + \epsilon\widehat{B}(\alpha(n))\right] \quad (20)$$

where throughout this paper, for any vector mod2π is defined element-wise. The "penalization" ρ is a suitably large positive constant. Another alternative to (20) is to update λ via a multiplier (augmented Lagrangian) algorithm (see Section 3.2). A third alternative is to fix λ. For sufficiently large ρ, $\alpha(n)$ will converge to $\alpha(\infty)$ which is in a pre-specified ball around a local minimum α^*.

Step 4. Set $n = n + 1$ and go to Step 1.

The gradient estimators $\widehat{\nabla_\alpha C}(\alpha(n))$ in Step 2 are given in (53). These estimators are strongly consistent in N. Various variants of the gradient estimates will be considered for $\widehat{\nabla_\alpha B}(\alpha(n))$. As explained in Section 6, if (53) is also used for the gradient of the constraints, the algorithm is asymptotically sub-optimal, and the bias depends on the length N of the update intervals: if updating is performed relatively infrequently then this bias may not be too large. In Section 5.3 we propose alternative implementations, aiming to build an estimator which is strongly consistent in n, regardless of the size of N. The first is to use a running (or moving) average for the sample average $\hat{C}_n$ in (53), the second is to use a sliding window of size $W > N$ to estimate this average, and finally, one can also use past observations to estimate the bias as the process evolves and add a correction term to the update (19).

The algorithm using canonical coordinates can be derived similarly using (52). For Step 3, a penalty function primal dual method or augmented Lagrangian method similar to above can be used, details are omitted.

Remark: If the true parameters of the MDP jump change at infrequent intervals, then iterate averaging [21] (as long as the minimal window of averaging is smaller than the jump change time) and adaptive step size algorithms similar to [20] can be implemented in the above stochastic approximation algorithms to improve efficiency and tracking capabilities. We will explore these variations of the basic algorithm elsewhere.

2 Deterministic Algorithms for Unconstrained MDPs

As mentioned above, to find the optimal value θ^* defined in (18), our plan is to use a stochastic approximation algorithm of the form (16) with appropriate mechanisms to satisfy the constraints. In this section, we consider the **unconstrained** MDP (12), (14) (i.e., the MDP constraints (13) are omitted). The canonical and spherical parameterizations

defined in Section 1.2 will be used for dealing with the parameter constraints (14). The methodology can easily be extended to other parameterizations of the action probabilities.

A key result in stochastic approximation theory (averaging theory), see for example [21], states that under suitable regularity and stability conditions, the behavior of the stochastic approximation algorithm is captured by a deterministic dynamical system (differential/difference equation or inclusion) as the step size ϵ goes to zero. Thus to design the stochastic approximation algorithms and give insight into their performance, we will first focus on designing *deterministic* dynamical systems (ODEs) for solving the unconstrained MDP (12)-(14). By deterministic we mean that the objective function and all higher order derivatives can be exactly computed due to complete knowledge of the parameters of the MDP. We will construct suitable ordinary differential equations (ODEs) whose stable points will be Kuhn-Tucker points of the optimization problem. We will show how to build the updates in a descent direction, hence the limit points of the ODE will be local minima.

Once the deterministic algorithm has been designed, the corresponding stochastic approximation algorithm follows naturally by replacing the gradient in the deterministic algorithm with the gradient estimate (which is computed from the sample path of the Markov chain). In the un-constrained MDP case, convergence of the stochastic approximation algorithms to the deterministic algorithms is well studied in the stochastic approximation literature, see Section 6.

2.1 Canonical Coordinates $\psi = \theta$ and Generalized Gradients

Let $t \in \mathbb{R}^+$ denote continuous time. Consider the design of an ODE in the canonical coordinates θ such that the ODE's trajectory $\theta(t) \in \Theta$ for all time t. In order to ensure that $\theta(t) \in \Theta$, [30] proposes a projection along the Lagrangian associated with the optimization problem (12),(14). As explained in [30], a scheme that projects the gradient to the feasible surface can be built from the Lagrangian. The *generalized gradient* operator is defined as:

$$\mathcal{G}_{iu} = \frac{\partial}{\partial \theta_{iu}} - \sum_{a \in \mathcal{A}(i)} \theta_{ia} \frac{\partial}{\partial \theta_{ia}}, \tag{21}$$

and the result (Lemma 1 in [30]) is stated as follows.

Lemma 1 *For any $i \in S$, let the multidimensional function $\{\theta_u(t), u \in \mathcal{A}(i)\}$ be the solution of the ODE:*

$$\frac{d\theta_u(t)}{dt} = -\theta_u(t)\, \mathcal{G}_{iu}[C(\theta(t))] \tag{22}$$

with initial condition satisfying $\theta_u(0) > 0, \sum_{u \in \mathcal{A}(i)} \theta_u(0) = 1$. Then for all $t > 0$, $\theta(t) \in \Theta$ is a feasible probability vector in the interior of the set Θ and the stable points are all Kuhn-Tucker points (also called stationary points) of the problem (12),(14). Furthermore, they cannot be local maxima, because the drift of the ODE is a descent direction for (12),(14).

Proof: The complete proof appears in [30]. To gain insight into the result, we now show that the RHS of the ODE is a feasible descent direction. The feasibility is straightforwardly shown as follows: From (22) and (21), the aggregate drift is

$$\sum_{u\in\mathcal{A}(i)} \frac{d\theta_u}{dt} = -\sum_{u\in\mathcal{A}(i)} \theta_u(t)\,\frac{\partial}{\partial\theta_{iu}}[C(\theta(t))] + \left(\sum_{u\in\mathcal{A}(i)} \theta_u(t)\right) \sum_{a\in\mathcal{A}(i)} \theta_a(t)\,\frac{\partial}{\partial\theta_{ia}}[C(\theta(t))].$$

Thus if at any time t $\theta \in \Theta$ then the aggregate drift is zero. By assumption the initial condition of the ODE is in the feasible set, therefore the value $\sum_u \theta_u(t) = 1$ remains constant. To show that the RHS is a feasible descent direction it suffices now to show that it is a descent direction. Call $M(t) = C[\theta(t)]$ the actual cost of the trajectory, then using the chain rule of differentiation and (21)

$$\begin{aligned}\frac{dM(t)}{dt} &= \sum_{u\in\mathcal{A}(i)} \frac{\partial}{\partial\theta_{iu}} C[\theta(t)]\,\frac{d}{dt}\theta_{iu}(t)\\ &= -\sum_{u\in\mathcal{A}(i)} \theta_u(t)\,\left(\frac{\partial}{\partial\theta_{iu}} C[\theta(t)]\right)^2\\ &\quad + \sum_{u\in\mathcal{A}(i)} \frac{\partial}{\partial\theta_{iu}} C[\theta(t)]\,\theta_u(t) \sum_{a\in\mathcal{A}(i)} \theta_a(t)\,\frac{\partial}{\partial\theta_{ia}}[C(\theta(t))].\end{aligned}$$

Fix any t and to simplify notation, call $\theta_u(t) = p_u$, $\frac{\partial C[\theta(t)]}{\partial\theta_{iu}} = C_u$. From the proof of feasibility, it follows that $\sum_u p_u = 1$, which is a probability on the set $\mathcal{A}(i)$ so that the drift of $M(t)$ is the negative of the variance of the quantities C_u with respect to the probabilities p_u, namely: $-\sum_u p_u C_u^2 + (\sum_u p_u C_u)^2$. This implies that $dM(t)/dt \leq 0$, in other words the drift is a descent direction for the cost. □

Remark: If other linear equality constraints define the state space for the control, weights can be introduced in the definition of the generalized gradient to make $\sum_{u\in\mathcal{A}(i)} \mathcal{G}_{iu} = 0$ for all i. The result, namely convergence to local optima, will hold true.

In Section 4 consistent estimators for the generalized gradient are presented. This gives rise to the stochastic approximation scheme:

$$\theta_{iu}(n+1) = \theta_{iu}(n) - \epsilon\theta_{iu}(n)\,\widehat{\mathcal{G}_{iu}}[C(\theta(n))] \tag{23}$$

where ϵ denotes the fixed step size and $\widehat{\mathcal{G}_{iu}}[C(\theta(n))]$ is a consistent estimator of $\mathcal{G}_{iu}[C(\theta(n))]$. We propose to use decentralized operation of the stochastic approximation procedures as follows: a "control agent" is associated to each state $i \in S$. Every time the system visits state i, the estimation of $\mathcal{G}_{i\cdot}$ is updated, and after a predetermined number of such observations, the controller at i updates the vector θ_i following (23).

Remark: It is interesting to notice that above generalized gradient approach is equivalent to the parameterization of the action probabilities proposed in [4]. In [4], a control variable $\psi_{i,a} \in \mathbb{R}$ is introduced with

$$\theta_{ia}(\psi) = \frac{e^{\psi_{ia}}}{\sum_{u\in\mathcal{A}(i)} e^{\psi_{iu}}}.$$

Then all possible action probabilities are represented by this transformation, so that $\theta(\psi^*) = \theta^*$ is the true optimum, and the optimization over ψ is unconstrained. The exponential parameterization satisfies

$$\frac{\partial\theta_{iu}}{\partial\psi_{ia}} = \begin{cases} \theta_{iu}(1-\theta_{iu}) & u = a \\ -\theta_{iu}\theta_{ia} & u \neq a. \end{cases}$$

Using the chain rule of differentiation, for any function $C(\theta)$:

$$\begin{aligned} \frac{\partial}{\partial\psi_{ia}} C[\theta(\psi)] &= \sum_{u\in\mathcal{A}(i)} \frac{\partial}{\partial\theta_{ia}} C(\theta) \left(\frac{\partial\theta_{iu}}{\partial\psi_{ia}}\right) \\ &= \theta_{ia}\left(\frac{\partial}{\partial\theta_{ia}} C(\theta) - \sum_{u\in\mathcal{A}(i)} \theta_{iu} \frac{\partial}{\partial\theta_{ia}} C(\theta)\right). \end{aligned}$$

2.2 Spherical coordinates $\psi = \alpha$

The gradient projection method in Section 2.1 ensures that updates remain at all times in the feasible set satisfying the constraints (5), but they can exhibit slow convergence, particularly when the optimal probability vector θ^* is degenerate. As can be seen from (23), when a component has value zero it remains there. Because the drift of the update is proportional to the size of the updated component, as a component approaches zero, the magnitude of future updates decreases (to prevent crossing outside the feasible set). This mechanism may slow down convergence of the gradient algorithm using canonical coordinates $\psi = \theta$, particularly close to the optimal solution if this has components representing pure strategies, as is often the case.

Consider now the parameterization $\psi = \alpha$. Note that the function $C(\alpha)$, with $\theta_{ia} = \lambda_{ia}^2(\alpha)$ will be periodic in α, because of the symmetry between quadrants in spherical coordinates.

Lemma 2 *For each $i \in S$ consider the $d(i)$ dimensional ODE:*

$$\frac{d}{dt}\alpha_i(t) = -\nabla_{\alpha_i} C[\alpha(t)], \tag{24}$$

with initial condition $(\alpha_{ip}) \in \boldsymbol{\alpha}$, $p \in \mathcal{P}(i) = \{1, \ldots, d(i)\}$. Define $\theta_{ia}(t) = \lambda_{ia}^2[\alpha(t)]$. Then all limit points of $\theta(t)$ are Kuhn-Tucker points of the optimization problem (12),(14).

Proof: The proof of the lemma is based on local optimality of $\lim_{t\to\infty} \alpha(t) = \alpha^*$, and application of the Implicit Function Theorem. □

The stochastic approximation algorithm

$$\alpha(n+1) = \left[\alpha(n) - \epsilon\widehat{\nabla_\alpha C}[\alpha(n)]\right] \bmod 2\pi, \tag{25}$$

where $\widehat{\nabla_\alpha C}[\alpha(n)]$ is a consistent estimator of $\nabla_\alpha C[\alpha(n)]$ will yield asymptotically the locally optimal control (under some regularity assumptions that we will verify in Section 6). The projection mod2π ensures $\alpha(n) \in \boldsymbol{\alpha}$ and does not affect the dynamics of α since $C(\alpha)$, $B(\alpha)$ are periodic in α.

2.3 Numerical example comparing parameterizations

Here we present a numerical example that compares convergence of the generalized gradient algorithm versus spherical coordinates algorithm for an un-constrained MDP. The parameters chosen were: $S = \{0, 1\}$, there are three actions, $\mathcal{A}(i) = \{0, 1, 2\}$, $i \in S$ (i.e., $d(0) = d(1) = 2$),

$$A(0) = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}, A(1) = \begin{pmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \end{pmatrix}, A(2) = \begin{pmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{pmatrix}.$$

The cost matrix $(c(i,a))$ and parameter at $\theta(t) = (\theta_{ia}(t))$ initialized as $t = 0$ were chosen as

$$(c(i,a)) = -\begin{bmatrix} 50.0 & 200.0 & 10.0 \\ 3.0 & 500.0 & 0.0 \end{bmatrix}, \quad \theta(0) = \begin{bmatrix} 0.1 & 0.1 & 0.8 \\ 0.3 & 0.5 & 0.2 \end{bmatrix}.$$

Given this exact knowledge of the MDP parameters, the gradients $\nabla_\psi C(\psi)$, can be exactly computed. In Figure 1, we compare the evolution of the cost $C(\psi(t))$ versus time for $\psi(t) = \theta(t)$ and $\psi(t) = \alpha(t)$. A first order discretization of the ODEs (22) and (24) with discretization step size 10^{-3} was used.

This and other numerical studies not presented here suggest that the algorithms using the spherical coordinates α converge significantly faster than those using the canonical coordinates θ. For this reason, in the rest of the paper, we mainly present algorithms with spherical coordinates.

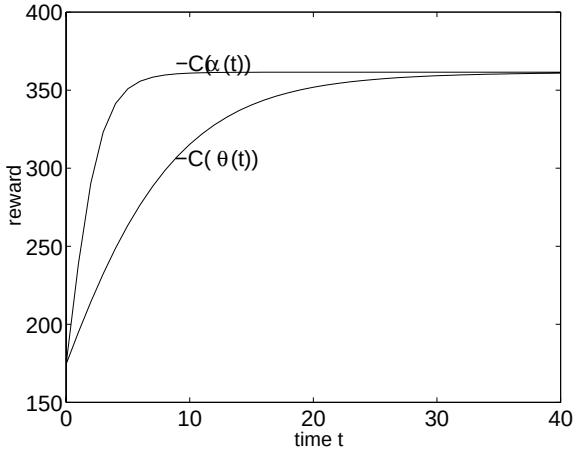

Figure 1: Comparison of algorithms using spherical and spherical coordinates

3 Deterministic Algorithms for Constrained MDP

In this section we consider the MDP problem (12) with MDP constraints (13) and parameter constraints (14). As in Section 2, we present deterministic dynamical systems (ODEs). The corresponding stochastic approximation algorithms for the constrained MDP are given by replacing $\nabla_\alpha C$, $\nabla_\alpha B$, B in the deterministic algorithms presented below with the estimated gradients $\widehat{\nabla_\alpha C}$, $\widehat{\nabla_\alpha B}$, $\widehat{B}$. These estimates are computed using the parameter free gradient estimation algorithms given in Section 5. The proofs of convergence of the resulting stochastic approximation algorithms are given in Section 6.2.

