Using Randomization to Break the Curse of Dimensionality

John Rust
Yale University
jrust@gemini.econ.yale.edu
Final revision, November, 1996

Abstract: This paper introduces random versions of successive approximations and multigrid algorithms for computing approximate solutions to a class of finite and infinite horizon Markovian decision problems (MDPs). We prove that these algorithms succeed in breaking the "curse of dimensionality" for a subclass of MDPs known as *discrete decision processes* (DDPs).[1]

Keywords: dynamic programming, curse of dimensionality, Bellman operator, random Bellman operator, computational complexity, maximal inequalities, empirical processes

---

[1] This paper benefited from helpful comments from co-editor Peter Robinson and two referees, and discussions with H. Amman, D. Andrews, W. Brock, G. Chamberlain, W. Davis Dechert, V. Hajivassiliou, K. Judd, J. Kuelbs, T. Kurtz, A. Pakes, D. Pollard, J.F. Traub, C. Sims, J. Tsitsiklis, and H. Woźniakowski. I am especially grateful to J. Tsitsiklis for pointing out the need for an error bound that holds for all N, to D. Pollard for showing how maximal inequalities for empirical processes can be used to derive such a bound, and to J. Traub and H. Woźniakowski for their comments on the relationship between these results and the theory of information-based complexity.

1. Introduction

This paper introduces random versions of successive approximations and multigrid algorithms for computing approximate solutions to Markovian decision problems (MDPs). An MDP is a mathematical model of a decision maker who is in state s_t at time $t = 1, \ldots, T$ ($T \leq \infty$) and takes an action a_t that determines current utility $u(s_t, a_t)$ and affects the distribution of next period's state s_{t+1} via a Markov transition probability $p(s_{t+1}|s_t, a_t)$. The problem is to determine an *optimal decision rule* α that solves $V(s) \equiv max_\alpha E_\alpha \left\{ \sum_{t=0}^{T} \beta^t u(s_t, a_t) | s_0 = s \right\}$ where E_α denotes expectation with respect to the controlled stochastic process $\{s_t, a_t\}$ induced by the decision rule $\alpha \equiv \{\alpha_1, \ldots, \alpha_T\}$, and $\beta \in (0, 1)$ denotes the discount factor. The method of *dynamic programming* (a term coined by Richard Bellman in his (1957) text) provides a constructive, recursive procedure for computing α using the *value function* V as a "shadow price" to decentralize a complicated stochastic/multiperiod optimization problem into a sequence of simpler deterministic/static optimization problems. [2] As is well known, (Blackwell, (1965), Denardo, (1967)) a stationary, infinite horizon MDP can be viewed as multidimensional generalization of a geometric series whose solution is mathematically equivalent to the solution to a particular functional equation known as *Bellman's equation.*

Unfortunately, it is quite rare that one can explicitly solve Bellman's equation and derive analytical or "closed-form" solutions for either the optimal decision rule α or the value function V so most DP problems must be solved numerically on digital computers. From the standpoint of computation, there is an important distinction between *discrete MDPs* where the state and control variables can assume only a finite number of possible values and *continuous MDPs* where the state or control variables can assume a continuum of possible values. Discrete MDP problems can be solved exactly (modulo rounding error in arithmetic operations), whereas the solutions to continuous MDP problems can only be approximated to within some arbitrarily small solution tolerance ϵ. There is a well developed literature on solution algorithms for discrete MDPs. Although a large number of algorithms have been proposed, simple backward induction and variations of the method of successive approximations and policy iteration (Newton's method) are the most commonly used solutions methods. [3] The most commonly used numerical methods for solving continuous MDPs is to solve a "nearby" discrete MDP problem. The key restriction is that the "nearby" problem lives in a finite-dimensional space so it can be solved on a digital computer. [4]

---

[2] In finite horizon problems V actually denotes an entire sequence of value functions, $V \equiv \{V_0, \ldots, V_T\}$, and α denotes the corresponding sequence of decision rules. In the stationary infinite-horizon case, $T = \infty$, and the solution (V, α) reduces to a pair of functions of the current state s.

[3] See Puterman, (1990), (1994) for a survey, and Puterman and Brumelle (1979) for a proof that the Howard/Bellman policy iteration algorithm is equivalent to the Newton/Kantorovich method.

[4] Discretization is not the only way to do this, however. See Rust (1996) and Judd (1996) for a survey of "parametric approximation" and "projection" methods that include the Bellman *et. al.* (1963) approach of approximating V by polynomials or Daniel's (1976) suggestion of using spline functions to approximate V.

A minimum requirement of any sensible approximation procedure is that it be *consistent:* i.e. an approximate solution V_N ought to converge to the true solution V as $N \to \infty$, where N is some parameter indexing the accuracy of the solution. There is a large theoretical literature analyzing the convergence of various *deterministic* discretization and parametric approximation algorithms for the approximate solution of various operator equations (see, Anselone (1971), Anselone and Ansorge (1981), Krasnosel'skii *et. al.* (1972)), as well as a more specialized literature on the approximate solution of Bellman's equation for MDP problems (see, e.g. Fox, (1973), Bertsekas, (1975), Santos and Vigo, (1996)). However except for a recent numerical study by Keane and Wolpin (1994) there is relatively little theoretical literature analyzing the convergence of *random* algorithms for solving the Bellman equation in either finite or infinite horizon MDPs.[5]

There is an important practical limitation to one's ability to solve continuous MDPs arbitrarily accurately, Bellman's *curse of dimensionality.*[6] This is the well-known exponential rise in the time and space required to compute an approximate solution to an MDP problem as the dimension (i.e. the number of state and control variables) increases. Although one typically thinks of the curse of dimensionality as arising from the discretization of continuous MDPs, it also occurs in discrete MDPs that have many state and control variables. An important unresolved question is whether we can somehow circumvent the curse of dimensionality through a clever choice of solution algorithm, perhaps for a restricted class of problems exhibiting special structure.

Yudin and Nemirovsky (1976a), (1977) were the first to prove the negative result that the static nonlinear optimization problem (a special case of the MDP problem when $\beta = 0$ and the state space S contains a single element) is subject to an inherent curse of dimensionality irregardless of whether deterministic or random algorithms are used. However Yudin and Nemirovsky (1976b) showed that it is possible to break the curse of dimensionality for certain subclasses of problems such as convex optimization problems. They showed that the number of function evaluations required to approximate a solution to a d-dimensional *convex* optimization problem only increases linearly in d on a worst case basis. A number of important developments in theoretical computer science in the last twenty years has enabled formal proofs of lower bounds on the *computational complexity* of solving various continuous multivariate mathematical problems including nonlinear optimization, numerical integration, function approximation, and recently, MDP problems. There are two main branches of complexity theory, corresponding to discrete and continuous problems.

---

[5] There is a recent literature analyzing the convergence of stochastic reinforcement learning algorithms such as "real time dynamic programming" (Barto, Bradtke and Singh, (1995)) and "Q-learning" (Tsitsiklis, (1994)). The latter paper shows that Q-learning is a type of stochastic approximation algorithm. Some of these methods are actually being used in applications, see e.g., Pakes and McGuire (1996). Hammersley and Handscombe (1992) describe a number of applications of monte carlo methods including solution of large linear systems. Although such methods would seem to have a direct application to solving the large linear systems arising from policy iteration methods, to our knowledge nobody has ever actually used or advocated this approach for solving infinite horizon MDPs.

[6] It is not clear whether Bellman originated this term, although the earliest references to this phrase that we are aware of appears in the preface to Bellman's (1957) text, and also in Bellman and Dreyfus (1962), p. 322: "Let us now discuss on of the most promising techniques for overcoming the 'curse of dimensionality', the approximation of functions by polynomials."

Discrete computational complexity applies to finite problems that can be solved exactly such as the traveling salesman problem. The *size* of a discrete problem is indexed by an integer d and the (worst case) complexity, $comp(d)$, denotes the minimal number of computer operations necessary to solve the hardest possible problem of size d, (or ∞ if there is no algorithm capable of solving the problem). Continuous computational complexity theory applies to continuous problems such as multivariate integration, function approximation, nonlinear programming, and continuous MDP problems. None of these problems can be solved exactly, but in each case the true solution can be approximated to within some arbitrarily small solution tolerance ϵ. Problem size is indexed by an integer d denoting the dimension of the space that the continuous variable variable lives in (typically R^d), and the complexity, $comp(\epsilon, d)$, is defined as the minimal computational cost of solving the hardest possible d-dimensional problem with a maximum error of ϵ.

Complexity theory provides a simple way of formalizing what we mean by the curse of dimensionality: we say that a discrete MDP problem is subject to a curse of dimensionality if $comp(d) = \Omega(2^d)$, i.e. if the lower bound on computational complexity grows exponentially fast as the dimension d increases. Similarly a continuous MDP problem is subject to the curse of dimensionality if $comp(\epsilon, d) = \Omega(1/\epsilon^d)$.[7] In the computer science literature a problem which is subject to the curse of dimensionality is said to be *intractable.*[8] On the other hand, if we can show that the computational complexity is bounded above by a polynomial function of d and ϵ then the MDP problem is not subject to a curse of dimensionality. Computer scientists refer to polynomial-time problems as *tractable.*[9]

The discrete and continuous complexity bounds $comp(d)$ and $comp(\epsilon, d)$ depend on the model of computation used (parallel vs. serial, real vs. Turing), the type of algorithm used (deterministic vs. stochastic), the relevant metric for measuring the error ϵ in the approximate solution (worst case vs. average case complexity), and the class of problems being considered (general MDPs versus linear-quadratic problems and other restricted subclasses). We do not have space here to present a review of the general theory of continuous (or information-based) computational complexity and refer the interested reader to the excellent monograph by Traub, Wasilkowski and Woźniakowski, (1988). The main results of this theory and the previous literature on the complexity of the MDP problem that are relevant to our analysis in this paper can be summarized as follows. Discrete MDPs with $|S|$ states and $|A|$ actions can

---

[7] The symbols O, Ω denote upper and lower asymptotic bounds, respectively. Thus, $f(\epsilon) = O(g(\epsilon))$ if $\overline{\lim}_{\epsilon \to 0} |f(\epsilon)/g(\epsilon)| < \infty$. We say $f(\epsilon) = \Omega(g(\epsilon))$ if $g(\epsilon) = O(f(\epsilon))$.

[8] We use the terminology "curse of dimensionality" due to its historic association with dynamic programming noted above. Those who are unfamiliar with the technical terminology of computer science may also confuse the term "intractable" with "unsolvable", which is an entirely different concept. Computer scientists have a specific terminology for unsolvable problems (i.e. problems for which there is no algorithm that is capable of solving any particular instance of the problem in a finite amount of time): these problems have infinite complexity, and are classified as *non-computable.* However even though intractable problems are computable problems in the computer science terminology, as the problem grows large the lower bound on the solution time grows so quickly that large problems are not computable in any practical sense.

