An Optimal One-Way Multigrid Algorithm for Discrete-Time Stochastic Control

Chee-Seng Chow, *Member, IEEE,* and John N. Tsitsiklis, *Member, IEEE*

Abstract—**We consider the numerical solution of discrete-time stationary infinite-horizon discounted stochastic control problems, for the case where the state space is continuous and the problem is to be solved approximately, within a desired accuracy. After a discussion of problem discretization, we introduce a multigrid version of the successive approximation algorithm that proceeds "one way" from coarse to fine grids, and analyze its computational requirements as a function of the desired accuracy and of the discount factor. We also study the effects of a certain mixing (ergodicity) condition on the algorithm's performance. We show that the one-way multigrid algorithm improves upon the complexity of its single-grid variant and is, in a certain sense, optimal.**

I. Introduction and Summary

THIS paper deals with the computational aspects of continuous-state discounted-cost Markov decision problems (MDP's), as they arise in discrete-time stochastic control [4], [6], [7], [18] and is a continuation of a research effort [24], [25] aimed at understanding the computational complexity of control problems.

In a typical MDP, we are given a controlled discrete-time system that evolves in a state space $S \subset \boldsymbol{R}^n$ and we are interested in computing a fixed point J^* of the dynamic programming operator T (acting on a space of functions on the set S) defined by

$$(TJ)(x) = \inf_{u \in C} \left[g(x,u) + \alpha \int_S J(y) P(y \mid x, u)\, dy \right], \qquad \forall x \in S.$$

Here, $C \subset \boldsymbol{R}^m$ is the control space, $g(x, u)$ is the cost incurred if the current state is x, and control u is applied, $\alpha \in (0, 1)$ is a discount factor, and $P(y \mid x, u)$ is a stochastic kernel that specifies the probability distribution of the next state y, when the current state is x and control u is applied. Then, $J^*(x)$ is interpreted as the value of the expected discounted cost, starting from state x, and provided that the control actions are chosen optimally (see Section II for more details). Unfortunately, even if the problem data (the functions g and P) are given in closed form, the equation $TJ^* = J^*$ does not usually admit closed form solutions and must be solved numerically. This can be accomplished by discretizing the continuous problem to obtain an MDP with finite state and control spaces. Then the resulting discrete problem can be solved by means of several algorithms such as successive approximation (value iteration), policy iteration, or linear programming [4]. Furthermore, there are bounds available on how fine the discretization should be in order to achieve a desired accuracy (see, e.g., [18] and [31]).

The computational requirements of continuous-state MDP's are substantial and for this reason, past research has focused on the finite-state case (see, e.g., [16] and the references therein). However, the availability of more powerful computer hardware might make the solution of such problems feasible, including real-time applications (e.g., in robotics [28]).

We analyze the "worst-case" complexity of the problem. A novelty in our complexity analysis is that we simultaneously consider the dependence on the desired accuracy parameter ϵ and on the discount factor α (as $\epsilon \downarrow 0$ and $\alpha \uparrow 1$). The dependence on α is interesting for both theoretical and practical reasons. From the theoretical point of view, when α approaches 1, the problem converges in a certain sense to an "average cost" problem [4]. From the practical point of view, if one discretizes the time in a continuous-time discounted stochastic control problem, the discount factor in the resulting discrete-time problem approaches 1 as the discretization step becomes finer. It will be shown that the dependence of the complexity on α is significantly affected by the presence or absence of a certain mixing (ergodicity) condition.

The main contribution of this paper is the introduction of a multigrid variant of the successive approximation algorithm, together with a detailed analysis of its computational requirements. This algorithm proceeds by solving the problem approximately on a coarse grid and by using the coarse grid solution as a starting point for the solution on a finer grid. The algorithm, in contrast to the more general multigrid algorithms which move up and down the grids, proceeds "one-way" from coarse to fine grids, hence, the name "one-way multigrid algorithm." Thus, most of the work takes place on coarse grids with a complexity reduction

Manuscript received April 25, 1989; revised March 31, 1990 and March 26, 1991. Paper recommended by Past Associate Editor, A. Haurie. This work was supported by the National Science Foundation under Grant ECS-8552419, with matching funds from Bellcore and Du Pont, and by the ARO under Grant DAAL03-86-K-0171. A preliminary version of this paper was presented at the 27th IEEE Conference on Decision and Control.

C.-S. Chow was with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139. He is now with the IBM T. J. Watson Research Center, Yorktown Heights, MA 10598.

J. N. Tsitsiklis is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9101161.

0018-9286/91/0800-0898$01.00

resulting. In particular, the dependence of the complexity of our algorithm on ϵ is optimal. Also, the complexity dependence on α is optimal in the presence of a mixing condition, and close to optimal otherwise.

Multigrid methods have been studied extensively, primarily for the numerical solution of partial differential equations and have been found, both theoretically and experimentally, to offer substantial computational savings (see, e.g., [8], [13], and [15]). In the context of stochastic control, multigrid methods have been independently introduced in [1] and [19]. (Also, see [5] and [21] for related works.) However, our work is different in a number of important respects to be discussed in Section VII-D.

Outline of the Paper

In Section II, we introduce our notation and review some basic facts about monotone contraction operators. We state our assumptions and define the problem of interest. In Section III, we describe a discretization procedure related to the one introduced in [31], and quantify the resulting approximation error. In Section IV, we introduce an ergodicity condition that is a continuous-state formulation of a "scrambling-type" condition discussed in [14] and [27]. We show that the ergodicity condition leads to faster convergence of successive approximation methods and to better discretization error bounds. We also show that if a continuous-state problem satisfies an ergodicity condition, then this property is inherited by the discretized version of the problem. In Section V, we review some error bounds for the successive approximation algorithm, introduce our model of computation, and develop some estimates on the computational cost of a typical iteration. In Section VI, we analyze the complexity of the classical (single-grid) successive approximation algorithm. The analysis in this section is carried out twice: for general problems, as well as for problems satisfying the ergodicity condition. In Section VII, we introduce our multigrid version of the successive approximation algorithm and analyze its complexity. We also discuss the optimality of our methods. In Section VIII, we consider the computation of a policy whose cost is within ϵ of the optimal. Finally, in Section IX, we discuss several extensions and generalizations of our results and suggest certain directions for future research.

II. Markov Decision Problems

In this section, we give a precise definition of Markov decision problems (MDP's) and state our assumptions. We start by introducing some notation and with a review of some basic concepts.

A. Notation, Norms, and Operators

Let S be a Borel measurable subset of the Euclidean space $\boldsymbol{R}^n$. We use $\mathscr{B}(S)$ [respectively, $\mathscr{C}(S)$] to denote the space of all bounded Borel measurable (respectively, bounded continuous) functions on S. When comparing two functions J, $J' \in \mathscr{B}(S)$, we use the notation $J \leq J'$ which is to be interpreted as $J(x) \leq J'(x)$ for all $x \in S$. Furthermore, any scalar c may also denote the constant function on S of value c; in particular, $J + c$ denotes the function with value $J(x) + c$ at $x \in S$.

We view the Euclidean space $\boldsymbol{R}^n$ as a normed vector space by endowing it with the sup-norm $\|\cdot\|_\infty$. We will also use $\|\cdot\|_\infty$ to denote the sup-norm on $\mathscr{B}(S)$. It is well known [2], [20] that $\mathscr{B}(S)$ and $\mathscr{C}(S)$ are Banach spaces with respect to the sup-norm.

We define

$$\|J\|_S = \sup_{x \in S} J(x) - \inf_{x \in S} J(x), \qquad \forall J \in \mathscr{B}(S). \quad (2.1)$$

The function $\|\cdot\|_S$ is called the *span norm* and is actually a seminorm [14], [27]. In particular, it satisfies the triangle inequality. Moreover, the span norm also satisfies

$$\|J\|_S = 2 \min_{c \in \boldsymbol{R}} \|J + c\|_\infty, \qquad J \in \mathscr{B}(S) \quad (2.2)$$

where the minimum is attained by letting $c = -[\sup_x J(x) + \inf_x J(x)]/2$. It follows from (2.2) that $\|J\|_S \leq 2\|J\|_\infty$, for all $J \in \mathscr{B}(S)$.

An operator $A: \mathscr{B}(S) \mapsto \mathscr{B}(S)$ is called a *monotone operator* if $J \leq J'$ implies $AJ \leq AJ'$. Furthermore, if there exists some $\alpha \in (0, 1)$ such that $\|AJ - AJ'\|_\infty \leq \alpha \|J - J'\|_\infty$ for all J, $J' \in \mathscr{B}(S)$, then A is called a *contraction operator* on $\mathscr{B}(S)$, with contraction factor α. Operators that satisfy both properties are called *monotone contraction operators* [12], [31]. Finally, for any nonnegative integer k, let A^k denote the composition of k copies of A, with A^0 denoting the identity operator.

B. Specification of a Markov Decision Problem

An MDP is defined as follows. We are given a state space $S \subset \boldsymbol{R}^n$ on which a controlled stochastic process evolves, and a control space $C \subset \boldsymbol{R}^m$ from which control actions will be chosen. We assume that S and C are bounded and measurable, and, without loss of generality, we can make the further assumption that $S \subset [0, 1]^n$ and $C = [0, 1]^m$. The dynamics of the system are described by a Borel measurable function $P: S \times S \times C \mapsto [0, \infty)$. In particular, $P(y \mid x, u)$ is to be interpreted as the probability density of the next state y when the current state is x and the control u is applied.

We incorporate state-dependent constraints in our formulation. In particular, for each $x \in S$, we are given a nonempty set $U(x) \subset C$ of admissible controls. Define $\Gamma = \{(x, u) \mid x \in S \text{ and } u \in U(x)\}$. We assume that Γ is the intersection of a closed subset of $\boldsymbol{R}^n \times \boldsymbol{R}^m$ with the set $S \times C$. (That is, Γ is closed with respect to the induced topology on $S \times C$.) It follows that for any $x \in S$, $U(x)$ is compact.

If at some stage k, the state is x and control u is applied, then a cost $\alpha^k g(x, u)$ is incurred, where $g: S \times C \mapsto \boldsymbol{R}$ is a bounded measurable function, and $\alpha \in (0, 1)$ is the *discount factor*. A Markov decision problem is specified by the tuple $(S, C, \{U(x)\}_{x \in S}, P, g, \alpha)$.

C. Assumptions

We assume that there exists a constant $K \geq 1$ such that:

A.1: $|g(x, u) - g(x', u')| \leq K\|(x, u) - (x', u')\|_\infty$, for all x, $x' \in S$ and u, $u' \in C$;

A.2: $|P(y \mid x, u) - P(y' \mid x', u')| \le K\|(y, x, u) - (y', x', u')\|_\infty$, for all $x, x', y, y' \in S$ and $u, u' \in C$;

A.3: for any $x, x' \in S$ and any $u' \in U(x')$, there exists some $u \in U(x)$ such that $\|u - u'\|_\infty \le K\|x - x'\|_\infty$;

A.4: $0 \le P(y \mid x, u) \le K$ and $\int_S P(y \mid x, u)\, dy = 1$, for all $x, y \in S$ and $u \in C$.

The first two assumptions state that g and P are Lipschitz continuous. The third is the same as an assumption used in [3], and is a continuity condition on the point-to-set mapping $x \mapsto U(x)$. The last assumption reflects the fact that $P(\cdot \mid x, u)$ is a probability density. Unless otherwise stated, assumptions A.1–A.4 will always be in effect. (These assumptions are somewhat restrictive; possible generalizations are discussed in Section IX.) Under our assumptions, an MDP is a special case of the *lower semicontinuous model* studied in [6]; it is also a special case of a model in [18].

D. Policies and the Optimal Cost Function

Define

$$\Pi = \{\mu: S \mapsto C \mid \mu \text{ is Borel measurable and } \mu(x) \in U(x), \quad \forall x \in S\}. \tag{2.3}$$

Let Π^∞ be the set of all sequences $\pi = (\mu_0, \mu_1, \cdots)$ of elements of Π. Each element of Π^∞ is called a *policy* and is interpreted as a prescription for choosing control actions as a function of time and of the current state. In particular, if the state at time t is equal to some x and policy π is used, then control $\mu_t(x)$ is applied. Once a particular policy is fixed, we can construct a Markov process $\{x_t^\pi \mid t = 0, 1, \cdots\}$ by letting $P(\cdot \mid x_t^\pi, \mu_t(x_t^\pi))$ be the probability density function of x_{t+1}^π, conditioned on x_t^π.

For any policy $\pi \in \Pi^\infty$, we define its cost $J_\pi(x)$, as a function of the initial state, by letting

$$J_\pi(x) = E\left\{\sum_{t=0}^{\infty} \alpha^t g(x_t^\pi, \mu_t(x_t^\pi)) \mid x_0^\pi = x\right\}, \qquad x \in S. \tag{2.4}$$

The *optimal cost function* $J^*: S \mapsto \boldsymbol{R}$ is defined by

$$J^*(x) = \inf_{\pi \in \Pi^\infty} J_\pi(x), \qquad x \in S. \tag{2.5}$$

Accordingly, a policy π is called *optimal* if $J_\pi = J^*$. Furthermore, if $\epsilon > 0$, $\|J - J^*\|_\infty \le \epsilon$, we will call J an *ϵ-approximation* of J^*, or an *ϵ-optimal cost function*. The problem considered in this paper is precisely the computation of such a function.

E. The Dynamic Programming Operator and Bellman's Equation

We define the dynamic programming operator $T: \mathscr{B}(S) \mapsto \mathscr{B}(S)$, by letting

$$TJ(x) = \min_{u \in U(x)} \left\{g(x, u) + \alpha \int_S J(y) P(y \mid x, u)\, dy\right\}, \qquad x \in S. \tag{2.6}$$

(We have used "min" instead of "inf" in (2.6) because the minimum is always attained by some $u \in U(x)$. This follows from the continuity assumptions A.1–A.2 and the compactness of $U(x)$.)