Because the spherical coordinate parameterization α gives better convergence, we focus here only on the optimization w.r.t. α. In Section 3.1 we present a penalty function based primal dual algorithm. In Section 3.2 we present an augmented Lagrangian algorithm Both methods perform extremely well in numerical examples. For completeness we also present in Section 3.3 a primal algorithm based on gradient projection. This primal algorithm requires higher computational complexity.

3.1 First-Order Primal Dual Algorithm

A widely used deterministic optimization method (with extension to stochastic approximation in [18, pg.180]) for handling constraints is based on the Lagrange multipliers and uses a first-order primal dual algorithm [6, pg 446]. First convert the inequality MDP constraints (13) to equality constraints by introducing the variables $z = (z_1, \ldots, z_L)$ so that $B_l(\alpha) + z_l^2 = 0$, $l = 1, \ldots, L$. Define $\psi \stackrel{\triangle}{=} (\alpha, z)$, $B_l(\psi) \stackrel{\triangle}{=} B_l(\alpha) + z_l^2$. Define the Lagrangian

$$\mathcal{L}(\psi, \lambda) \stackrel{\triangle}{=} C(\alpha) + \sum_{l=1}^{L} \lambda_l B_l(\psi) \tag{26}$$

In order to converge, a primal dual algorithm operating on the Lagrangian requires the Lagrangian to be locally convex at the the optimum, i.e., Hessian to be positive definite at the optimum (which is much more restrictive that the second order sufficiency condition of Assumption 2 in Section 1.3). Numerical examples show that this positive definite condition on the Hessian, which Luenberger [22, pp.397] terms "local convexity" seldom holds in the MDP case.

We can "convexify" the problem by adding a penalty term to the objective function (12). The resulting problem is:

$$\min_{\psi \in \Psi} C(\alpha) + \frac{\rho}{2} \sum_{l=1}^{L} (B_l(\psi))^2$$

subject to (13), (14). Here ρ denotes a large positive constant.

As shown in [22, pg.429], the optimum of the above problem is identical to that of (12), (13), (14). Define the augmented Lagrangian,

$$\mathcal{L}_\rho(\psi, \lambda) \stackrel{\triangle}{=} C(\alpha) + \sum_{l=1}^{L} \lambda_l B_l(\psi) + \frac{\rho}{2} \sum_{l=1}^{L} (B_l(\psi))^2 \tag{27}$$

Note that although the original Lagrangian may not be convex near the solution (and hence the primal dual algorithm does not work), for sufficiently large ρ, the last term in $\mathcal{L}_\rho$ "convexifies" the Lagrangian. Indeed, for sufficiently large ρ, [6] shows that the augmented Lagrangian is locally convex.

After some further calculations detailed in [6, pg.396], the primal dual algorithm operating on $\mathcal{L}_\rho(\psi(n), \lambda(n))$ reads:

$$\alpha^\epsilon(n+1) = \left[\alpha^\epsilon(n) - \epsilon\Big(\nabla_\alpha C(\alpha^\epsilon(n)) + \nabla_\alpha B(\alpha^\epsilon(n))\left(\lambda^\epsilon(n) + \rho B(\alpha^\epsilon(n))\right)\Big)\right] \bmod 2\pi \tag{28}$$

$$\lambda^\epsilon(n+1) = \max\left[0, \lambda^\epsilon(n) + \epsilon B(\alpha^\epsilon(n))\right] \tag{29}$$

where $\epsilon > 0$ denotes the step size.

Lemma 3 *Under Assumptions 1 and 2 for sufficiently large $\rho > 0$, there exists $\bar{\epsilon} > 0$, such that for all $\epsilon \in (0, \bar{\epsilon}]$, the sequence $\{\alpha^\epsilon(n), \lambda^\epsilon(n)\}$ generated by the primal dual algorithm (28) is attracted to a local KT pair (α^*, λ^*).*

Proof: Since $\mathcal{L}_\rho$ is convex for sufficiently large $\rho > 0$ [22], the proof straightforwardly follows from Proposition 4.4.2 in [6]. □

Let $T \in \mathbb{R}^+$ denote a fixed constant and $t \in [0, T]$ denote the continuous time. Define the piecewise constant interpolated continuous-time process

$$\alpha^\epsilon(t) = \alpha^\epsilon(n) \quad t \in [n\epsilon, (n+1)\epsilon) \tag{30}$$

$$\lambda^\epsilon(t) = \lambda^\epsilon(n) \quad t \in [n\epsilon, (n+1)\epsilon) \tag{31}$$

The above lemma implies $\{\lambda^\epsilon(n)\}$ lies in a compact set Λ for all n. Note $\alpha^\epsilon(n) \in \boldsymbol{\alpha}$ by definition where $\boldsymbol{\alpha}$ is compact. Then the following result directly follows from the above lemma and [18] where the convergence of the stochastic version is proved.

Theorem 1 *Under Assumptions 1 and 2, the interpolated process $\{\alpha^\epsilon(t), \lambda^\epsilon(t)\}$ defined in (30), (31) converges uniformly as $\epsilon \to 0$ to the process $\{\alpha(t), \lambda(t)\}$, i.e.,*

$$\lim_{\epsilon \downarrow 0} \limsup_{0<t\leq T} |\alpha^\epsilon(t) - \alpha(t)| = 0, \ \lim_{\epsilon \downarrow 0} \limsup_{0<t\leq T} |\lambda^\epsilon(t) - \lambda(t)| = 0, \tag{32}$$

where $\{\alpha(t), \lambda(t)\}$ satisfy the ODEs

$$\begin{aligned} \frac{d}{dt}\alpha(t) &= -\nabla_\alpha C(\alpha(t)) - \nabla_\alpha B(\alpha(t))(\lambda(t) + \rho B(\alpha(t)) \\ \frac{d}{dt}\lambda_l(t) &= \begin{cases} B_l(\alpha(t)) & \text{if } \lambda_l(t) \geq 0 \\ 0 & \text{if } \lambda_t(t) = 0 \ \text{ and } \frac{d}{dt}\lambda_l(t) < 0 \end{cases}, \quad l = 1, \ldots, L. \end{aligned} \tag{33}$$

The attraction point of this ODE is the local KT pair (α^*, λ^*).

3.2 Augmented Lagrangian (Multiplier) Algorithms

We outline three augmented Lagrangian (multiplier) algorithms.

1. Inexact Primal Minimization Multiplier Algorithm: The augmented Lagrangian approach (also know as a multiplier method) consists of the following coupled ODE and difference equation:

$$\frac{d\alpha^{(n+1)}(t)}{dt} = -\nabla_\alpha C(\alpha^{(n+1)}(t)) - \nabla_\alpha B(\alpha^{(n+1)}(t)) \left(\lambda(n) + \rho B(\alpha^{(n+1)}(t)\right) \tag{34}$$

$$\lambda_l(n+1) = \max\left[0, \ \lambda_l(n) + \rho B_l(\alpha^{(n+1)}(\infty))\right], \quad l = 1, \ldots, L \tag{35}$$

where $\alpha^{(n+1)}(\infty)$ denotes the fixed point of the ODE (34). Iteration (35) is a first order update for the multiplier, while (34) represents an ODE which is attracted to the minimum of the augmented Lagrangian $\mathcal{L}_\rho$. The max in (35) arises in dealing with the inequality

constraints, see [6, pp.396]. [6, Proposition 4.2.3] shows that if $(\alpha^{(0)}(0), \lambda_0(0))$ lies in the domain of attraction of a local KT pair (α^*, λ^*), then (34), (35) converges to this KT pair.

A practical alternative to the above exact primal minimization is first order *in-exact minimization* of the primal. The iterative version of the algorithm reads [6, pg.406]: At time $n+1$ set $\alpha^{(0)}(n+1) = \alpha(n)$. Then run $j = 0, \ldots, J-1$ iterations of the following gradient minimization of the primal

$$\alpha^{(j+1)}(n+1) = \left[\alpha^{(j)}(n+1) - \epsilon\Big(\nabla_\alpha C(\alpha^{(j)}(n)) + \nabla_\alpha B(\alpha^{(j)}(n))\left(\lambda(n) + \rho B(\alpha^{(j)}(n))\right)\Big)\right] \bmod 2\pi, \tag{36}$$

$\alpha(n+1) = \alpha^{(J)}(n+1)$ followed by a first order multiplier step

$$\lambda_l(n+1) = \max\left[0,\ \lambda_l(n) + \rho B_l(\alpha(n+1))\right], \quad l = 1, \ldots, L \tag{37}$$

Iteration (36) represents a first order fixed step size inexact minimization of the augmented Lagrangian $\mathcal{L}_\rho$ (*inexact* because (36) is terminated after a finite number of steps J). It is shown in [5] that as long as the inexact minimization of the primal is done such that the error tolerances are decreasing with n but summable, then the algorithm converges to a Kuhn Tucker point. More recently, [14] show that even if the error tolerances are fixed (i.e., non-decreasing), convergence can be shown for $n \to \infty$. In numerical examples presented in Section 7.1, we found that often choosing $J \leq 10$ results in convergence to a KT pair.

2. Multiplier Algorithm of [18]: A second possibility is to use the multiplier algorithm presented and analysed in [18]. For the case of equality constraints, i.e., $B_l(\alpha) = 0$ (with obvious generalization to the inequality case using surplus variables), the algorithm at each time updates $\alpha(n)$ for $J = 1$ iteration followed by updating λ according to

$$\lambda(n+1) = -(\nabla_\alpha B'(\alpha(n))\nabla_\alpha B(\alpha(n)))^{-1}\nabla_\alpha B'(\alpha(n))\nabla_\alpha C(\alpha(n))$$

It is shown in [18] that the resulting ODE is attracted to the Kuhn Tucker points. However, to devise a stochastic approximation algorithm, it is not possible to directly substitute the estimates $\widehat{\nabla_\alpha B}(\alpha(n))$ for $\nabla_\alpha B(\alpha(n))$, because in general $\widehat{\nabla_\alpha B}'(\alpha(n))\widehat{\nabla_\alpha B}(\alpha(n))$ is not an unbiased estimator for $\nabla_\alpha B'(\alpha(n))\nabla_\alpha B(\alpha(n))$. To estimate the inverse of the matrix consistently is even a harder problem. Due to this technical difficulty we have preferred not to implement this algorithm.

3. Fixed Multiplier: A trivial case of the multiplier algorithm is to fix $\lambda(n) = \bar{\lambda}$ for all time n and only update α according to (36) with $I = 1$ iteration at each time instant. This is clearly equivalent to the primal update (28) with fixed $\lambda(n) = \bar{\lambda}$. From Theorem 1, the interpolated trajectory of this algorithm converges uniformly as $\epsilon \to 0$ to the trajectory of the ODE

$$\frac{d}{dt}\alpha(t) = -\nabla_\alpha C(\alpha(t)) - \nabla_\alpha B(\alpha(t))(\bar{\lambda} + \rho B(\alpha(t)) \tag{38}$$

The following result in [6] shows that the attraction point of this ODE is close to α^* for sufficiently large ρ, resulting in a near optimal solution. First convert the L inequality constraints to equality constraints $B_l(\psi) = 0, < l = 1, \ldots, L$ as defined in Section 3.1. Let ψ^*, λ^* denote the corresponding KT pair where $\psi^* = (\alpha^*, z^*)$.

Result 1 *[6, Proposition 4.2.3]. Let $\bar{\rho} > 0$ be scalar such that $\nabla_\psi{}^2 \mathcal{L}_\rho(\psi^*, \lambda^*) > 0$. Then there exist positive scalars δ and K such that for $(\bar{\lambda}, \rho) \in D \in \mathbb{R}^{L+1}$ defined by*

$$D = \{(\lambda, \rho) : \|\lambda - \lambda^*\| < \delta\rho,\ \rho \geq \bar{\rho}\}$$

the attraction point of the ODE $\alpha^{\lambda,\rho}$ is unique. Moreover

$$\|\alpha^{\lambda,\rho} - \alpha^*\| \leq K \frac{\|\bar{\lambda} - \lambda^*\|}{\rho}$$

A fixed multiplier algorithm is seemingly much simpler to implement than the inexact minimization one. However, in practice a bad choice of the pair $(\bar{\lambda}, \rho)$ may yield a noticeable bias. In Section 7.1, numerical examples compare the inexact primal minimization with the fixed multiplier algorithm.

3.3 Projected Gradient Primal Algorithm

In this subsection, we present for completeness, a "primal algorithm" [22] for solving the constrained MDP problem. This primal algorithms requires larger computational cost than the algorithms given above. Also, our numerical studies show that the augmented Lagrangian and primal dual algorithms presented above have better numerical properties and superior convergence rates compared to the primal method presented here.

The primal algorithm we present here is a version of the gradient projection method (see [8]) that updates along the gradient direction and then projects the resulting vector onto the constraints surface (if the resulting vector is infeasible).

$$\begin{aligned} \tilde{\alpha}(n+1) &= \alpha(n) - \epsilon \nabla_\alpha C[\alpha(n)] \\ \alpha(n+1) &= \Pi_X[\tilde{\alpha}(n+1)], \end{aligned} \tag{39}$$

where $\Pi_X(\cdot)$ is the projection of the vector onto the space:

$$X = \{\alpha \in \mathbb{R}^{d-1} \colon B_l(\alpha) \leq 0, l = 1, \ldots, L\}.$$

Evaluation of the projection operation is usually the main computational hindrance of the algorithm. Because we are interested in applying a stochastic version of the method, we propose here an approximation that is provably convergent under some stability conditions. Our method is based on the observation that if $x = \Pi_X(c)$, then x is the vector in X that minimizes the distance $\|x - c\|^2$, or equivalently, x is the solution to the quadratic optimization problem:

$$\min_{x \in X} \left(\frac{1}{2} x'x - c'x \right).$$

As illustrated in Example 3.4.3 of [8], when the constraint set is linear the dual of this problem leads to a much simpler optimization problem where the new constraints are only positivity constraints. Because the points $\alpha(n), \tilde{\alpha}(n+1)$ are close when ϵ is small, using the fact that the constraints are analytical functions of α (polynomial and trigonometric functions are analytic) a Taylor approximation of $B(x)$ can be used to obtain the following (approximative) optimization problem with linear constraints:

$$\begin{aligned} &\text{minimize} \quad \left(\frac{1}{2}x'x - c'x\right), \\ &\text{subject to:} \quad B(\alpha(n)) + \nabla_\alpha B(\alpha(n))'(x - \alpha(n)) \leq 0. \end{aligned}$$

Let $\mu = (\mu_j, j = 1, \ldots, L)$ be the dual variable. Then the dual to this problem is:

$$\min_{\mu \geq 0} \left(\frac{1}{2}\mu' \nabla_\alpha B(\alpha(n))' \nabla_\alpha B(\alpha(n))\mu - [B(\alpha(n)) + \nabla_\alpha B(\alpha(n))'(\tilde{\alpha}(n+1) - \alpha(n)]'\mu\right), \quad (40)$$

and the optimal solutions of the dual and primal problems satisfy $x^* = \tilde{\alpha}(n+1) - \nabla_\alpha B(\alpha(n))'\mu^* \approx \Pi_X(\tilde{\alpha}(n+1))$. Although a linearization has been used to find the projection, as $\epsilon \to 0$, one would hope that this procedure converges to the optimal feasible point.

