February 1988 LIDS-P-1751

An Optimal Multigrid Algorithm for Continuous State Discrete Time Stochastic Control[1]

Chee-Seng Chow[2]
John N. Tsitsiklis[2]

[1]Research supported by an NSF PYI award, with matching funds from Bellcore Inc., and by the ARO under grant DAAL03-86-K-0171

[2]Dept. of Electrical Engineering and Computer Science, Massachussets Institute of Technology, Cambridge, MA 02139.

Abstract

This paper studies the application of multigrid methods to a class of discrete time, continuous state, discounted, infinite horizon dynamic programming problems. We analyze the computational complexity of computing the optimal cost function to within a desired accuracy of ϵ, as a function of ϵ and of the discount factor α. Using an advesary argument, we obtain lower bound results on the computational complexity for this class of problems. We also provide a multigrid version of the successive approximation algorithm whose computational requirements are (as a function of α and ϵ) within a constant factor from the lower bounds when a certain *mixing* condition is satisfied (hence the algorithm is optimal).

1 Introduction

In this paper, we analyze how the computational effort in computing an approximation to the optimal cost function of a discounted dynamic programming problem depends on the discount factor and the accuracy of the approximation, and how multigrid ideas can be used to improve the running time of a successive approximation algorithm in dynamic programming.

To illustrate the issues, consider a system with state space $S = [0, 1]$. At the beginning of each time period, the state of the system is measured exactly. Based on the observed state, a decision, out of N possible choices, is made. Then, within the time period a (bounded) cost, is incurred and the system probabilistically changes state. The cost and the one-step transition probabilities are functions of only the current state and the current decision. This process is repeated in each subsequent time period. Future costs are discounted by a discount factor $\alpha \in (0, 1)$. The goal is to find the minimum long term (infinite horizon) total cost as a function of the initial state, and when the best decision is made at every stage. This cost function is called the *optimal cost function.*

The above problem is a discrete time, infinite horizon, perfectly observable, Markov Decision Problem (MDP), with discounted cost (e.g. see [Hartl 80]). It can be shown under certain assumptions that the optimal cost function exists and is unique ([Denar 67]). Moreover if the cost and the transition probability density functions are Lipschitz continuous, then for any $\epsilon > 0$, one can obtain a (piecewise constant) function which is within ϵ of the optimal cost function, viz. an *ϵ-optimal cost function.* This is done by discretizing the continuous problem to get another discounted cost problem with finite state space and solving for the optimal cost function of this discretized problem. There are explicit bounds on how fine the discretization should be in order to achieve the desired accuracy ([Whitt 78] and [Whitt 79]).

For the discretized discounted cost problem, there are many iterative algorithms for computing the optimal cost functon, e.g. successive approximation, policy iteration, and

linear programming. For any $\epsilon > 0$, one can compute an ϵ-optimal cost function for the discounted cost problem using a finite number of arithmetic operations. Intuitively, it is clear that as ϵ decreases to 0 (the more accurate we want the answer), the finer we have to discretize the continuous problem, so more computation is needed. Similarly, as the discount factor tends to 1, the discretization error gets amplified more, and a finer discretization is needed. We want to know how the computational effort depends on the accuracy desired and the discount factor.

Multigrid algorithms use two or more grid levels (see [Hackb 81], [Hackb 85],[Brand 86]), with different iterations taking place on different grids. For many practical problems, the multigrid version, if properly implemented, converges significantly faster. In some cases, by using an appropriate model of computation the resultant algorithm can be shown to be *optimal*, i.e. the running time of the algorithm is at worst within a constant factor of the fastest algorithm possible.

The model of computation we use is a ***real number computer***: a machine that operates on real numbers with the four basic arithmetic operations and the three comparison tests. Manipulating the individual digits of a number is disallowed. This model of computation has been used by other researchers (e.g. see [Traub 80] and [Traub 83]). The complexity (total work) of a computation is the sum of the work in ***reading the input*** (e.g. a unit work per input) and the work in ***computing the output*** (e.g. a unit work per operation). Using an adversary argument, lower bounds on the complexity of a problem can be proved with this model.

We will show how multigrid methods can be incorporated into a successive approximation algorithm in dynamic programming to obtain significant improvement in the running time. When a certain ***mixing condition*** is satisfied, the running time of the multigrid version is within a constant factor of the lower bound on complexity. Hence the algorithm is optimal. (This is in contrast to the single grid successive approximation algorithm.)

The purpose of this paper is to illustrate the key issues involved and the central ideas. To this effect, we restrict to the special case where the state space is one-dimensional and the control space is finite. Results for more general cases are previewed in Section 5 and will be reported in detail elsewhere.