It is well known and is easily shown that T is a monotone contraction operator (see, e.g., [12] and [31]) and satisfies

$$T(J + c) = TJ + \alpha c, \qquad \forall J \in \mathscr{B}(S), \forall c \in \boldsymbol{R}, \tag{2.7}$$

$$\|TJ - TJ'\|_S \le \alpha \|J - J'\|_S, \qquad \forall J, J' \in \mathscr{B}(S). \tag{2.8}$$

It is also easily shown (see, e.g., [6] or [18]) that, under our assumptions, T maps $\mathscr{B}(S)$ into $\mathscr{C}(S)$, in particular, T maps $\mathscr{C}(S)$ into itself. Since T is a contraction operator on the Banach space $\mathscr{C}(S)$, the *dynamic programming equation (Bellman's equation)* $J = TJ$ has a unique solution in $\mathscr{C}(S)$. Furthermore, by [6, Corollary 9.17.2], the solution is the same as J^*. Therefore, J^* is a continuous function.

F. Stationary Policies and Associated Operators

For any $\mu \in \Pi$, a policy of the form $\pi = (\mu, \mu, \cdots)$ is called a *stationary policy*. When dealing with a stationary policy, we abuse notation and use μ to denote the policy and J_μ to denote its expected cost function (instead of using π and J_π, respectively). For any $\mu \in \Pi$, we define the operator $T_\mu: \mathscr{B}(S) \mapsto \mathscr{B}(S)$, by letting

$$T_\mu J(x) = g(x, \mu(x)) + \alpha \int_S J(y) P(y \mid x, \mu(x))\, dy, \qquad x \in S. \tag{2.9}$$

Similarly with T, T_μ is a monotone contraction operator, and satisfies (2.7)–(2.8).

It follows from [6, Proposition 7.29] that T_μ maps $\mathscr{B}(S)$ into itself. Again, since T_μ is a contraction operator on the Banach space $\mathscr{B}(S)$, T_μ must have a unique fixed point in $\mathscr{B}(S)$; it is then easily shown that the unique fixed point of T_μ is J_μ. Furthermore, it follows from a measurable selection theorem [6, Proposition 7.33] that for any $J \in \mathscr{B}(S)$ there exists a $\mu \in \Pi$ such that $T_\mu J = TJ$; that is, the minimum in TJ is attained by μ.

More importantly, it is shown in [6, Corollary 9.17.2] that there exists an optimal *stationary* policy. Thus, we can restrict attention to stationary policies and from now on, the word "policy" should be interpreted as "stationary policy."

III. Discretization Procedures

The computation of an ϵ-approximation of J^* is usually accomplished by "discretizing" the original problem and by constructing a new MDP that has finite state and control spaces. However, since we will be comparing functions corresponding to different discretization levels, it is both conceptually and notationally simpler for us to consider MDP's that involve simple functions on S rather than functions on finite subsets of S. In this section, we construct such a discretization and estimate the resulting inaccuracy as a function of the grid-spacing and of the discount parameter α.

A. Discretization of the State and Control Spaces

Let $h \in (0, 1]$ be a scalar that parameterizes the coarseness of our discretizations; we call h the *grid size* or the *grid level*. We start by partitioning the unit interval $I = [0, 1]$

into a collection $\mathcal{I}_h$ of subsets. In particular, $\mathcal{I}_h$ consists of the set $[0, h]$ together with all nonempty sets of the form $(ih, (i+1)h] \cap I$, $i = 1, 2, \ldots$. We then partition the unit n-dimensional cube $[0, 1]^n$ into a collection $\mathcal{I}_h^n$ of subsets defined by

$$\mathcal{I}_h^n = \{I_1 \times \cdots \times I_n \mid I_i \in \mathcal{I}_h\}.$$

We discretize the state space by partitioning it into a finite collection of subsets. Each set in this partition is the intersection of S with an element of $\mathcal{I}_h^n$. More precisely, we let $\mathcal{S}_h$ be the set of all nonempty sets σ of the form $\sigma = S \cap \iota$, $\iota \in \mathcal{I}_h^n$, and these sets form the desired partition. We choose a representative element from each $\sigma \in \mathcal{S}_h$ and we let $\tilde{S}_h$ be the set of all representatives. For any $x \in S$, we let σ_x be the element of $\mathcal{S}_h$ to which x belongs. We also use $\check{\sigma}_x$ to denote the representative of the set σ_x.

The control space is discretized by letting $\tilde{C}_h$ be the set of all $(u_1, \cdots, u_m) \in C$ such that each u_i is an integer multiple of h. The set of admissible discretized controls is defined by

$$\tilde{U}_h(x) = \left\{\tilde{u} \in \tilde{C}_h \mid \|u - \tilde{u}\|_\infty \leq \frac{h}{2} \quad \text{for some } u \in U(\check{\sigma}_x)\right\}, \qquad x \in S. \quad (3.1)$$

For any $x \in S$, the set $U(\check{\sigma}_x)$ is nonempty, by assumption. Furthermore, using the definition of $\tilde{C}_h$, for any $u \in U(\check{\sigma}_x)$ there exists some $\tilde{u} \in \tilde{C}_h$ such that $\|u - \tilde{u}\|_\infty \leq h/2$. Thus, the set $\tilde{U}_h(x)$ is nonempty for each $x \in S$. It is also easy to see that

$$\tilde{U}_h(x) = \tilde{U}_h(x') = \tilde{U}_h(\check{\sigma}_x), \qquad \forall x \in S, \forall x' \in \sigma_x. \quad (3.2)$$

B. Discretization of the Cost and the Dynamics

We are primarily interested in the case where h is small. We can therefore assume that $h \leq 1/2K$, where $K \geq 1$ is the constant of Assumptions A.1–A.4. Given some $h \in (0, 1/2K]$, we define the functions $\tilde{g}_h: S \times \tilde{C}_h \mapsto \boldsymbol{R}$ and $\tilde{P}_h: S \times S \times \tilde{C}_h \mapsto [0, \infty)$ by letting

$$\tilde{g}_h(x, \tilde{u}) = g(\check{\sigma}_x, \tilde{u}), \quad (3.3)$$

$$\tilde{P}_h(y \mid x, \tilde{u}) = \frac{P(\check{\sigma}_y \mid \check{\sigma}_x, \tilde{u})}{\int_S P(\check{\sigma}_z \mid \check{\sigma}_x, \tilde{u})\, dz}. \quad (3.4)$$

An easy calculation [9] verifies that the denominator in (3.4) is nonzero; so, $\tilde{P}_h$ is well-defined. We note that for each $(x, \tilde{u}) \in S \times \tilde{C}_h$, the function $\tilde{P}_h(\cdot \mid x, \tilde{u})$ is a probability density on S. Furthermore, $\tilde{P}_h(y \mid x, \tilde{u})$ can be viewed as a sample of $P(\cdot \mid \cdot, \tilde{u})$ at the points $\check{\sigma}_y$, $\check{\sigma}_x$, except that the samples are suitably normalized. (This normalization is needed in our subsequent complexity analysis.)

C. The Discretized Dynamic Programming Equation

Given the partition $\mathcal{S}_h$ of the state space S, we say that a function f with domain S is a *simple function on* $\mathcal{S}_h$ if f is constant on each element of $\mathcal{S}_h$. That is, $f(x) = f(x')$ for every $\sigma \in \mathcal{S}_h$ and every $x, x' \in \sigma$.

We have so far constructed a discretized MDP $(S, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{P}_h, \alpha)$. The dynamic programming operator $\tilde{T}_h: \mathcal{B}(S) \mapsto \mathcal{B}(S)$ corresponding to this problem is defined by

$$\tilde{T}_h J(x) = \min_{\tilde{u} \in \tilde{U}_h(x)} \left\{\tilde{g}_h(x, \tilde{u}) + \alpha \int_S J(y) \tilde{P}_h(y \mid x, \tilde{u})\, dy\right\}, \qquad J \in \mathcal{B}(S). \quad (3.5)$$

Similarly with T, $\tilde{T}_h$ is also a monotone contraction operator and satisfies (2.7)–(2.8). We also define the following set of policies:

$$\tilde{\Pi}_h = \{\tilde{\mu}_h: S \mapsto \tilde{C}_h \mid \tilde{\mu}_h \text{ is a simple function on } \mathcal{S}_h \text{ and } \tilde{\mu}_h(x) \in \tilde{U}_h(x), \quad \forall x \in S\}.$$

For any fixed $\tilde{u} \in \tilde{C}_h$, the functions $\tilde{g}_h(\cdot, \tilde{u})$ and $\int_S J(y) \tilde{P}_h(y \mid \cdot, \tilde{u})\, dy$ are simple on $\mathcal{S}_h$. It follows from (3.5) that for any $J \in \mathcal{B}(S)$, $\tilde{T}_h J$ is a simple function on $\mathcal{S}_h$, and that the minimum is attained by a policy in the set $\tilde{\Pi}_h$.

Since simple functions on $\mathcal{S}_h$ form a complete normed space, the fixed point of $\tilde{T}_h$ must also be a simple function on $\mathcal{S}_h$, in particular, there exists a unique simple function on $\mathcal{S}_h$, denoted by $\tilde{J}_h^*$, that solves the discretized Bellman equation $J = \tilde{T}_h J$.

It is clear that the discretized problem $(S, \tilde{C}_h, \{\tilde{U}_h(x)\}, \tilde{g}_h, \tilde{P}_h, \alpha)$ is equivalent to an MDP whose state space is the finite set $\tilde{S}_h$. To this latter problem, we can associate an optimal cost function $\hat{J}_h^*: \tilde{S}_h \mapsto \boldsymbol{R}$ and we have the relation $\hat{J}_h^*(\check{\sigma}_x) = \tilde{J}_h^*(x)$, $\forall x \in S$. For our purposes, however, it is easier to work with the state space S, rather than $\tilde{S}_h$, because $\tilde{J}_h^*$ and J^* are defined on the same set S and can be directly compared.

D. Discretization Error Bounds

Our main discretization error estimate is the following:

Theorem 3.1: There exist constants K_1 and K_2 (depending only on the constant K of assumptions A.1–A.4) such that for all $h \in (0, 1/2K]$ and all $J \in \mathcal{B}(S)$

$$\|TJ - \tilde{T}_h J\|_\infty \leq (K_1 + \alpha K_2 \|J\|_S) h. \quad (3.6)$$

Furthermore,

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{1}{1-\alpha}(K_1 + \alpha K_2 \|J^*\|_S) h. \quad (3.7)$$

Proof: We start with the following lemma.

Lemma 3.1: There exists a constant K_P (depending only on the constant K of assumptions A.1–A.4) such that

$$|\tilde{P}_h(y \mid x, \tilde{u}) - P(y \mid x, \tilde{u})| \leq K_P h, \quad \forall (y, x, \tilde{u}) \in S \times S \times \tilde{C}_h, \qquad \forall h \in (0, 1/2K]. \quad (3.8)$$

Proof: The result follows from assumption A.2 (Lipschitz continuity of P) and a simple calculation [9]. Q.E.D.

We continue with the proof of Theorem 3.1. Fix some $J \in \mathcal{B}(S)$ and some $x \in S$. We define

$$H(u) = g(x, u) + \alpha \int_S J(y) P(y \mid x, u)\, dy, \quad u \in C, \tag{3.9}$$

$$\tilde{H}_h(\tilde{u}) = \tilde{g}_h(x, \tilde{u}) + \alpha \int_S J(y) \tilde{P}_h(y \mid x, \tilde{u})\, dy, \ \tilde{u} \in \tilde{C}_h. \tag{3.10}$$

It is clear from assumptions A.1–A.2 that $H(\cdot)$ is a continuous function. Since $U(x)$ is compact, the minimum in $\min_{u \in U(x)} H(u)$ is attained by some $u \in U(x)$.

So let $v \in U(x)$ be such that $H(v) = \min_{u \in U(x)} H(u)$. Using assumption A.3, there exists some $v' \in \tilde{U}(\breve{\sigma}_x)$ such that $\| v - v' \|_\infty \le K \| x - \breve{\sigma}_x \|_\infty \le Kh$. Finally, choose some $\tilde{v} \in \tilde{U}_h(\breve{\sigma}_x) = \tilde{U}_h(x)$ such that $\| v' - \tilde{v} \|_\infty \le h/2$. (This is possible because of the way that $\tilde{U}_h(\breve{\sigma}_x)$ is defined.) Therefore, $\| v - \tilde{v} \|_\infty \le (K + \frac{1}{2})h$. We now have

$$\begin{aligned} \tilde{T}_h J(x) - TJ(x) &= \min_{\tilde{u} \in \tilde{U}_h(x)} \tilde{H}_h(\tilde{u}) - \min_{u \in U(x)} H(u) \\ &\le | \tilde{H}_h(\tilde{v}) - H(v) | \\ &\le | \tilde{H}_h(\tilde{v}) - H(\tilde{v}) | \\ &\quad + | H(\tilde{v}) - H(v) |. \end{aligned} \tag{3.11}$$

Using the definitions (3.9)–(3.10), the Lipschitz continuity of P (assumption A.2), Lemma 3.1, and a straightforward calculation [9], we obtain $\tilde{T}_h J(x) - TJ(x) \le (K_1 + \alpha 2 K_2 \| J \|_\infty)h$, where K_1 and K_2 are suitable constants. By a symmetrical argument, we obtain the same bound for $TJ(x) - \tilde{T}_h J(x)$; thus, $| TJ(x) - \tilde{T}_h J(x) | \le (K_1 + \alpha 2 K_2 \| J \|_\infty)h$. Taking the supremum over all $x \in S$, we obtain $\| TJ - \tilde{T}_h J \|_\infty \le (K_1 + \alpha 2 K_2 \| J \|_\infty)h$.