[9] Here again it is important to note the difference between the common meaning of the term "tractable" and the computer science definition. Even so-called "tractable" polynomial-time problems can quickly become computationally infeasible if complexity satisfies $comp(d) = \Omega(d^{\,b})$ for some large exponent b. However it seems to be a fortunate act of nature that the maximum exponent b for most common polynomial time problems is fairly small; typically $b \in [1, 4]$.

be solved exactly in polynomial time using a variety of algorithms.[10] Certain subclasses of continuous MDPs such as linear quadratic MDPs can also be solved in polynomial time using algebraic methods to solve the matrix Ricatti equation[11] The upper bound on the complexity of solving a discrete finite horizon MDP problem with $|S|$ states and $|A|$ decisions is $cT|A||S|^2$ where c is the time cost per arithmetic operation. This upper bound turns out to be the key to understanding the complexity bounds for continuous MDPs established by Chow and Tsitsiklis (1989), (1991) who proved that a "one way multigrid" algorithm is an approximately optimal algorithm in the sense that its complexity is within a $O\left(1/|\log(\beta)|\right)$ factor of the lower bound on worst case complexity given by

$$comp(\epsilon, d_s, d_a, \beta) = \Theta\left(\frac{T}{\left((1-\beta)^2\epsilon\right)^{2d_s+d_a}}\right), \tag{1.1}$$

where the Θ symbol in equation (1.1) denotes the fact that expression in parentheses is an asymptotic upper and lower bound on complexity.[12] This bound can be understood as follows: in order to guarantee that the final step of the backward induction process yields an approximate value function $\hat{V}$ that is within ϵ of the true value function V, we need to insure that the maximum error incurred in each step of the backward induction process is no greater than $(1-\beta)^2\epsilon$. In order to guarantee that any deterministic discretization procedure yields this accuracy requires a minimum of $|A| = \Omega\Big(1/((1-\beta)^2\epsilon)^{d_a}\Big)$ discretized decisions and $|S| = \Omega\Big(1/((1-\beta)^2\epsilon)^{d_s}\Big)$ discretized states. Since backward induction on the resulting discrete MDP problem requires $O(T|A||S|^2)$ operations, it follows that complexity bound for solution of continuous MDPs is given by the expression in equation (1.1).[13]

This paper considers the question of whether one can break the curse of dimensionality by using random instead of deterministic algorithms. Randomization is known to break the curse of dimensionality of certain mathematical problems. The most prominent example is multivariate integration of a function f defined on the d-dimensional unit cube $[0,1]^d$ that is $r > 0$ times differentiable. The worst-case deterministic complexity of this problem is $comp^{\textit{wor-det}}(\epsilon, d) = \Theta(1/\epsilon^m)$ where $m = d/r$, so the integration problem is subject to a curse of dimensionality if we only allow deterministic algorithms. However consider monte carlo integration of f using random uniform draws from $[0,1]^d$. It is easy to show that the worst case *randomized complexity* of the multivariate integration problem is $comp^{\textit{wor-ran}}(\epsilon, d) = O(1/\epsilon^2)$, so randomization succeeds in breaking the curse of dimensionality of the multivariate

[10] This result assumes that we index the size of the MDP by $(|S|, |A|)$. There is a curse of dimensionality if we index the size of the MDP problem by (d_s, d_a) where d_s is the number of state variables and d_a is the number of control variables, since in that case the total size of the MDP problem is indexed by $(|S|^{d_s}, |A|^{d_a})$ which increases exponentially fast in d_s and d_a.

[11] See Anderson, Hansen, McGratten and Sargent, (1996).

[12] Formally, we say that $f(\epsilon) = \Theta(g(\epsilon))$ if $f(\epsilon) = O(g(\epsilon))$ and $f(\epsilon) = \Omega(g(\epsilon))$. This implies that there exist positive constants c_1 and c_2 such that $c_1|g(\epsilon)| \leq |f(\epsilon)| \leq c_2|g(\epsilon)|$ for sufficiently small ϵ.

[13] Note that in infinite horizon problems $T = \Theta(\log(1/(1-\beta)\epsilon)/\log(\beta))$ contraction steps are sufficient to find an ϵ-approximation to a fixed point of the Bellman operator. Remarkably, in the Chow-Tsitsiklis complexity bound T represents a bounding constant that does not tend to infinity as $\beta \to 1$ or $\epsilon \to 0$.

integration problem. However randomization does not always succeed in breaking the curse of dimensionality: Yudin and Nemirovsky (1977) showed that randomization doesn't help in solving general multivariate nonlinear programming problems, Traub, Wasilkowski and Woźniakowski (1988) showed that randomization doesn't help in multivariate function approximation and interpolation, and Werschulz (1991) showed that randomization doesn't help in solving multivariate elliptic partial differential equations or Fredholm integral equations of the second kind. Indeed, the fact that the general nonlinear optimization problem is a special case of the general MDP problem implies that randomization cannot break the curse of dimensionality for general MDP problems with an action space A that contains a continuum of possible choices.

We prove that randomization does succeed in breaking the curse of dimensionality for a particular subclass of MDPs known as *Discrete Decision Processes* (DDP's). These are MDPs with a continuous multi-dimensional state space S, but a finite action set A. DDP's arise frequently in economic applications such as optimal replacement of durable goods, optimal retirement behavior, optimal search, and many other situations (for a review of empirical applications of DDP's see Rust, (1994)). The fact that DDP's have finite action sets implies that the main work involved in carrying out the backward induction process is the numerical integration of the value function at each given point in the state space. Since randomization succeeds in breaking the curse of dimensionality of numerical integration, it seems plausible that it might also be able to break the curse of dimensionality for this class of problems. However rather than calculating a single multivariate integral, the DDP problem requires calculation of an infinite number of multivariate integrals at each possible conditioning state $s \in S$. The DDP problem is also nonlinear in the sense that the current value function V equals the maximum of the conditional expectation of the future value function.[14] While randomization can be shown to break the curse of dimensionality in certain classes of linear problems such as integration or solution of ODE's, it generally is not able to break the curse of dimensionality in nonlinear problems. So it is perhaps not immediately obvious that randomization really can succeed in breaking the curse of dimensionality of DDP problems.

A recent study by Keane and Wolpin (1994) used monte carlo integration to find approximate solutions to large scale DDP problems that would be computationally intractable using standard deterministic algorithms. Their computational results are quite encouraging, suggesting that random algorithms might have considerable promise in a wide range of applications. However apart from computer simulations, Keane and Wolpin did not provide any theoretical analysis of the convergence properties of their algorithm. We do not provide any computer simulations of the random algorithms we propose, although we do provide a fairly complete mathematical characterization of the convergence properties of our algorithms. Although the particular algorithms we analyze here are quite different from the one Keane and Wolpin proposed, our hope is that the tools we introduce will be useful for analyzing a wider class

[14] In particular, if one uses unbiased monte carlo integration to compute the conditional expectation of the future value function, Jensen's inequality implies that the resulting estimate of the current value function will be biased upward.

of random algorithms including the Keane-Wolpin algorithm. In particular, one can show (see Rust (1996)) that the Keane-Wolpin algorithm does not succeed in breaking the curse of dimensionality for DDP problems. The reason is that their algorithm involves a multivariate function approximation subproblem but as we noted above multivariate function approximation is subject to a curse of dimensionality regardless of whether deterministic or random algorithms are used. We are able to avoid the curse of dimensionality inherent in multivariate function approximation since the random algorithm we propose is *self-approximating* in a sense that will be made precise in section 3.

Section 2 provides a brief review of MDPs, and presents the main inequalities and convergence bounds that will be used in subsequent sections. The random algorithms analyzed in this paper are all based on the *random Bellman operator* $\tilde{\Gamma}_N$ defined by:

$$\tilde{\Gamma}_N(V)(s) = \max_{a \in A(s)} \left[u(s,a) + \frac{\beta}{N} \sum_{k=1}^{N} V(\tilde{s}_k) p(\tilde{s}_k | s, a) \right], \tag{1.2}$$

where B is the space of continuous functions on the k-dimensional unit cube, $B = C([0,1]^d)$, and N is the number of *IID* uniformly distributed sample points $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$ from $[0,1]^d$ at which the monte carlo integral (i.e. the sample average) is evaluated. The convergence and complexity properties of the random successive approximations and multigrid algorithms depends on the error in using the random Bellman operator $\tilde{\Gamma}_N$ to approximate the true Bellman operation Γ. We appeal to a maximal inequality for empirical processes due to Pollard (1989) to show that:

$$\sup_{p \in BL(K_p)} \sup_{\|V\| \leq K_v} E\left\{ \|\tilde{\Gamma}_N(V) - \Gamma(V)\| \right\} \leq \frac{\gamma(d) K_p K_v}{\sqrt{N}}, \tag{1.3}$$

where $\gamma(d) < \infty$ is a bounding constant that satisfies $\gamma(d) = O(d)$. Section 4 introduces random versions of successive approximations and multigrid algorithms for solving finite and infinite horizon DDP's. Using the results of section 3 we are able to derive the following upper bound on the computational complexity of the DDP problem:

$$comp^{\textit{wor-ran}}(\epsilon, d) = O\left(\frac{d^4}{(1-\beta)^8 \epsilon^4} \right). \tag{1.4}$$

Since the Chow-Tsitsiklis complexity bound implies that the deterministic worst case complexity of the DDP problem is given by

$$comp^{\textit{wor-det}}(\epsilon, d) = \Theta\left(\frac{1}{[(1-\beta)^2 \epsilon]^{2d}} \right), \tag{1.5}$$

it follows that randomization succeeds in breaking the "curse of dimensionality" for this class of problems.[15] Section 5 presents concluding remarks and some conjectures and suggestions for further research in this area.

---

[15] Note that the complexity bounds in (1.4) and (1.5) should be interpreted as holding for ϵ sufficiently small and β sufficiently large. For example, if β times the expectation of the value function was uniformly less than ϵ, then an ϵ-approximation to the value function could be obtained without any numerical integration by just solving a static maximization problem for each s, which involves only $|A|$ operations, where $|A|$ is an upper bound on the number of actions in each $A(s)$. The proof of the complexity bound in Chow and Tsitsiklis (1989) requires $\beta \geq 1/2$. Our bounds require the condition that $\beta K_u > (1-\beta)\epsilon$, where K_u is a Lipschitz bound on u to be introduced in section 3. I am grateful to H. Woźniakowski for pointing this out.

2. Bellman Operators

This section reviews some basic facts about MDPs and defines the *Bellman operator* Γ. We also define the subclass of DDP's and provide some key inequalities involving the Bellman operator that will be used in our subsequent analysis.

Definition 2.1: *A* ***Markovian Decision Process*** *consists of the following objects:*

- A *time index* $t \in \{0, 1, 2, \ldots, T\},\ T \leq \infty$
- A *state space* S,
- An *action space* A,
- A family of *constraint sets* $s \to A(s) \subseteq A$,
- A *utility function* $u(s, a)$,
- A Markov *transition density* $p(s'|s, a)$,
- A *discount factor* $\beta \in [0, 1)$.

We will impose explicit topological structure on S and A and smoothness conditions on u and p in section 3 so we refrain from adding any measure-theoretic qualifications at this point. The agent's optimization problem is to find an optimal decision rule $\alpha^* = \{\alpha_0, \ldots, \alpha_T\}$ given by:

$$\begin{aligned} \alpha^* &= \underset{(\alpha_0,\ldots,\alpha_T)}{argmax}\ E_\alpha\{U_T(\mathbf{s}, \mathbf{d})\} \\ &\equiv \int_{s_0} \cdots \int_{s_T} \left[\sum_{t=0}^{T} \beta^t u(s_t, \alpha_t(s_t))\right] \prod_{t=1}^{T} p(ds_t|s_{t-1}, \alpha_{t-1}(s_{t-1}) p_0(ds_0), \end{aligned} \tag{2.1}$$

where p_0 is a probability distribution over the initial state s_0.[16]

In finite-horizon problems ($T < \infty$), dynamic programming amounts to calculating the optimal decision rule $\alpha^* = (\alpha_0, \ldots, \alpha_T)$ by backward induction starting at the terminal period, T. The backward recursion must be done for each time period $t = T, T-1, \ldots, 0$ and for each possible state s_t using the following recursions. In the terminal period V_T and α_T are defined by:

$$\alpha_T(s_T) = \underset{a_T \in A(s_T)}{argmax}\ u\big(s_T, a_T\big), \tag{2.2}$$

[16] We have assumed that u and p do not depend on calendar time for notational simplicity. It should be obvious that all our results on finite horizon dynamic programming go through if u and p depend on time, provided each u_t and p_t satisfy the Lipschitz conditions in section 3.

$$V_T(s_T) = \max_{a_T \in A(s_T)} u\big(s_T, a_T\big). \tag{2.3}$$

In periods $t = 0, \ldots, T-1$, V_t and α_t are recursively defined by:

$$\alpha_t(s_t) = \underset{a_t \in A(s_t)}{argmax} \left[u(s_t, a_t) + \beta \int V_{t+1}(s_{t+1}) p(ds_{t+1} \big| s_t, a_t) \right], \tag{2.4}$$

$$V_t(s_t) = \max_{a_t \in A(s_t)} \left[u(s_t, a_t) + \beta \int V_{t+1}(s_{t+1}) p(ds_{t+1} \big| s_t, a_t) \right]. \tag{2.5}$$