2 Problem Formulation

2.1 Description of The Model

Consider the discounted dynamic programming problem introduced in Section I. It has state space $S = [0, 1]$, and control space $C = \{1, 2, \ldots, N\}$, which represents all allowed actions. A function $\mu : S \mapsto C$ is called a stationary policy and prescribes the action $\mu(x)$ whenever

the system is in state x. Let Π be the set of all stationary policies.[1]

The dynamics of the system P are described by a non-negative function on $S \times S \times C$. For each $x \in S, u \in C$, $P(\cdot|x,u)$ is a probability density on S. Moreover, $P(y|x,u)dy$ is the probability that the next state lies between $y - \frac{dy}{2}$ and $y + \frac{dy}{2}$ given that the current state is x and control u is applied. We are also given a bounded continuous function, $g : S \times C \mapsto \mathbf{R}$, called the cost per stage. In particular, $g(x,u)$ is interpreted as the immediate cost if the state of the system is x and action u is applied. Let α be a constant belonging to $(0,1)$ called the discount factor. The cost incurred each time period later is discounted by α.

The cost $J_\mu : S \mapsto \mathbf{R}$ corresponding to a stationary policy μ is defined by

$$J_\mu(x_0) = E\left[\sum_{n=0}^{\infty} \alpha^n g(x_i, \mu(x_i))\right],$$

where $x_0, x_1, \ldots$ is the random trajectory generated when the initial state is x_0 and policy μ is used. The optimal cost function $J_* : S \mapsto \mathbf{R}$ is defined by

$$J_*(x_0) = \inf_\mu J_\mu(x_0), \ \forall x_0 \in S.$$

The problem we consider is the following: Given an $\epsilon > 0$, we would like to find a J_*' such that $\|J_* - J_*'\|_\infty \leq \epsilon$, where $\|\cdot\|_\infty$, is the supremum norm defined as follows:

$$\|J\|_\infty = \sup_{x \in S} |J(x)|.$$

In addition to $\|\cdot\|_\infty$, we will also use the following quasi-norm:

$$\|J\|_Q = \sup_{x \in S} J(x) - \inf_{x \in S} J(x).$$

The discounted MDP is specified by $(\alpha, \epsilon, S, C, g, P,)$.

2.2 Assumptions

To solve the discounted MDP, we need to make some Lipschitz continuity assumptions about g and P, and, without loss of generality, with Lipschitz constant 1.

Assumption 2.1 *For all $x, x', y, y' \in S, u, u' \in C$, the following hold*

1. $|g(x,u) - g(x',u)| \leq |x - x'|$
2. $\int_{y \in S} |P(y|x,u) - P(y|x',u)|dy \leq |x - x'|$

[1]For the purposes of this paper, measurability issues are ignored. In fact, under the smoothness assumptions to be imposed in Section 2.2, measurability issues are easily handled.

3. $|P(y|x,u) - P(y'|x,u)| \leq |y - y'|$

The *accessibility rate* ρ of the problem is defined as follows:

$$\rho = \int_S \min_{x \in S, u \in C} P(y|x,u)dy > 0.$$

If $\rho > 0$, we say that the problem satisfies a *1-stage accessibility condition.*

Assumption 2.2 *There holds* $\rho > 0$.

This is our *mixing condition.* Intuitively, it means the existence of a set of states such that under all policies and from all states there is some minimum, non-zero, probability of reaching those states. (This assumption is actually stronger than necessary for our results.)

2.3 Dynamic Programming Equation

Let $\mathcal{B}(S)$ denote the set of all bounded continuous functions on S. Under **Assumption 2.1** the minimizing operator T, defined by

$$TJ(x) = \min_{u \in C}\{g(x,u) + \alpha \int_{y \in S} P(y|x,u)J(y)dy\},$$

can be shown to map $\mathcal{B}(S) \mapsto \mathcal{B}(S)$. Moreover, it is a contraction operator ($\|TJ - TJ'\|_\infty \leq \alpha\|J - J'\|_\infty$). Since $\mathcal{B}(S)$ is complete, T has a unique fixed point in $\mathcal{B}(S)$ and the fixed point of T is J_*([Denar 67]). Hence, the optimal cost function J_* exists and is the unique solution to the dynamic programming equation

$$TJ = J.$$

2.4 Error Bounds and Successive Approximation

From the contraction property of T, starting with any initial estimate $J \in \mathcal{B}(S)$, one can obtain an approximation to J_* by successive application of the T operator on J. Moreover the following inequalities between J_* and $T^k J$ (T^k is the k-times composition of T) are known ([Berts 87]):

$$\begin{aligned} J_* &\leq T^k J + \frac{\alpha}{1-\alpha} \max_{x \in S}\{(T^k J - T^{k-1}J)(x)\}, \\ J_* &\geq T^k J + \frac{\alpha}{1-\alpha} \min_{x \in S}\{(T^k J - T^{k-1}J)(x)\}. \end{aligned}$$