Tto complete the proof of the first part of the theorem, since T and $\tilde{T}_h$ satisfy (2.7), we have

$$\begin{aligned} \| TJ - \tilde{T}_h J \|_\infty &= \| T(J + c) - \tilde{T}_h (J + c) \|_\infty \\ &\le (K_1 + \alpha 2 K_2 \| J + c \|_\infty)h. \end{aligned} \tag{3.12}$$

Since c is arbitrary, we can choose c to minimize the right-hand side of (3.12); using (2.2), we obtain (3.6).

Lemma 3.2: Let $T_1 : \mathcal{B}(S) \mapsto \mathcal{B}(S)$ be a contraction operator with contraction factor α and fixed point $J_1 \in \mathcal{B}(S)$. Then, for any $J \in \mathcal{B}(S)$,

$$\| J - J_1 \|_\infty \le \frac{1}{1 - \alpha} \| J - T_1 J \|_\infty.$$

Proof: See [12, Theorem 1]. Q.E.D.

We now use Lemma 3.2 and (3.6) (with $J = J^*$, $T_1 = \tilde{T}_h$, $J_1 = \tilde{J}_h^*$) to obtain

$$\begin{aligned} \| J^* - \tilde{J}_h^* \|_\infty &\le \frac{1}{1 - \alpha} \| J^* - \tilde{T}_h J^* \|_\infty \\ &\le \frac{1}{1 - \alpha} (K_1 + \alpha K_2 \| J^* \|_S) h \end{aligned}$$

which completes the proof of the theorem. Q.E.D.

Let $J^0(x) = 0$ for all $x \in S$. It follows from assumption A.1 that $\| TJ^0 \|_S \le 2K$. Since T is a contraction operator with respect to the span norm $\| \cdot \|_S$ [cf. (2.8)], we have

$$\begin{aligned} \| J^* \|_S = \| TJ^* - J^0 \|_S &\le \| TJ^* - TJ^0 \|_S \\ &\quad + \| TJ^0 - J^0 \|_S \le \alpha \| J^* \|_S + \| TJ^0 \|_S \end{aligned}$$

which implies that

$$\| J^* \|_S \le \frac{2K}{1 - \alpha}. \tag{3.13}$$

By an identical argument, we also get

$$\| \tilde{J}_h^* \|_S \le \frac{2K}{1 - \alpha}. \tag{3.14}$$

Using (3.13) in the discretization error bound of Theorem 3.1 yields the following.

Corollary 3.1: Let $K' = K_1 + 2KK_2$. Then for every $h \in (0, 1/2K]$ we have

$$\| J^* - \tilde{J}_h^* \|_\infty \le \frac{K'h}{(1 - \alpha)^2}. \tag{3.15}$$

In the next section, we show that under an ergodicity assumption $\| J^* \|_S$ can be bounded by a constant independent of α in which case the bound of (3.15) can be sharpened.

The following result will be needed later. Its proof is a simple calculation and is omitted.

Lemma 3.3: If $J \in \mathcal{B}(S)$ and $h > 0$, then $\| \tilde{T}_h J \|_S \le 2K + \alpha \| J \|_S$. Furthermore, if $\| J \|_\infty \le 2K/(1 - \alpha)$, then $\| \tilde{T}_h J \|_\infty \le 2K/(1 - \alpha)$.

E. Remarks

1) Our discretization procedure is similar to those in [31]. Despite some differences, the bounds of Theorem 3.1 are similar to those in [31, Theorem 6.1]. A main difference is that our discretized problems are defined on the same state space S (unlike [31]) and all of the operators $\tilde{T}_h$ act on the same function space $\mathcal{B}(S)$. This greatly facilitates the grid-level changes in the multigrid algorithms to be introduced later. For example, in our framework, two iterations on different grids correspond to the application of an operator of the form $\tilde{T}_h \tilde{T}_{h'}$. In contrast, in the framework in [31], a grid-level change requires the application of certain interpolation and projection operators.

2) Our discretization scheme is also similar to an approximation scheme in [18], and the discretization bounds are also similar. One difference is that we discretize the control space (whereas the scheme in [18] does not).

3) The assumptions that the control space C is equal to $[0, 1]^m$ and that the functions g and P are defined (and are Lipschitz continuous) on the entire set C, allow us to use a uniform discretization of C, independent of the constraint sets $U(x)$. This idea was used in [3], where the additional requirement $\tilde{U}_h(\tilde{x}) \subset U(\tilde{x})$ was imposed. However, such a requirement is not necessary in our framework.

IV. The Ergodicity Condition

In this section, we consider a special case where the dynamics satisfy a certain mixing condition, which we call a *k-stage ergodicity condition*. We show that this condition leads to a faster convergence in the successive approximation algorithm (to be introduced in Section V) and results in a more accurate problem discretization. We also show that an ergodicity condition in the continuous problem is inherited by the discretized problems, for sufficiently fine discretizations.

A. A k-Stage Ergodicity Condition

For any $\mu \in \Pi$, let $P_\mu(y \mid x) = P(y \mid x, \mu(x))$. Let $\mu_0, \mu_1, \cdots$ be a sequence of elements of Π. We define a function $P_{\mu_0}P_{\mu_1} \cdots P_{\mu_i}: S \times S \mapsto \boldsymbol{R}$ by means of the recursive formula

$$(P_{\mu_0}P_{\mu_1} \cdots P_{\mu_i})(x_{i+1} \mid x_0) = \int_S (P_{\mu_0} \cdots P_{\mu_{i-1}})(x_i \mid x_0) P_{\mu_i}(x_{i+1} \mid x_i)\, dx_i. \quad (4.1)$$

We can interpret $(P_{\mu_0}P_{\mu_1} \cdots P_{\mu_i})(x_{i+1} \mid x_0)$ as the probability density of the state x_{i+1} at time $i+1$, given that the initial state is x_0 and that policy $\pi = (\mu_0, \mu_1, \cdots)$ is used.

Definition 4.1: Let k be some positive integer. We say that P satisfies a *k-stage ergodicity condition with ergodicity rate* $\rho > 0$ if for all $x, x' \in S$, $\mu_0, \cdots, \mu_i, \mu'_0, \cdots, \mu'_i \in \Pi$,

$$\int_S \min\left[P_{\mu_0}P_{\mu_1} \cdots P_{\mu_i}(y \mid x), P_{\mu'_0}P_{\mu'_1} \cdots P_{\mu'_i}(y \mid x')\right] dy \geq \rho. \quad (4.2)$$

The definition of a k-stage ergodicity condition for the discretized dynamics $\tilde{P}_h$ is the same, except that we use $\tilde{P}_{\tilde{\mu}}$ instead of P_μ.

This condition is a k-stage generalization of an ergodicity condition in [18] and is a continuous-state formulation of a "scrambling-type" recurrence condition discussed in [14] and [27]. It contains a number of important conditions as special cases (see [14, Theorem 4]), and typically holds when the system being controlled is "sufficiently noisy." (Moreover, ergodicity conditions are often assumed when studying average-cost problems; see, e.g., [4], [18].) We caution, however, that for problems arising from the discretization of continuous-time problems, the value of k may be impractically large, and the value of ρ may be impractically small.

B. Ergodicity and Span Norm Contraction

We now show that a k-stage ergodicity condition leads to an additional span norm contraction factor (independent of α) in the dynamic programming operator and results in a better discretization error bounds.

Theorem 4.1: If P satisfies a k-stage ergodicity condition with ergodicity rate ρ, then

a) $\|T^kJ - T^kJ'\|_S \leq \alpha^k(1-\rho)\|J - J'\|_S$, for all J, $J' \in \mathcal{B}(S)$;

b) $\|J^*\|_S \leq 2kK/\rho$.

Similarly, if $\tilde{P}_h$ satisfies a k-stage ergodicity condition with ergodicity rate ρ, then

a') $\|\tilde{T}_h^kJ - \tilde{T}_h^kJ'\|_S \leq \alpha^k(1-\rho)\|J - J'\|_S$, for all J, $J' \in \mathcal{B}(S)$;

b') $\|\tilde{J}_h^*\|_S \leq 2kK/\rho$.

Proof: We first prove part a). Recall (from Section II-F) that for any $J \in \mathcal{B}(S)$, there exists $\mu \in \Pi$ such that $TJ = T_\mu J$. So let $\mu_0, \mu_1, \cdots, \mu_{k-1}$ (respectively, $\mu'_0, \mu'_1, \cdots, \mu'_{k-1}$) be the policies that attain the minimum in T^kJ (respectively, T^kJ'); that is

$$T^kJ = T_{\mu_0}T_{\mu_1} \cdots T_{\mu_{k-1}}J;$$
$$T^kJ' = T_{\mu'_0}T_{\mu'_1} \cdots T'_{\mu_{k-1}}J'.$$

The rest of the proof is exactly the same as the proof (for discrete-state problems) of [14, Theorem 5]. (Alternatively, see [4, pp. 318–320].)

To prove that $\|J^*\|_S \leq 2kK/\rho$, let $J^0(x) = 0$, for all $x \in S$. Using the triangle inequality

$$\|J^*\|_S \leq \|T^kJ^* - T^kJ^0\|_S + \|T^kJ^0 - J^0\|_S \leq (1-\rho)\|J^*\|_S + \|T^kJ^0\|_S$$

which implies that $\|J^*\|_S \leq \|T^kJ^0\|_S/\rho$. Finally, the result follows from

$$\|T^kJ^0\|_S \leq \sum_{i=1}^{k} \|T^iJ^0 - T^{i-1}J^0\|_S \leq \sum_{i=1}^{k} \alpha^{i-1}\|TJ^0\|_S \leq 2kK.$$

For the discretized dynamics, we use the finiteness of $\tilde{\Pi}_h$ to conclude that for any $J \in \mathcal{B}(S)$, the minimum of $\tilde{T}_hJ$ is attained by some $\tilde{\mu} \in \tilde{\Pi}_h$. The rest of the proof of parts a') and b') is identical and is omitted. Q.E.D.

Theorem 4.1 states that, under an ergodicity condition, $\tilde{T}_h$ is a k-stage contraction operator with respect to the span norm $\|\cdot\|_S$ and the contraction factor is independent of α, even if α increases to 1. Furthermore, the bound $\|J^*\|_S \leq 2kK/\rho$, together with Theorem 3.1, leads to a tighter discretization error bound.

Corollary 4.1—Under a k-Stage Ergodicity Condition): There exists some constant K'' (depending only on the constant K of assumptions A.1–A.4, k, and ρ) such that for every $h \in (0, 1/2K]$ we have

$$\|J^* - \tilde{J}_h^*\|_\infty \leq \frac{K''h}{1-\alpha}.$$

C. Ergodicity and Discretization

We now show that an ergodicity condition on the continuous problem is inherited by the discretized problems, when the discretization is sufficiently fine.

Theorem 4.2: Suppose that P satisfies a k-stage ergodicity condition with ergodicity rate 2ρ. Then there exists some $h_a > 0$ (depending only on K, k, and ρ) such that for all

$h \in (0, h_a]$, $\tilde{P}_h$ satisfies a k-stage ergodicity condition with rate ρ.

Proof: We will need the following lemma.

Lemma 4.1: For any $\tilde{\mu} \in \tilde{\Pi}_h$, there exists some $\mu \in \Pi$ such that $\| \tilde{\mu} - \mu \|_\infty \le (K+1)h$.

Proof: Fix some $\tilde{\mu} \in \tilde{\Pi}_h$ and some $x_0 \in S$. The partition that contains x_0 is σ_{x_0} and its representative is $\breve{\sigma}_{x_0}$. Let $\tilde{u}_0 = \tilde{\mu}(x_0) = \tilde{\mu}(\breve{\sigma}_{x_0})$, where the second quality holds because $\tilde{\mu}$ is constant on the set σ_{x_0}. By the definition of $\tilde{\Pi}_h$, there exists some $u_0 \in U(\breve{\sigma}_{x_0})$ such that $\| u_0 - \tilde{u}_0 \|_\infty \le h/2$.

Let $G = \{ u \in C \mid \| u - \tilde{u}_0 \|_\infty \le (K + 1/2)h \}$. By assumption A.3, $G \cap U(x)$ is nonempty, for all $x \in \sigma_{x_0}$. Thus, for every $x \in \sigma_{x_0}$, we can choose some $\mu(x) \in U(x)$ such that $\| \mu(x) - \tilde{\mu}(x) \|_\infty = \| \mu(x) - \tilde{u}_0 \|_\infty \le (K + 1/2)h$. By repeating this argument for each set in the partition of S, we obtain a function μ that satisfies the desired inequality. There is one final issue that has to be dealt with: according to the definition of Π, μ must be a measurable function. This can be accomplished by appealing to a suitable measurable selection theorem ([6, Proposition 7.33]). Q.E.D.

We now proceed to the proof of the theorem. Let $\tilde{\mu}_0, \tilde{\mu}_1, \cdots, \tilde{\mu}_{k-1}$ be a sequence of elements of $\tilde{\Pi}_h$. Let $\mu_0, \mu_1, \cdots, \mu_{k-1}$ be elements of Π such that $\| \mu_i - \tilde{\mu}_i \|_\infty \le (K+1)h$, $i = 0,1,\cdots, k-1$. (They exist by Lemma 4.1.)