Definition 2.2: *The **Bellman operator** $\Gamma : B(S) \to B(S)$ is a mapping on the Banach space $B(S)$ of measurable functions of $s \in S$ (under the supremum norm) defined by:*

$$\Gamma\big(W\big)(s) \equiv \max_{a \in A(s)} \left[u(s,a) + \beta \int W(s') p(ds' \big| s, a) \right], \tag{2.6}$$

Using the definition of the Bellman operator, we can write the recursion (2.5) more compactly as:

$$V_{T-t} = \Gamma^t(V_T), \quad t = 0, \ldots, T. \tag{2.7}$$

In the infinite horizon case $T = \infty$ so there is no "last" period from which to start the backward induction to carry out the dynamic programming algorithm described above. However if the per period utility functions u are uniformly bounded and the discount factor β is in the $[0, 1)$ interval, then we can approximate the solution to the infinite horizon problem arbitrarily closely by the method of *successive approximations*. This is equivalent to using the solution to a long, but finite horizon MDP problem to approximate the solution to the infinite horizon problem. Removing time subscripts from equation (2.4), we obtain the following equation for the optimal stationary decision rule α:

$$\alpha(s) = \underset{a \in A(s)}{argmax} \left[u(s,a) + \beta \int V(s') p(ds' \big| s, a) \right], \tag{2.8}$$

where V is the solution to *Bellman's equation:*

$$V(s) = \max_{a \in A(s)} \left[u(s,a) + \beta \int V(s') p(ds' \big| s, a) \right]. \tag{2.9}$$

Bellman's equation can be re-written more compactly as a fixed point condition on the Bellman operator:

$$V = \Gamma\big(V\big), \tag{2.10}$$

There is a standard approach to establishing the existence and uniqueness of a solution to Bellman's equation due to Denardo 1967, that recognizes that Γ is a contraction mapping on a Banach space B. This immediately implies the existence and uniqueness of the fixed point V to the Bellman operator. It also follows that the the method of successive approximations is globally convergent from any initial starting value. The contraction property of the Bellman operator holds under the following regularity conditions:

1. S and A are compact metric spaces,

2. $s \to A(s)$ is a continuous correspondence,

3. $u(s,a)$ is jointly continuous in (s,a),

4. $\beta \in [0,1)$.

We impose these regularity conditions in the subsequent analysis, so hereafter B will denote the Banach space $C(S)$ of all continuous, bounded functions $f\colon S \to R$ under the supremum norm, $\|f\| = \sup_{s\in S} |f(s)|$. We now state a few key inequalities that will be useful in the subsequent analysis. The first inequality provides bounds on the difference between the fixed point V to a contraction mapping Γ and the fixed point V_N to a slightly perturbed contraction mapping Γ_N. We say a contraction mapping Γ has *modulus* β if $\|\Gamma(V) - \Gamma(W)\| \leq \beta\|V - W\|$ for all $V, W \in B$.

Lemma 2.1: *Suppose $\{\Gamma_N\}$ are a family of contraction mappings on a Banach space B with common modulus β that that converge pointwise to a contraction mapping Γ with modulus β: i.e. $\forall W \in B$ we have*

$$\lim_{N\to\infty} \Gamma_N(W) = \Gamma(W). \tag{2.11}$$

Then $V_N = \Gamma_N(V_N) \to V$ where V is the fixed point of Γ and $\|V_N - V\|$ satisfies the error bound:

$$\|V_N - V\| \leq \frac{\|\Gamma_N(V) - \Gamma(V)\|}{(1-\beta)}. \tag{2.12}$$

The proof of this lemma is a simple application of the triangle inequality:

$$\begin{aligned} \|V_N - V\| &= \|\Gamma_N(V_N) - \Gamma(V)\| \\ &\leq \|\Gamma_N(V_N) - \Gamma_N(V)\| + \|\Gamma_N(V) - \Gamma(V)\| \\ &\leq \beta\|V_N - V\| + \|\Gamma_N(V) - \Gamma(V)\|. \end{aligned} \tag{2.13}$$

The next lemma provides some additional inequalities bounding the rate of convergence of the method of successive approximations.

Lemma 2.2: *Let Γ be a contraction mapping on a Banach space B with fixed point $V = \Gamma(V)$. If W is an arbitrary element of B the following inequalities hold:*

$$\|W - \Gamma(W)\| \leq (1+\beta)\|V - W\| \tag{2.14}$$

$$\|\Gamma^t(W) - V\| \leq \beta^t\|\Gamma(W) - W\|/(1-\beta) \tag{2.15}$$

Error bound (2.14) shows that if W is close to the fixed point V then W must also be close to $\Gamma(W)$. Inequality (2.15) is an *a priori* error bound that shows the converse result: the maximum error in a sequence of successive

approximations $\{\Gamma^t(W)\}$ starting from W is a geometrically declining function of the initial error $\|V - \Gamma(W)\|$. These two inequalities will be used to establish the convergence of the various parametric approximation methods presented in section 3. The next lemma is a useful result that implies that the fixed point to Bellman's equation always lies within a maximum distance $K/(1-\beta)$ of the origin.

Lemma 2.3: *Let Γ be a Bellman operator. Then we have:*

$$\Gamma : B\left(0, \frac{K}{(1-\beta)}\right) \rightarrow B\left(0, \frac{K}{(1-\beta)}\right), \tag{2.16}$$

where $B(0,r) = \{V \in B \Big| \ \|V\| \leq r\}$ and the constant K is given by:

$$K \equiv \sup_{s \in S} \sup_{a \in A(s)} |u(s,a)|. \tag{2.17}$$

The final lemma of this section extends Lemma 2.1 to provide a simple sufficient condition guaranteeing that in the finite horizon case the sequence of value functions resulting from backward induction using an approximate Bellman operator Γ_N will be uniformly close to the true value functions produced by backward induction using the true Bellman operator Γ.

Lemma 2.4: *Let $\{\Gamma_N\}$ be a family of contraction mappings that converge pointwise to a contraction mapping Γ. Suppose there exists an integer $\overline{N}(\epsilon, \beta)$ such that for all $N \geq \overline{N}(\epsilon, \beta)$ we have:*

$$\left\| \hat{\Gamma}_N(W) - \Gamma(W) \right\| \leq (1-\beta)\epsilon, \tag{2.18}$$

uniformly for all $W \in B\Big(0, K/(1-\beta)\Big)$ where the constant K is given by:

$$K \equiv \sup_{s \in S} \sup_{a \in A(s)} |u(s,a)|. \tag{2.19}$$

If we begin the backward induction using any estimate of the terminal value function $\hat{V}_T$ satisfying:

$$\begin{aligned} &\|\hat{V}_T - V_T\| \leq \epsilon, \\ &\|\hat{V}_T\| \leq K/(1-\beta), \end{aligned} \tag{2.20}$$

then $\hat{V}_t$ will be uniformly within ϵ of V_t for all t:

$$\max_{t \in \{1, \ldots, T\}} \|\hat{V}_t - V_t\| \leq \epsilon. \tag{2.21}$$

The proof of Lemma 2.4 is given in the appendix. We conclude this section with a definition of the class of DDPs, the subclass of MDPs which will be the focus of the remainder of the paper:

Definition 2.2: *A **Discrete Decision Process** (DDP) is an MDP with the following property:*

- There is a finite set A such that $A(s) \subset A$ for each $s \in S$.

For simplicity, section 3 will make the further assumption that $A(s) = A$ for all $s \in S$. This apparent restriction actually does not involve any loss of generality, since we can mimic the outcome of a problem with state-dependent choice sets $A(s)$ by a problem with a state-independent choice set A by choosing the utility function $u(s,a)$ so that the utility of any "infeasible" action $a \in A \cap A(s)^c$ is so low that it will never be chosen.

3. Random Bellman Operators

This section derives bounds on the approximation error of using a "random Bellman operator", $\tilde{\Gamma}_N$ in place of the true Bellman operator Γ defined in equation (2.6) of section 2. Since evaluation of the true Bellman operator involves multivariate integration, it generally can only be approximated whereas the random Bellman operator we propose requires only a finite number of algebraic operations and is thus quite simple to evaluate. In section 4 we propose random versions of successive approximations and multigrid algorithms that simply involve replacing iterations of the true Bellman operator Γ by the random Bellman operator $\tilde{\Gamma}_N$. Using the error bound derived in Lemma 2.1 of section 2, we will be able to derive a bound on the expected error between the true value function V (the fixed point to the true Bellman operator Γ) and the random value function $\tilde{V}_N$ (an approximation to the fixed point to the random Bellman operator $\tilde{\Gamma}_N$) in terms of the expected error of $\|\tilde{\Gamma}_N(V) - \Gamma(V)\|$. This bound, which holds uniformly for all $V \in B(0,K)$ for some constant K, is the key to the derivation of the complexity bounds in section 4. We begin by presenting some preliminary definitions and inequalities and establishing the asymptotic properties of the random Bellman operator. Section 3.2 uses a maximal inequality of Pollard (1989) to derive a uniform bound on $E\{\|\tilde{\Gamma}_N(V) - \Gamma(V)\|\}$. Proofs of all results are given in the appendix.

3.1 Preliminary Definitions and Inequalities

Definition 3.1 *The **random Bellman operator** $\tilde{\Gamma}_N : B \to B$ (where $B = C[0,1]^d$) is given by:*

$$\tilde{\Gamma}_N(V)(s) \equiv \max_{a \in A} \left[u(s,a) + \frac{\beta}{N} \sum_{k=1}^{N} V(\tilde{s}_k) p(\tilde{s}_k | s, a) \right], \tag{3.1}$$

where $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$ are IID draws with respect to Lebesgue measure λ from the unit hypercube $[0,1]^d$.

Note that the operator $\tilde{\Gamma}_N$ is *self-approximating:* for any function V one can evaluate $\tilde{\Gamma}_N(\hat{V})(s)$ at any point $s \in S$ without requiring any explicit interpolation of the values of $\tilde{\Gamma}_N(\hat{V})(s)$ at the random sample points $s \in \{\tilde{s}_1, \ldots, \tilde{s}_N\}$.

[17] In particular, if u and p are continuous functions of s, then $\tilde{\Gamma}_N(V)$ is a (random) continuous function of s and evaluation of this function at any particular point $s \in [0,1]^d$ involves nothing more than evaluating the simple formula on the right hand side of equation (3.1).

Our proof of the convergence of $\tilde{\Gamma}_N(V)$ to $\Gamma(V)$ is based on the linear operators $\tilde{\Gamma}_{a,N}$ and Γ_a defined by:

$$\begin{aligned} \tilde{\Gamma}_{a,N}(V)(s) &= u(s,a) + \frac{\beta}{N}\sum_{k=1}^{N} V(\tilde{s}_k)p(\tilde{s}_k|s,a) \\ \Gamma_a(V)(s) &= u(s,a) + \beta \int V(s')p(s'|s,a)\lambda(ds'). \end{aligned} \tag{3.2}$$

The operators $\tilde{\Gamma}_N(V)$ and $\Gamma(V)$ can be regarded as "envelopes" of the linear operators $\tilde{\Gamma}_{a,N}(V)$ and $\Gamma_a(V)$ in the sense that:

$$\begin{aligned} \tilde{\Gamma}_N(V) &= \max_{a\in A} \tilde{\Gamma}_{a,N}(V) \\ \Gamma(V) &= \max_{a\in A} \Gamma_a(V), \end{aligned} \tag{3.3}$$

where the maximization on the right hand side of (3.3) is performed pointwise for each $s \in S$.