The subsidiary optimization problem (40) must be solved at each iteration of the procedure (39). Because this is a quadratic minimization problem with non-negative constraints, a gradient search can also be used for the dual problem, which would entail using J iterations for each n:

$$\begin{aligned} \mu_{j+1}(n+1) = \max\Big(0,\ &\mu_j(n) - \bar{\epsilon} \times \\ &\left[\nabla_\alpha B(\alpha(n))' \nabla_\alpha B(\alpha(n))\mu_j(n) - B(\alpha(n)) + \nabla_\alpha B(\alpha(n))'(\tilde{\alpha}(n+1) - \alpha(n))\right]\Big) \end{aligned} \quad (41)$$

Here $\bar{\epsilon} > 0$ denotes a small step size. Our method considers $\mu_0(n+1) = \mu_1(n)$, to start the new subsidiary problem. Because the updates in (39) take small steps and all functions F, B_j are continuously differentiable, one expects that the ensuing procedure will also converge as $\epsilon \to 0, n \to \infty$ with $\epsilon n = t$, to the solution of a projected ODE.

4 Measure-Valued Gradient Estimation

In this section we focus on the derivation of the general formulas for gradient estimation for Markov chains. We also discuss implementation aspects of the ensuing formulas. In Section 5 we will use these formulas to devise parameter free (learning) gradient estimators for $\widehat{\nabla_\alpha C}$ and $\widehat{\nabla_\alpha B}$.

4.1 Measure-Valued Gradient Estimators and Implementation

Estimators of the generalized gradient (21) for a queueing system were established in [29]. We derive now the general form of the estimation w.r.t. any parameterization for the Markov Decision Process using the measured valued differentiation method of [12]. Recall that the transition probability of the chain $\{Z_n\}$ given by (10) is parametrized by ψ. Denote by ∇_ψ the gradient w.r.t. the multidimensional parameter ψ. In [12] it is shown that the weak derivative of the n-th step transition can be calculated using the chain rule for differentiation, just as in ordinary calculus; that is, for any test function $F : \mathcal{Z}^n \to \mathbb{R}$,

$$\begin{aligned}\nabla_\psi \mathbb{E}_\psi[F(\bar{Z})] &= \nabla_\psi \left(\sum_{\bar{i}\in S^n} \sum_{\bar{u}\in\mathcal{A}^n} F(\bar{i},\bar{u}) \prod_{k=1}^{n} P_{i_{k-1},u_{k-1},i_k,u_k}(\psi) \right) \\ &= \sum_{k=1}^{n} \left(\sum_{\bar{i}\in S^n} \sum_{\bar{u}\in\mathcal{A}^n} F(\bar{i},\bar{u}) \prod_{l=1}^{k-1} P_{i_{l-1},u_{l-1},i_l,u_l}(\psi)\, \nabla_\psi P_{i_{k-1},u_{k-1},i_k,u_k}(\psi) \prod_{l=k+1}^{n} P_{i_{l-1},u_{l-1},i_l,u_l}(\psi) \right),\end{aligned} \tag{42}$$

where $\bar{Z} = (Z_1, \ldots, Z_n)$, $\bar{i} = (i_1, \ldots, i_n)$, $i_k \in S$, $\bar{u} = (u_1, \ldots, u_n)$, $u_k \in \mathcal{A}$, and each component of $\nabla_\psi P(\psi)$ is the weak derivative of the kernel $P(\psi)$ w.r.t. each component of ψ, as we explain shortly. While it is a matrix, it does not define a transition probability (the rows do not add up to one, they add up to zero) so it is not possible to interpret the expression above directly in terms of "transitions" to states i_{k+1}, u_{k+1} starting at i_k, u_k. Using the concept of weak derivatives (see [23]), the transition kernels $\nabla_\psi P_{i_{k-1},u_{k-1},i_k,u_k}(\psi)$ can be interpreted as the weighted *difference* between two transition probabilities for the random variable i_k, u_k, as we now show.

The problem is to find a closed formula for the derivative of the one-step expectation $\mathbb{E}_\psi[f(Z_{k+1})|Z_k = i_k]$ for *any* real valued test function $f\colon \mathcal{Z} \to \mathbb{R}$. Using (10), we have

$$\nabla_\psi \mathbb{E}_\psi[f(Z_{k+1}) \mid Z_k = (i_k, u_k)] = \nabla_\psi \mathbb{E} \left(\sum_{i\in S} f(i, u_{k+1}) A_{i_k,i}(u_k) \mid Z_k = (i_k, u_k) \right) \tag{43}$$

where a conditioning argument akin to the method in [12] has been used to isolate the dependency on $\theta(\psi)$: given the state and action pair Z_k the only dependency on ψ is in the distribution of u_{k+1}. It then suffices to evaluate $\nabla_\psi \mathbb{E} f(i, u)$ for each fixed value of i, with $\mathbb{P}[u = a] = \theta_{ia}(\psi)$.

In the case of spherical coordinates, the action u_{k+1} (given $i_{k+1} = i$) has a distribution:

$$u_{k+1} \;=\; \begin{cases} 0 & \text{w.p. } \cos^2(\alpha_{i1}) \\ Y_1 & \text{w.p. } \sin^2(\alpha_{i1}) \end{cases} \tag{44}$$

$$Y_1 = \begin{cases} 1 & \text{w.p. } \cos^2(\alpha_{i2}) \\ Y_2 & \text{w.p. } \sin^2(\alpha_{i2}) \end{cases}$$

$$\vdots$$

$$Y_{d(i)-1} = \begin{cases} d(i)-1 & \text{w.p. } \cos^2(\alpha_{i,d(i)}) \\ d(i) & \text{w.p. } \sin^2(\alpha_{i,d(i)}) \end{cases}$$

For convenience we will call $Y_{d(i)} = d(i)$. Clearly $Y_p, p = 1, \ldots, d(i)$ can only have one of the values $p, \ldots, d(i)$. Because α_{ip} does not affect the distribution of u_{k+1} if $i_{k+1} \neq i$, the gradient is non null only when $i_{k+1} = i$, in which case we have:

$$\begin{aligned} \frac{\partial}{\partial \alpha_{ip}} \mathbb{E} f(i,u) &= \frac{\partial}{\partial \alpha_{ip}} \left(f(i,p-1)\cos^2(\alpha_{ip}) + f(i,Y_p)\sin^2(\alpha_{ip}) \right) \prod_{k=1}^{p-1} \sin^2(\alpha_{ik}) \\ &= -2\sin(\alpha_{ip})\cos(\alpha_{ip}) \prod_{k=1}^{p-1} \sin^2(\alpha_{ik}) \mathbb{E}[f(i,p-1) - f(i,Y_p)], \end{aligned}$$

because the terms $f(i,k), k < p-1$ have weights which are independent of α_{ip}. The random variable Y_p has a distribution on $\{p, \ldots, d(i)\}$ corresponding to

$$\mathbb{P}(Y_p = a) = \frac{\theta_{ia}(\psi)}{\prod_{k=1}^{p-1} \sin^2(\alpha_{ik})}, \quad a \geq p \tag{45}$$

Notice that for $p = d(i)$ the random variable $Y_{d(i)} = d(i)$ is degenerate.

Theorem 2 *Fix state i. Let $\{Z_n = (X_n, u_n)\}$ be a MDP governed by (5) and (10). For any $p = 1, \ldots, d(i) - 1$, and for each k let $\{Z_n(k) = (X_n(k), u_n(k)); n \geq 0\}$ be a Markov decision process that follows the same transition rules (5) and (10), but with initial state $Z_0^-(k) = (i, Y_p)$ where Y_p is randomly generated according to (45). Then:*

$$\frac{\partial}{\partial \alpha_{ip}} \left[\frac{1}{N} \sum_{n=1}^{N} c(Z_n) \right] = -\frac{2\tan(\alpha_{ip})}{N} \sum_{k=1}^{N} \delta_{ip}(k) \, \mathsf{E} \left(\sum_{n=0}^{\tau(k) \vee N - k} [c(Z_{n+k}) - c(Z_n(k))] \right), \tag{46}$$

where $\delta_{ip}(k) = \mathbf{1}_{\{X_k = i, u_k = p-1\}}$ and $\tau(k)$ is the time until coupling:

$$\tau(k) = \min\{n > 0 : Z_n(k) = Z_{n+k}\}.$$

Let now $\{Z_n(k) = (X_n(k), u_n(k))\}$ have initial state $X_0(k) = i$ and

$$u_0(k) = \begin{cases} d(i)-1 & u_k = d(i) \\ d(i) & u_k = d(i)-1 \end{cases}.$$

$$\frac{\partial}{\partial \alpha_{i,d(i)}}\left[\frac{1}{N}\sum_{n=1}^{N} c(Z_n)\right] = \frac{2\cos(\alpha_{i,d(i)})\sin(\alpha_{i2})}{N}\sum_{k=1}^{N}[\delta_{i,d(i)}(k) - \delta_{i,d(i)-1}(k) \times \mathsf{E}\left(\sum_{n=0}^{\tau(k)\vee N-k}[c(Z_{n+k}) - c(Z_n(k))]\right),$$

where $\tau(k)$ *is the time until coupling* $\nu(k) = \min\{n > 0 : Z_n(k) = Z_{n+k}\}$.

Proof: Consider $p < d(i)$ and call $F(\bar{Z})$ the sample average cost $(1/N)\sum_n c(Z_n)$. It follows from the chain rule and the development of the one-step transition derivative kernel that:

$$\frac{\partial}{\partial \alpha_{ip}}\left(\mathbb{E}_\psi[F(\bar{Z})]\right) = -2\sin(\alpha_{ip})\cos(\alpha_{ip})\prod_{k=1}^{p-1}\sin^2(\alpha_{ik})\sum_{k=1}^{N}\mathbf{1}_{\{X_k=i\}}\mathsf{E}[F(\bar{Z}^+(k)) - F(\bar{Z}^-(k))],$$

where for each k, $\{Z_n^\pm(k), n \le k\}$ is a Markov process with transition matrix $P(\theta)$. Next, the "plus" and "minus" processes have actions $u_k^+ = p-1$ and $u_k^- = Y_p$ (with distribution as in (45)). Then the evolution of the processes follows: $\mathbb{P}[Z_{k+1}^\pm(k) = (j,a')|Z_k^\pm(k) = (i, Y^\pm)] = A_{ij}(Y^\pm)\theta_{ja}$, and $\{Z_n(k), n > k\}$ again is a Markov process with transition matrix $P(\theta)$. For each k, an instance of the paths up to step k is the nominal process itself, therefore choosing $Z_n^\pm(k) = Z_n, n \le k$ will yield the same expectation for the gradient, and the first terms in the difference of sample averages cancel out.

Equivalently, the plus and minus processes can be stated as MDP's where the decision at step k is "forced" to have the values $p-1, Y_p$ respectively, that is, for each k the MDP $Z_n^\pm(k) = (X_n^\pm(k), u_n^\pm(k)$ evolves according to (5) and (10). Using this particular representation, the trajectories "split" the decisions: the *nominal* decision is the one observed: $u_k \sim \theta_i(\psi)$, the decision in the "plus" system is $u_k(k)^+ = p-1$ and the decision in the "minus" system is distributed according to $\tilde{\theta}_i^{(a)}$. From there on, all processes follow the same dynamics for the MDP, namely equations (5) and (1).

Sample now the nominal process to obtain an instance of the "plus" process, that is, whenever $u_k = p-1$ we consider the nominal process as the "plus" process. Because this observation has a sampling rate of $\theta_{i,p-1}(\psi) = \cos^2(\alpha_{ip})\prod_{k=1}^{p-1}\sin^2(\alpha_{ik})$, then

$$\frac{\partial}{\partial \alpha_{ip}}\left(\mathbb{E}_\psi[F(\bar{Z})]\right) = -\frac{2\sin(\alpha_{ip})\cos(\alpha_{ip})}{N}\sum_{k=1}^{N}\mathbf{1}_{\{Z_k=(i,a)\}}\times \mathsf{E}\left\{\sum_{n=k}^{N}[c(X_n^+(k), u_n^+(k)) - c(X_n^-(k), u_n^-(k))]\right\}.$$

The proof is complete now, identifying $X_n(k) = X^-_{k+n}(k), u_n(k) = u^-(k)_{n+k}; n \geq 0$, and using the fact that for each k, after the coupling time $\tau(k)$ both MDP's have the same distribution.

□

Remark: Evaluating the phantom processes k only for those steps where $u_k = p - 1$ in the nominal path not only saves computational time, but as will be discussed Section 5.1, is the basis for model-free estimation. For those steps, the initial state of the plus system will have the same decision as the observed one, that is, $u^+_k(k) = u_k = p - 1$ and only one other random variable $u^-_k(k) = Y_p$ has to be simulated.

Using a similar methodology the MVD estimator for the canonical coordinates can be calculated, as summarized in the following result.

Theorem 3 *Let $\{Z_n = (X_n, u_n)\}$ be a MDP governed by (5) and (1), and for each $k \in \{1, \ldots, N\}$, define the "phantom" MDP $\{Z_n(k) = (X_n(k), u_n(k)); n \geq 0\}$ as a Markov decision process that follows the same transition rules (5) and (1), but with initial state $Z_0(k) = (i, u_0(k))$ where $u_0(k)$ is randomly generated according to $u_0(k) \sim \tilde{\theta}^{(a)}_i$, and*

$$\tilde{\theta}^{(a)}_{i,u} = \frac{\theta_{iu}}{(1 - \theta_{ia})}, \quad u \in \mathcal{A}(i) \setminus \{a\}. \tag{47}$$

Let $\delta_{ia}(k) = \mathbf{1}_{\{X_n = i, u_n = a\}}$. Then

$$\mathcal{G}_{ia}\left(\mathbb{E}_\psi\left[\frac{1}{N}\sum_{n=1}^{N} c(Z_n)\right]\right) = \frac{(1 - \theta_{ia})}{N\theta_{ia}} \sum_{k=1}^{N} \delta_{ia}(k) \times \mathsf{E}\left(\sum_{n=0}^{\tau(k)\vee(N-k)} [c(Z_{n+k}) - c(Z_n(k))]\right), \tag{48}$$

where $\tau(k)$ is the time until coupling:

$$\tau(k) = \min\{n > 0 : Z_n(k) = Z_{n+k}\}.$$

Remark: Instead of working with the augmented process $\{Z_n\}$, an equivalent approach is to consider the Markov chain $X_n(\psi)$ with transition kernel $P_{ij}(\psi) \equiv \mathbb{P}[X_{n+1}(\psi) = j | X_n(\psi) = i] = \sum_{a \in \mathcal{A}(i)} \theta_{ia}(\psi) A_{ij}(a)$. Denote its invariant measure as $\tilde{\pi}_\psi$. The cost function is then $C(\psi) = \mathbb{E}_{\tilde{\pi}_\psi}[\tilde{c}(\cdot, \psi)] = \sum_{a \in \mathcal{A}(i)} \theta_{ia}(\psi) c(i, a)$ where $\tilde{c}(i, \psi) = \sum_{a \in \mathcal{A}(i)} \theta_{ia}(\psi)\, c(i, a)$. The gradient estimation algorithm obtained is identical to that presented above.

4.2 Rapproachment with other Gradient Estimation Methods

The aim of this subsection is to briefly compare the gradient estimation formula in Theorem 3 with two other widely used gradient estimators in the literature, namely, realization perturbation factors and the score function estimator.