Let μ and $\tilde{\mu}$ be elements of Π and $\tilde{\Pi}_h$, respectively. Using Lemma 3.1, there exists a constant K_P such that

$$| \tilde{P}_h(y \mid x, \tilde{\mu}(x)) - P(y \mid x, \tilde{\mu}(x)) | \le K_P h, \qquad \forall x, y \in S. \quad (4.3)$$

Furthermore, by assumption A.2

$$| P(y \mid x, \tilde{\mu}(x)) - P(y \mid x, \mu(x)) | \le K | \tilde{\mu}(x) - \mu(x) | \le K \| \mu - \tilde{\mu} \|_\infty, \qquad \forall x, y \in S. \quad (4.4)$$

Combining (4.3) and (4.4), we have $| \tilde{P}_{\tilde{\mu}}(y \mid x) - P_\mu(y \mid x) | \le K_P h + K \| \mu - \tilde{\mu} \|_\infty$, $\forall x, y \in S$. In particular

$$| \tilde{P}_{\tilde{\mu}_i}(y \mid x) - P_{\mu_i}(y \mid x) | \le (K_P + K^2 + K)h, \qquad \forall x, y \in S, i = 0, 1, \cdots, k-1.$$

Using this inequality and the definition of $P_{\mu_0} P_{\mu_1} \cdots P_{\mu_{k-1}}$ it follows easily that there exists a constant K_4 (depending only on K and k) such that

$$(\tilde{P}_{\tilde{\mu}_0} \cdots \tilde{P}_{\tilde{\mu}_{k-1}})(y \mid x) \ge (P_{\mu_0} \cdots P_{\mu_{k-1}})(y \mid x) - K_4 h/2 \qquad \forall x, y \in S.$$

Using a similar argument, we have for any $\tilde{\mu}'_0, \cdots, \tilde{\mu}'_{k-1} \in \tilde{\Pi}_h$, there exist $\mu'_0, \cdots, \mu'_{k-1} \in \Pi$ such that

$$(\tilde{P}_{\tilde{\mu}'_0} \cdots \tilde{P}_{\tilde{\mu}'_{k-1}})(y \mid x) \ge (P_{\mu'_0} \cdots P_{\mu'_{k-1}})(y \mid x) - K_4 h/2, \qquad \forall x, y \in S.$$

Hence, we have, as required, for any $x, x' \in S$, $\tilde{\mu}_0, \cdots, \tilde{\mu}_{k-1}, \tilde{\mu}'_0, \cdots, \tilde{\mu}'_{k-1} \in \tilde{\Pi}_h$,

$$\begin{aligned} \int_S \min \left[\tilde{P}_{\tilde{\mu}_0} \cdots \tilde{P}_{\tilde{\mu}_{k-1}}(y \mid x), \tilde{P}_{\tilde{\mu}'_0} \cdots \tilde{P}_{\tilde{\mu}'_{k-1}}(y \mid x') \right] dy \\ \ge \int_S \min \left[P_{\mu_0} \cdots P_{\mu_{k-1}}(y \mid x), P_{\mu'_0} \cdots P_{\mu_{k-1}}(y \mid x') \right] dy - \int_S K_4 h \, dy \\ \ge 2\rho - K_4 h \ge \rho \end{aligned}$$

provided that $h \le h_a$, where we define $h_a = \min(1/2K, \rho/K_4)$. Q.E.D.

Finally, we prove a result analogous to Lemma 3.3; it will be used later.

Lemma 4.2: Suppose that P satisfies a k-stage ergodicity condition with ergodicity rate 2ρ and that $J \in \mathcal{B}(S)$ and $h \in (0, h_a]$. If $\| J \|_S \le 2(k+1)K/\rho$, then

$$\| \tilde{T}_h^{kl+1} J \|_S \le 2(k+1)K/\rho, \qquad \forall l = 1, 2, \cdots, . \quad (4.5)$$

Proof: Let $J^0(x) = 0$ for all $x \in S$. First, we note from Theorems 4.1 and 4.2 that, for $h \in (0, h_a]$, $\tilde{T}_h$ is a k-stage span norm contraction operator with contraction factor $(1-\rho)$; from the proof of Theorem 4.1, that $\| \tilde{T}_h^k J^0 \|_S \le 2kK$; and from Lemma 3.3, that $\| \tilde{T}_h J \|_S \le 2K + \| J \|_S$.

We now prove (4.5) by induction on l. For $l = 1$, we have

$$\begin{aligned} \| \tilde{T}_h^{k+1} J \|_S &\le \| \tilde{T}_h^{k+1} J - \tilde{T}_h^k J^0 \|_S + \| \tilde{T}_h^k J^0 - J^0 \|_S \\ &\le (1-\rho) \| \tilde{T}_h J \|_S + \| \tilde{T}_h^k J^0 \|_S \\ &\le (1-\rho)\left[2(k+1)K/\rho + 2K \right] + 2kK \\ &\le 2(k+1)K/\rho. \end{aligned}$$

Now assuming that the result holds for l, we will prove it for $l+1$. We have

$$\begin{aligned} \| \tilde{T}_h^{(l+1)k+1} J \|_S &\le \| \tilde{T}_h^{(l+1)k+1} J - \tilde{T}_h^k J^0 \|_S + \| \tilde{T}_h^k J^0 \|_S \\ &\le (1-\rho) \| \tilde{T}_h^{lk+1} J \|_S + 2kK \\ &\le (1-\rho)\left[2(k+1)K/\rho \right] + 2kK \\ &\le 2(k+1)K/\rho \end{aligned}$$

as required Q.E.D.

V. Successive Approximation Algorithms

In this section, we introduce the successive approximation algorithm. We review some known bounds on its speed of convergence and study the effects of the ergodicity condition. We then introduce a model of computation and analyze the computational requirements of a typical iteration of the algorithm.

A. Successive Approximation Error Bounds

The successive approximation algorithm for a discretized problem proceeds as follows. We start with some function $J \in \mathcal{B}(S)$ which is simple on $\mathcal{S}_h$, and we compute $\tilde{T}_h^t J$ $(t = 1,2,\cdots)$. Since $\tilde{T}_h$ is a contraction operator (with contraction factor α) and since $\tilde{J}_h^*$ is (by definition) a fixed

point of $\tilde{T}_h$, we have

$$\| \tilde{J}_h^* - \tilde{T}_h^t J \|_\infty \leq \alpha^t \| \tilde{J}_h^* - J \|_\infty. \tag{5.1}$$

In particular, $\tilde{T}_h^t J$ converges to $\tilde{J}_h^*$. A further consequence of the contraction property of $\tilde{T}_h$ is the following well-known error bound [4], [12]:

$$\begin{aligned} \| \tilde{J}_h^* - \tilde{T}_h^t J \|_\infty &\leq \frac{\alpha}{1-\alpha} \| \tilde{T}_h^t J - \tilde{T}_h^{t-1} J \|_\infty \\ &\leq \frac{\alpha^t}{1-\alpha} \| \tilde{T}_h J - J \|_\infty. \end{aligned} \tag{5.2}$$

Since $\tilde{T}_h$ is also a monotone operator and satisfies (2.7), the convergence rate of the algorithm can be accelerated by using the following error bounds (see, e.g., [4]), that are valid for any $J \in \mathscr{B}(S)$:

$$\tilde{J}_h^* \leq \tilde{T}_h^{t+1} J + \frac{\alpha}{1-\alpha} \max_{x \in S} \{ (\tilde{T}_h^{t+1} J - \tilde{T}_h^t J)(x) \}, \tag{5.3}$$

$$\tilde{J}_h^* \geq \tilde{T}_h^{t+1} J + \frac{\alpha}{1-\alpha} \min_{x \in S} \{ (\tilde{T}_h^{t+1} J - \tilde{T}_h^t J)(x) \}. \tag{5.4}$$

(We have used "max" and "min" because $\tilde{T}_h^t J$ and $\tilde{T}_h^{t+1} J$ are simple functions.) The following is an appoximation to $\tilde{J}_h^*$ that exploits the bounds of (5.3)-(5.4):

$$\begin{aligned} J^{t+1} = \tilde{T}_h^{t+1} J + \frac{\alpha}{2(1-\alpha)} \Big[&\min_x (\tilde{T}_h^{t+1} J - \tilde{T}_h^t J)(x) \\ &+ \max_x (\tilde{T}_h^{t+1} J - \tilde{T}_h^t J)(x) \Big]. \end{aligned} \tag{5.5}$$

We subtract (5.3) or (5.4) from (5.5) to obtain

$$\begin{aligned} \| \tilde{J}_h^* - J^{t+1} \|_\infty &\leq \frac{\alpha}{2(1-\alpha)} \| \tilde{T}_h^{t+1} J - \tilde{T}_h^t J \|_S \\ &\leq \frac{\alpha^{t+1}}{2(1-\alpha)} \| \tilde{T}_h J - J \|_S. \end{aligned} \tag{5.6}$$

This bound is not much better than the bound of (5.2). However, if we assume that a k-stage ergodicity condition holds, Theorem 4.1 yields

$$\begin{aligned} \| \tilde{T}_h^{tk+1} J - \tilde{T}_h^{tk} J \|_S &\leq \alpha^{tk} (1-\rho)^t \| \tilde{T}_h J - J \|_S \\ &\leq (1-\rho)^t \| \tilde{T}_h J - J \|_S. \end{aligned} \tag{5.7}$$

Combining with (5.6), we obtain

$$\| \tilde{J}_h^* - J^{tk+1} \|_\infty \leq \frac{(1-\rho)^t}{2(1-\alpha)} \| \tilde{T}_h J - J \|_S. \tag{5.8}$$

Thus, the distance of J^t from $\tilde{J}_h^*$ contracts by a factor of at least $(1-\rho)$ every k iterations. In particular, the convergence rate has an upper bound independent of α.

Before we can analyze the computational requirements of a typical iteration of the algorithm, we must first define our model of computation, which is done in the next section.

B. Model of Computation

Given that we are dealing with problems involving continuous variables, discrete models of computation (such as Turing machines [22]) are not suitable. We shall use instead a continuous model in which arithmetic operations are performed on infinite precision real numbers (see [23] and [29] for related models).

Our model consists of three components:

a) A Mechanism for Reading the Input: The input to the computation is provided by means of an "oracle" that works as follows.

i) To obtain information about S, a computer submits to the oracle "queries" consisting of an element $\iota \in \mathscr{I}_h^n$. If $\iota \cap S$ is empty then the oracle returns a special symbol to indicate this fact; otherwise, the oracle returns an element in $\iota \cap S$ and the volume $\lambda_n(\iota \cap S)$ of that set, where $\lambda_n(\cdot)$ stands for the Lebesque measure.

ii) To obtain information about $U(x)$, a computer submits to the oracle a pair (h, x) and the oracle returns a list of the elements of the set $\tilde{U}_h(x)$.

iii) Finally, to obtain values of g and P at some specific points, the computer submits to the oracle a triple (y, x, u), and the oracle returns the values of $P(y \mid x, u)$ and $g(x, u)$.

b) The Nature of the Allowed Computations: We consider a computing machine, or simply a "computer" that has the capability of performing comparisons and elementary arithmetic operations on infinite precision real numbers. Furthermore, the computer can use the results of earlier computations to decide what queries to submit to the oracle. The rules by which the computing machine decides at each step what to do next will be referred to as an "algorithm."

c) A Format for Representing the Output of the Computation: In our case, the output of the computation is a function J_h which is simple on $\mathscr{S}_h$, where the discretization parameter h is to be decided by the computer itself. One possible format is the following. The computer first outputs the value of h, which implicitly specifies the partition $\mathscr{S}_h$ of S. It then outputs the pair $(\tilde{x}, J_h(\tilde{x}))$, for every $\tilde{x} \in \tilde{S}_h$.

There are some additional assumptions that have to be made in our particular context: the computer is provided the values of m and n (the dimensins of C and S, respectively), the discount factor α, the desired accuracy ϵ, and the constant K of assumptions A.1-A.4. Furthermore, if a k-stage ergodicity condition is assumed, the computer is also given the values of k and of the ergodicity rate.

The computational cost of an algorithm (also called its complexity) will be counted in a very simple manner: each query to the oracle costs one unit; similarly, each arithmetic operation or comparison costs one unit. (In a variation of this model, a query asking for the elements of a set $\tilde{U}_h(x)$ could have cost equal to the cardinality of the set returned by the oracle. Our complexity estimates, however, are not sensitive to minor variations of this type.)

Let us fix the dimensions m and n, the constant K of assumptions A.1-A.4, and the constants k and ρ involved in the ergodicity condition of Section IV. Once these parameters

are fixed, let $\mathscr{P}(\alpha)$ be the set of all MDP's with discount factor α and let $\mathscr{P} = \cup_{\alpha\in(0,1)} \mathscr{P}(\alpha)$. Let us consider an algorithm γ that given any $\epsilon > 0$ and any MDP in $\mathscr{P}$, returns an ϵ-optimal cost function. We use $C(\gamma;\ \alpha,\ \epsilon)$ to denote the worst case running time of this algorithm for a particular value of ϵ and where the worst case is taken over all MDP's belonging to $\mathscr{P}(\alpha)$. We then define the complexity $\mathscr{C}(\alpha,\ \epsilon)$ of solving MDP's as the minimum of $C(\gamma;\ \alpha,\ \epsilon)$ over all algorithms γ with the aforementioned properties.

There is a similar definition of the complexity of solving MDP's that satisfy a k-stage ergodicity condition with ergodicity rate ρ. The details of this definition are analogous to the one in the preceding paragraph. We use $\mathscr{C}_{\text{mix}}(\alpha,\ \epsilon)$ to denote this complexity.

It is convenient to only consider order of magnitude estimates when arguing about algorithm or problem complexity. We thus introduce the following notation:

a) Let $f,\ g{:}(0,\ 1] \mapsto [0,\ \infty)$ be functions of the grid size h. We write $f = O(g)$ if there exist constants c and $h_0 > 0$ such that $f(h) \le cg(h)$ for all $h \in (0,\ h_0]$. We also write $f = \Omega(g)$ if $g = O(f)$.

b) Let $f,\ g{:}(0,\ 1) \times (0,\ 1] \mapsto [0,\ \infty)$ be functions of α and ϵ. We write $f = O(g)$, if there exist constants c, $\epsilon_0 > 0$, and $\alpha_0 < 1$ such that $f(\epsilon,\ \alpha) \le cg(\epsilon,\ \alpha)$, for all $\epsilon \in (0,\ \epsilon_0]$ and $\alpha \in [\alpha_0,\ 1)$. We also write $f = \Omega(g)$ if $g = O(f)$.