Note that $\tilde{\Gamma}_N$ is not guaranteed to be a contraction mapping with probability 1 since $\sum_{i=1}^{N} p(s_k|s,a)/N$ does not necessarily sum to 1. However a simple application of the uniform strong law of large numbers shows that the sum converges 1 with probability 1 uniformly for $s \in [0,1]^d$, so that $\tilde{\Gamma}_N$ will be a contraction for N sufficiently large for any $\beta \in [0,1)$. However since we want error bounds that hold for all N, we show that by using a simple normalization we can construct a closely related random Bellman operator, $\hat{\Gamma}_N$ that is guaranteed to be a contraction mapping for all N and all sample points $\{s_1, \ldots, s_N\}$:

$$\hat{\Gamma}_N(V)(s) = \max_{a\in A}\left[u(s,a) + \beta\sum_{k=1}^{N} V(s_k)p_N(s_k|s,a)\right], \tag{3.4}$$

where p_N is defined by:

$$p_N(s_k|s,a) = \frac{p(s_k|s,a)}{\sum_{i=1}^{N} p(s_i|s,a)}, \tag{3.5}$$

if $\sum_{i=1}^{N} p(s_i|s,a) > 0$, or 0 otherwise. It turns out to be much simpler to analyze the asymptotic properties of the sequence $\{\tilde{\Gamma}_{a,N}(V)\}$ since it is a simple sample average of *IID* random elements. Thus, the initial analysis will focus on the random linear operators $\tilde{\Gamma}_{a,N}$. However we will show that error bounds on $\tilde{\Gamma}_{a,N}(V)$ and $\tilde{\Gamma}_N(V)$ can be used to derive corresponding error bounds for the normalized operators $\hat{\Gamma}_{a,N}(V)$ and $\hat{\Gamma}_N(V)$. In particular, the normalized operators also share the $\sqrt{N}$ rate of convergence to the true operators $\Gamma_a(V)$ and $\Gamma(V)$, which implies that backward induction or successive approximation based on the normalized random Bellman operator $\hat{\Gamma}_N$ succeeds in breaking the curse of dimensionality.

---

[17] The random Bellman operator was inspired by a deterministic self-approximating linear operator introduced by Tauchen and Hussey, (1991).

The following lemma is the key to the subsequent analysis. It bounds the approximation error in the nonlinear operator $\|\tilde{\Gamma}_N(V) - \Gamma(V)\|$ by the maximum of the approximation errors in the linear operators $\|\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)\|$.

Lemma 3.1 $\forall N \geq 1$ we have:

$$\|\tilde{\Gamma}_N(V) - \Gamma(V)\| = \| \max_{a \in A} \tilde{\Gamma}_{a,N}(V) - \max_{a \in A} \Gamma_a(V)\| \leq \max_{a \in A} \|\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)\| \leq \sum_{a \in A} \|\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)\|. \tag{3.6}$$

Notice that inequality (3.6) holds *everywhere* i.e. for any $N \geq 1$ and any set of sample points $\{s_1, \ldots, s_N\} \in [0,1]^{dN}$, and not just with probability 1. It is easy to verify that inequality (3.6) also holds for the normalized operators $\hat{\Gamma}_N$ and $\hat{\Gamma}_{a,N}$.

Before we can proceed further, we need to specify the regularity conditions defining the subclass of DDP problems for which our subsequent results apply. The regularity conditions amount to the requirement that u and p are Lipschitz continuous functions of s.

(A1) $S = [0,1]^d$.

(A2) $\exists K_u < \infty, \forall a \in A, \forall s, s' \in S$ *we have:* $|u(s,a) - u(s',a)| \leq K_u |s - s'|$.

(A3) The transition probability has a density $p(s'|s,a)$ with respect to Lebesgue measure λ on $[0,1]^d$.

(A4) $\forall a \in A, \forall s', s, t \in S \;\; |p(s'|s,a) - p(s'|t,a)| \leq K_p(s')\|s - t\|$ *where* $K_p(s')$ *satisfies:*

$$\int K_p^2(s')\lambda(ds') < K_p^2 < \infty.$$

For notational convenience we will also assume that the following inequalities hold:

$$\begin{aligned} \max_{a \in A} \sup_{s \in S} |u(s,a)| &\leq K_u, \\ \max_{a \in A} \sup_{s' \in S} \sup_{s \in S} \; p(s'|s,a) &\leq K_p. \end{aligned} \tag{3.7}$$

Definition 3.2 *Let* $BL(K) \subset B$ *denote the set of uniformly Lipschitz functions with Lipschitz bound* K*, i.e.*

$$BL(K) = \left\{ f \in B \middle| \; |f(s) - f(s')| \leq K\|s - s'\|, \; \|f\| \leq K \right\}. \tag{3.8}$$

The Arzelà-Ascoli theorem implies that $BL(K)$ is a compact subset of B. Compactness is a key to the error bounds derived below. The next lemma shows that the random Bellman operator maps elements of B into the compact set $BL(K)$ for some $K < \infty$.

Lemma 3.2 $\forall K > 0, \forall N \geq 1$ *the operators* $\tilde{\Gamma}_N$, $\tilde{\Gamma}_{a,N}$, $\hat{\Gamma}_N$, $\hat{\Gamma}_{a,N}$, Γ, *and* Γ_a *map* $B(0,K)$ *into* $BL(K_u + \beta K K_p)$*, where* $B(0,K)$ *is the ball of radius* K *in* B.

Lemma 3.2 implies that the random elements $\tilde{\Gamma}_N(V)$ and $\tilde{\Gamma}_{a,N}(V)$ are concentrated with probability 1 on the compact subset $BL(K_u + \beta\|V\|K_p) \subset B$.

3.2 Error Bounds for Random Bellman Operators

In this section we show that the expected error in using the random Bellman operator $\tilde{\Gamma}_N$ to approximate the true Bellman operator Γ decreases at rate $1/\sqrt{N}$ independent of the dimension d. To gain some insight into why this might be so, note that by Lemma 3.1, the approximation error $\|\tilde{\Gamma}_N(V) - \Gamma(V)\|$ is bounded above by the sum of the approximation errors in the random linear operators $\tilde{Z}_{a,N}(V)$ defined by

$$\tilde{Z}_{a,N}(V) \equiv \sqrt{N}[\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)] \tag{3.9}$$

Observe that for each $V \in B$ $\tilde{Z}_{a,N}(V)$ is a Lipschitz function of $t \in [0,1]^d$, i.e. a random element of $C([0,1]^d)$. Let $N(\epsilon)$ be the minimal number of balls of radius ϵ which cover $[0,1]^d$. Since $N(\epsilon) = O(1/\epsilon^d)$ the d-dimensional hypercube has finite *metric entropy:*

$$\int_0^1 \sqrt{\log(N(\epsilon))}d\epsilon < \infty. \tag{3.10}$$

The Jain-Marcus (1975) central limit theorem implies that $\tilde{Z}_{a,N}(V) \Longrightarrow_w \tilde{Z}_a(V)$ where $\tilde{Z}_a(V)$ is a Gaussian random element of $C([0,1]^d)$. This, together with Lemma 3.1 implies the following result:

Theorem 3.1: *For each $V \in B$ we have:*

$$\begin{aligned} \sqrt{N}\|\Gamma(V) - \tilde{\Gamma}_N(V)\| &= O_p(1) \\ \sqrt{N}\|\Gamma(V) - \hat{\Gamma}_N(V)\| &= O_p(1). \end{aligned} \tag{3.11}$$

Thus, the error in approximating $\Gamma(V)$ by $\tilde{\Gamma}_N(V)$ is $O_p(1/\sqrt{N})$. The fact that the rate of convergence is independent of the dimension d suggests that that iterations based on the random Bellman operator might be capable of breaking the curse of dimensionality. However Theorem 3.1 is not sufficient to prove the result since we need to show that the expectation of the $O_p(1/\sqrt{N})$ approximation errors in (3.11) do not increase exponentially fast in d. Indeed, in order to derive the complexity bounds in section 4, we need an bound on the expected error $E\{\|\tilde{\Gamma}_N(V) - \Gamma(V)\|\}$ that holds *uniformly,* i.e. for all $N \geq 1$, for all $V \in B(0,K)$, and for all u and p satisfying $(A2), \ldots, (A4)$.

To establish the latter bound, we appeal to a maximal inequality for empirical processes due to Pollard (1989). The reason empirical processes play a role in this problem is due to the fact that the random linear operators $\tilde{Z}_{a,N}(V) \equiv \sqrt{N}[\tilde{\Gamma}_{a,N} - \Gamma_a(V)]$ can be represented as stochastic integrals with respect to the empirical process $\tilde{B}_N$:

$$\tilde{Z}_{a,N}(V)(t) = \beta \int V(s)p(s|t,a)\tilde{B}_N(ds), \tag{3.12}$$

where $\tilde{B}_N(s) = \sqrt{N}[\lambda_N(s) - \lambda(s)]$, $\lambda(s)$ is Lebesgue measure of the set $[0, s]$, and $\lambda_N(s)$ is the empirical CDF

$$\lambda_N(s) = \frac{1}{N}\sum_{i=1}^{N} I\{s_i \leq s\}. \tag{3.13}$$

The maximal inequality provides a bound on the expectation of the supremum norm of $\tilde{Z}_{a,N}(V)$, i.e. a bound on $E\{\|\tilde{Z}_{a,N}(V)\|\}$ which holds for all $N \geq 1$. Since these maximal inequalities are derived from somewhat simpler maximal inequalities for Gaussian processes (e.g. Theorem 3.2 in Pollard, (1989)), it is convenient to derive the bound in two steps: we first derive a bound on $E\{\|\tilde{Z}_a(V)\|\}$ where $\tilde{Z}_a(V)$ is the limiting Gaussian process, and then use Pollard's "symmetrization method" to show that this bound also applies to $E\{\|\tilde{Z}_{a,N}(V)\|\}$ at the cost of a slight increase in the bounding constant. The maximal inequality can be defined in terms of a covering integral similar to (3.10), except that we replace $N(\epsilon)$ by $N(\epsilon/2)$ and the covering number $N(\epsilon)$ is defined in terms of a metric $\rho(t, s)$ on $[0, 1]^d$ satisfying

$$E\left\{|\tilde{Z}_a(V)(s) - \tilde{Z}_a(V)(t)|^2\right\} \leq \rho(s, t)^2. \tag{3.14}$$

In order to define $\rho(s, t)$ and the implied bound on $E\{\|\tilde{Z}_a(V)\|\}$, it is helpful to have an explicit representation for the limiting Gaussian process $\tilde{Z}_a(V)$. As is well known from the literature on empirical processes, $\tilde{B}_N$ converges weakly to $\tilde{B}$, where $\tilde{B}$ is the Brownian bridge process on $[0, 1]^d$ (see, e.g. Gänßler (1983)). This implies that the Gaussian stochastic process $\tilde{Z}_a(V)$ also has a representation as a stochastic integral with respect to the limiting Brownian Bridge process:

$$\tilde{Z}_a(V)(t) = \beta \int V(s)p(s|t, a)\tilde{B}(ds). \tag{3.15}$$

Using this representation, it is easy to see that the metric $\rho(s, t) = \beta K_p\|V\|\|s-t\|$ satisfies (3.14) and the corresponding covering integral satisfies:

$$\int_0^{\delta} \sqrt{\log\left(N(\epsilon/2)\right)}d\epsilon \leq d\sqrt{\pi}\beta K_p\|V\|, \tag{3.16}$$

where $\delta = \sup_{s\in[0,1]^d} \rho(s, 0) = \sqrt{d}\beta K_p\|V\|$.

Theorem 3.2 *Let $\tilde{Z}_a(V)$ be the Gaussian process defined in equation (3.15). Then $E\{\|\tilde{Z}_a(V)\|\}$ satisfies the following bound:*

$$\begin{aligned} E\left\{\|\tilde{Z}_a(V)\|\right\} &\leq E\left\{\left|\tilde{Z}_a(V)(0)\right|\right\} + C\int_0^{\delta} \sqrt{\log\left(N(\epsilon/2)\right)}d\epsilon \\ &\leq \beta\left[1 + d\sqrt{\pi}C\right] K_p\|V\|, \end{aligned} \tag{3.17}$$

where C is a universal constant independent of $\tilde{Z}_a(V)$.

Corollary: *The bound in Theorem 3.3 holds uniformly for all $p \in BL(K_p)$ and all $\|V\| \in B(0, K_v)$:*

$$\sup_{p\in BL(K_p)} \sup_{V\in B(0,K_v)} E\{\|\tilde{Z}_a(V)\|\} \leq \beta\left[1 + d\sqrt{\pi}C\right] K_pK_v. \tag{3.18}$$

The next step is to apply the symmetrization method of Pollard (1989) to show that a version of the inequality (3.18) not only holds in the limit as $N \to \infty$, but also for all $N \geq 1$ at the cost of a slight increase in the bounding constant — by a factor of $\sqrt{\pi/2}$.