Realization Perturbation Factors: The implementation in Theorem 3 of the generalized gradient was called "the swapping phantoms" in [29], because the phantom decisions (those of the minus process) are swapped. In that paper the setting is a queueing system with general state space and the terms in the sum were called the "difference process". In the context of a finite state Markov chain, these were called "realization perturbation" factors in [9], when $N \to \infty$. To recover the result therein, use the following argument. In the long run, the fraction of steps where $\delta_{ia}(k) = 1$ is $\theta_{ia}\,\pi_i(\theta)$, that is, the stationary probability of the chain $Z_n = (X_n, u_n)$, which can be used to show that

$$\begin{aligned}\mathcal{G}_{ia}\left(\mathbb{E}_\psi[c(Z)]\right) =& \tilde{\pi}_i(\theta)(1-\theta_{ia}) \times \\ & \mathsf{E}\left\{\sum_{n=0}^{\tilde{\nu}}[\mathbb{E}_\psi[c(Z_n) \mid Z_0 = (i,a)] - \mathbb{E}_\psi[c(Z_n) \mid Z_0 = (i,\tilde{u})]\right\},\end{aligned}$$

where $\tilde{u} \sim \tilde{\theta}_i^{(a)}$, and $\tilde{\nu}$ is the coupling time of the two Markov chains. Using a conditioning argument on the values of u, the above expression is equivalent to:

$$\begin{aligned}\mathcal{G}_{ia}\left(\mathbb{E}_{\pi(\psi)}[c(Z)]\right) = & \sum_{u\in\mathcal{A}(i)} \theta_{ia}\,\tilde{\pi}_i(\theta) \times \\ & \mathbb{E}_\psi\left\{\sum_{n=0}^{\tilde{\tau}(u)}[\mathsf{E}[c(Z_n) \mid Z_0 = (i,a)] - \mathsf{E}[c(Z_n) \mid Z_0 = (i,u)]\right\},\end{aligned}$$

where (abusing notation) $\tilde{\tau}(u)$ is now the corresponding coupling time of the processes started at (i,a) and (i,u). This is the familiar formula for the weight of the perturbation realization factors in terms of stationary probabilities, in [9] and [10], see also [23, Lemma 3.75, pp.203]

Score Function Estimator of Bartlett & Baxter [4]: The Score Function estimator can also be derived using the measure-valued approach. Let a random variable Z have value i with probability $p_\theta(i)$. Then

$$\frac{\partial}{\partial\theta_{ia}}F(Z) = \frac{\partial}{\partial\theta_{ia}}\sum_i F(i)\,p_\theta(i) = \sum_i\left(\frac{\partial}{\partial\theta_{ia}}\ln[p_\theta(i)]\right)F(i)p_\theta(i) = \mathbb{E}_\psi[F(Z)S(\theta_{ia},Z)],$$

where $S(Z,\theta_{ia})$ is known as the Score Function. In particular when estimating (43), conditioning on Z_n the method would yield

$$S(Z_{k+1},\theta_{ia},Z_k) = \frac{1}{\theta_{ia}}A_{i_k,i}(u_k)\mathbf{1}_{\{Z_{k+1}=(i,a)\}}$$

which requires knowledge of the transition probabilities. These can (of course) be estimated as the Markov chain evolves. However more seriously, the estimator is not uniformly

bounded in θ: when one or more components of the control parameter tend to zero (which they do when a policy is pure instead of randomized) the estimator blows up. To overcome this problem a Score Function estimator is used in [4] with the exponential parameterization, which gives a direct estimation of the scaled generalized gradient $\theta_{ia}\mathcal{G}_{ia}$. When inserted in the formula for the chain rule, the Score Function estimator for the Markov Chain is of the form $\sum_n S(\theta_{ia}, Z_n)$ and it is a well known problem that the variance increases with time. Numerous variance reduction techniques have been proposed in the literature [23] including regenerative estimation, finite horizon approximations and more recently [4] propose to use a forgetting factor for the derivative estimator. Their method suffers therefore of a variance/bias trade-off, while our estimation method is consistent and has uniformly bounded variance (in N), as will be shown shortly.

5 Learning based Constrained MDP solution

In this section we show how to modify the gradient estimation algorithms of Theorems 2 and 3 to make them parameter free. That is, the algorithms presented below are simulation based and do not require explicit knowledge of the transition probabilities of the constrained MDP. As mentioned in the Section 1 this makes the stochastic approximation algorithm (16) a viable alternative to neuro-dynamic programming methods (which typically cannot handle constrained MDPs) and do not have fixed step size to track time varying parameters.

5.1 Gradient Estimators: The Doeblin and the Frozen Phantoms

Several implementations of the estimator (48) or (46) can be used. Because of Markov chain coupling, once a phantom system is in the same state as the nominal system, their evolution is identical in distribution. The differences between the phantom chains and the nominal will become identically zero after coupling and we say then that the perturbation from that phantom system "dies".

In the literature (see [10, 9] and references therein) it is usually understood that a simulation can be performed off-line. Efficiency of these estimators is mostly affected by computational time, which depends on how many parallel phantom systems exists at a given iteration, as well as the CPU time required to generate each of the new states in these phantom systems. Because the processes start at different state values (i, a) and $(i, \tilde{u})$, it is not straightforward to find a particular coupling that will minimize the time for coupling $\tau(k)$, as explained in [10]. We experimented using independent and antithetic decisions with no remarkable difference in performance. We call these phantoms as the Doeblin phantoms, since the states of the phantom MDP's are generated independently of the current state. Figure 2 depicts the idea of coupling in this setting.

In our programs, a random number of parallel phantoms are living at any given iteration, but the number of states was fixed, so we generate the next state $X_{n+1}(k)$ from each possible state $i \in S$ and keep an array: all those phantom systems for which $X_n(k) = i$ have already coupled into one system.

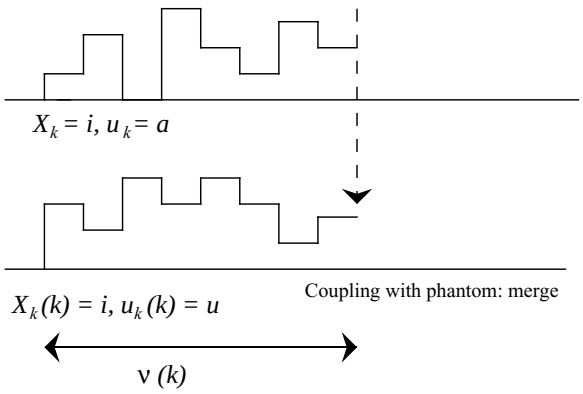

Figure 2: Doeblin phantoms evolve in parallel until coupling with nominal

While the computational problems of the Doeblin phantoms are important, the most important problem with this method is that it is based on off-line simulations of the phantom chains. Recall that we are interested in a closed formula that can be evaluated as a function of the observation of the nominal process $\{X_n(\theta), u_n(\theta)\}$ and without explicit knowledge of $A_{ij}(\cdot)$. Formally, the Doeblin phantom processes $\{Z^-(k) = (X^-(k), u^-(k))\}$ are not adapted to the natural filtration of the process $\{Z_n\}$ and parallel simulations are required to reproduce a version of each phantom using (5) and (1). When $A_{ij}(u)$ is not known, it is of course still possible to try to estimate the transition matrix concurrently with the gradients. Instead, we propose a new method that overcomes this difficulty and gives the basis for indirect control of the MDP.

Cut and paste techniques [13] are clever ways to use the observations instead of simulating off-line: if a phantom system starts at state i and a given decision $\tilde{u} \in \mathcal{A}(i) \setminus \{a\}$, the history of the process may be used as a stochastic version of this system: for example one can wait until the nominal process has state-decision pair $(i, \tilde{u})$. From then on the cost of the phantom system can be observed from the nominal path.

Our methodology is as follows: First, a cut-and-paste argument is used. The phantom system is 'frozen' at the initial state for ν iterations, until the nominal system hits this phantom state (which happens in finite time a.s.). Once the systems couple they follow identical paths until the nominal system has completed N stages, at which point the phantom system must complete the ν remaining steps. The novel idea that we introduce is to filter these dynamics instead of simulating the remaining ν steps of the phantom systems. Filtering the phantoms implies averaging out the dynamics of the phantom system so no further detailed simulation is required. In addition to making our estimation model-free, it turns out to be more efficient than the usual off-line simulations because no extra CPU time is required.

This filtering of frozen phantoms is the motivation for our main result below.

Theorem 4 *Let $\{Z_n\}$ be a MDP governed by (5) and (10), with initial condition $Z_0 = (i,a)$. Call $\delta_{ia}(n) = \mathbf{1}_{\{Z_n=(i,a)\}}$. Let $\{\tilde{u}(k)\}$ be a sequence of iid random variables with distribution $\theta_i^{(a)}$ in (47) or Y_k in (44), independent of $\mathfrak{F}_n$. Furthermore, define:*

$$\nu(k) = \min\{n \geq 1 : Z_{k+n} = (i, \tilde{u}(k))\}, \text{ and } \hat{C}_N = \frac{1}{N}\sum_{n=1}^{N} c(Z_n). \tag{49}$$

Let

$$L \stackrel{\triangle}{=} \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \delta_{ia}(k) \times \mathsf{E}\left(\sum_{n=k}^{N}[c(Z_n) - c(Z_n(k))]\right), \tag{50}$$

where the expectation is w.r.t. $\tilde{u}(k)$, and $Z_n(k) = (X_n(k), u_n(k))$ is started at the point $(i, \tilde{u}(k))$. Then a consistent estimator of $\mathbb{E}[L]$, which occurs in (48) and (46) is

$$\frac{1}{N}\sum_{k=1}^{N} \delta_{ia}(k) \left([c(i,a) - c(i,\tilde{u}(k)] + \sum_{n=k+1}^{k+\nu(k)} \tilde{c}(Z_n) - \nu(k)\,\hat{C}_N\right) \tag{51}$$

Remark: The following are consistent estimators for the generalized gradient (48) and gradient in spherical coordinates (46), respectively:

$$\hat{G}_N(i,a) = \frac{(1-\theta_{ia})}{N\theta_{ia}}\sum_{k=1}^{N} \delta_{ia}(k) \left([c(i,a) - c(i,\tilde{u}(k)] + \sum_{n=k+1}^{k+\nu(k)} \tilde{c}(Z_n) - \nu(k)\,\hat{C}_N\right) \tag{52}$$

$$\hat{G}_N(i,p) = \frac{2\tan(\alpha_{ip})}{N}\sum_{k=1}^{N} \delta_{i,p-1}(k) \left([c(i,p-1) - c(i,\tilde{u}(k)] + \sum_{n=k+1}^{k+\nu(k)} \tilde{c}(Z_n) - \nu(k)\,\hat{C}_N\right) \tag{53}$$

for $p < d(i)$. The derivative w.r.t. $\alpha_{i,d(i)}$ is obtained in an analogous manner and will be omitted.

Proof: Consider the homogeneous Markov chain $\{Z_n\}$. Because the chain is aperiodic and irreducible in a finite state space, it is geometrically ergodic with a unique stationary distribution. Hence there is a constant $0 < \rho < 1$ so that for any function $d : S \to \mathbb{R}$, there is a positive constant $K_d < \infty$ such that

$$|\mathbb{E}[d(Z_n) \mid X_0] - \mathbb{E}[d(Z_\infty)| \leq K_d\rho^n \text{ a.s.},$$

where Z_∞ has the stationary distribution of the chain. This in turn implies that for each k, the sum of the difference processes is absolutely summable:

$$\sum_{n=1}^{\infty} |\mathbb{E}[d(Z_n) \mid Z_0 = (i,a)] - \mathbb{E}[d(Z_n) \mid Z_0 = (i,\tilde{u})]| \leq 2K_d \sum_{n\geq 0} \rho^n < \infty.$$

This together with the dominated convergence theorem can be used to interchange limits and expectations in (50) and establish that:

$$\mathbb{E}[L] = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\left(\delta_{ia}(k) \sum_{n=k}^{N} (c(Z_n) - \mathsf{E}[c(Z_n(k)])\right)$$

where $Z_n(k) = (X_n(k), u_n(k))$ is started at the point $(i, \tilde{u}(k))$ which follows (5) and (1). Construct a version of this process via the nominal process $\{Z_n\}$:

$$Z_n(k) = \begin{cases} (X_n, u_n) & n < k \\ (i, \tilde{u}) & n = k \text{ (if } X_n = i) \\ (X_{n+\nu(k)}, u_{n+\nu(k)}) & n > k \end{cases}$$

The idea of the hitting time until the nominal reaches a phantom system is illustrated in Figure 3.

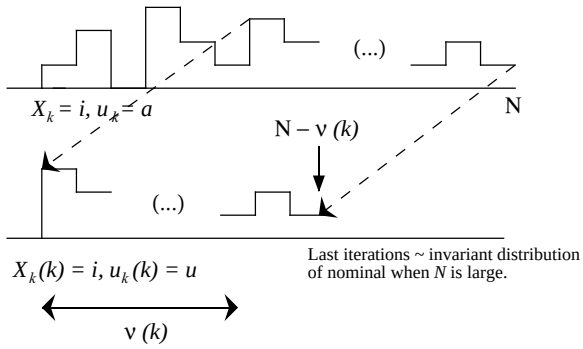

Figure 3: Frozen phantoms wait until nominal hits the initial state.