C. The Complexity of Evaluating $\tilde{T}_h J$

We estimate here the complexity of evaluating $\tilde{T}_h J$ according to the formula

$$\tilde{T}_h J(x) = \min_{\tilde{u}\in\tilde{U}_h(x)} \left\{ \tilde{g}_h(x,\ \tilde{u}) + \alpha \int_S J(y)\tilde{P}_h(y \mid x,\ \tilde{u})\ dy \right\} \quad (5.9)$$

for the case where J is a simple function on $\mathscr{S}_h$. Since $\tilde{T}_h J$ also turns out to be a simple function on $\mathscr{S}_h$, we only need to determine the values of $\tilde{T}_h J$ for $\tilde{x} \in \tilde{S}_h$. Thus, $\tilde{T}_h J$ is determined by

$$\tilde{T}_h J(\tilde{x}) = \min_{\tilde{u}\in\tilde{U}_h(\tilde{x})} \left\{ \tilde{g}_h(\tilde{x},\ \tilde{u}) + \alpha \sum_{\tilde{y}\in\tilde{S}_h} J(\tilde{y})\tilde{P}_h(\tilde{y} \mid \tilde{x},\ \tilde{u})\lambda_n(\sigma_{\tilde{y}}) \right\}, \qquad \tilde{x} \in \tilde{S}_h \quad (5.10)$$

where λ_n stands for the n-dimensional Lebesque measure.[1]

We make the following observations. Since $|\tilde{S}_h| = O(h^{-n})$, and $|\tilde{U}_h(\tilde{x})| \le |\tilde{C}_h| = O(h^{-m})$, there are $O(h^{-(n+m)})$ different pairs $(\tilde{x},\ \tilde{u})$. Also, for any fixed $\tilde{x}$ and $\tilde{u}$, the right-hand side of (5.10) can be computed with $O(h^{-n})$ operations, with most of the work needed for the summation. Thus, the total time spent in arithmetic operations and comparisons is $O(h^{-(2n+m)})$. (The number of arithmetic operations needed in the normalization of $\tilde{P}_h$ is of the same order.) Furthermore, $O(h^{-(2n+m)})$ oracle queries are sufficient for obtaining the required values of the functions $\tilde{g}_h$, $\tilde{P}_h$, and of the elements of the sets $\tilde{U}_h(\tilde{x})$. We have therefore proved the following lemma.

Lemma 5.1: If J is a simple function on $\mathscr{S}_h$, then the complexity of computing $\tilde{T}_h J$ is $O(h^{-(2n+m)})$.

In our estimates, we have assumed that the minimization with respect to $\tilde{u}$ is carried out by exhaustive enumeration. In practice, the dependence on u may have a special structure that can be exploited to reduce the computational requirements. Nevertheless, our analysis will be carried out for the general case where no special structure is assumed.

VI. Single-Grid Successive Approximation and Its Complexity

In this section, we describe the single-grid successive approximation algorithm and analyze its complexity using the model of computation of Section V-B. We consider separately: i) the general case, where the problem is not assumed to satisfy an ergodicity condition, and ii) the special case, where the problem is assumed to satisfy a k-stage ergodicity condition with ergodicity rate 2ρ.

The basic idea in single-grid successive approximation is that we choose a grid size h_f so that $\|J^* - \tilde{J}^*_{h_f}\|_\infty$ is small. We then keep applying the operator $\tilde{T}_{h_f}$ until a sufficiently accurate approximation of $\tilde{J}^*_{h_f}$ is obtained.

A. The General Case

Let ϵ be the desired accuracy. From the discretization error bound of Corollary 3.1, we have

$$\|J^* - \tilde{J}^*_{h_f}\|_\infty \le \frac{K'}{(1-\alpha)^2} h_f. \quad (6.1)$$

Thus, if we let

$$h_f = \frac{(1-\alpha)^2\epsilon}{2K'} \quad (6.2)$$

we obtain $\|J^* - \tilde{J}^*_{h_f}\|_\infty \le \epsilon/2$. (Actually, Corollary 3.1 has the condition $h \le 1/2K$. This is of no concern because we are interested in the cases where $\epsilon \downarrow 0$ and/or $\alpha \uparrow 1$. In these cases, (6.2) shows that h_f becomes arbitrarily small.) With our choice of h_f, the complexity of evaluating $\tilde{T}_{h_f} J$, for some J that is simple on $\mathscr{S}_h$, is $O([(1-\alpha)^2\epsilon]^{-(2n+m)})$ (cf. Lemma 5.1).

Let $J^0(x) = 0$ for all $x \in S$, and apply $\tilde{T}_{h_f}$ on J^0 for t times, where t is the smallest integer satisfying

$$\frac{\alpha^t}{2(1-\alpha)} \|\tilde{T}_h J^0\|_S \le \frac{\epsilon}{2}.$$

Let J^t be as defined in (5.5). Then, (5.6) yields $\|\tilde{J}^*_{h_f} - J^t\|_\infty \le \epsilon/2$, and the triangle inequality shows that $\|J^* - J^t\|_\infty \le \|J^* - \tilde{J}^*_{h_f}\|_\infty + \|\tilde{J}^*_{h_f} - J^t\|_\infty \le \epsilon$, as desired.

[1] This formula should explain why we have assumed that the oracle can provide information on the volume of certain sets [see item a)-i) in Section V-B]. If such volume information were not directly available, then it should be somehow estimated. Although this could be an important issue in practice, its theoretical aspects are somewhat tangential to the present work.

We now bound the complexity of this algorithm. Since $\|\tilde{T}_h J^0\|_S \le 2K$, it is seen that

$$t \le \frac{\log[2K/((1-\alpha)\epsilon)]}{|\log \alpha|} + 1 = O\left(\frac{\log\frac{1}{(1-\alpha)\epsilon}}{|\log\alpha|}\right).$$

Therefore, the complexity of the algorithm is

$$O\left(\frac{\log\frac{1}{(1-\alpha)\epsilon}}{|\log\alpha|}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right). \quad (6.3)$$

B. The Special Case

We now impose an ergodicity condition, with ergodicity rate 2ρ. Corollary 4.1 yields $\|J^* - \tilde{J}^*_{h_f}\|_\infty \le K'' h_f/(1-\alpha)$. We wish to have $\|J^* - \tilde{J}^*_{h_f}\|_\infty \le \epsilon/2$ and this can be accomplished by letting $h_f = (1-\alpha)\epsilon/(2K'')$. Accordingly, the complexity of each iteration is $O(1/((1-\alpha)\epsilon)^{2n+m})$.

Let again $J^0(x) = 0$ for all $x \in S$, and apply $\tilde{T}_{h_f}$ on J^0 for $lk+1$ times. Equation (5.8) yields

$$\|\tilde{J}^*_{h_f} - J^{lk+1}\|_\infty \le \frac{(1-\rho)^l}{2(1-\alpha)}\|\tilde{T}_h J^0\| \le \frac{(1-\rho)^l}{2(1-\alpha)}2K. \quad (6.4)$$

We now bound the complexity of the algorithm. We desire to have $\|\tilde{J}^*_{h_f} - J^{lk+1}\|_\infty \le \epsilon/2$ and, from (6.4), this can be achieved with

$$l \le \frac{\log\frac{2K}{(1-\alpha)\epsilon}}{|\log(1-\rho)|} + 1 = O\left(\log\frac{1}{(1-\alpha)\epsilon}\right).$$

So, the complexity of the algorithm is

$$O\left(\log\frac{1}{(1-\alpha)\epsilon}\left[\frac{1}{(1-\alpha)\epsilon}\right]^{2n+m}\right). \quad (6.5)$$

VII. Complexity of One-Way Multigrid Successive Approximation

In this section, we introduce a "one-way" multigrid version of the algorithm of Section VI and estimate its complexity. The first iterations of this algorithm are executed with a relatively large value of h (coarse grid) and the value of h is gradually reduced (grid refinement) as the algorithm proceeds, hence the name "one-way." Because most of the iterations are executed on relatively coarse grids, the resulting complexity is smaller than that of the single-grid algorithm, by a factor of $\log(1/((1-\alpha)\epsilon))$. Furthermore, our method has certain optimality properties, to be discussed in Section VII-C.

General multigrid methods have been extensively studied in the context of partial differential equations, and have been found to lead to substantially faster convergence both theoretically and in practice [8], [13], [15]. The use of one-way multigrid algorithms has been suggested by various authors (see, e.g., [17], [26]). However, our detailed and rigorous complexity analysis of one-way multigrid algorithms, in the context of dynamic programming, seems new. Some alternative methods [1], [19] are discussed in Section VII-D.

A. The General Case

The algorithm starts by fixing an appropriate coarsest grid-level (discretization parameter) h_0. The choice of h_0 is independent of α and ϵ, but we require that $h_0 \le 1/2K$, so that the discretization error bound of Corollary 3.1 applies. We then compute the function $\tilde{J}^*_{h_0}$ exactly, and let $J^F_{h_0} = \tilde{J}^*_{h_0}$. We switch to a new grid-level by replacing h_0 by $h_0/2$, and use $J^F_{h_0}$ to initialize the computations at the new grid-level.

More generally, at any grid-level h, we do the following. We start with an initial estimate J^I_h and we compute $\tilde{T}^t_h J^I_h$, $t = 1, 2, \cdots, t(h)$, where $t(h)$ is the smallest positive integer such that

$$\|\tilde{T}^{t(h)}_h J^I_h - \tilde{T}^{t(h)-1}_h J^I_h\|_S \le \frac{2K'h}{\alpha(1-\alpha)}. \quad (7.1)$$

(The fact that such a $t(h)$ exists is evident because $\tilde{T}^t_h J^I_h$ converges.) At that point, we let

$$\tilde{J}^F_h = \tilde{T}^{t(h)}_h J^I_h + \frac{\alpha}{2(1-\alpha)}\left[\min_x (\tilde{T}^{t(h)}_h J^I_h - \tilde{T}^{t(h)-1}_h J^I_h)(x) + \max_x (\tilde{T}^{t(h)}_h J^I_h - \tilde{T}^{t(h)-1}_h J^I_h)(x)\right] \quad (7.2)$$

which is our final estimate at the current grid-level. Then, (5.6) yields

$$\|\tilde{J}^*_h - J^F_h\|_\infty \le \frac{\alpha}{2(1-\alpha)}\|\tilde{T}^{t(h)}_h J^I_h - \tilde{T}^{t(h)-1}_h J^I_h\|_S \le \frac{\alpha^{t(h)}}{2(1-\alpha)}\|\tilde{T}_h J^I_h - J^I_h\|_S. \quad (7.3)$$

If

$$\frac{K'h}{(1-\alpha)^2} \le \frac{\epsilon}{2} \quad (7.4)$$

the algorithm terminates. Otherwise, we replace h by $h/2$ and use the final function J^F_h of the current grid-level to initialize the computations at the next grid level. That is, $J^I_{h/2} = J^F_h$. It is clear that after a finite number of grid-level changes, (7.4) will be satisfied, and this shows that the algorithm eventually terminates.

We now verify the correctness of the algorithm. Let h_f be the final grid level at which the algorithm terminates. Using Corollary 3.1, we have

$$\|\tilde{J}^*_{h_f} - J^*\|_\infty \le \frac{K'h_f}{(1-\alpha)^2} \le \frac{\epsilon}{2}. \quad (7.5)$$

Furthermore, (7.1) and (7.3) yield

$$\| \tilde{J}^*_{h_f} - J^F_{h_f} \|_\infty \le \frac{\alpha}{2(1-\alpha)} \| \tilde{T}^{t(h_f)}_{h_f} J^I_{h_f} - \tilde{T}^{t(h_f)-1}_{h_f} J^I_{h_f} \|_S \le \frac{K' h_f}{(1-\alpha)^2} \le \frac{\epsilon}{2} \tag{7.6}$$

Equations (7.5)–(7.6) and the triangle inequality yield $\| J^* - J^F_{h_f} \|_\infty \le \epsilon$, as desired.

In order to develop a complexity estimate, we need to bound the number $t(h)$ of iterations at each grid level. This is done in the following two lemmas.

Lemma 7.1: for $h \in \{h_0/2, h_0/4, \cdots, h_f\}$, and every $t \in \{1, \cdots, t(h)\}$, we have $\| \tilde{T}^t_h J^I_h \|_S \le 2K/(1-\alpha)$.

Proof: The proof proceeds by induction. We have $\| J^I_{h_0/2} \|_S = \| \tilde{J}^*_{h_0} \|_S \le 2K/(1-\alpha)$, by (3.14). Assume that $\| J^I_h \|_S \le 2K/(1-\alpha)$ for some $h \in \{h_0/2, h_0/4, \cdots, h_f\}$. Then, using Lemma 3.3., $\| \tilde{T}_h J^I_h \|_S \le 2K/(1-\alpha)$ and, continuing inductively, the same bound holds for $\| \tilde{T}^t_h J^I \|_S$, $t = 1, \cdots, t(h)$. It is seen from (7.2) that $\| J^I_{h/2} \|_S = \| J^F_h \|_S = \| \tilde{T}^{t(h)}_h J^I_h \|_S \le 2K/(1-\alpha)$. Q.E.D.

Lemma 7.2: There exists a constant c, independent of α and ϵ, such that $t(h) \le c/|\log \alpha|$, for $h = h_0/2, h_0/4, \cdots, h_f$.