Theorem 3.3 *For each $N \geq 1$ we have:*

$$E\left\{\|\tilde{Z}_{a,N}(V)\|\right\} = E\left\{\sqrt{N}\|\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)\|\right\} \leq \sqrt{\frac{\pi}{2}}\beta\left[1 + d\sqrt{\pi}C\right] K_p\|V\|. \tag{3.19}$$

Corollary *For each $n \geq 1$ the following uniform bound holds:*

$$\sup_{p\in BL(K_p)} \sup_{V\in B(0,K_v)} E\{\|\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)\|\} \leq \frac{\gamma(d)K_pK_v}{\sqrt{N}}, \tag{3.20}$$

where the constant $\gamma(d)$ is given by:

$$\gamma(d) = \sqrt{\frac{\pi}{2}}\beta\left[1 + d\sqrt{\pi}C\right]. \tag{3.21}$$

Using inequality (3.20) and inequality (3.6) from Lemma 3.1, we obtain the main result of this section.

Theorem 3.4 *For each $N \geq 1$ the expected error in the random Bellman operator satisfies the uniform bound:*

$$\sup_{p\in BL(K_p)} \sup_{V\in B(0,K_v)} E\left\{\|\tilde{\Gamma}_N(V) - \Gamma(V)\|\right\} \leq \frac{\gamma(d)|A|K_pK_v}{\sqrt{N}}, \tag{3.22}$$

where the bounding constant $\gamma(d)$ is given in equation (3.21).

Corollary *The expected error in the normalized Bellman operator $\hat{\Gamma}_N$ satisfies:*

$$\sup_{p\in BL(K_p)} \sup_{V\in B(0,K_v)} E\left\{\|\hat{\Gamma}_N(V) - \Gamma(V)\|\right\} \leq 2\frac{\gamma(d)|A|K_pK_v}{\sqrt{N}}, \tag{3.23}$$

where the bounding constant $\gamma(d)$ is given in equation (3.21).

4. Complexity Bounds for Random Successive Approximations and Random Multigrid Algorithms

Using the error bound in Theorem 3.4, it is now straightforward to show that the random Bellman operator succeeds in breaking the curse of dimensionality for finite and infinite horizon DDP problems. We begin by considering the complexity of the random version of the successive approximations algorithm. Given a desired solution tolerance ϵ we choose a number of simulations N sufficiently large that the expected error in the solution will be less than ϵ. Then we draw *IID* uniform random sample points $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$ which will subsequently remain fixed in each of T backward induction steps. Backward induction begins with a value function $\hat{V}_T \in R^N$ given by:

$$\hat{V}_T(\tilde{s}_i) = \underset{a \in A}{argmax}\, u(\tilde{s}_i, a) \quad i = 1, \ldots, N. \tag{4.1}$$

Subsequent value functions V_{T-t}, $t = 0, \ldots, T$ are generated by successive application of the $\hat{\Gamma}_N$ operator:

$$\hat{V}_{T-t}(\tilde{s}_i) = \hat{\Gamma}_N^t(V_T)(\tilde{s}_i), \quad t = 0, \ldots, T, \; s_i = 1, \ldots, N. \tag{4.2}$$

Recall from the introduction that the total work involved in carrying out this backward induction is $O(T|A|N^2)$. Thus, the solution the algorithm produces are approximations to the $(T+1)$ value functions V_t, $t = 0, \ldots, T$ evaluated at N randomly selected points in S, i.e. a $(T+1) \times N$ array $\{\hat{V}_t(\tilde{s}_i) \,|\, i = 1, \ldots, N, \; t = 0, \ldots, T\}$ satisfying $E\{|\hat{V}_t(\tilde{s}_i) - V_t(\tilde{s}_i)|\} < \epsilon$. The complexity functions in the theorems below are upper bounds on the total number of arithmetic operations required to 1) calculate the "information", i.e. evaluate the u and p functions at all $s \in \{\tilde{s}_1, \ldots, \tilde{s}_N\}$ and for all $a \in A$: $\{u(\tilde{s}_i, a), p(\tilde{s}_i|\tilde{s}_j, a)|i, j = 1, \ldots, N \; a = 1, \ldots, |A|\}$, and 2) calculate the $(T+1)$ approximate value functions $\hat{V}_t(\tilde{s}_i)$ at the N randomly selected points $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$. The self-approximating property of the random Bellman operator implies that the algorithm can also be used to find solutions $\hat{V}_t(s)$ at points $s \notin \{\tilde{s}_1, \ldots, \tilde{s}_N\}$, and the uniform convergence bounds in Theorem 3.4 guarantees that if $N = \Theta(d^2|A|^2K_p^2K_u^2/\epsilon^2)$, then $E\{\sup_{s \in S} |\hat{V}_t(s) - V_t(s)|\} < \epsilon$. The marginal cost of calculating $\hat{V}_t(s)$ at an additional point $s \notin \{\tilde{s}_1, \ldots, \tilde{s}_N\}$ is $O(|A|N^2)$.

Finally, note that under our assumptions, if K_v is an upper bound on V_t, then $\beta \int V_t(s')p(s'|s,a)\lambda(ds') < \beta K_v$, and if this latter term is less than ϵ we can compute an ϵ-approximation to V_t in $O(|A|N)$ operations by simply solving the static optimization problem $max_{a \in A(s_i)}[u(s_i, a)]$ at N points $\{s_1, \ldots, s_N\}$ in S and ignoring the problem of calculating $\beta \int V_t(s')p(s'|s,a)\lambda(ds')$. Therefore the complexity bounds given below are valid only for (β, ϵ) satisfying $K_v > \epsilon/\beta$, i.e. for β sufficiently large and ϵ sufficiently small. In the infinite horizon case, $K_v = K_u/(1-\beta)$ and the condition becomes $\beta K_u > (1-\beta)\epsilon$. We implicitly assume that these inequalities hold in order to simplify the statements of Theorems 4.1 and 4.2 below.

Theorem 4.1 *Randomization breaks the curse of dimensionality of solving finite horizon DDP's: i.e. an upper bound on the worst case complexity of the class of randomized algorithms for solving a T-period DDP*

problem with $|A|$ possible actions, discount factor $\beta \in (0,1)$ and utility function and transition probability (u,p) satisfying $(A1),\ldots,(A4)$ is given by:

$$comp^{\textsf{wor-ran}}(\epsilon,d) = O\left(\frac{Td^4|A|^5K_u^4K_p^4}{(1-\beta)^8\epsilon^4}\right). \tag{4.3}$$

Corollary *Randomization breaks the curse of dimensionality of solving infinite horizon DDPs: i.e. an upper bound on the worst case randomized complexity of the infinite horizon DDP problem is given by:*

$$comp^{\textsf{wor-ran}}(\epsilon,d) = O\left(\frac{\log\big(1/(1-\beta)\epsilon\big)d^4|A|^5K_u^4K_p^4}{|\log(\beta)|(1-\beta)^8\epsilon^4}\right). \tag{4.4}$$

We now consider a random extension of the oneway multigrid algorithm introduced by Chow and Tsitsiklis (1991). We show that for infinite horizon DDPs, this "random multigrid algorithm" reduces the upper bound on the worst case randomized complexity of infinite horizon DDP problems by a factor $\log(1/(1-\beta)\epsilon)$. The random multigrid algorithm consists of a number of "outer" iterations $k = 1,2,\ldots,$ where a number N_k of uniform random sample points $\{\tilde{s}_1,\ldots,\tilde{s}_{N_k}\}$ is drawn at each iteration k independently of the sample points drawn at previous iterations $k-1,k-2,\ldots$ of the multigrid algorithm. The basic idea is to start at iteration $k=0$ with a relatively small number of sample points N_0 and successively increase the number of sample points drawn at each iteration by a factor of 4:

$$N_k = 2^{2k}N_0. \tag{4.5}$$

Within each outer iteration k, a number $T(k)$ of inner successive approximation steps are taken using the random Bellman operator $\hat{\Gamma}_{N_k}$. Let $\hat{V}_k$ denote the value function produced at the termination of the $T(k)$ successive approximation steps at outer iteration k. The starting point for successive approximations in outer iteration k is the value function $\hat{V}_{k-1}$ produced at outer iteration $k-1$. Thus we have the recursion:

$$\hat{V}_k = \hat{\Gamma}_{N_k}^{T(k)}(\hat{V}_{k-1}), \tag{4.6}$$

where the starting point for iteration 0 of the multigrid algorithm is given by (4.1). Since the expected error between $\hat{\Gamma}_{N_k}$ and Γ is given by:

$$E\left\{\left\|\hat{\Gamma}_{N_k}(V)-\Gamma(V)\right\|\right\} \le \frac{\gamma(d)|A|K_uK_p}{\sqrt{N_k}(1-\beta)} \equiv \frac{K}{\sqrt{N_k}(1-\beta)}, \tag{4.7}$$

we choose the following stopping rule for the inner successive approximation steps: $T(k)$ is the smallest integer t satisfying:

$$E\left\{\left\|\hat{\Gamma}_{N_k}^{t}(\hat{V}_{k-1})-\hat{\Gamma}_{N_k}^{t-1}(\hat{V}_{k-1})\right\|\right\} \le \frac{K}{\sqrt{N_k}\beta(1-\beta)}. \tag{4.8}$$

Given a desired solution tolerance of $\epsilon > 0$ the stopping criterion for the outer iterations is the smallest value k^* satisfying:

$$N_{k^*} \geq \frac{K^2}{(1-\beta)^4 \epsilon^2}. \tag{4.9}$$

Since N_k is increasing by a factor of 4 at each outer iteration, (and therefore the expected error is being halved at each outer iteration of the multigrid algorithm), it is clear that the multigrid algorithm will terminate after a finite number of iterations k. Let $\hat{V}_{N_{k^*}}$ denote the fixed point of the operator $\hat{\Gamma}_{N_{k^*}}$ at the final iteration of the multigrid algorithm. Inequality (2.15) and the stopping rules for $T(k^*)$ given in equation (4.8) imply the following bound:

$$E\left\{\left\|\hat{V}_{k^*} - \hat{V}_{N_{k^*}}\right\|\right\} \leq \epsilon. \tag{4.10}$$

Inequality (2.12) and the stopping rule for k^* given in inequality (4.9) imply that the expected error between $\hat{V}_{N_{k^*}}$ and the true fixed point $V = \Gamma(V)$ is given by:

$$E\left\{\left\|\hat{V}_{N_{k^*}} - V\right\|\right\} \leq \frac{E\left\{\left\|\hat{\Gamma}_{N_{k^*}}(V) - \Gamma(V)\right\|\right\}}{(1-\beta)} \leq \epsilon. \tag{4.11}$$

Using the triangle inequality and inequalities (4.10) and (4.11) we have:

$$E\left\{\left\|\hat{V}_{k^*} - V\right\|\right\} \leq 2\epsilon. \tag{4.12}$$

Thus, the multigrid algorithm terminates after a finite number of iterations k^* with an expected error of 2ϵ. The next lemma shows that the stopping rule (4.8) for the successive approximation steps guarantees that $T(k) = O(1/|\log(\beta)|)$ independent of ϵ, and k.

Lemma 4.1 *There exists a constant c independent of β and ϵ such that:*

$$T(k) \leq \frac{c}{|\log(\beta)|}, \quad k = 1, \ldots, k^*. \tag{4.13}$$

Theorem 4.2 *An upper bound on the worst case complexity of infinite horizon DDP problems is given by:*

$$comp^{\textsf{wor-ran}}(\epsilon, d) = O\left(\frac{|A|^5 d^4 K_u^4 K_p^4}{|\log(\beta)|(1-\beta)^8 \epsilon^4}\right). \tag{4.14}$$

5. Extensions and Conjectures

This paper has established upper bounds on the randomized complexity of finite and infinite horizon DDP problems. We showed that randomization succeeds in breaking the curse of dimensionality for a subclass of DDP problems satisfying a Lipschitz condition. We assumed, for simplicity, that the Lipschitz bounds K_u and K_p on (u,p) are constants independent of the problem dimension, d. It is easy to show that all our results go through provided that K_u and K_p increase polynomially in d. A much more difficult problem is to establish tight upper and lower bounds on the randomized complexity of the DDP problem. We conjecture that an integration algorithm similar to Bakhvalov's 1959 algorithm will be an optimal random algorithm for the DDP problem, with a complexity exponent equal to the square of the exponent $m = d/r$ (where r denotes the highest degree of differentiability of u and p), indicating that smoother DDP problems enjoy faster rates of convergence. However since $m = d/r$, the gain in using more sophisticated randomization schemes over the simple monte carlo algorithm will be small when the problem dimension d is large relative to the degree of smoothness r.