Using this version of the process, for each k, the difference of the finite horizon sum in the inner brackets is:

$$\begin{aligned} &c(i,a) - c(i,\tilde{u}(k)) + \sum_{n=k+1}^{N} c(Z_n) - \sum_{n=k+1+\nu(k)}^{N+\nu(k)} c(Z_n) \\ &= c(i,a) - c(i,\tilde{u}(k)) + \sum_{n=k+1}^{(k+\nu(k))\vee N} c(Z_n) - \mathbf{1}_{\{k+\nu(k)<N\}} \sum_{n=N+1}^{N+\nu(k)} c(Z_n). \end{aligned}$$

We now show that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \delta_{ia}(k) \mathbb{E}\left(\mathbf{1}_{\{k+\nu(k)>N\}} \sum_{N+1}^{N+\nu(k)} |c(Z_n)| \right) = 0. \tag{54}$$

First, notice that $|c(Z_n)| < K_1$ is uniformly bounded because the state space is finite. Define $\tau(u) = \inf\{k \geq 1,\ Z_k = (i,u)\}$ as the first return time to state (i,u). Clearly for each value of $u \in \mathcal{A}(i)$, the hitting time until the first return to (i,u) (starting from (i,a)) is bounded a.s. by $\tau(u)$. Because $\nu(k)$ is a first hitting time for some $\tilde{u}(k) \in \mathcal{A}(i)$, then $\nu(k) \leq \tau$ a.s., where $\tau = \max(\tau(u)\colon u \in \mathcal{A}(i))$. This implies that for any N, the sum in (54) is bounded by:

$$K_1 \,\mathbb{E}\left(\frac{\tau}{N}\sum_{k=1}^{N}\delta_{ia}(k)\mathbf{1}_{\{\tau>N-k\}}\right).$$

The Markov chain $\{Z_n\}$ is positive recurrent, implying that τ is a.s. finite and $\mathbb{E}\tau = \theta_{iu}\,\pi_i(\theta) < \infty$ for some $u \in \mathcal{A}(i)$. Therefore for any $\epsilon > 0$ there exists $b_\epsilon < \infty$ (independent of N) such that $\mathbb{E}[\tau \mathbf{1}_{\{\tau>b\}}] < \epsilon$. Take $N > b_\epsilon$ and calculate:

$$\begin{aligned}
&K_1 \,\mathbb{E}\left(\frac{\tau}{N}\sum_{k=1}^{N}\delta_{ia}(k)\mathbf{1}_{\{\tau>N-k\}}\right)\\
&\leq K_1 \,\mathbb{E}\left(\frac{\tau}{N}\sum_{k=1}^{N-b_\epsilon}\mathbf{1}_{\{\tau>N-k\}} + \sum_{k=N-b_\epsilon+1}^{N}\mathbf{1}_{\{\tau>N-k\}}\right)\\
&\leq K_1 \,\mathbb{E}\left(\frac{\tau}{N}\sum_{k=1}^{N-b_\epsilon}\mathbf{1}_{\{\tau>b_\epsilon\}} + \sum_{k=0}^{b_\epsilon}\mathbf{1}_{\{\tau>k\}}\right)\\
&\leq \epsilon + \frac{b_\epsilon \mathbb{E}\tau}{N},
\end{aligned}$$

which shows (54). With (54), it follows that:

$$\mathbb{E}[L] = \lim_{N\to\infty}\frac{1}{N}\mathbb{E}\left[\sum_{k=1}^{N}\delta_{ia}(k)\left(c(i,a) - c(i,\tilde{u}(k)) + \sum_{n=k+1}^{k+\nu(k)}c(Z_n) - \sum_{n=N+1}^{N+\nu(k)}c(Z_n)\right)\right].$$

By the Markov property, the last sum above can also be expressed by $\sum_{m=0}^{\infty} c(Z_n^{(N)})\mathbf{1}_{\{m<\nu(k)\}}$, where $Z_0^{(N)} \sim \mathsf{P}^N$ has the distribution of the N-step transition of the nominal chain. This distribution converges to the invariant distribution as N grows. Because $\{Z_k\}$ is time homogeneous, the distribution of $\nu(k)$ (see (49)) is independent of k. Therefore,

$$\lim_{N\to\infty}\mathbb{E}_\psi\sum_{m=0}^{\infty}c(Z_n^{(N)})\mathbf{1}_{\{m<\nu(k)\}} = \mathsf{E}\sum_{m=0}^{\infty}C(\theta)\mathbf{1}_{\{m<\nu(k)\}} = C(\theta)\mathsf{E}[\nu(k)],$$

which completes the proof, because $\nu(k)$ are uniformly bounded in N and $\hat{C}_N \to C(\theta)$ a.s. so the expectation of the product will converge to $C(\theta)\mathsf{E}[\nu(k)]$. □

Notice that the estimator $\hat{G}_N$ is not measurable w.r.t. $\mathfrak{F}_N(\theta)$, because for those values of $k \approx N$ for which $k + \nu(k) > N$ it requires observations "into the future" or off-line simulations. Nonetheless, from the proof of (54) it follows that the truncated estimator:

$$\frac{1}{N}\sum_{k=1}^{N} \delta_{ia}(k) \left([c(i,a) - c(i,\tilde{u}(k)] + \sum_{n=k+1}^{(k+\nu(k))\vee N} \tilde{c}(X_n,\theta) - \nu(k)\,\hat{C}_N \right)$$

is also a consistent estimator of L_N. In order to apply stochastic approximation, however we will not truncate the estimation.

5.2 Numerical Comparison of Gradient Estimators

System Parameters

We simulated the system

$$A(0) = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}, A(1) = \begin{pmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \end{pmatrix}, A(2) = \begin{pmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{pmatrix}.$$

The action probability matrix $(\theta(i,a))$ and cost matrix $(c(i,a))$ were chosen as:

$$(\theta(i,a)) = \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.4 & 0.4 & 0.2 \end{bmatrix}, \quad (c(i,a)) = -\begin{bmatrix} 50.0 & 200.0 & 10.0 \\ 3.0 & 500.0 & 0.0 \end{bmatrix}$$

Gradient Estimates in Canonical Coordinates

Using canonical coordinates, the theoretical values of the generalized gradient for this problem are:

$$\mathcal{G}[C(\theta)] = \begin{pmatrix} -9.010 & 18.680 & -9.670 \\ -45.947 & 68.323 & -22.377 \end{pmatrix}. \tag{55}$$

When computing the generalized gradient estimators it is often necessary to normalize the result: even if $\sum_a \mathcal{G}_{ia}[F(\theta)] = 0$ for all F, numerical errors often make the estimation biased. Because it is important to maintain feasibility, we normalized the results of the estimation using:

$$y_{ia} = \hat{Y}_{ia} - |\hat{Y}_{ia}| \frac{\sum_{a\in\mathcal{A}(i)} \hat{Y}_{ia}}{\sum_{a\in\mathcal{A}(i)} |\hat{Y}_{ia}|}$$

where $\hat{Y}$ are the observations and y the corresponding normalization. The normalized estimation using (48) with Doeblin phantoms, is called $\hat{G}_N^{\text{Doeblin}}$, and that of (52) is called $\hat{G}_N^{\text{Frozen}}$. Confidence intervals were estimated at level 0.05 using the normal approximation with 100 batches. For batch sizes $N = 100$ and 1000, respectively the gradient estimates are (to be compared with the theoretical values in (55)):

$$\hat{G}_{100}^{\text{Doeblin}} = \begin{pmatrix} -7.548 \pm 0.580 & 16.594 \pm 0.602 & -9.046 \pm 0.548 \\ -45.209 \pm 1.620 & 67.315 \pm 1.519 & -22.106 \pm 1.678 \end{pmatrix}$$

$$\hat{G}_{1000}^{\text{Doeblin}} = \begin{pmatrix} -7.997 \pm 0.198 & 17.351 \pm 0.195 & -9.354 \pm 0.186 \\ -46.738 \pm 0.456 & 69.668 \pm 0.462 & -22.929 \pm 0.534 \end{pmatrix}$$

$$\hat{G}_{100}^{\text{Frozen}} = \begin{pmatrix} -7.851 \pm 0.618 & 17.275 \pm 0.664 & -9.425 \pm 0.594 \\ -44.586 \pm 1.661 & 66.751 \pm 1.657 & -22.164 \pm 1.732 \end{pmatrix}$$

$$\hat{G}_{1000}^{\text{Frozen}} = \begin{pmatrix} -8.361 \pm 0.215 & 17.928 \pm 0.240 & -9.566 \pm 0.211 \\ -46.164 \pm 0.468 & 68.969 \pm 0.472 & -22.805 \pm 0.539 \end{pmatrix}$$

The variance of the unnormalized estimators is shown in Table 1, together with the corresponding CPU time. Clearly to achieve the same precision the Doeblin phantoms require doubling the simulation time. The obvious advantage of our estimator is that calculations do not require knowledge of the transition probabilities but only the observation of the process. In addition, as clearly shown in Table 1, the extra computation required for our estimation is negligible. we performed a series of results using different parameters and all behaved in the same manner.

$N = 1000$	$\mathsf{Var}[\hat{G}_N^{\text{Doeblin}}]$			$\mathsf{Var}[\hat{G}_N^{\text{Frozen}}]$		
$i = 0$	1.022	0.995	0.908	1.180	1.506	1.159
$i = 1$	5.420	5.562	7.444	5.700	5.800	7.565
CPU	4 secs.			2 secs.		

Table 1: Variance of Doeblin phantom vs Frozen phantom, canonical coordinates

As mentioned in Section 1.5 the closest approach to the algorithms in this paper is that in [4] which uses a score function method to estimate the gradients. We simulated this score function method of [4] for gradient estimation with the following parameters: forgetting factor 1 (otherwise the estimates are biased), batch sizes of $N = 1000$ and 10000. In both cases a total number of $10,000$ batches were simulated which required CPU times of 1348 seconds and 13492 seconds, respectively. The variance of the estimates are given Table 2.

$$\hat{G}_{10000}^{\text{Score}} = \begin{pmatrix} -3.49 \pm 5.83 & 16.91 \pm 7.17 & -13.42 \pm 5.83 \\ -41.20 \pm 14.96 & 53.24 \pm 15.0 & -12.12 \pm 12.24 \end{pmatrix}$$

$$\hat{G}_{1000}^{\text{Score}} = \begin{pmatrix} -6.73 \pm 1.84 & 19.67 \pm 2.26 & -12.93 \pm 1.85 \\ -31.49 \pm 4.77 & 46.05 \pm 4.75 & -14.55 \pm 3.88 \end{pmatrix}$$

$N = 1000$	$\mathsf{Var}[\hat{G}_N^{\text{Score}}]$		
$i = 0$	89083	135860	89500
$i = 1$	584012	593443	393015
CPU	1374 secs.		

$N = 10000$	$\mathsf{Var}[\hat{G}_N^{\text{Score}}]$		
$i = 0$	876523	1310900	880255
$i = 1$	5841196	5906325	3882805
CPU	13492 secs.		

Table 2: Variance of Score Function estimator

Gradient Estimates in Spherical Coordinates

For the same data, the parameterization in spherical coordinates yields the following theoretical values for the gradient, using again $N = 100, 1000$ for the batch size, and 10 batches for the estimation of variances.

$$\nabla_\alpha C_N(\alpha) = \begin{pmatrix} 45.05 & -55.07 \\ 187.58 & -159.91 \end{pmatrix}$$

The gradient estimates are

$$\widehat{\nabla C}_{100}^{\text{Doeblin}} = \begin{pmatrix} 41.720 \pm 4.695 & -55.096 \pm 1.464 \\ 197.166 \pm 8.062 & -164.701 \pm 4.610 \end{pmatrix}$$

$$\widehat{\nabla C}_{1000}^{\text{Doeblin}} = \begin{pmatrix} 41.703 \pm 1.278 & -53.667 \pm 0.455 \\ 191.249 \pm 2.858 & -167.048 \pm 1.228 \end{pmatrix}$$

$$\widehat{\nabla C}_{100}^{\text{Frozen}} = \begin{pmatrix} 43.333 \pm 4.791 & -54.191 \pm 2.096 \\ 196.956 \pm 8.951 & -161.200 \pm 4.918 \end{pmatrix}$$

$$\widehat{\nabla C}_{1000}^{\text{Frozen}} = \begin{pmatrix} 44.322 \pm 1.323 & -53.656 \pm 0.712 \\ 189.549 \pm 2.829 & -164.329 \pm 1.231 \end{pmatrix}$$

The variance matrix is in Table 3. Although estimation of the gradients required the same amount of CPU time as the generalized gradient, when implementing an on-line control some extra computational time must be added in calculating the trigonometric expressions necessary to change coordinates every time that an update of the control variables is performed.

$N = 1000$	$\mathsf{Var}[\widehat{\nabla C}_N^{\text{Doeblin}}]$		$\mathsf{Var}[\widehat{\nabla C}_N^{\text{Frozen}}]$	
$i = 0$	42.558	5.404	45.604	13.206
$i = 1$	212.74	39.26	208.43	39.431
CPU	4 secs.		2 secs.	

Table 3: Variance of Doeblin phantom vs Frozen phantom, spherical coordinates

It is clear from our experiments that the frozen phantom implementation not only is robust (thus more appropriate when the underlying parameters are unknown) but also

more efficient: the CPU time is about half of that using Doeblin phantoms, yet their variances are comparable.

5.3 Statistical Properties of frozen phantom estimators

We will assume that the model satisfies the Assumptions 1 and 2. as well as:

Assumption 3 *The random time for coupling has uniformly integrable variance:*

$$\sup_{\alpha} \mathbb{E}_{\psi} \left(|\nu_{i,a}(k)|^2 \right) < \infty,$$

where the hitting time is defined by:

$$\nu_{i,a}(k) = 1 + \min\{n > 0 \colon X_{k+n} = i, u_{k+1} = a \mid X_k = i, u_k = a\}.$$

For any function of the state $f(Z)$, we will use a capital letter to denote the invariant average, namely:

$$F(\alpha) = \sum_{(i,a) \in \mathcal{Z}} \pi_{i,a}(\alpha) f(i,a).$$

An *estimation interval* is an interval of N steps of the underlying MDP $\{Z_k\}$ during which the observations are averaged to form the various estimation terms. The n-th "batch" for the estimation comprises all steps k such that

$$k \in I_n \equiv \{nN + 1, \ldots (n+1)N\} \tag{56}$$

The *sample average* of batch n will be denoted:

$$\hat{F}_n = \frac{1}{N} \sum_{k=nN+1}^{(n+1)N} f(Z_k), \tag{57}$$

the *total running average* $\hat{\hat{F}}_n$ and *exponentially discounted running average* $\hat{\hat{F}}^{\delta}_n$ up to batch n will be denoted by

$$\hat{\hat{F}}_n = \frac{1}{n} \sum_{m=0}^{n-1} \hat{F}_m = \hat{\hat{F}}_{n-1} + \frac{1}{n} \left(\hat{F}_{n-1} - \hat{\hat{F}}_{n-1} \right), \tag{58}$$

$$\hat{\hat{F}}^{\delta}_n = \hat{\hat{F}}^{\delta}_{n-1} + \delta \left(\hat{F}_{n-1} - \hat{\hat{F}}^{\delta}_{n-1} \right), \quad \delta \in (0,1) \tag{59}$$

Define the index of the first "living phantom" at time n by:

$$\phi(n) = \min\{k \leq n \colon \nu(k) + k \leq n\}$$

From the results in Section 5 the frozen phantom estimators of the various components (i, p) of the gradient are all of the generic form:

$$\widehat{F'}_{i,p}(n) = \frac{K_{ip}(\alpha)}{N} \sum_{k=\phi(nN+1)}^{(n+1)N} \delta_{ip}(k) \left[f(i,a) - f(i,\tilde{u}_k) + \sum_{j=(k+1)\wedge(nN+1)}^{(k+\nu(k))\vee(n+1)N} f(Z_j) - \nu(k)\widehat{\Gamma}_n \mathbf{1}_{\{\nu(k)+k\in I_n\}} \right] \tag{60}$$

where $a = p - 1$ and we have omitted the subscript $i, \tilde{u}_k$ from $\nu(k)$. The constant $K_{ip}(\alpha)$ is related to the derivative of $\theta(\alpha)$ and the indicator δ_{ip} is related to the sampling frequencies for phantom production. The term $\widehat{\Gamma}_n$ refers to an estimator of $F(\alpha)$ that must consider observations of $f(X_n, u_n)$ in I_n (56) and may also consider past observations. The truncation of the sum is illustrated in Figure 4.