Proof: Fix some $h \in \{h_0/4, h_0/8, \cdots, h_f\}$ and let $\hat{J} = \tilde{T}^{t(2h)-1}_{2h} J^I_{2h}$. (Thus, $\hat{J}$ is the function available just before the last iteration at grid level $2h$.) Then, (7.1) yields

$$\| \tilde{T}_{2h} \hat{J} - \hat{J} \|_S \le \frac{4K'h}{\alpha(1-\alpha)}. \tag{7.7}$$

Using the triangle inequality, the fact that $\| \cdot \|_S \le 2 \| \cdot \|_\infty$, (7.7), and Theorem 3.1, we have

$$\begin{aligned} &\| \tilde{T}_h \tilde{T}_{2h} \hat{J} - \tilde{T}_{2h} \hat{J} \|_S \\ &\quad \le \| \tilde{T}_h \tilde{T}_{2h} \hat{J} - T \tilde{T}_{2h} \hat{J} \|_S + \| T \tilde{T}_{2h} \hat{J} - \tilde{T}_{2h} \tilde{T}_{2h} \hat{J} \|_S \\ &\qquad + \| \tilde{T}_{2h} \tilde{T}_{2h} \hat{J} - \tilde{T}_{2h} \hat{J} \|_S \\ &\quad \le 2\big(\| \tilde{T}_h \tilde{T}_{2h} \hat{J} - T \tilde{T}_{2h} \hat{J} \|_\infty \\ &\qquad + \| T \tilde{T}_{2h} \hat{J} - \tilde{T}_{2h} \tilde{T}_{2h} \hat{J} \|_\infty \big) + \alpha \| \tilde{T}_{2h} \hat{J} - \hat{J} \|_S \\ &\quad \le 2\big((K_1 + \alpha K_2 \| \tilde{T}_{2h} \hat{J} \|_S) h \\ &\qquad + (K_1 + \alpha K_2 \| \tilde{T}_{2h} \hat{J} \|_S) 2h \big) \\ &\qquad + \alpha \frac{4K'h}{\alpha(1-\alpha)}. \end{aligned} \tag{7.8}$$

By Lemma 7.1, we have $\| \tilde{T}_{2h} \hat{J} \|_\infty \le 2K/(1-\alpha)$. Using this inequality in (7.8), we obtain

$$\| \tilde{T}_h \tilde{T}_{2h} \hat{J} - \tilde{T}_{2h} \hat{J} \|_S \le 6 \left(K_1 + \alpha K_2 \frac{2K}{1-\alpha} \right) h + \frac{4K'h}{(1-\alpha)} \le \frac{10K'}{1-\alpha} h \tag{7.9}$$

where the last inequality follows from the fact $K' = K_1 + 2K_2K$ [cf. Corollary 3.1]. Note that the left-hand side of (7.9) is equal to $\| \tilde{T}_h J^I_h - J^I_h \|_S$. Using (7.9) and the fact that $\tilde{T}_h$ is a contraction operator, with contraction factor α, with respect to the span norm $\| \cdot \|_S$, we obtain

$$\| \tilde{T}^t_h J^I_h - \tilde{T}^{t-1}_h J^I_h \|_S \le \alpha^{t-1} \| \tilde{T}_h J^I_h - J^I_h \|_S \le \alpha^{t-1} \frac{10K'}{1-\alpha} h. \tag{7.10}$$

In particular, if t is chosen so that $10\alpha^t \le 2$, then the termination condition of (7.1) is satisfied. This shows that $t(h)$ is no larger than the smallest t such that $10\alpha^t \le 2$ and, therefore, $t(h) \le c/|\log \alpha|$, where $c = 5$.

The proof for the case $h = h_0/2$ is identical, provided that we define $\hat{J} = J^I_{h_0/2} = \tilde{J}^*_{h_0}$. We than have $\| \tilde{T}_{2h} \hat{J} - \hat{J} \|_S = \| \tilde{T}_{h_0} \tilde{J}^*_{h_0} - \tilde{J}^*_{h_0} \|_S = 0$ and (7.7) is trivially true. The rest of the argument holds without any changes. Q.E.D.

Note that at each grid level h we start with a function J^I_h that is simple on $\mathcal{S}_{2h}$ and, therefore, simple on $\mathcal{S}_h$. Since only simple functions are involved, Lemma 5.1 provides an estimate of the complexity of each iteration. Using also Lemma 7.2 to estimate the number of iterations at each grid level, the total complexity of the algorithm is

$$\begin{aligned} &\boldsymbol{C}(\alpha, \epsilon) \\ &\quad = O\left(\frac{1}{|\log \alpha|} \left[(1/h_f)^{2n+m} + (1/2h_f)^{2n+m} + (1/4h_f)^{2n+m} + \cdots \right] \right) \\ &\quad = O\left(\frac{1}{|\log \alpha|} \left[\frac{1}{h_f} \right]^{2n+m} \left[1 + \frac{1}{2} + \frac{1}{4} + \cdots \right] \right) \\ &\quad = O\left(\frac{1}{|\log \alpha|} \left[\frac{1}{(1-\alpha)^2 \epsilon} \right]^{2n+m} \right). \end{aligned} \tag{7.11}$$

(The last step in (7.11) uses the relation $h_f = \Omega(\epsilon(1-\alpha)^2)$ which is a consequence of the termination criterion (7.4).) Note that we have ignored the computations involved at the first grid level h_0. This is justifiable because we can compute $\tilde{J}^*_{h_0}$ with a number of operations that is independent of α and ϵ (e.g., using linear programming or policy iteration) and let $J^F_{h_0} = \tilde{J}^*_{h_0}$. In practice, we might only compute an appxoximation of $\tilde{J}^*_{h_0}$ (e.g., by using the successive approximation algorithm at grid level h_0). It is easily verified that such a modification does not change our complexity estimate.

B. The Special Case

We now assume that the problem satisfies a k-stage ergodicity condition with ergodicity rate 2ρ. The algorithm is almost the same except for the following differences. The initial grid size h_0 is chosen to satisfy $h_0 \le h_a$, where h_a is the constant of Theorem 4.2. Furthermore, the termination criterion of (7.4) is replaced by

$$\frac{K'' h}{1-\alpha} \le \frac{\epsilon}{2} \tag{7.12}$$

where K'' is the constant of Corollary 4.1.

The proof of termination is the same as in Section VII-A. Correctness of the algorithm also follows similarly, except that we have to invoke Corollary 4.1 instead of Corollary 3.1. We now bound the number of iterations at each grid level.

Lemma 7.3: For $h \in \{h_0/2, h_0/4, \cdots, h_f\}$ and every $t \in \{1, 2, \cdots, t(h)\}$, we have $\| \tilde{T}_h^{tk+1} J_h^I \|_S \leq 2(k+1)K/\rho$. In particular, $\| J_h^F \|_S \leq 2(k+1)K/\rho$.

Proof: The proof is similar to the proof of Lemma 7.1 except that we use Lemma 4.2 instead of Lemma 3.3 and the fact that $\| J_{h_0/2}^I \|_S = \| \tilde{J}_{h_0}^* \|_S \leq 2(k+1)K/\rho$.

Lemma 7.4: Under the ergodicity condition, there exists a constant c, independent of α and ϵ, such that $t(h) \leq c$, for $h = h_0/2, h_0/4, \cdots, h_f$.

Proof: The proof is identical with the proof of Lemma 7.2. The only difference is that, under the k-stage ergodicity condition, (7.10) gets replaced by

$$\| \tilde{T}_h^{kl+1} J_h^I - \tilde{T}_h^{kl} J_h^I \|_S \leq (1-\rho)^l \| \tilde{T}_h J_h^I - J_h^I \|_S.$$

As α is replaced by the absolute constant $1-\rho$, it follows that $t(h)$ is also bounded by an absolute constant independent of α. Q.E.D.

We now use Lemma 7.3 to estimate the complexity of the algorithm. We obtain

$$\begin{aligned} C_{\text{mix}}(\alpha, \epsilon) &= O\Big((1/h_f)^{2n+m} + (1/2h_f)^{2n+m} \\ &\quad + (1/4h_f)^{2n+m} + \cdots \Big) \\ &= O\left(\left[\frac{1}{h_f} \right]^{2n+m} \left[1 + \frac{1}{2} + \frac{1}{4} + \cdots \right] \right) \\ &= O\left(\left[\frac{1}{(1-\alpha)\epsilon} \right]^{2n+m} \right). \end{aligned} \tag{7.13}$$

We have used in the last step the fact $h_f = \Omega(\epsilon(1-\alpha))$ which is a consequence of (7.12).

C. Lower Bounds and the Optimality of Multigrid Successive Approximation

The following lower bounds on the complexity of the solution of MDP's have been established in [10]:

$$C(\alpha, \epsilon) = \Omega\left(\left[\frac{1}{(1-\alpha)^2 \epsilon} \right]^{2n+m} \right), \tag{7.14}$$

$$C_{\text{mix}}(\alpha, \epsilon) = \Omega\left(\left[\frac{1}{(1-\alpha)\epsilon} \right]^{2n+m} \right). \tag{7.15}$$

We stress that these lower bounds apply to any conceivable algorithm (within our model of computation), not necessarily of the successive approximation type.

By comparing these lower bounds to the complexity of our algorithm [(7.11) and (7.13)], we note the following.

1) For problems satisfying the ergodicity condition, our algorithm is optimal; that is, its complexity is within a constant factor of the lower bound.

2) Without an ergodicity condition, our algorithm is always within a factor of $O(1/|\log \alpha|) = O(1/(1-\alpha))$ of the optimal. In particular, if α is fixed and we concentrate on the dependence on ϵ, our algorithm is again optimal.

Let us also mention that the above lower bounds hold for any problem discretization. In fact, these are lower bounds on the number of oracle queries needed in order to obtain sufficient information to compute an ϵ-approximation of J^* [10]. We then notice that the number of oracle queries in the multigrid algorithm is equal to the lower bound. This implies that our discretization scheme is optimal (within a constant factor), in the sense that no discretization using a smaller number of queries could accomplish the desired goal.

Finally, it is shown in [11] that if we restrict to iterative algorithms of the successive approximation type that only use the family of contraction mappings $\{\tilde{T}_h\}$, then, under certain simplifying assumptions, non-one-way multigrid algorithms cannot improve the complexity of our one-way multigrid algorithm. This suggests that if the $O(1/(1-\alpha))$ complexity "gap" is to be closed, we have to exploit some additional structure of the problem and consider a radically different method.

D. A Comparison With Other Algorithms

We compare our multigrid algorithm with the algorithms reported in [1] and [19]. The main differences are as follows.

1) The problems solved in these references involve continuous time and lead to an elliptic partial differential equation, while we are dealing with discrete-time problems that lead to an integral equation.

2) The algorithms of [1] and [19] are based on policy iteration whereas we use successive approximation. The policy iteration algorithm involves a "policy evaluation" step which amounts to solving the linear equation $T_\mu J_\mu = J_\mu$, where μ is a certain policy. It is then suggested that the solution of this equation be carried out using a multigrid algorithm. Whereas an algorithm similar to ours might be suitable for that task, the multigrid algorithm of [1] and [19] is radically different. Ours proceeds from coarser to finer grids, whereas the algorithm in these references moves repeatedly up and down between different grids.

3) The complexity analysis in [1] is carried out only for a specific example. Furthermore, the analysis is based on a heuristic correspondence between policy iteration and Newton's method, together with an implicit assumption that Newton's method converges very fast. (There is again no complexity analysis in [19], only the proof of convergence is shown.)

There may be good reasons for choosing policy iteration over successive approximation, because it often converges faster in practical problems. On the other hand, the complexity of policy iteration algorithms is very difficult to analyze, in general; this is the reason why we have focused on successive approximation methods.

Another important reason why our method is so different from those of [1] and [19] is that it is designed to solve

fundamentally different problems: discrete time rather than continuous time. When one discretizes continuous-time problems (e.g., controlled diffusion processes), smaller time steps lead to a contraction factor that approaches 1 and to an ergodicity rate that approaches zero; in a sense, the conditioning of the problem worsens. By moving up and down the grids and using a "coarse-grid correction," multigrid algorithms can remove this ill-conditioning and result in a contraction factor bounded above by a constant smaller than 1, and independent of the grid size [13], [15]. Such multigrid algorithms depend heavily on the available special structure. Our results indicate that, in the absence of such special structure, our one-way multigrid algorithm does not leave much room for improvement. For example, suppose that we are starting with a true discrete-time discounted problem, in which α is given constant (rather than a discretization-dependent parameter); alternatively, suppose that the ergodicity rate ρ is not too close to 0. Then, the aforementioned advantage of full multigrid methods is not pertinent because the problem is well-conditioned to start with. Even if the problem is ill-conditioned, our algorithms can be improved upon only if there is some additional structure to be exploited.

VIII. Computing ϵ-Optimal Policies

In this section, we consider the computation of an *ϵ-optimal policy*, that is, a stationary policy whose expected cost is within ϵ of the optimal. The main result of this section is that the upper and lower bounds of Section VII are applicable to this problem as well; furthermore, computing an ϵ-optimal policy is "as hard as" computing an ϵ-optimal cost function (that is, the cost of compting the former is within a constant factor of the cost of computing the latter, and vice versa).

A. Definition of ϵ-Optimal Policies

Given a value of the discretization parameter h, we consider the set of $\tilde{\Pi}_h$ of all policies at grid-level h, as defined in Section III-C. These policies are easy to deal with computationally because they are simple functions on $\mathcal{S}_h$. (Note that if $\tilde{\mu} \in \tilde{\Pi}_h$, we must have $\tilde{\mu}(x) \in \tilde{U}_h(x)$ for all $x \in S$ but this does not imply that $\tilde{\mu} \in \Pi$.)

To each $\tilde{\mu} \in \tilde{\Pi}_h$ we associate the operator $T_{\tilde{\mu}}: \mathcal{B}(S) \mapsto \mathcal{B}(S)$ defined by

$$T_{\tilde{\mu}} J(x) = g\big(x, \tilde{\mu}(x)\big) + \alpha \int_S J(y) P\big(y \mid x, \tilde{\mu}(x)\big)\, dy. \tag{8.1}$$

We also associate to $\tilde{\mu}$ the operator $\tilde{T}_{\tilde{\mu}}: \mathcal{B}(S) \mapsto \mathcal{B}(S)$ defined by

$$\tilde{T}_{\tilde{\mu}} J(x) = \tilde{g}_h\big(x, \tilde{\mu}(x)\big) + \alpha \int_S J(y) \tilde{P}_h\big(y \mid x, \tilde{\mu}(x)\big)\, dy. \tag{8.2}$$

Similarly to T_μ, $T_{\tilde{\mu}}$ and $\tilde{T}_{\tilde{\mu}}$ are monotone contraction operators and satisfy (2.7)-(2.8). Let $J_{\tilde{\mu}}$ and $\tilde{J}_{\tilde{\mu}}$ be the fixed points of $T_{\tilde{\mu}}$ and $\tilde{T}_{\tilde{\mu}}$, respectively. Note that $J_{\tilde{\mu}}$ (respectively, $\tilde{J}_{\tilde{\mu}}$) can be interpreted as the expected cost functions associated with stationary policy $\tilde{\mu}$ for the original MDP (respectively, for the discretized MDP).