The analysis in this paper assumes the *real number model of computation.* That is, we have assumed that all calculations are carried out in infinite-precision arithmetic and that the computer is capable of generating truly *IID* random uniform draws from the unit cube $[0,1]^d$. Of course, actual computers are only capable of finite-precision arithmetic and generate *pseudorandom* uniform draws from $[0,1]^d$ using deterministic algorithms. However we agree with the view expressed in Traub, Wasilikowski and Woźniakowski 1988 that "pseudo-random computation may be viewed as a close approximation of random computation, and that randomness is a very powerful tool for computation even if implemented on deterministic computers" (p. 414). Indeed, Traub and Woźniakowski 1992 have shown that monte carlo algorithms based on a linear congruential generator of period m with a uniformly distributed initial seed "behaves as for the uniform distribution and its expected error is roughly $n^{-1/2}$ as long as the number n of function values is less than m^2." (p. 323).

An important open question is whether one can break the curse of dimensionality of the DDP problem on an average case basis: i.e. when the error in the algorithm is evaluated relative to a *prior distribution* over the space (u,p) of possible DDP problems (for details on how average case complexity is defined, see see Traub, Wasilikowski and Woźniakowski, 1988). Certain problems including multivariate integration have been shown to be tractable on an average case basis even though they are intractable on a worst case basis. The difficulty in applying an average case analysis to the DDP problem is to find a reasonable prior over the space of admissible transition probabilities. The typical prior used in multivariate integration problems, folded Wiener sheet measure, does not ensure that the transition probabilities p are nonnegative and integrate to 1.

Recent research has shows that the sample average integration algorithm (3.7) based on *deterministic* sample points such as the Hammersley, Halton, and Sobol' points, approaches the lower bound on the average case complexity

of numerical integration (see, e.g. Woźniakowski, 1991 who showed that a shifted version of the Hammersley points actually attains the lower bound on average case complexity). We can gain some insight into why these deterministic integration methods work well from the *Koksma-Hlwaka inequality:*

$$\left| \frac{1}{N} \sum_{i=1}^{N} f(s_i) - \int f(s)\lambda(ds) \right| \leq V(f) D_N^*(s_1, \ldots, s_N), \tag{5.1}$$

where λ is Lesbesgue measure on $[0,1]^d$, $V(f)$ is the total variation in f in the sense of Hardy and Krause (see page 19 of Neiderreiter for a definition), and D_N^* is the *discrepancy:*

$$D_N^*(s_1, \ldots, s_N) \equiv \sup_{B \in \mathcal{B}} |\lambda_N(B) - \lambda(B)|, \tag{5.2}$$

where $\mathcal{B}$ is the class of (open) suborthants of $[0,1]^d$, ($\mathcal{B} = \{[0,s)^d \subset [0,1]^d | s \in [0,1]^d\}$) and λ_N is the empirical CDF corresponding to the sample points $(s_1, \ldots, s_N)$. This inequality suggests that one can obtain maximal inequalities for a class of functions of uniformly bounded $V(f)$-variation using a maximal inequality for the empirical process indexed by the class $\mathcal{B}$. Moreover, the Koksma-Hlwaka inequality suggests that we can get more accurate estimates of the integral by using deterministic sequences of points $(s_1, \ldots, s_N)$ for which the discrepancy is small or even minimal. An interesting literature on *discrepancy bounds* (surveyed in Neiderreiter 1992) has derived upper and lower bounds on the rate of decrease of certain "low discrepancy" point sets such as the Hammersley, Halton, and Sobol' points. For example Roth's 1954 lower bound on the discrepancy of any set of points $(s_1, \ldots, s_N)$ in $[0,1]^d$ is given by:

$$D_N^*(s_1, \ldots, s_N) \geq K(d) \frac{(\log N)^{(d-1)/2}}{N}$$

where $K(d)$ is a universal constant that only depends on the dimension of the hypercube, d. An upper bound for the N-element Hammersley point set P is given by:

$$D_N^*(P) \leq \frac{\gamma(d)(\log N)^{d-1}}{N} + O(N^{-1}(\log N)^{d-2}).$$

This means that multivariate integration using deterministic sequences such as Hammersley points do much better than random samples since the rate of decrease in the expected error in the latter is at the slower rate $1/\sqrt{N}$. Recent numerical experiments comparing the accuracy of numerical integration using deterministic integration points such as the Halton and Sobol' points versus standard monte carlo integration in Paskov (1996) and Paskov and Traub (1996) confirms that for certain classes of functions, the deterministic algorithms provide more accurate estimates in less cpu time. The intuitive reason for the superior performance of deterministic, low discrepancy sequences over pseudo-random sequences is that the former are more evenly distributed about the unit cube, with fewer "gaps" and "clusters". This is visually apparent in figures 5.1 and 5.2 which compare successive points generated by Sobol's sequences to those generated by a linear congruential pseudo random number generator. In future work we plan

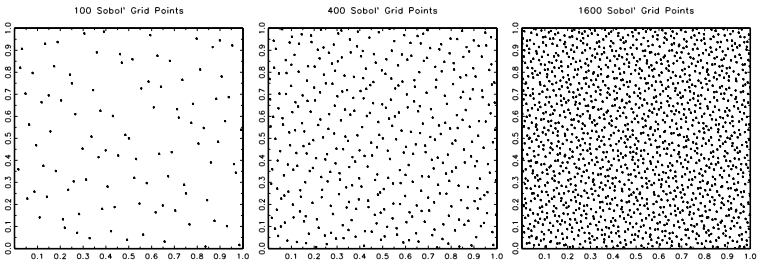

Figure 5.1 Example of Successive Grids Generated by Sobol Multigrid Algorithm

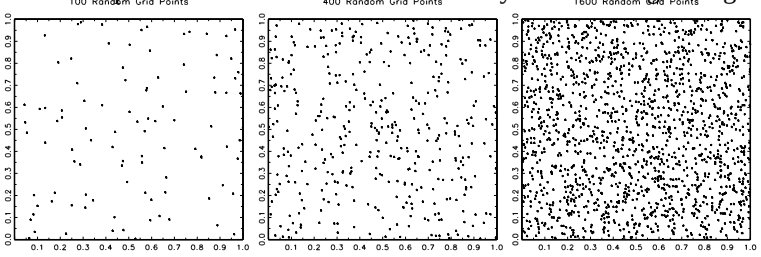

Figure 5.2 Example of Successive Grids Generated by Random Multigrid Algorithm

to investigate versions of successive approximations and the multigrid algorithms for solving DDP problems using the Sobol' points and other low discrepancy sequences. Our conjecture is that the deterministic versions of these algorithms could significantly outperform the "random" versions for a wide variety of DDP problems.

6. Appendix: Proofs of Propositions

Proof of Lemma 2.4: We prove the result by induction on t. Starting at $t = T-1$ suppose we choose N and $\hat{V}_T$ satisfying (2.18), (2.19), and (2.20). Then we have:

$$\begin{aligned}\left\|\hat{V}_{T-1} - V_{T-1}\right\| &= \left\|\hat{\Gamma}_N(\hat{V}_T) - \Gamma(V_T)\right\| \\ &= \left\|\hat{\Gamma}_N(\hat{V}_T) - \hat{\Gamma}_N(V_T) + \hat{\Gamma}_N(V_T) - \Gamma(V_T)\right\| \\ &\leq \left\|\hat{\Gamma}_N(\hat{V}_T) - \hat{\Gamma}_N(V_T)\right\| + \left\|\hat{\Gamma}_N(V_T) - \Gamma(V_T)\right\| \\ &\leq \beta\epsilon + (1-\beta)\epsilon = \epsilon.\end{aligned} \tag{6.1}$$

This argument can be repeated for each $t = T-1, T-2, \ldots, 1$ provided we can show that for each t we have $\|\hat{V}_t\| \leq K/(1-\beta)$. However this follows follows from Lemma 2.3 and the assumption that the approximate Bellman operator Γ_N is a contraction mapping with modulus β.

Proof of Lemma 3.1: Fix $s \in S$. Define the decision rules α and α_N by:

$$\begin{aligned}\alpha(s) &= \underset{a \in A}{argmax}\, \Gamma_a(V)(s) \\ \alpha_N(s) &= \underset{a \in A}{argmax}\, \tilde{\Gamma}_{a,N}(V)(s).\end{aligned} \tag{6.2}$$

Then we have:

$$\begin{aligned}\Gamma(V)(s) &= \Gamma_{\alpha(s)}(V)(s) \geq \Gamma_{\alpha_N(s)}(V)(s), \\ \tilde{\Gamma}_N(V)(s) &= \tilde{\Gamma}_{\alpha_N(s),N}(V)(s) \geq \tilde{\Gamma}_{\alpha(s),N}(V)(s).\end{aligned} \tag{6.3}$$

Suppose that $\tilde{\Gamma}_N(V)(s) \geq \Gamma(V)(s)$. Then we have the following inequality:

$$\tilde{\Gamma}_{\alpha_N(s),N}(V)(s) = \tilde{\Gamma}_N(V)(s) \geq \Gamma(V)(s) = \Gamma_{\alpha(s)}(V)(s) \geq \Gamma_{\alpha_N(s)}(V)(s). \tag{6.4}$$

It follows that

$$0 \leq \tilde{\Gamma}_N(V)(s) - \Gamma(V)(s) \leq \tilde{\Gamma}_{\alpha_N(s),N}(V)(s) - \Gamma_{\alpha_N(s)}(V)(s) \leq \underset{a \in A}{max}\, |\tilde{\Gamma}_{a,N}(V)(s) - \Gamma_a(V)(s)|. \tag{6.5}$$

Using an identical argument when $\Gamma(V)(s) \geq \tilde{\Gamma}_N(V)(s)$ we get the following inequality:

$$0 \leq \Gamma(V)(s) - \tilde{\Gamma}_N(V)(s) \leq \Gamma_{\alpha(s)}(V)(s) - \tilde{\Gamma}_{\alpha(s),N}(V)(s) \leq \underset{a \in A}{max}\, |\Gamma_a(V)(s) - \tilde{\Gamma}_{a,N}(V)(s)|. \tag{6.6}$$

In either case we have:

$$|\tilde{\Gamma}_N(V)(s) - \Gamma(V)(s)| \leq \underset{a \in A}{max}\, |\tilde{\Gamma}_{a,N}(V)(s) - \Gamma_a(V)(s)|. \tag{6.7}$$

Taking suprema over $s \in S$ we have:

$$\begin{aligned}
\|\tilde{\Gamma}_N(V) - \Gamma(V)\| &= \sup_{s\in S} |\tilde{\Gamma}_N(V)(s) - \Gamma(V)(s)| \\
&\leq \sup_{s\in S} \max_{a\in A} |\tilde{\Gamma}_{a,N}(V)(s) - \Gamma_a(V)(s)| \\
&= \max_{a\in A} \sup_{s\in S} |\tilde{\Gamma}_{a,N}(V)(s) - \Gamma_a(V)(s)| \\
&= \max_{a\in A} \|\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)\| \\
&\leq \sum_{a\in A} \|\tilde{\Gamma}_{a,N}(V) - \Gamma_a(V)\|.
\end{aligned} \tag{6.8}$$