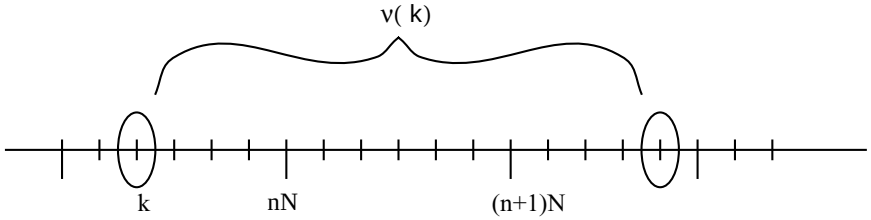

Figure 4: Estimation by batches

Lemma 4 *If $\widehat{\Gamma}_n$ is replaced by the actual average $F(\alpha)$ in (60) then $\mathbb{E}_{\pi(\psi)}[\hat{F}^{(0)}] = \nabla_\alpha F(\alpha)$.*

Proof: Notice that under the stationary measure consecutive estimators have the same distribution (although they are not independent), because the invariant distribution of the number of living phantoms at the start of the interval is independent of n. From the ergodicity of the underlying MDP, $\mathbb{E}_{\pi(\psi)}[\hat{F}^{(0)}(n)] = \lim_{n\to\infty}(1/n)\sum_{m=0}^{(n-1)} \hat{F}^{(0)}(m)$. Because the estimation by batches considers breaking up the partial sums of the estimation, then the difference:

$$\frac{1}{n}\sum_{m=0}^{(n-1)} \widehat{F'}_{i,p}(m) - \frac{K_{ip}(\alpha)}{nN} \sum_{k=1}^{nN} \delta_{ip}(k) \left[f(i,a) - f(i,\tilde{u}_k) + \sum_{j=(k+1)}^{k+\nu(k)} f(Z_j) - \nu(k)F(\alpha) \right]$$

tends to zero in absolute value, a.s., which follows from Assumption 3. From the results in Section 5.1 the RHS converges a.s. to $\nabla_\alpha F(\alpha)$ as $n \to \infty$, for any fixed value of the batch size N, which establishes the claim. □

The above claim is the key to the characterization of the asymptotic bias of the stochastic approximation and it can be used as well to implement diverse variants of the estimation aimed at reducing the corresponding bias. Which variant to use depends mostly on the application.

Corollary 1 *Assume that* $\mathbb{E}_{\pi(\psi)}\widehat{\Gamma}_n = F(\alpha)$ *is an unbiased estimator under the invariant measure of the process. Then the bias of the estimation per batch is given by:*

$$\mathbb{E}_{\pi(\psi)}[\widehat{F'}(n)] - \nabla_\alpha F(\alpha) = \frac{K_{ip}(\alpha)}{N} \sum_{k:\ k+\nu(k)\in I_n} \delta_{ip}(k)\, \mathrm{Cov}_{\pi(\psi)}(\nu(k), \widehat{\Gamma}_n). \tag{61}$$

Implementation Issues:

- If N is sufficiently large, then the sample average estimator $\hat{F}_n$ (57) has a small variance (of order $1/N$) and using this for $\widehat{\Gamma}_n$ in (60) may result in a negligible bias.
- If frequent updates are required for the algorithm, one may want to favor small batch sizes, in which case the bias may be reduced using a series of alternative solutions:
  - Shift the estimator, setting $\widehat{\Gamma}_n = \hat{F}_{n-s}$ as the sample average obtained s batches before. Because of the Markovian structure of the process the correlation will decrease as s increases. However this may yield slow adaptation behavior in practice.
  - Use $\widehat{\Gamma}_n = \hat{\hat{F}}_n$ (58) as the total running average. Although this would yield an asymptotically ($n \to \infty$) unbiased estimator, the use of running averages may not be very convenient when tracking.
  - Adapt the running average to either a truncated window average of the form $\widehat{\Gamma}_n = \frac{1}{s}\sum_{m=n-s}^{n} \hat{F}_m$, or use the exponentially discounted average $\hat{\Gamma}_n = \hat{\hat{F}}_n^{\delta}$ (58).

6 Stochastic Approximation Algorithms and Convergence

In this section we present weak convergence proofs for the learning algorithms that use the parameter free gradient estimators (frozen phantoms) of Section 5.1. These learning algorithms are stochastic versions of the deterministic algorithms presented in Section 2 (unconstrained case) and Section 3 (constrained case). The algorithms assume that the transition probabilities are *unknown*, and only observations of the states of the dynamic process are available. Under these conditions the cost and constraint functions are not analytically available, and neither are their gradients. Using the simulation-based formula (60) for the estimation of the gradients with the local sample averages for the estimation of the cost and constraint functions may lead to a bias, thus making the algorithm suboptimal.

The focus of this section is to point out the actual bias as well as indications to reduce or eliminate this bias based on the results of Section 5.3, mainly from Corollary 1.

Notation: In the sequel we will use the following expectation operators: $\mathbb{E}_{\pi(\psi)}$ denotes expectation w.r.t measure $\pi(\psi)$ parameterized by ψ; $\mathbb{E}_\psi$ denotes expectation w.r.t the underlying probability measure of $\{Z_1, \ldots, Z_N\}$, where $N \geq 1$ is a fixed positive integer (see also Remark following Theorem 3). In Section 4 we will use E to denote expectation w.r.t. to the distribution of certain simulated random variables called phantoms.

6.1 Convergence Proof for Unconstrained MDP

Weak convergence of the stochastic approximation algorithms (23), (25) to the deterministic ODEs (22), (24), respectively, can be established by well known results in stochastic approximations, see [21]. To simplify the exposition assume that estimation of the various gradients is performed over observation intervals of $N \geq 1$ transitions, during which ψ is kept constant. At the end of these estimation intervals the control values ψ are updated.

For a weak convergence proof of the generalized gradient stochastic approximation algorithm (23) simply substitute $\psi(n) = \theta(n)$, $\psi(t) = \theta(t)$, $Y(n) = \theta(n)\widehat{\mathcal{G}_{iu}}[C(\theta(n))]$, $g(\psi)$ by RHS of the ODE (22) in the following theorem. For a weak convergence proof of the spherical coordinate stochastic approximation algorithm substitute $\psi(n) = \alpha(n)$, $\psi(t) = \varphi(t)$, $Y(n) = \widehat{\nabla_\alpha C}[\alpha(n)]$, $g(\psi)$ by the RHS of the ODE (24).

Theorem 5 *Consider the recursion:*

$$\psi(n+1) = \psi(n) + \epsilon\, Y(n), \tag{62}$$

where

- $Y(n)$ *is* $\mathfrak{F}_{nN}$*-measurable and uniformly integrable,*
- *there exists a continuous and locally bounded function* $g(\psi)$ *such that:*

$$\lim_{m\to\infty} \frac{1}{m}\mathbb{E}_\psi \sum_{n=1}^{m} Y(n) = g(\psi), \textit{and}$$

- *The ODE*

$$\frac{d\psi(t)}{dt} = g(\psi(t)) \tag{63}$$

  has a unique solution for every initial condition and the set of stable points is non null.

Then as the constant step size $\epsilon_i \to 0$*, the interpolated process* $\psi^\epsilon(\cdot)$ *converges weakly (in distribution) to the solution of the ODE (63), where:*

$$\psi^\epsilon(t) = \psi(n), \ \textit{for all } t \in [n\epsilon, (n+1)\epsilon).$$

Numerous variants of Theorem 5 exist, taking account of weaker conditions, decentralized operation of the control agents (components of θ or α), etc. For our purposes Theorem 5 will suffice to characterize the asymptotic (local) optimality of our learning algorithms.

6.2 Convergence Proof for Constrained MDP

Here we present the weak convergence proofs of the stochastic versions of the three deterministic algorithms presented in Section 3.1, Section 3.2 and Section 3.3 for constrained MDPs.

For notational convenience we only consider equality constraints here (as mentioned in Section 3.2 the inequality constraints can be handled with minor modifications) so that the constrained MDP problem (12), (13), (14) reads

$$\min_{\alpha_i \in \mathbb{R}^{d(i)}, i \in S} C(\alpha)$$
$$\text{s.t. } B_l(\alpha) = 0, \quad l = 1, \ldots, L$$

A control "agent" is associated with each of the possible visited states $i \in S$. The control parameter for this agent is the vector $(\alpha_{ip}, p \in \mathcal{P}(i))$, plus an agent for the (artificial control) variable representing the Lagrange multiplier λ. The scheme works by observing the process over an estimation interval, during which the value of the control parameter does not change and the estimators $\hat{C}^\epsilon(n), \hat{B}^\epsilon(n)$ for the sample averages are calculated, along with the gradient estimators (60) of the form $\widehat{\nabla C}^\epsilon(n), \widehat{\nabla B}^\epsilon(n)$. Notice that the computation of each partial derivative can be made locally at each controller. Throughout this section suppose that Assumptions 1, 2 and 3 hold. In addition, let:

$$h_0(\alpha) = \mathbb{E}_{\pi(\psi)}[\widehat{\nabla C}^\epsilon(n)], \quad h_l(\alpha) = \mathbb{E}_{\pi(\psi)}[\widehat{\nabla B}_l^\epsilon(n)],$$

be the invariant averages of the batch estimation (refer to Section 5.3).

6.2.1 First-Order Primal Dual Algorithm. Consider the stochastic approximation procedure where the control parameter (α, λ) is updated with (c.f. (28), (29))

$$\alpha^\epsilon(n+1) = \alpha^\epsilon(n) - \epsilon \left(\widehat{\nabla C}^\epsilon(n) + \sum_{l=1}^{L} \left(\lambda_l^\epsilon(n) + \rho \hat{B}_l^\epsilon(n) \right) \widehat{\nabla B}_l^\epsilon(n) \right) \tag{64}$$

$$\lambda^\epsilon(n+1) = \lambda^\epsilon(n) + \epsilon \, \hat{B}^\epsilon(n) \tag{65}$$

Proposition 1 *Define the interpolated process $\alpha^\epsilon(t)$ as in (30) and analogously for $\lambda^\epsilon(t)$. Then as $\epsilon \to 0$, the interpolated process converges in distribution to the solution of the ODE (c.f. (33))*

$$\frac{d}{dt}\alpha(t) = -\left(h_0[\alpha(t)] + \sum_{l=1}^{L} (\bar{\lambda}_l + \rho \, B_l[\alpha(t)]) \, h_l[\alpha(t)] + \kappa[\alpha(t)] \right) \tag{66}$$

$$\frac{d}{dt}\lambda(t) = B[\alpha(t)],$$

where the added drift is defined as:

$$\kappa(\alpha) = \rho \lim_{n \to \infty} \sum_{l=1}^{L} \mathrm{Cov}_{\pi(\psi)}[\hat{B}_l^\epsilon(n), \widehat{\nabla B}_l^\epsilon(n)].$$

Proof: The result follows by direct application of Theorem 5, see also [18]. Indeed, the uniform integrability of the updates is ensured by Assumption 3, while continuity of the invariant expectations follows from Assumption 1. To characterize the drift functions, use:

$$\mathbb{E}_{\pi(\psi)}\left(\hat{B}_l^\epsilon(n)\widehat{\nabla B}_l^\epsilon(n)\right) = \mathbb{E}_{\pi(\psi)}(\hat{B}_l^\epsilon(n))\,\mathbb{E}_{\pi(\psi)}(\widehat{\nabla B}_l^\epsilon(n)) + \mathrm{Cov}_{\pi(\psi)}[\hat{B}_l^\epsilon(n), \widehat{\nabla B_l^\epsilon}(n)],$$

which establishes the result. □

Remark: Trade-off Between Bias and Tracking Ability. The three sources of bias in the stochastic approximation algorithm (64), (65) are the bias in the estimates $\widehat{\nabla B}$, $\widehat{\nabla C}$ and $\hat{B}$. In [2] we present several numerical examples that study the effect of this bias.

A quick mathematical artifice (with little practical value) for eliminating the bias is to use batch sizes $N(\epsilon) \to \infty$ as $\epsilon \to 0$. Then the ODE (66) becomes identical to (33). In the numerical examples of [1] we chose $N = 1000$ – recall for finite N the bias is $O(1/N)$. Although choosing $N(\epsilon) \to \infty$ is theoretically appealing, it is of no practical use since the algorithm will not respond to changes in the optimal policy caused by time variations in the parameters of the MDP. In [2] we us estimation intervals of length $N = 5, 10$ observations to update often, and our results can be implemented even for $N = 1$.

Indeed, the bias in the gradient estimates $\widehat{\nabla B}$, $\widehat{\nabla C}$ can be eliminated asymptotically using the running averages in (58). Then the only source of bias is $\hat{B}$ – this gives rise to the term $\kappa(\alpha)$ in the ODE (66). Of course the bias in $\hat{B}$ can also be eliminated asymptotically using the running average (58) in which case $\kappa(\alpha) = 0$.

On the other hand, the running averages (58) do not respond to changes in the underlying parameters (e.g. transition probabilities) of the MDP since they are decreasing step size algorithms. Hence they cannot be used for tracking time varying optimal policies. To handle this tracking case, we use the exponentially weighted running averages of (58). Using a two time scale stochastic approximation argument it is easily shown [19] that if $\epsilon/\delta \to 0$, e.g., if $\delta = \sqrt{\epsilon}$, and $\epsilon \to 0$, then the estimates are unbiased. In practical implementation, for non zero δ, the estimates are biased.

6.2.2 Augmented Lagrangian Multiplier Algorithm. Consider the following stochastic approximation version of the multiplier algorithm (37):

$$\alpha^\epsilon(n+1) = \alpha^\epsilon(n) - \epsilon\left(\widehat{\nabla C}^\epsilon(n) + \sum_{l=1}^{L}((\lambda_l^\epsilon(n) + \rho B[\alpha^\epsilon(n)])\,\widehat{\nabla B_l}^\epsilon(n)\right). \tag{67}$$

where $\{\lambda^\epsilon(n), \epsilon > 0, n \in \mathbb{N}\}$ is any tight sequence. A trivial example is when $\lambda^\epsilon(n)$ is a bounded constant (a.s.).

The following result regarding the weak convergence of (67) is proved in the appendix.

Proposition 2 *Assume that* $\{\lambda^\epsilon(n), \epsilon > 0, n \in \mathbb{N}\}$ *is tight. Define the interpolated process* $\alpha^\epsilon(t)$ *of (67) as in (30). Then as* $\epsilon \to 0$*, the interpolated process* $\alpha^\epsilon(t)$ *converges in distribution to the solution of the ODE:*

$$\frac{d\alpha(t)}{dt} = -\left[h_0[\alpha(t)] + \sum_{l=1}^{L}(\bar{\lambda}_l + \rho\, B_l[\alpha(t)])\, h_l[\alpha(t)] + \kappa[\alpha(t)]\right], \tag{68}$$

where $\bar{\lambda}_l$ *is an accumulation point of the sequence* $\{\bar{\lambda}(\epsilon), \epsilon > 0\}$ *of (convergent) Cesaro sums:*

$$\bar{\lambda}(\epsilon) \equiv \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} \lambda^\epsilon(n),$$

and the added drift is defined as:

$$\kappa(\alpha) = \rho \lim_{n\to\infty} \sum_{l=1}^{L} \mathrm{Cov}_{\pi(\psi)}[\hat{B}_l^\epsilon(n), \widehat{\nabla B}_l^\epsilon(n)].$$

Remark: In particular, if one uses $\hat{\Gamma}^\epsilon(n) = \hat{\hat{F}}$ in the gradient estimation and also $\hat{G}^\epsilon(n) = \hat{\hat{B}}_n$ (58), then the ODE (68) reduces to (38) – which is the ODE for the deterministic fixed multiplier algorithm. Then Result 1 of Section 3.2 applies which implies that α_n converges weakly to a near optimal, provided that the pair $(\bar{\lambda}, \rho)$ is well chosen.