Definition 8.1: Let $\epsilon > 0$. A function $\tilde{\mu}: S \mapsto C$ is called an *ϵ-optimal policy* if there exists some $h > 0$ such that $\tilde{\mu} \in \tilde{\Pi}_h$, $\| J_{\tilde{\mu}} - J^* \|_\infty \le \epsilon$, and $\| \tilde{J}_{\tilde{\mu}} - J^* \|_\infty \le \epsilon$.

We now proceed to analyze the complexity of computing an ϵ-optimal policy.

B. Upper Bounds for Computing ϵ-Optimal Policies

We will show that computing an ϵ-optimal policy is "no harder than" (within a constant factor in cost of) computing an ϵ-optimal cost function; thus, the upper bounds of (7.11) and (7.13) apply to the computation of an ϵ-optimal policy as well. To show this, we use the well-known fact that the policy used in the final iteration of successive approximation algorithm is basically an ϵ-optimal policy. The proof of this result depends on the following lemma.

Lemma 8.1: Let $\hat{J} \in \mathcal{B}(S)$ and $\tilde{\mu} \in \tilde{\Pi}_h$ be a policy that satisfies $\tilde{T}_h \hat{J} = \tilde{T}_{\tilde{\mu}} \hat{J}$, then for all $h \in (0, 1/2K]$, there hold:

a) $\| \tilde{J}_h^* - \tilde{J}_{\tilde{\mu}} \|_\infty \le \dfrac{\alpha}{1-\alpha} \| \tilde{T}_h \hat{J} - \hat{J} \|_S$;

b) $\| \tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}} \|_\infty \le \dfrac{1}{1-\alpha} (K_1 + \alpha K_2 \| J_{\tilde{\mu}} \|_S) h$, where K_1 and K_2 are the constants of Theorem 3.1.

Proof:

a) The result follows easily from the triangle inequality and (5.2) and (2.2).

b) It is clear from the definition of $\tilde{T}_{\tilde{\mu}}$ and $T_{\tilde{\mu}}$ and the proof of Theorem 3.1 that

$$\| \tilde{T}_{\tilde{\mu}} J - T_{\tilde{\mu}} J \|_\infty \le \big(K_1 + \alpha K_2 \| J \|_S\big) h, \qquad \forall J \in \mathcal{B}(S).$$

Using Lemma 3.2 yields the desired result. Q.E.D.

We now apply Lemma 8.1 to the general case. Suppose that we compute an ϵ-optimal cost function using the multigrid successive approximation algorithm of Section VII-A. Let $\hat{J} = \tilde{T}_{h_f}^{t(h_f)-1} J_{h_f}^I$, so that $\tilde{T}_{h_f} \hat{J}$ corresponds to the last successive approximation iteration [cf. (7.6)]. Let $\tilde{\mu}$ be a policy that attains the minimum in $\tilde{T}_{h_f} \hat{J}$. Then by Lemma 8.1a)

$$\| \tilde{J}_{h_f}^* - \tilde{J}_{\tilde{\mu}} \|_\infty \le \frac{\alpha}{1-\alpha} \| \tilde{T}_{h_f} \hat{J} - \hat{J} \|_S \le \epsilon \tag{8.3}$$

where the last inequality follows from (7.6). Furthermore, since $\| J_{\tilde{\mu}} \|_S \le 2K/(1-\alpha)$ [cf. (3.13)] we see from Lemma 8.1b), Corollary 3.1, and (7.5) that

$$\| \tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}} \|_\infty \le \frac{1}{1-\alpha} \big(K_1 + \alpha K_2 \| J_{\tilde{\mu}} \|_S\big) h_f \le \frac{K'}{(1-\alpha)^2} h_f \le \frac{\epsilon}{2}. \tag{8.4}$$

Lastly, the choice of h_f [cf. (7.5)] ensures that the discretization error

$$\| J^* - \tilde{J}_{h_f}^* \|_\infty \le \frac{\epsilon}{2}. \tag{8.5}$$

Using the triangle inequality and (8.3)–(8.5), we conclude that

$$\| J^* - \tilde{J}_{\tilde{\mu}} \|_\infty \leq \| J^* - \tilde{J}^*_{h_f} \|_\infty + \| \tilde{J}^*_{h_f} - \tilde{J}_{\tilde{\mu}} \|_\infty \leq \frac{3}{2}\epsilon, \tag{8.6}$$

$$\| J^* - J_{\tilde{\mu}} \|_\infty \leq \| J^* - \tilde{J}_{\tilde{\mu}} \|_\infty + \| \tilde{J}_{\tilde{\mu}} - J_{\tilde{\mu}} \|_\infty \leq 2\epsilon. \tag{8.7}$$

Thus, (8.6)–(8.7) show that $\tilde{\mu}$ is a 2ϵ-optimal policy. We note that a similar reasoning yields the bounds of (8.6)–(8.7) for the special case, where the ergodicity condition is assumed.

We conclude that the work needed to compute an ϵ-optimal policy is no greater than that of computing an $\epsilon/2$-optimal cost function, and the upper bounds of (7.11) and (7.13) apply to the computation of an ϵ-optimal policy.

Let us now consider the problem of computing an ϵ-optimal *admissible* policy, that is, a policy $\mu \in \Pi$ such that $\| J_\mu - J^* \|_\infty \leq \epsilon$. This can be done, in principle, by first computing an ϵ-optimal policy (for some smaller ϵ) and approximating it by an element of Π, due to the following lemma.

Lemma 8.2: Let $J \in \mathcal{B}(S)$, $\mu \in \Pi$, $\tilde{\mu} \in \tilde{\Pi}_h$, and let K be the constant of Assumptions A.1–A.4. Then

$$\| T_\mu J - T_{\tilde{\mu}} J \|_\infty \leq (K + \alpha K \| J \|_S) \| \mu - \tilde{\mu} \|_\infty.$$

Furthermore,

$$\| J_\mu - J_{\tilde{\mu}} \|_\infty \leq \frac{1}{1-\alpha}(K + \alpha K \| J_{\tilde{\mu}} \|_S) \| \mu - \tilde{\mu} \|_\infty.$$

Proof: The first part of the lemma follows from assumptions A1 and A.2; the second part follows from Lemma 3.2. Q.E.D.

For the general case, the computation of an ϵ-optimal admissible policy μ proceeds as follows (A similar argument applies to the special case.) We first choose a discretization parameter h which is small enough so that the discretization error $K'h/(1-\alpha)^2$ is no greater than $\epsilon/8$. We use the multigrid successive approximation algorithm to compute an $\epsilon/4$-optimal cost function and, according to our earlier discussion, we obtain as a by-product an $\epsilon/2$-optimal policy $\tilde{\mu} \in \tilde{\Pi}_h$; that is, $\| J^* - J_{\tilde{\mu}} \|_\infty \leq \epsilon/2$.

We note from Lemma 4.1 that there exists some $\mu \in \Pi$ such that $\| \mu - \tilde{\mu} \|_\infty \leq (K+1)h$; so, by Lemma 8.2, $\| J_\mu - J_{\tilde{\mu}} \|_\infty \leq \frac{1}{1-\alpha}(K + \alpha K \| J_{\tilde{\mu}} \|_S)(K+1)h$. It can be seen from the proof of Theorem 3.1 that $K(K+1)$ is less than K_1 and K_2. Proceeding as in (8.4), we obtain $\| J_\mu - J_{\tilde{\mu}} \|_\infty \leq \epsilon/8$. So, by the triangle inequality

$$\| J^* - J_\mu \|_\infty \leq \| J^* - J_{\tilde{\mu}} \|_\infty + \| J_{\tilde{\mu}} - J_\mu \|_\infty \leq \frac{\epsilon}{2} + \frac{\epsilon}{8} = \frac{5}{8}\epsilon.$$

Thus, J_μ is indeed an ϵ-optimal admissible policy, as desired.

If the method in the preceding paragraph is to be used, we must be able, given any $\tilde{\mu} \in \tilde{\Pi}_h$, to compute an admissible $\mu \in \Pi$ such that $\| \tilde{\mu} - \mu \|_\infty \leq (K+1)h$. In general, this is impossible under our model of computation; in fact, it is even impossible, in general, to represent an element of Π using a finite data structure. On the other hand, for problems that arise in practice, the sets $U(x)$ often have a simple structure and this task is feasible. In those cases, the computation of an ϵ-optimal admissible policy is no harder (within a constant factor) than the computation of an ϵ-optimal cost function.

C. Lower Bounds for Computing ϵ-Optimal Policies

We observe that an ϵ-optimal policy, by definition, determines the optimal cost function J^* to within ϵ, so, the lower bounds of Section VII-C [(7.14)–(7.15)] apply to the computation of an ϵ-optimal policy as well. (See [10] for more details.) It remains to argue that computing an ϵ-optimal policy is "no easier than" computing an ϵ-optimal cost function (that is, the cost of computing the latter is within a constant factor of the cost of computing the former).

For the special case where an ergodicity condition is imposed, the upper bound for computing an ϵ-optimal cost function is within a constant factor of the lower bound [cf. (7.13) and (7.15)]. We conclude that computing an ϵ-optimal policy is no easier than computing an ϵ-optimal cost function. Thus, we have shown for problems satisfying an ergodicity condition that computing an ϵ-optimal policy is "as hard as" computing an ϵ-optimal cost function.

We now consider the general case. For α fixed and concentrating on the dependence on ϵ, the upper bound for computing an ϵ-optimal cost function is within a constant factor of the lower bound [cf. (7.11) and (7.14)]. Arguing as in the preceding pargraph, we conclude that, with respect to the dependence on ϵ, computing an ϵ-optimal policy is as hard as computing an ϵ-optimal cost function. But because of the "gap" of $O(1/(1-\alpha))$ between the upper and lower bounds, we cannot draw the same conclusion for the dependence on α. A different argument, with some additional assumptions, is needed.

We will give only the main idea of the argument (see [9] for details). We show that if an $\epsilon/2$-optimal policy is available, then an ϵ-optimal cost function can be quickly computed (with complexity better than the lower bound). Thus, an algorithm can first compute an $\epsilon/2$-optimal policy, then use the policy to compute an ϵ-optimal cost function with total computational cost within some constant factor of the cost of computing the policy. It follows that computing an ϵ-optimal policy is no easier than computing an ϵ-optimal cost function. However, this argument requires the following assumptions: i) the dimension of the control space $m \geq 1$, and ii) the ϵ-optimal policy to be computed belongs to $\tilde{\Pi}_h$ with $h = \Omega((1-\alpha)^2\epsilon)$, namely, the policy is not "unnecessarily" complicated.

IX. Extensions

We discuss here certain extensions of our results. We will only present the main ideas and the reader is referred to [9] for more details.

A. Piecewise Lipschitz Continuous Dynamics

Assumption A.2 requires $P(y \mid x, u)$ to be Lipschitz continuous. This assumption is unnecessarily restrictive, and rules out many interesting examples. In fact, our results remain valid if $|g(x, u)| \leq K$ for all $x \in S$, $u \in C$, and Assumption A.2 is replaced by the following.

Assumption B.2: There exists a constant $K \leq 1$ such that:

i) $\int_S |P(y \mid x, u) - P(y \mid x', u')|\, dy \geq K \|(x, u) - (x', u')\|_\infty$, for all $x, x' \in S$ and $u, u' \in C$;

ii) for every $x \in S$ and $u \in C$, $P(y \mid x, u)$ is a "piecewise Lipschitz continuous" function of y.

By P being "piecewise Lipschitz continuous," we mean that we can partition the state space S into a finite collection of disjoint subsets U_i such that $P(\cdot \mid x, u)$ is Lipschitz continuous, with Lipschitz constant K, on each set U_i. Furthermore, to rule out pathological cases, we require that the sets U_i have "piecewise smooth" boundary. (An even more general formulation, in terms of measures and stochastic kernels, can be found in [9]; we have chosen not to present the most general formulation in this paper to simplify the presentation and emphasize the complexity results.)

With assumption A.2 replaced by assumption B.2, it can be shown that the discretizations of Section III again satisfy (cf. Lemma 3.1) $\int_S |P(y \mid x, u) - \tilde{P}_h(y \mid x, u)|\, dy \leq K_P h$, $\forall h \in (0, h_0]$, and this property is the key to the discretization error bounds of Theorem 3.1. Furthermore, any ergodicity condition in the continuous problem is again inherited by the discretized problem (cf. Theorem 4.2). As a consequence, all subsequent results, as well as the complexity analysis, remain valid.

B. The Case Where P Is Not a Probability Measure

Suppose that $|g(x, u)| \leq K$ for all $x, \in S$, $u \in C$. We can relax Assumption A.4 by assuming instead that there exists some constant $K \geq 1$ such that for all $x, y \in S$ and $u \in C$ we have

a) $\int_S P(y \mid x, u)\, dy \leq 1$,

b) $P(y \mid x, u) \in [0, K]$.

Such an assumption can be used to model those MDP's in which the system has some nonzero probability of entering a zero-cost absorbing state.

In an even more general class of problems, we can assume that

a') $\int_S |P(y \mid x, u)|\, dy \leq 1$,

b') $|P(y \mid x, u)| \in [0, K]$.