Proof of Theorem 3.2: Define a metric $\rho(s,t)$ on $S = [0,1]^d$ by

$$\rho(s,t) = \beta K_p \|V\| \, \|s - t\|, \tag{6.9}$$

where $\|s-t\|$ is the usual Euclidean distance between the points s and t, $\|s-t\|^2 = (s_1 - t_1)^2 + \cdots (s_d - t_d)^2$. Since the sample paths of $\tilde{Z}_a(V)$ are ρ-continuous, we can apply the maximal inequality in Theorem 3.2 of Pollard 1989 provided that we can show that inequality (3.14) holds. This turns out to be an easy consequence of the representation of $\tilde{Z}_a(V)$ in equation (3.15):

$$\begin{aligned}
&E\{\|\tilde{Z}_a(V)(t) - \tilde{Z}_a(V)(s)\|^2\} = \beta^2 E\left\{\left|\int V(s')[p(s'|t,a) - p(s'|s,a)]\tilde{B}(ds')\right|^2\right\} \\
=&\beta^2 \int V^2(s')[p(s'|s,a) - p(s'|t,a)]^2 \lambda(ds') - \beta^2 \left[\int V(s')[p(s'|s,a) - p(s'|t,a)]\lambda(ds')\right]^2 \\
\leq&\beta^2 K_p^2 \|V\|^2 \|s-t\|^2 = \rho(s,t)^2.
\end{aligned} \tag{6.10}$$

Hölder's inequality and the normality of $\tilde{Z}_a(V)(s_0)$ imply that the first term on the right hand side of inequality (3.14) is bounded by $\beta K_p \|V\|$:

$$E\{|\tilde{Z}_a(V)(s_0)|\} = \beta E\left\{\left|\int V(s) p(s|s_0,a)\tilde{B}(ds)\right|\right\} \leq \beta K_p \|V\|. \tag{6.11}$$

Using the fact that

$$H\left(\frac{\epsilon}{2}, \rho\right) = H\left(\frac{\epsilon}{2\beta K_p \|V\|}, \|\cdot\|\right), \tag{6.12}$$

we can derive an upper bound for $H(\epsilon/2, \rho)$ by partitioning $S = [0,1]^d$ into cubes of length $2\epsilon/\sqrt{d}$ on each side. It is easy to see that the diameter of these cubes is 2ϵ and there are $N = (\sqrt{d}/2\epsilon)^d$ such cubes in the partition. If we let $\{s_1, \ldots, s_N\}$ denote the centers of these cubes, the N balls with radius ϵ constitute an ϵ-net of $[0,1]^d$, so it follows that

$$H(\epsilon, S, \|\cdot\|) \leq \left(\frac{\sqrt{d}}{2\epsilon}\right)^d. \tag{6.13}$$

Integrating the entropy function using the above inequalities we have:

$$\int_0^{\delta} \log\left(H\big(\frac{x}{2\beta K_p\|V\|}, S, \|\cdot\|\big)\right)^{1/2} dx \leq \sqrt{d}\int_0^{\delta} \log\left(\frac{\sqrt{d}\beta K_p\|V\|}{x}\right)^{1/2} dx. \tag{6.14}$$

Let $s_0 = 0 \in [0,1]^d$. A straightforward calculation shows that $\delta = \sup_{s\in[0,1]} \rho(s, s_0) = \sqrt{d}\beta K_p\|V\|$. Using the change of variables $\sqrt{d}\beta K_p\|V\|y = x$, we can evaluate the last integral in (6.14) as:

$$\sqrt{d}\int_0^{\delta} \log\left(\frac{\sqrt{d}\beta K_p\|V\|}{x}\right)^{1/2} dx = d\beta K_p\|V\|\int_0^1 \log\left(\frac{1}{y}\right)^{1/2} dy = d\sqrt{\pi}\beta K_p\|V\|. \tag{6.15}$$

Substituting inequalities (6.11), (6.14), and (6.15) into the maximal inequality in Theorem 3.2 of Pollard 1989 we obtain the inequality for $E\{\|\tilde{Z}_a(V)\|\}$ given in (3.17).

Proof of Corollary to Theorem 3.2: Inequality (3.17) applies to an arbitrary $p \in BL(K_p)$, so inequality (3.17) must hold uniformly for all $p \in BL(K_p)$:

$$\sup_{p\in BL(K_p)} E\{\|\tilde{Z}_a(V)\|\} \leq \beta\left[1 + d\sqrt{\pi}C\right] K_p\|V\|. \tag{6.16}$$

If we substitute K_v for $\|V\|$ in the above inequality, it follows that

$$\sup_{V\in B(0,K_v)}\sup_{p\in BL(K_p)} E\{\|\tilde{Z}_a(V)\|\} = \sup_{p\in BL(K_p)}\sup_{V\in B(0,K_v)} E\{\|\tilde{Z}_a(V)\|\} \leq \beta\left[1 + d\sqrt{\pi}C\right] K_pK_v. \tag{6.17}$$

Proof of Theorem 3.4: This follows from the bound in Theorem 3.3 and a straightforward modification of inequality (7) in Pollard, 1989.

Proof of Corollary to Theorem 3.4: Using the definition of the normalized random Bellman operator in equation (3.4) we can write the following expression for difference between $\hat{\Gamma}_{a,N}(V)(s)$ and $\tilde{\Gamma}_{a,N}(V)(s)$:

$$\hat{\Gamma}_{a,N}(V)(s) - \tilde{\Gamma}_{a,N}(V)(s) = \frac{1}{\frac{1}{N}\sum_{i=1}^N p(s_i|s,a)}\left[1 - \frac{1}{N}\sum_{i=1}^N p(s_i|s,a)\right]\left[\frac{\beta}{N}\sum_{i=1}^N V(s_i)p(s_i|s,a)\right]. \tag{6.18}$$

Equation (6.18) implies the following bound on the difference between $\hat{\Gamma}_{a,N}(V)$ and $\tilde{\Gamma}_{a,N}(V)$:

$$\|\hat{\Gamma}_{a,N}(V) - \tilde{\Gamma}_{a,N}(V)\| \leq \beta\|V\| \sup_{s\in[0,1]^d}\left|1 - \frac{1}{N}\sum_{i=1}^N p(s_i|s,a)\right|. \tag{6.19}$$

Applying the maximal inequality for empirical processes of Theorem 3.4 to that special case when $V \equiv 1$ and $\beta = 1$ we have:

$$E\left\{\sup_{s\in[0,1]^d}\left|1 - \frac{1}{N}\sum_{i=1}^N p(s_i|s,a)\right|\right\} \leq \frac{\gamma(d)K_p}{\sqrt{N}}, \tag{6.20}$$

where $\gamma(d)$ is given in equation (3.21). Together, inequalities (6.19) and (6.20) imply the following uniform bound on the expected error in $\|\hat{\Gamma}_{a,N}(V) - \tilde{\Gamma}_{a,N}(V)\|$:

$$\sup_{p\in BL(K_p)} \sup_{V\in B(0,K_v)} E\left\{\|\hat{\Gamma}_{a,N}(V) - \tilde{\Gamma}_{a,N}(V)\|\right\} \leq \frac{\gamma(d)K_pK_v}{\sqrt{N}}. \tag{6.21}$$

Inequality (3.6) implies the following bound on the expected error in $\|\hat{\Gamma}_N(V) - \tilde{\Gamma}_N(V)\|$:

$$\sup_{p\in BL(K_p)} \sup_{V\in B(0,K_v)} E\left\{\|\hat{\Gamma}_N(V) - \tilde{\Gamma}_N(V)\|\right\} \leq \frac{\gamma(d)|A|K_pK_v}{\sqrt{N}}. \tag{6.22}$$

To obtain the final uniform bound on the expected error in $\|\hat{\Gamma}_N(V) - \Gamma(V)\|$, use inequalities (6.21), (6.22), and the triangle inequality

$$\sqrt{N}\|\hat{\Gamma}_N(V) - \Gamma(V)\| \leq \sqrt{N}\|\hat{\Gamma}_N(V) - \tilde{\Gamma}_N(V)\| + \sqrt{N}\|\tilde{\Gamma}_N(V) - \Gamma(V)\|, \tag{6.23}$$

we obtain inequality (3.23).

Proof of Theorem 4.1: Recall that by Lemma 2.4 if $\hat{V}_T$ is within ϵ of the true term value function V_T given in equation (2.3) of section 2 and N is chosen sufficiently large that $\|\hat{\Gamma}_N(V) - \Gamma(V)\| \leq (1-\beta)\epsilon$ for all $V \in B(0, K_u/(1-\beta))$ then $\hat{V}_{T-t}$ is guaranteed to be within ϵ of the true solution V_{T-t} for all $t = 0, \ldots, T$. Although the random Bellman operator generates a random sequence $\{\hat{V}_0, \ldots, \hat{V}_T\}$ of value functions, it is easy to see that the results of Lemma 4.1 continue to hold when expectations are taken. Thus, if N is chosen sufficiently large so that $E\{\|\hat{\Gamma}_N(V) - \Gamma(V)\|\} \leq (1-\beta)\epsilon$ uniformly for all $V \in B(0, K_u/(1-\beta))$, then we have

$$E\left\{\left\|\hat{V}_{T-t} - V_{T-t}\right\|\right\} \leq \epsilon, \quad t = 0, \ldots, T. \tag{6.24}$$

The error bound in the corollary to Theorem 3.5 allows us to guarantee that $E\{\|\hat{\Gamma}_N(V) - \Gamma(V)\|\} \leq (1-\beta)\epsilon$ uniformly for $p \in BL(K_p)$ and $V \in B(0, K_u/(1-\beta))$ provided the number of random sample points N satisfies:

$$N \geq \left[\frac{2\gamma(d)|A|K_uK_p}{(1-\beta)^2\epsilon}\right]^2, \tag{6.25}$$

where $\gamma(d)$ is defined in (3.21). Since the work involved in solving a T-period DDP problem with N possible discrete states and $|A|$ possible actions by backward induction is $O(T|A|N^2)$, it follows that an upper bound on the complexity of randomized algorithms for solving the DDP problem is given by equation (4.3). Furthermore, Theorem 3.5 guarantees that choosing N to satisfy (6.25) guarantees that the expected error $E\{\|\hat{V}_{T-t} - V_{T-t}\|\} \leq \epsilon$ for any DDP problem (u, p) satisfying $(A1), \ldots, (A4)$. Thus, the complexity bound (4.3) does in fact provide an upper bound on the worst case complexity of solving this class of DDP problems using random algorithms.

Proof of Corollary to Theorem 4.1: Lemma 2.1 guarantees that if $E\{\|\hat{\Gamma}_N(V) - \Gamma(V)\|\} \leq (1-\beta)\epsilon$ uniformly for $V \in B(0, K_u/(1-\beta))$ then $E\{\|\hat{V}_N - V\|\} \leq \epsilon$ where $\hat{V}_N$ is the (random) fixed point to the random contraction

operator $\hat{\Gamma}_N$ and V is the fixed point V to Γ. By inequality (2.15) of Lemma 2.2, a total of $T(\beta,\epsilon)$ successive approximation steps are required to guarantee that the final iteration of any contraction mapping of modulus β is within ϵ of the fixed point, where $T(\beta,\epsilon)$ is given by

$$T(\beta,\epsilon) = \Theta\left(\frac{\log\left(1/(1-\beta)\epsilon\right)}{|\log(\beta)|}\right). \tag{6.26}$$

Thus, setting N to satisfy inequality (6.25) and T to satisfy inequality (2.15) guarantees that the expected error in the final iterate $\hat{V}_0 = \hat{\Gamma}_N^T(\hat{V}_T)$ satisfies:

$$E\left\{\left\|\hat{V}_0 - V\right\|\right\} \leq 2\epsilon. \tag{6.27}$$

Setting N and T to the minimal values satisfying inequalities (2.15) and (6.26) yields the upper bound on randomized complexity (4.4).