Consider the following update of the multiplier λ in (67). Define $J = \lfloor 1/\epsilon \rfloor$ and consider the recursion

$$\lambda(n+1) = \lambda(n) + \bar{B}(n/J)\mathbf{1}_{\left\{\frac{n}{J}\in\mathbb{N}\right\}} \tag{69}$$

together with (67). Thus the multiplier is updated once every J time points. Here

$$\bar{B}(n/J) = \frac{1}{J}\sum_{j=(n-1)J+1}^{nJ} \hat{B}(j)$$

If $\hat{\Gamma}_n = \hat{\hat{F}}_n$ in (60) and $\hat{B}_l^\epsilon(n) = \hat{\hat{B}}_n$ in (67), then as $\epsilon \to 0$, the algorithm (69), (67) converges weakly to the deterministic system (34), (35) which is the exact multiplier algorithm. As mentioned in Section 3.2 this in turn converges to a local KT point. In a practical implementation, one would choose J as a large positive integer, In our numerical examples even a choice of $J = 10$ resulted in convergence to a KT point.

6.2.3 Projected Gradient Primal Algorithm. Consider now the stochastic version of the projection algorithm:

$$\tilde{\alpha}^\epsilon(n+1) = \alpha^\epsilon(n) - \epsilon \widehat{\nabla C}^\epsilon(n) \tag{70}$$

$$\mu^\epsilon_{j+1}(n) = \left(\mu^\epsilon(n) - \bar{\epsilon}\left[\widehat{\nabla B}^\epsilon(n)' \widehat{\nabla B}^\epsilon(n)\mu^\epsilon_j(n) - \hat{B}^\epsilon(n) + \widehat{\nabla B}^\epsilon(n)(\tilde{\alpha}(n+1) - \alpha(n))\right]\right)_+ \tag{71}$$

$$\mu^\epsilon_0(n+1) = \mu^\epsilon_J(n)$$

$$\alpha^\epsilon(n+1) = \tilde{\alpha}^\epsilon(n+1) - \widehat{\nabla B}^\epsilon(n)\mu(n+1). \tag{72}$$

Using the known methods of [21], if the projection was accurate, then a modification of Theorem 5 would yield the result that the interpolation process converges weakly as $\epsilon \to 0$ to the solution of the corresponding projected ODE. By "accurate" we mean that $B(\alpha^\epsilon(n)), \nabla B(\alpha^\epsilon(n))$ be exact in the updates of $\mu^\epsilon(n)$. This follows because even if $B(\cdot)$ has been linearized to solve for the projection, from uniform boundedness of the gradients in Assumption 1 there is a small enough value of ϵ that bounds the error of the linear approximation within any specified tolerance.

The only way that accuracy can be obtained for $\mu^\epsilon(\cdot)$ to converge to the projection (within a tolerated error) is that the estimation be *strongly consistent*. That is, for the terms $\widehat{\nabla B}^\epsilon(n)$ one need to implement (60) in a manner such that for the fixed α process,

$$\widehat{\nabla B}^\epsilon(n)' \widehat{\nabla B}^\epsilon(n) \to \nabla B(\alpha)' \nabla B(\alpha), \quad \text{a.s}$$

A straightforward solution to this estimation problem is simply to use running averages in (58) as well as for the gradient itself.

7 Numerical Examples

Extensive numerical studies of the stochastic approximation algorithms for optimizing the constrained MDP are given in [1] and [2]. Here we report a few of these examples.

7.1 Example 1: Learning Algorithm for MDP with 2 constraints

Here we consider the stochastic adaptive control of the following constrained MDP:

$S = \{0, 1\}$, $\mathcal{A}(i) = \{0, 1, 2\}$, $i \in S$ (i.e., $d(0) = d(1) = 2$),

$$A(0) = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}, \quad A(1) = \begin{pmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \end{pmatrix}, \quad A(2) = \begin{pmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{pmatrix}.$$

The cost matrix $(c(i,a))$, two constraints ($L = 2$) matrices $(\beta_1(i,a))$, $(\beta_2(i,a))$ are

$$(c(i,a)) = -\begin{bmatrix} 50 & 200 & 10 \\ 3 & 500 & 0 \end{bmatrix}, \quad \beta_1 = \begin{bmatrix} 20 & 100 & -8 \\ -3 & 4 & -10 \end{bmatrix}, \; \beta_2 = \begin{bmatrix} 10 & -20 & 22 \\ -19 & 17 & -15 \end{bmatrix}.$$

The optimal control policy incurs a cost of -111.80 (or equivalently a reward of 111.80) and is randomized with probabilities (18)

$$\theta^* = \begin{bmatrix} 0 & 0.2 & 0.8 \\ 0 & 0.28 & 0.72 \end{bmatrix}.$$

We illustrate the performance of the learning algorithm using the stochastic approximation versions of the multiplier algorithm (Section 6.2.2) and gradient projection algorithm (Section 6.2.3). These algorithms assume no knowledge of the transition probabilities. The batch size used for the frozen phantoms to estimate the gradient is $N = 1000$. The algorithms run for $n = 1, \ldots, 10,000$ batches. The algorithms were initialized with initial parameter $\theta(0) = (\theta_{ia}(0))$ as

$$\theta(0) = \begin{bmatrix} 0.1 & 0.1 & 0.8 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}.$$

Since the performance of the deterministic algorithms for the constrained MDP form the basis for the stochastic versions, we also illustrate the performance of the corresponding deterministic algorithms presented in Section 3. These assume full knowledge of the parameters and theoretical values of the derivatives.

Figure 5 shows the reward (negative cost) trajectories of the fixed multiplier algorithm with $\rho = 5$, $\epsilon = 5 \times 10^{-6}$, $\lambda_l = 1.0$, $l = 1, 2$ We found the algorithm quite insensitive to choice of the multiplier λ. The policy estimate at $n = 10,000$ is

$$\hat{\theta}^* = \begin{bmatrix} 0 & 0.192 & 0.808 \\ 0 & 0.275 & 0.724 \end{bmatrix}$$

The performance of the inexact multiplier method was very similar and hence omitted.

Figure 6 shows the reward (negative cost) trajectories of the gradient projection primal algorithm with step sizes $\epsilon = 1 \times 10^{-6}$ and $\bar{\epsilon} = 1 \times 10^{-3}$. The algorithm gets trapped at a local maximum with cost approximately -95. In Figure 7 we initialized the gradient projection primal algorithm with policy

$$\theta(0) = \begin{bmatrix} 0.1 & 0.1 & 0.8 \\ 0.0 & 0.2 & 0.8 \end{bmatrix}.$$

The algorithm converges to the global maximum.

In general we found that the the primal dual, multiplier algorithm and gradient projection algorithms performed comparably.

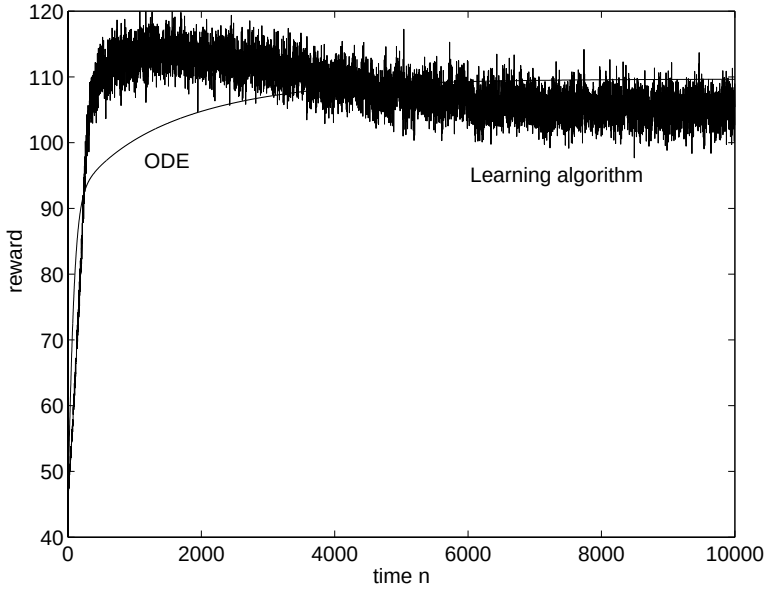

Figure 5: Fixed multiplier Algorithm

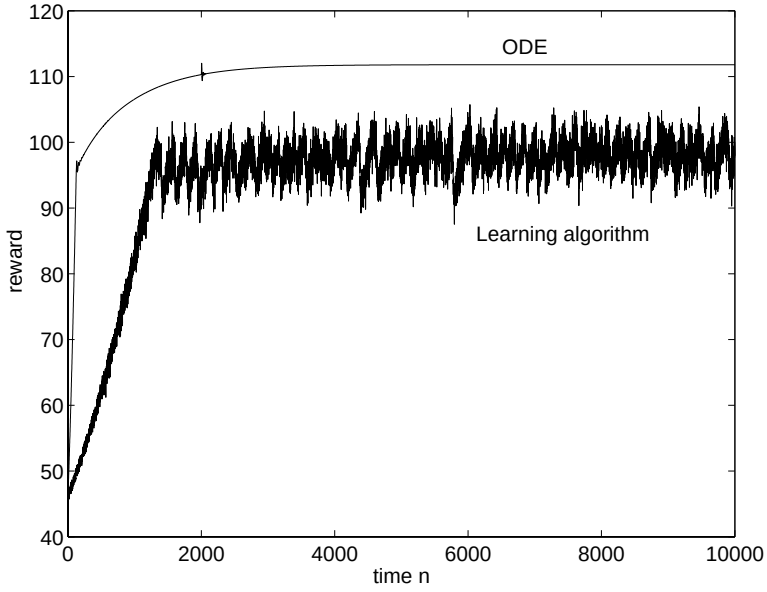

Figure 6: Projected Gradient Primal Algorithm

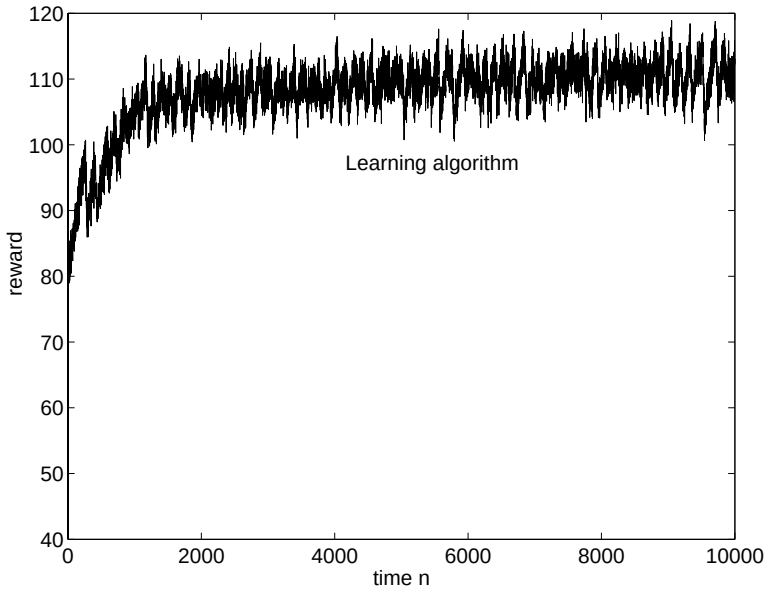

Figure 7: Projected Gradient Primal Algorithm with different initial condition

7.2 Example 2: Learning Algorithm for Constrained MDP with unknown time-varying transition probabilities

The previous section dealt with large batch sizes $N = 1000$. For this case, the bias in the estimates of the gradients and constraints is virtually zero and all three algorithms perform similarly. However, for tracking MDPs with time varying parameters (e.g. transition probabilities) often one needs the algorithm to operate on smaller batch sizes so as to respond faster Here we consider a smaller batch size $N = 10$ and examine the effect of exponential discounted weighting described in Section 5.3.

We consider adaptive stochastic control of the following time-varying constrained MDP: For time $n \leq 4000$, identical parameters to Section 7.1 with optimal cost -111.80. For $4000 < n \leq 12000$, the transition probabilities are

$$A(0) = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \qquad A(1) = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}, \qquad A(2) = \begin{bmatrix} 0.5 & 0.5 \\ 0.45 & 0.55 \end{bmatrix}$$

This has an optimal cost of -44.52 (i.e. reward of 44.52).

We only present results for the primal dual based stochastic gradient algorithm here – see the paper [2] which gives several other numerical examples with small batch sizes for the projected gradient and multiplier algorithm. The algorithm was initialized with randomized policy

$$\theta(0) = \begin{bmatrix} 0.1 & 0.1 & 0.8 \\ 0.0 & 0.2 & 0.8 \end{bmatrix}.$$

Figure 8 illustrates the performance of the primal dual based stochastic approximation algorithm. The parameters used are in the primal dual algorithm are $\rho = 100$, $\epsilon = 2 \times 10^{-7}$ (see (64), (65)).

The choice of the discounting factor δ (58) in the primal dual method clearly shows the trade off between bias and tracking ability in Figure 8. For $\delta = 1.0$, the algorithm has fast tracking properties but a large bias. For $\delta = 0.5$ and $\delta = 0.1$ the bias gets smaller.

Overall we found the primal dual and fixed multiplier based stochastic gradient algorithms easiest to use since they have very few parameters (step sizes) to tune.

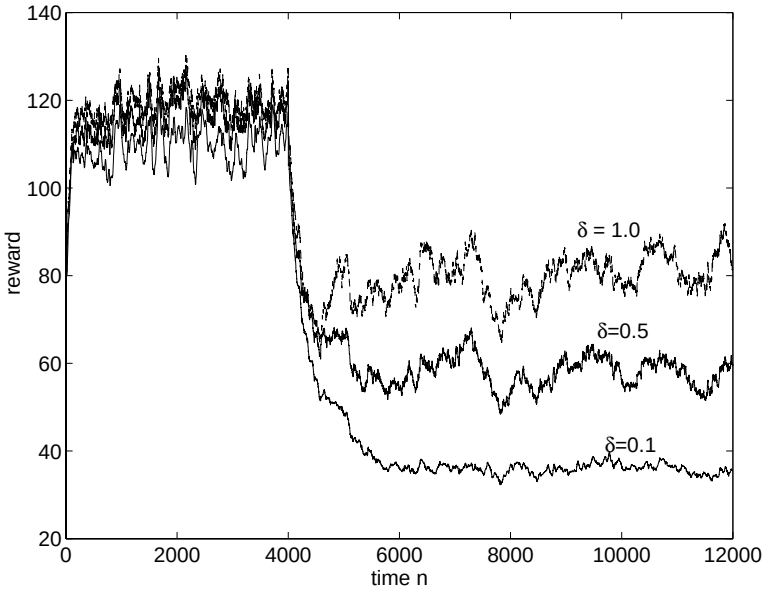

Figure 8: Primal dual algorithm based stochastic adaptive controller

8 Conclusions and Extensions

In this paper simulation based gradient algorithms have been presented for adaptively optimizing a constrained average cost finite state Markov decision process. First a parameterization of the randomized control policy using spherical coordinates was presented. Then a novel measure-valued gradient estimator using frozen phantoms was presented. As illustrated in Section 5.2, this gradient estimator has much smaller variance than score function based gradient estimators. The measure-valued gradient estimator was then used in a stochastic gradient algorithm with fixed step size in order to track time varying Markov decision processes with unknown transition probability matrices. For the unconstrained case, these algorithms are optimal in that they converge weakly to a local minimum. For the constrained case the algorithms are near optimal in that the algorithms converge weakly

to a local minimum with a bias – however, this bias is identifiable and can be made negligible. In extensive numerical examples throughout the paper (Section 2.3, Section 5.2, Section 7), the various steps in the design of the above simulation based adaptive controller are examined.