A convenient discretization rule for such problems is to define

$$\tilde{P}_h(y \mid x, \tilde{u}) = \begin{cases} P(\breve{\sigma}_y \mid \breve{\sigma}_x, \tilde{u}), \\ \quad \text{if } \int_S |P(\breve{\sigma}_z \mid \breve{\sigma}_x, \tilde{u})|\, dz \leq 1; \\ P(\breve{\sigma}_y \mid \breve{\sigma}_x, u) \Big/ \int_S |P(\breve{\sigma}_z \mid \breve{\sigma}_x, \tilde{u})|\, dz, \\ \quad \text{otherwise.} \end{cases}$$

Since P is allowed to be negative, it is clear that T and $\tilde{T}_h$ are now no longer monotone operators. However, they are still contraction operators, with contraction factor α, and the proof of Theorem 3.1 (discretization error bounds) remains valid provided that $\|\cdot\|_S$ is replaced by $2\|\cdot\|_\infty$. Using the algorithm of Section VII (with some minor modifications) it can be shown that the complexity of the multigrid successive approximation algorithm is

$$O\left(\frac{1}{1-\alpha}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right)$$

exactly as in the case of MDP's satisfying assumptions A.1–A.4. Unlike the case where P corresponds to a probability measure, we cannot improve this complexity estimate by imposing an ergodicity condition on P. In fact, the lower bound for the case where P is a nonnegative subprobability measure and satisfies an ergodicity condition is shown in [10] to be

$$\Omega\left(\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n+m}\right). \tag{9.1}$$

This may seem counterintuitive, given the fact that the case of a nonnegative subprobability measure can be always reduced to the case of a probability measure, by introducing an additional absorbing state to which all of the "missing" probability is channeled. The catch is that the ergodicity condition is destroyed in the course of this state augmentation.

C. Fredholm Equations of the Second Kind

A Fredholm equation of the second kind is an equation of the form

$$g(x) + \int_S G(y, x) J(y)\, dy = J(x)$$

where S is a bounded subset of $\boldsymbol{R}^n$, g and G are given functions, and J is the unknown.

The numerical solution of this equation has been well studied (see, e.g., [15], [26], [30]). Let us assume that G is a bounded function and that $\int_S |G(y, x)|\, dy \leq \alpha$ for all $x \in S$, where $\alpha \in (0, 1)$. If we let $P(y \mid x) = G(y, x)/\alpha$, it is clear that we are dealing with the problem discussed in Section IX-B, except that the control variable u is absent. (Thus, $m = 0$.) It follows that (under Lipschitz continuity assumptions) our multigrid algorithm can be used to compute an ϵ-approximation of the solution and has complexity

$$O\left(\frac{1}{1-\alpha}\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n}\right). \tag{9.2}$$

Furthermore, the lower bound of (9.1) becomes

$$\Omega\left(\left[\frac{1}{(1-\alpha)^2\epsilon}\right]^{2n}\right) \tag{9.3}$$

and therefore our algorithm is optimal as far as the dependence on ϵ is concerned.

Multigrid algorithms for Fredholm's equation can also be

found in [15] and [26], and they are different in the following respects. First, the algorithms in these references are more general because they do not require a contraction assumption. Furthermore, these algorithms perform computations on fine grids and then use certain coarse-grid corrections. This is in contrast to our method that only proceeds from coarse to fine grids. According to our results, for the problems we are considering, our method has optimal dependence on the accuracy parameter ϵ and close to optimal dependence on α. (Note that α can be viewed as a measure of ill-conditioning of the problem.) It is unclear whether the algorithms in [15] and [26] have any similar optimality properties (since they do not consider the dependence of the algorithm on the problem's ill-conditioning). To the best of our knowledge closing the $O(1/(1-\alpha))$ gap between the upper bound (9.2) and the lower bound (9.3) in solving Fredholm's equation of the second kind is still open.

D. Different Norms

Let us consider the L_p-norm on $\mathcal{B}(S)$ defined by

$$\| J \|_p = \left[\int_S | J(y) |^p \, dy \right]^{1/p}, \qquad p \in [1, \infty).$$

Since the volume of S is bounded by 1, it is easily shown that $\| J \|_p \leq \| J \|_\infty$ for any $J \in \mathcal{B}(S)$ and any $p \in [1, \infty)$. For this reason, the function J returned by our algorithms automatically satisfies $\| J - J^* \|_p \leq \epsilon$.

It also turns out [10] that the lower bounds on the computational complexity of the problem do not change when L_p-norms are used to measure the error $J - J^*$. It follows that such a different choice of norms does not affect the optimality properties of our algorithms.

E. Average Cost Problems

Our results can be extended to the case of average cost Markov decision problems [9]. In particular, under an ergodicity condition optimal algorithms can be obtained. On the other hand, without an ergodicity condition, average cost problems are, in general, ill-posed and have infinite computational complexity. It is an interesting research problem to find conditions that are weaker than ergodicity and that guarantee well-posedness.

F. Another Formulation of Discrete-Time Stochastic Control Problems

In an alternative formulation of discrete-time stochastic control, we are given a dynamical equation of the form $x_{t+1} = f(x_t, u_t, w_t)$, where t denotes the time index, x_t denotes the state, u_t the control, and w_t denotes a noise term with known probability density $Q(w_t \mid x_t, u_t)$. Even though such problems can be reformulated into our framework, the resulting density $P(\cdot \mid x_t, u_t)$ is, in general, not Lipschitz continuous. In particular, our results do not apply. An important special case in which our results are inapplicable is the case of deterministic systems where P corresponds to a singular measure, as opposed to a density. The problem of characterizing the best possible discretization error and the design of optimal (or close to optimal) algorithms for such problems is open to the best of our knowledge.

G. Some Practical Issues

Although our algorithm has excellent theoretical properties, a lot of systematic experimentation is needed to determine the classes of problems for which it may be practical. Furthermore, in a practical implementation, several modifications are worth investigating.

a) Different discretization or approximation procedures can be tried in an effort to exploit any additional smoothness in the problem data. One possiblity is to use piecewise linear or higher order, instead of the piecewise constant, interpolation. (For example, see [21] and the references therein.) Nonuniform grids might also prove useful.

b) Many practical problems involve unbounded state spaces, and ways must be found to handle such problems.

c) Whereas our algorithm uses *a priori* bounds to decide when to change grid level, one might be able to use information generated by the algorithm and improve performance. In particular, one might estimate the degree of smoothness of J^*, while the algorithm is running.

d) Finally, the implementation of the "oracle calls" could present several challenges. This is true especially for the oracle calls that provide volume estimates and that generate the sets $\tilde{U}_h(x)$ of admissible controls for the discretized problem.

e) In practice, the running time of successive approximation can be improved by using Gauss–Seidel iterations, and by doing a Jacobi iteration only when successive approximation error bounds are needed.

X. Conclusions

We have studied the computational requirements of continuous-state Markov decision problems and have obtained some fairly definite conclusions, by presenting algorithms with certain optimality properites. There are several problems that remain to be addressed, having to do with alternative formulations (Section IX-F), continuous-time formulations, algorithmic implementation issues (Section IX-G). We see our work as a contribution to the understanding of the computational issues associated with control theory. Such issues are important because they will ultimately determine the practicality of different facets of control theory.

References

[1] M. Akian, J. P. Quadrat, and J. P. Chancelier, "Dynamic programming complexity and application," in *Proc. 27th IEEE Conf. Decision Contr.*, Austin, TX, Dec. 1988, pp. 1551–1558.

[2] R. B. Ash, *Measure, Integration, and Functional Analysis.* New York: Academic, 1972

[3] D. P. Bertsekas, "Convergence of discretization procedures in dynamic programming," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 415–419, 1975.

[4] ——, *Dynamic Programming: Deterministic and Stochastic Models.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[5] D. P. Bertsekas and D. Castanon, "Adaptive aggregation methods for discounted dynamic programming," in *Proc. 25th IEEE Conf. Decision Contr.*, Athens, Greece, Dec. 1986, pp. 1840–1845.

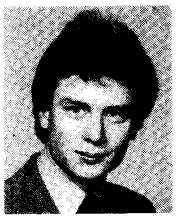
[6] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete Time Case.* New York: Academic, 1978.

[7] D. Blackwell, "Discounted dynamic programming," *Ann. Math. Statist.*, vol. 36, pp. 226–235, 1965.

[8] A. Brandt, "Multi-level approaches to large scale problems," in *Proc. Int. Congress Mathematicians*, Berkeley, CA, Aug. 1986, pp. 1319-1334.
[9] C.-S. Chow, "Multigrid algorithms and complexity results for discrete-time stochastic control and related fixed-point problems," Ph.D. dissertation, Lab. Informat. Decision Syst., Dep. Electrical Eng. Comp. Sci., M.I.T., Cambridge, MA, Tech. Rep. LIDS-TH-1934, Dec. 1989.
[10] C.-S. Chow and J. N. Tsitsiklis, "The complexity of dynamic programming," *J. Complexity*, vol. 5, pp 466-488, 1989.
[11] ——, "One-way multigrid algorithms and their optimality for a class of fixed-point problems," Lab. Informat. Decision Syst., M.I.T., Cambridge, MA, Tech. Rep. LIDS-P-1960, Mar. 1990.
[12] E. V. Denardo, "Contraction mappings in the theory underlying dynamic programming," *SIAM J. Review*, vol. 9, pp. 165-177, 1967.
[13] C. C. Douglas, "Multi-grid algorithms with applications to elliptic boundary-value problems," *SIAM Numerical Anal.*, vol. 21, pp. 236-254.
[14] A. Federgruen, P. J. Schweitzer, and H. C. Tijms, "Contraction mappings underlying undiscounted Markov decision problems," *J. Math. Anal. Appl.*, vol. 65, pp. 711-730, 1978.
[15] W. Hackbusch, *Multi-Grid Methods and Applications*. New York: Springer-Verlag, 1985.
[16] R. Hartley, L. C. Thomas, and D. J. White, Eds., *Recent Developments in Markov Decision Processes*. New York: Academic, 1980.
[17] A. Haurie and P. L'Ecuyer, "Approximation and bounds in discrete event dynamic programming," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 227-235, 1986.
[18] O. Hernández-Lerma, *Adaptive Markov Control Process*. New York: Springer-Verlag, 1989.
[19] R. H. W. Hoppe, "Multi-grid methods for Hamilton-Jacobi-Bellman equations," *Numerische Mathematik*, vol. 49, pp. 239-254, 1986.
[20] A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*. New York: Dover, 1970.
[21] P. L'Ecuyer, "Computing approximate solutions to Markov renewal programs with continuous state spaces," Dép. d'Informatique, Uni. Laval, Canada, Rep. DIUL-RR 8912.
[22] H. R. Lewis and C. H. Papadimitriou, *Elements of The Theory of Computation*. Englewood Cliffs, NJ: Prentice-Hall, 1981.
[23] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. New York: Wiley, 1983.
[24] C. H. Papadimitriou and J. N. Tsitsiklis, "Intractable problems in control theory," *SIAM J. Contr. Optimiz.*, vol. 24, pp. 639-654, 1986.
[25] ——, "The complexity of Markov decision processes," *Math. Operations Res.*, vol. 12, pp. 441-450, 1987.
[26] H. Schippers, "Multigrid techniques for the solution of Fredholm integral equations of the second kind," *Proceedings of a Colloquium on Numerical Treatment of Integral Equations*, T. Riele, Ed. Amsterdam, The Netherlands: Mathematisch Centrum, 1979.
[27] P. J. Schweitzer, "Contraction mappings underlying undiscounted Markov decision problems—II," *J. Math. Anal. Appl.*, vol. 132, pp. 154-170, 1988.
[28] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotics manipulators," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 491-500, 1986.
[29] J. F. Traub, G. W. Wasilkowski, and H. Wozniakowski, *Information-Based Complexity*. New York: Academic, 1988.
[30] A. G. Werschulz, "What is the complexity of the Fredholm problem of the second kind?," *J. Integral Equations*, vol. 9, pp. 213-241, 1985.
[31] W. Whitt, "Approximations of dynamic programs—I," *Math. Operat. Research*, vol. 3, pp. 231-243, 1978; and "Approximations of dynamic programs—II," *Math. Operat. Research*, vol. 4, pp. 179-185, 1979.

Chee-Seng Chow (S'89-M'89-S'89-M'90) was born in Malaysia, on February 22, 1961. He received the B.Sc. degrees in electrical engineering, computer science, mathematics, and physics, and the M.Sc. degree in electrical engineering and computer science in 1985, the Electrical Engineer degree in 1987, and the Ph.D. degree in electrical engineering and computer science in 1990, all from the Massachusetts Institute of Technology, Cambridge, MA.

He was a Research Student at Xerox Palo Alto Research Center, CA, during the Summers of 1982 till 1984, and the rest of 1984. Since 1989 he has been a Research Staff Member at IBM T.J. Watson Research Center, NY. His present research interests are in areas that combine computation, control, and communications.



John N. Tsitsiklis (S'80-S'81-M'81-M'83) was born in Thessaloniki, Greece, in 1958. He received the B.S. degree in mathematics, the B.S., M.S., and Ph.D. degrees in electrical engineering, all from the Massachusetts Institute of Technology, Cambridge, MA, in 1980, 1980, 1981, and 1984, respectively.

During the academic year 1983-1984 he was an Acting Professor of Electrical Engineering at Stanford University, Stanford, CA. Since 1984, he has been with the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology, where he is currently Associate Professor. His research interests are in the areas of parallel and distributed computation, systems and control theory, and operations reserch.

Dr. Tsitsiklis is the coauthor, with D. Bertsekas, of *Parallel and Distributed Computation: Numerical Methods* (1989). He has been a recipient of an IBM Faculty Development Award (1983), an NSF Presidential Young Investigator Award (1986), an Outstanding Paper Award by the IEEE Control Systems Society (for a paper coauthored with M. Athans, 1986), and of the Edgerton Faculty Achievement Award by M.I.T. (1989). He is an Associate Editor of *Applied Mathematics Letters* and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.