Proof of Lemma 4.1: We closely follow the proof given in Chow and Tsitsiklis 1991 for their non-random version of the multigrid algorithm, with notation adapted to the current problem formulation. Let $\hat{V}$ be the value function produced at successive approximation step $T(k-1)-1$ of iteration $k-1$ of the multigrid algorithm:

$$\hat{V} = \hat{\Gamma}_{N_{k-1}}^{T(k-1)-1}(\hat{V}_{k-2}). \tag{6.28}$$

Using the triangle inequality and inequality (4.7) we have:

$$\begin{aligned} E\left\{\left\|\hat{\Gamma}_{N_k}\left(\hat{\Gamma}_{N_{k-1}}(\hat{V})\right) - \hat{\Gamma}_{N_{k-1}}(\hat{V})\right\|\right\} &\leq E\left\{\left\|\hat{\Gamma}_{N_k}\left(\hat{\Gamma}_{N_{k-1}}(\hat{V})\right) - \Gamma\left(\hat{\Gamma}_{N_{k-1}}(\hat{V})\right)\right\|\right\} \\ &+ E\left\{\left\|\Gamma\left(\hat{\Gamma}_{N_{k-1}}(\hat{V})\right) - \hat{\Gamma}_{N_{k-1}}\left(\hat{\Gamma}_{N_{k-1}}(\hat{V})\right)\right\|\right\} \\ &+ E\left\{\left\|\hat{\Gamma}_{N_{k-1}}\left(\hat{\Gamma}_{N_{k-1}}(\hat{V})\right) - \hat{\Gamma}_{N_{k-1}}(\hat{V})\right\|\right\} \\ &\leq \frac{K}{\sqrt{N_k}(1-\beta)} + \frac{K}{\sqrt{N_{k-1}}(1-\beta)} + \frac{\beta K}{\sqrt{N_{k-1}}\beta(1-\beta)} \\ &= \frac{5K}{\sqrt{N_k}(1-\beta)}. \end{aligned} \tag{6.29}$$

Using the fact that $\hat{\Gamma}_{N_k}$ is a contraction mapping with modulus β we have:

$$E\left\{\left\|\hat{\Gamma}_{N_k}^t(\hat{V}_{k-1}) - \hat{\Gamma}_{N_k}^{t-1}(\hat{V}_{k-1})\right\|\right\} \leq \beta^{t-1} E\left\{\left\|\hat{\Gamma}_{N_k}(\hat{V}_{k-1}) - \hat{V}_{k-1}\right\|\right\} \leq \beta^{t-1}\frac{5K}{\sqrt{N_k}(1-\beta)}, \tag{6.30}$$

where the second inequality follows from (6.29) and the fact that $\hat{V}_{k-1} = \hat{\Gamma}_{N_{k-1}}(\hat{V})$. If we choose t so that $5\beta^t \leq 1$ it is easy to see from (6.30) that the stopping criterion (4.8) is satisfied. Thus it follows that the stopping rule $T(k)$ satisfies:

$$T(k) \leq \frac{\log(5)}{|\log(\beta)|}. \tag{6.31}$$

Proof of Theorem 4.2: By Lemma 4.1 at most $T(k) = \log(5)/|\log(\beta)|$ successive approximation steps are performed at any iteration of the multigrid algorithm so the total work at step k of the multigrid algorithm is of order $O(|A|N_k^2/|\log(\beta)|)$. It follows that the complexity of the multigrid algorithm is bounded by:

$$\begin{aligned} comp^{\textit{wor-ran}}(2\epsilon, d) &= O\left(\frac{|A|}{|\log(\beta)|}\left[N_{k^*}^2 + \left(\frac{N_{k^*}}{4}\right)^2 + \left(\frac{N_{k^*}}{16}\right)^2 + \cdots\right]\right) \\ &= O\left(\frac{|A|N_{k^*}^2}{|\log(\beta)|}\left[\sum_{i=0}^{\infty}\frac{1}{2^{4i}}\right]\right) \\ &= O\left(\frac{|A|N_{k^*}^2}{|\log(\beta)|}\right). \end{aligned} \tag{6.32}$$

Since N_{k^*} is chosen to satisfy (4.9) where the constant K is given in equation (4.7) simple substitution yields the form of the complexity bound in equation (4.14).

7. References

Anderson, E. Hansen, L. McGratten, E. and T. Sargent (1996) "Mechanics for Forming and Estimating Dynamic Linear Economies" in H. Amman, D. Kendrick and J. Rust (eds.) *Handbook of Computational Economics* Amsterdam, Elsevier, Chapter 4, 171–252.

Anselone, P.M. (1971) *Collectively Compact Operator Approximation Theory and Application to Integral Equations* Prentice Hall Series on Automatic Computation, Englewood Cliffs, New Jersey.

Anselone, P.M. and R. Ansorge (1981) "A Unified Framework for the Discretization of Nonlinear Operator Equations" *Numerical Functional Analysis and Optimization* **4-1** 61–99.

Bakhvalov, N.S. (1964) "On Optimal Estimates of the Rate of Convergence of Quadrature Processes and Integration Methods of the Monte Carlo Type" In the miscellany: *Numerical Methods for Solving Differential and Integral Equations and Quadrature Formulae* 'Nauka' Moscow, 5–63. (in Russian).

Barto, A.G., Bradtke, S.J. and Singh, S. (1995) "Learing to Act Using Real-Time Dynamic Programming" *Artificial Intelligence* **72** 81–138.

Bellman, R. (1957) *Dynamic Programming* Princeton University Press.

Bellman, R. and S. Dreyfus (1962) *Applied Dynamic Programming* Princeton University Press.

Bellman, R. Kalaba, R. and B. Kotkin (1963) "Polynomial Approximation: A New Technique in Dynamic Programming Allocation Processes" *Mathematics of Computation* **17** 155–161.

Bertsekas, D. (1975) "Convergence of Discretization Procedures in Dynamic Programming" *IEEE Transactions on Automatic Control* **20** 415–419.

Blackwell, D. (1965) "Discounted Dynamic Programming" *Annals of Mathematical Statistics* **36** 226–235.

Chow, C.S. and Tsitsiklis, J.N. (1989) "The Complexity of Dynamic Programming" *Journal of Complexity* **5** 466–488.

Chow, C.S. and Tsitsiklis, J.N. (1991) "An Optimal Multigrid Algorithm for Continuous State Discrete Time Stochastic Control" *IEEE Transactions on Automatic Control* **36-8** 898–914.

Daniel, J.W. (1976) "Splines and Efficiency in Dynamic Programming" *Journal of Mathematical Analysis and Applications* **54** 402–407.

Denardo, E.V. (1967) "Contraction Mappings Underlying the Theory of Dynamic Programming" *SIAM Review* **9** 165–177.

Fox, B.L. (1973) "Discretizing Dynamic Programming" *J. Opt. Theor. Appl.* **11** 228-234.

Gänßler, P. (1983) *Empirical Processes* Volume Three in Lecture Notes Series, Institute of Mathematical Statistics, Hayward, California.

Hammersley, J.J. and D.C. Handscomb (1992) *Monte Carlo Methods* Chapman and Hall, London.

Jain, N.C. and Marcus, M.B. (1975) "Central Limit Theorems for $C(S)$-valued Random Variables" *Journal of Functional Analysis* **19** 216–231.

Judd, K. (1996) "Approximation, Perturbation, and Projection Methods in Economic Analysis" in *Handbook of Computational Economics* ed. by H. Amman, D. Kendrick and J. Rust, Amsterdam, Elsevier-North Holland, Chapter 12, 511–585.

Keane, M. P. Wolpin, K. I. (1994) "The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence" *Review of Economics and Statistics* **76-4** 648–672.

Krasnosel'skii, M.A. Vainikko, G.M. Zabreiko, P.P. Rutitskii, Ya. B. and V. Ya. Stetsenko (1972) *Approximate Solution of Operator Equations* D. Louvish, translator. Wolters-Noordhoff Publishing, Groningen.

Niederreiter, H. (1992) *Random Number Generation and Quasi-Monte Carlo Methods* **63** SIAM CBMS-NSF Conference Series in Applied Mathematics, Philadelphia.

Pakes, A. and P. McGuire (1996) "Stochastic Algorithms for Dynamic Models: Markov Perfect Equilibrium and the 'Curse of Dimensionality"' manuscript, Department of Economics, Yale University.

Paskov, S. (1996) "New Methodologies for Valuing Securities" S. Pliska and M. Dempster (eds.) Isaac Newton Institute, Cambridge, England.

Paskov, S. and J.F. Traub (1996) "Faster Evaluation of Financial Derivatives" forthcoming, *Journal of Portfolio Management.*

Pollard, D. (1984) *Convergence of Stochastic Processes* Springer-Verlag, New York.

Pollard, D. (1989) "Asymptotics via Empirical Processes" *Statistical Science* **4-4** 341–386.

Porteus, E. L. (1980) "Overview of Iterative Methods for Discounted Finite Markov and Semi-Markov Decision Chains" in R. Hartley et. al. (eds.) *Recent Developments in Markov Decision Processes,* Academic Press.

Puterman, M. (1990) "Markov Decision Processes" in D.P. Heyman and M.J. Sobel (eds.) *Handbooks in Operations Research and Management Science* Volume 2, Amsterdam, Elsevier.

Puterman, M. (1994) *Markov Decision Processes* Wiley, New York.

Puterman, M.L. and Brumele, S. (1979) "On the Convergence of Policy Iteration in Stationary Dynamic Programming" *Mathematics of Operations Research* **4** 60–69.

Rall, L.B. (1969) *Computational Solution of Nonlinear Operator Equations* Wiley, New York.

Roth, K.F. (1954) "On Irregularities of Distribution" *Mathematika* **1** 73–79.

Rust, J. (1987) "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher" *Econometrica* **55-5** 999–1033.

Rust, J. (1988) "Maximum Likelihood Estimation of Discrete Control Processes" *SIAM Journal on Control and Optimization* **26-5** 1006–1023.

Rust, J. (1994) "Structural Estimation of Markov Decision Processes" chapter 51 in D. McFadden and R. Engle (eds.) *Handbook of Econometrics* **4** North Holland, 3082–3139.

Rust, J. (1996) "Numerical Dynamic Programming in Economics" in H. Amman, D. Kendrick, and J. Rust (eds.) *Handbook of Computational Economics* Elsevier-North Holland, Amsterdam, Chapter 14, 619–729.

Santos, M. and J. Vigo (1996) "Analysis of Error for a Numerical Programming Algorithm Applied to Economic Models" manuscript, ITAM, Mexico.

Shorack, G.R. and Wellner, J.A. (1986) *Empirical Processes with Applications to Statistics* Wiley, New York.

Skorohod, A.V. (1984) *Random Linear Operators* D. Reidel Publishing Company, Dordrecht.

Tauchen, G. Hussey, R. (1991) "Quadrature-Based Methods for Obtaining Approximate Solutions to Nonlinear Asset Pricing Models" *Econometrica* **59-2** 371–396.

Traub, J.F. and H.W. Woźniakowski (1994) "Breaking Intractability" *Scientific American* **270-1** 102–107.

Traub, J.F. Wasilkowski, G.W. and Woźniakowski, H. (1988) *Information-based Complexity* Academic Press.

Traub, J.F. and Woźniakowski, H. (1980) *A General Theory of Optimal Algorithms* Academic Press, New York.

Traub, J.F. and Woźniakowski, H. (1992) "The Monte Carlo Algorithm with a Psuedorandom Generator" *Mathematics of Computation* **58-197** 323–339.

Tsitsiklis, J.N. (1994) "Asynchronous Stochastic Approximation and Q-Learning" *Machine Learning* **16** 185–202.

Werschulz, A.G. (1991) *The Computational Complexity of Differential and Integral Equations* Oxford University Press, New York.

Woźniakowski, H. (1991) "Average Case Complexity of Multivariate Integration" *Bulletin of the American Mathematical Society* **24** 185–194.

Woźniakowski, H. (1992) "Average Case Complexity of Linear Multivariate Problems" *Journal of Complexity* **8** (Parts I and II) 372–392.

Yudin, D.B. and A.B. Nemirovsky (1976a) "Estimating the Computational Complexity of Mathematical Programming Problems" *Ekonomika i matem. metody* **XII-1** 128–142. (in Russian).

Yudin, D.B. and A.B. Nemirovsky (1976b) "Computational Complexity and Efficiency of Methods for Solving Convex Extremum Problems" *Ekonomika i matem. metody* **XII-2** 357–369. (in Russian).

Yudin, D.B. and A.B. Nemirovsky (1977) "Efficiency of Random Search in Control Problems" *Izvestiya Akad. Nauk SSSR, tekhnischeskaya kibernetika* **3** 3–17. (in Russian).