In a companion paper [2], a detailed numerical study of the frozen phantoms (parameter free gradient estimators) is conducted. The effect of moving averages and exponentially discounted averages on the bias is also studied. See also [1] for further numerical examples. In current work, we are examining applications of the techniques in this paper to admission control of wireless networks. As mentioned in Section 1, in this case the quality of service and blocking probability constraints naturally translate into constraints on the MDP.

Given that the proposed adaptive controller is a fixed step size stochastic approximation algorithm, several variations such as iterate averaging [21, Chapter 11], adaptive step size updating [21, Section 3.2] and decentralized asynchronous implementation [21, Chapter 12] are possible. It is also worthwhile examining the use of similar methods for partially observed Markov decision processes.

9 Appendix: Proof of Proposition 2

Proof: First, from tightness of $\{\lambda^\epsilon(n)\}$, it follows that the family $\{\bar{\lambda}(\epsilon)\}$ of (deterministic) averages lies in a compact set, thus the accumulation points $\bar{\lambda}$ exist.

From Assumption 3 and tightness of $\{\lambda^\epsilon(n)\}$, it follows that the sequences $\{(\alpha^\epsilon(n+1)-\alpha^\epsilon(n)/\epsilon)\}$ is uniformly integrable, which implies that $\{\alpha^\epsilon(n)\}$ is tight. Therefore for any sequence $(\alpha^{\epsilon_k}(\cdot), \lambda(\epsilon_k))$ there is at least one (weakly) convergent subsequence with a.s. Lipschitz continuous limit (refer to [21]). For the rest of the proof, until specified, assume that ϵ labels a weakly convergent subsequence (to avoid the ϵ_k cumbersome indexing). We will now identify the limits of such convergent subsequences and show that they all satisfy the same ODE. Also to ease the notation in the proof, call $Y_0^\epsilon(n) = \widehat{\nabla C}^\epsilon(n), Y_l^\epsilon(n) = \widehat{\nabla B_l^\epsilon}(n)$.

From the definition (30) it follows that:

$$\alpha^\epsilon(t+s) - \alpha^\epsilon(t) = -\epsilon \sum_{n=\lfloor t/\epsilon \rfloor}^{\lfloor (t+s)/\epsilon \rfloor - 1} \left[Y_0^\epsilon(n) + \sum_{l=1}^{L} \lambda_l^\epsilon(n+1)\, Y_l^\epsilon(n) \right].$$

Divide now the interval $(t, t+s]$ into subintervals of small size δ_ϵ containing each a number n_ϵ of updates, as shown in Figure 9.

Using the δ_ϵ grouping of subintervals, one obtains the telescopic sum:

$$\alpha^\epsilon(t+s) - \alpha^\epsilon(t) = -\sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor - 1} \delta_\epsilon \times \left(\frac{1}{n_\epsilon} \sum_{j=ln_\epsilon}^{(l+1)n_\epsilon - 1} \left[Y_0^\epsilon(j) + \sum_{l=1}^{L} \lambda_l^\epsilon(j+1)\, Y_l^\epsilon(j) \right] \right).$$

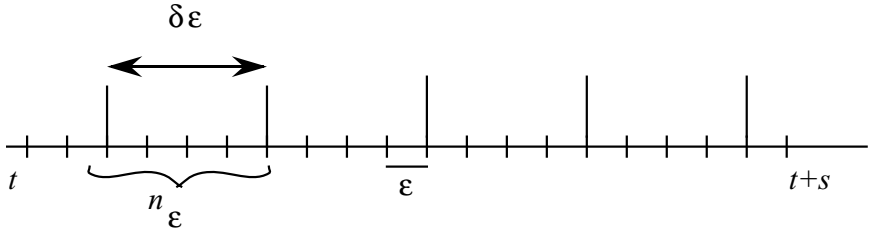

Figure 9: $\delta_\epsilon = \epsilon n_\epsilon$. Condition: $\delta_\epsilon \to 0, n_\epsilon \to \infty$ as $\epsilon \to 0$.

We now show that if $\mathfrak{F}^\epsilon(t)$ denotes the σ-algebra generated by the interpolated process up to time t, then:

$$\mathsf{E}(\alpha^\epsilon(t+s) - \alpha^\epsilon(t) \mid \mathfrak{F}^\epsilon(t)) \approx \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor - 1} \delta_\epsilon \left(h_0[\alpha^\epsilon(ln_\epsilon)] + \sum_{l=1}^{L} (\bar{\lambda}_l(\epsilon) + B_l[\alpha^\epsilon(ln_\epsilon)])\, h_l[\alpha^\epsilon(ln_\epsilon) + \kappa[\alpha^\epsilon(ln_\epsilon)] \right) \quad (73)$$

where the expectation of the absolute error in the vanishes as $\epsilon \to 0$. Let E_{ln_ϵ} denote the expectation conditioning on the information available up to the start of the current small subinterval of size δ_ϵ. Use now conditional expectations to express $\mathsf{E}(Y_l^\epsilon(j) \mid \mathfrak{F}^\epsilon(t)) = \mathsf{E}[\mathsf{E}_{ln_\epsilon}(Y_l^\epsilon(j)) \mid \mathfrak{F}^\epsilon(t)], ln_\epsilon \le j < (l+1)n_\epsilon$ for each term in the telescopic sum $l = 0, \ldots, L$. That is, we use a filter of the terms, focusing on each of the averages within subintervals.

Because one is interested in averages, any version of the process $\{\alpha^\epsilon(n)\}$ can be used to characterize these conditional expectations. In particular, Skorohod representation establishes that there is a process $\tilde{\alpha}^\epsilon(n)$ for each ϵ in the weakly convergent subsequence, such that $\tilde{\alpha}^\epsilon(n)$ has the same distribution as $\alpha^\epsilon(n)$ and it converges with probability 1 to the same a.s. continuous limit $\alpha(t)$ (see [21]). Because $\alpha(t)$ is Lipschitz continuous w.p.1, $\|\alpha(l\delta_\epsilon + \delta_\epsilon) - \alpha(l\delta_\epsilon)\| = \mathcal{O}(\delta_\epsilon)$ and since $\tilde{\alpha}^\epsilon(n)$ converges w.p. 1, it follows that

$$\sup_{ln_\epsilon \le j < (l+1)n_\epsilon} \|\tilde{\alpha}^\epsilon(j) - \alpha(l\delta_\epsilon)\| \to 0 \quad \text{a.s.}$$

which implies that the underlying distribution of the batch estimators $Y_l^\epsilon(j), j = ln_\epsilon, \ldots, (l+1)n_\epsilon - 1$ converges to that of the fixed-ψ MDP at the parameter value $\psi = \alpha(l\delta_\epsilon)$. Using the fact that $\alpha^\epsilon(ln_\epsilon)$ converges in distribution to $\alpha(l\delta_\epsilon)$, it follows that:

$$\mathsf{E}_{ln_\epsilon} \left\{ \frac{1}{n_\epsilon} \sum_{j=ln_\epsilon}^{(l+1)n_\epsilon - 1} \left[Y_0^\epsilon(j) + \sum_{l=1}^{L} \lambda_l^\epsilon(j+1)\, Y_l^\epsilon(j) \right] \right\}$$
$$= \mathsf{E}_{ln_\epsilon} \left\{ \frac{1}{n_\epsilon} \sum_{j=ln_\epsilon}^{(l+1)n_\epsilon - 1} \mathbb{E}_\psi \left[\widehat{\nabla C_0^\epsilon}(j) + \sum_{l=1}^{L} (\lambda_l^\epsilon(j) + \hat{B}^\epsilon(j))\, \widehat{\nabla B_l^\epsilon}(j) \right] \right\}.$$

The batch estimation procedure that we have suggested takes into account only new information on each estimation interval. Given the initial state value (with the aggregated information about the living phantoms), and the value of $\lambda_l^\epsilon(j)$, the expectation for the fixed ψ process of the gradient estimator $\widehat{\nabla B_l^\epsilon}(j)$ is independent of $\lambda_l^\epsilon(j)$. As $n_\epsilon \to \infty$, the underlying process $\{Z_n\}$ will have the stationary distribution for the j-th estimation batch, so that:

$$\begin{aligned}
&\mathsf{E}_{ln_\epsilon}\left\{\frac{1}{n_\epsilon}\sum_{j=ln_\epsilon}^{(l+1)n_\epsilon-1}\left[Y_0^\epsilon(j)+\sum_{l=1}^{L}\lambda_l^\epsilon(j+1)\,Y_l^\epsilon(j)\right]\right\}\\
&\quad= h_0[\alpha^\epsilon(ln_\epsilon)]+\sum_{l=1}^{L}\left(h_l[\alpha^\epsilon(ln_\epsilon)]\,\mathsf{E}_{ln_\epsilon}\left(\frac{1}{n_\epsilon}\sum_{j=ln_\epsilon}^{(l+1)n_\epsilon-1}\lambda_l^\epsilon(j)\right)+\rho\,\mathbb{E}_{\pi(\psi)}[\hat{B}_l^\epsilon(n)\,\widehat{\nabla B_l^\epsilon}(n)]\right)\\
&\quad\approx h_0[\alpha^\epsilon(ln_\epsilon)]+\sum_{l=1}^{L}(\bar\lambda(\epsilon)+\rho B[\alpha^\epsilon(ln_\epsilon)])h_l[\alpha^\epsilon(ln_\epsilon)]+\kappa[\alpha^\epsilon(ln_\epsilon)],
\end{aligned}$$

which establishes (73). Define now a piecewise constant function (on the δ_ϵ-subintervals):

$$\mathcal{G}^\epsilon(\alpha^\epsilon(t),\bar\lambda(\epsilon))=h_0[\alpha^\epsilon(ln_\epsilon)]+\sum_{l=1}^{L}(\bar\lambda_l(\epsilon)+\rho B[\alpha^\epsilon(ln_\epsilon)])h_l[\alpha^\epsilon(ln_\epsilon)]+\kappa[\alpha^\epsilon(ln_\epsilon)],$$

for $\delta_\epsilon \le t < (l+1)\delta_\epsilon$, then (73) implies that:

$$\mathsf{E}(\alpha^\epsilon(t+s)-\alpha^\epsilon(t)\mid\mathfrak{F}^\epsilon(t))\approx\int_t^{t+s}\mathcal{G}^\epsilon[\alpha^\epsilon(s),\bar\lambda(\epsilon)]\,ds \tag{74}$$

which implies that the limit process is a martingale with zero quadratic variation. For a detailed presentation of this methodology the reader is referred to [21]. Taking now the limit along the weakly convergent subsequence, $\alpha^\epsilon(t)\to\alpha(t), \bar\lambda(\epsilon)\to\bar\lambda$ establishes the limiting ODE for this subsequence.

□

References

[1] F. Vazquez Abad, V. Krishnamurthy, I. Baltcheva, and K. Martin. Self learning control of constrained Markov decision processes – a gradient approach. In *IEEE Conference on Decision and Control*, Las Vegas, 2002.

[2] F. Vazquez Abad, V. Krishnamurthy, and K. Martin. Implementation of gradient estimation to a constrained Markov decision problem. In *IEEE Conference on Decision and Control*, 2003. submitted.

[3] E. Altman. *Constrained Markov Decision Processes*. Stochastic Modelling. Chapman and Hall, London, 1999.

[4] J. Baxter and P. Bartlett. Direct gradient-based reinforcement learning: I. Gradient estimation algorithms. Technical report, Computer Sciences Laboratory, Australian National University, http://discus.anu.edu.au/ml/index.html, 1999.

[5] D. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[6] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA., 2000.

[7] D. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, 1996.

[8] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, Belmont, Massachusetts, 1995.

[9] X. R. Cao. The relations amongst potentials, perturbation analysis and Markov decision processes. *J DEDS*, 8:71–87, 1998.

[10] L. Dai. Perturbation analysis via coupling. *IEEE Trans. Auto. Control*, 41(4):614–628, 2000.

[11] P. Glasserman. *Gradient Estimation via Perturbation Analysis*. Kluwer Academic Publishers, 1991.

[12] B. Heidergott and F.J. Vázquez-Abad. Measure valued differentiation for stochastic processes: The finite horizon case. preprint.

[13] Y.-C. Ho and X.-R. Cao. *Discrete Event Dynamic Systems and Perturbation Analysis*. Kluwer Academic, Boston, 1991.

[14] C Humes Jr., P.J.S. Silva, and B.F. Svaiter. Some inexact hybrid proximal augmented lagrangian algorithms. Technical report, Instituto de Matematica e Estatisitica, http://www.ime.usp.br/ rsilva/index.en.html, 2001.

[15] M.N. Katehakis and A.F. Veinott Jr. The multi-armed bandit problem: Decomposition and computation. *Mathematics of Operations Research*, 12(2):262–268, 1987.

[16] V. Krishnamurthy and A. Logothetis. Iterative and recursive estimators for hidden Markov errors–in–variables models. *IEEE Transactions on Signal Processing*, 44(3):629–639, 1996.

[17] P.R. Kumar and P. Varaiya. *Stochastic systems – Estimation, Identification and Adaptive Control*. Prentice-Hall, New Jersey, 1986.

[18] H.J. Kushner and D.S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, 1978.

[19] H.J. Kushner and A. Shwartz. Weak convergence and asymptotic properties of adaptive filters with constant gains. *IEEE Trans. Inform. Theory*, IT-30:177–182, 1984.

[20] H.J. Kushner and J. Yang. Analysis of adaptive step-size SA algorithms for parameter tracking. *IEEE Transactions in Automatic Control*, 40(8):1403–1410, August 1995.

[21] H.J. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.

[22] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison Wesley, Second edition, 1984.

[23] G. Pflug. *Optimization of Stochastic Models: The Interface between simulation and Optimization*. Kluwer Academic Publishers, 1996.

[24] A.S. Poznyak, K. Najim, and E. Gomez-Ramirez. *Self-Learning Control of Finite Markov Chains*. Marcel Dekker, NY, 2000.

[25] M. Puterman. *Markov Decision Processes*. John Wiley, 1994.

[26] K.W. Ross. *Multiservice Loss Models for Broadband Telecommunication Networks*. Springer-Verlag, 1995.

[27] K.W. Ross and R. Varadarajan. Markov decision processes with sample path constraints: The communication case. *Operations Research*, 37(5):780–790, Sept-Oct 1989.

[28] S. Singh, V. Krishnamurthy, and H.V. Poor. Integrated voice/data call admission control for wireless DS-CDMA systems with fading. *IEEE Trans. Signal Proc.*, 50(6):1483–1495, June 2002.

[29] F. Vazquez-Abad. The case of the swapping and of the disappearing phantoms. In IEE, editor, *Proceedings of WODES98*, pages 102–107, Cagliari, August 1998.

[30] F. Vazquez-Abad. Strong points of weak convergence: a study using RPA gradient estimation for automatic learning. *Automatica*, 35(7):1255–1274, 1999.

[31] F. Vazquez-Abad and L.G. Mason. Adaptive control of deds under non-uniqueness of the optimal control. *Journal of Discrete Event Dynamic Systems, Theory and Applications*, 6(4):323–359, 1996.