After k iterations, if we let

$$J_k = T^k J + \frac{\alpha}{1-\alpha} \frac{\min_x\{(T^k J - T^{k-1}J)(x)\} + \max_x\{(T^k J - T^{k-1}J)(x)\}}{2},$$

we obtain the following error bound between J_* and J_k:

$$\|J_* - J_k\|_\infty \leq \frac{\alpha}{2(1-\alpha)} \|T^k J - T^{k-1} J\|_Q. \tag{1}$$

Under **Assumption 2.2**, it can be shown that we get a contraction factor independent of α when the quasi-norm is used:

$$\begin{aligned} \|T^{k+1} J - T^k J\|_Q &\leq \alpha(1-\rho)\|T^k J - T^{k-1} J\|_Q \\ &\leq \alpha^k (1-\rho)^k \|TJ - J\|_Q \\ &\leq (1-\rho)^k \|TJ - J\|_Q. \end{aligned}$$

Combining with (1), we obtain the following error bound equation (c.f. [Odoni 69]):

$$\|J_* - J_k\|_\infty \leq \frac{(1-\rho)^{k-1}}{2(1-\alpha)} \|TJ - J\|_Q, \tag{2}$$

which is the basis of the successive approximation algorithms in Section 3.

2.5 Discrete Dynamic Programming Equation

The continuous state problem can be discretized to give a discrete state problem. The discrete problem leads to discrete analogs to the T operator, the dynamic programming equation, and the error bound equations. There are many ways of discretizing S. For simplicity, but without loss of generality, we will use uniform discretizations. For any positive integer k, let $S^{\delta_k} = \{0, \frac{1}{2^k}, \frac{2}{2^k}, \ldots, 1\}$ and $\delta_k = \frac{1}{2^k}$. We call δ_k the diameter of discretization. Each k represents a different level of discretization. To de-emphasize the dependence on a particular grid level, we use δ and S^δ instead. For convenience, δ will also denote the discretization.

Given a discretization of the state space, the cost per stage and the dynamics can be discretized. Let their discretizations be denoted by $\tilde{g}$, and $\tilde{P}$ respectively. The cost function is discretized by pointwise sampling of g and the dynamics are discretized by pointwise sampling of P except that it has to be normalized to ensure that $\tilde{P}(\cdot|\tilde{x}, u)$ remains a probability measure on S. The minimizing operator for the discrete problem, $T^\delta : \mathcal{B}(S^\delta) \mapsto \mathcal{B}(S^\delta)$, is defined below:

$$T^\delta J^\delta(\tilde{x}) = \min_{u \in C} \{\tilde{g}(\tilde{x}, u) + \alpha \sum_{\tilde{z} \in S^\delta} \tilde{P}(\tilde{z}|\tilde{x}, u) J^\delta(\tilde{z})\}, \forall \tilde{x} \in S^\delta.$$

Let J_*^δ denote the optimal cost function for the discrete problem. It can be shown [Berts 87] that J_*^δ exists and is the unique solution of the discrete dynamic programming equation:

$$T^\delta J^\delta = J^\delta.$$

Since there are discrete analogs to all of the error bounds discussed earlier (the definitions of norms and accessibility condition are easily adapted to the discrete case), one can find an approximation to J_*^δ by using the discrete successive approximation algorithm. It remains to bound the discretization error between J_*^δ and J_*.

2.6 Discretization Error Bound

If the continuous problem has an accessibility rate 2ρ, then it can be shown that by choosing a fine enough discretization the discrete problem also satifies an accessibility condition with accessibility rate at least ρ. So we may as well assume that the continuous problem and its discretized problems (for all discretizations δ_1,δ_2, ...), have an accessibility rate at least ρ. Let K, K', and K'' be some constants independent of α and δ. It can be shown that for all δ,

$$\|J_*^\delta\|_Q \leq \frac{K}{1-\alpha(1-\rho)} \leq K'.$$

By an application of Theorem 6.1 of [Whitt 78] we can bound the difference between J_*^δ and J_* by

$$\|J_* - e(J_*^\delta)\|_\infty \leq \frac{K''\delta}{1-\alpha}, \tag{3}$$

where $e(J_*^\delta)$ denotes the extrapolation of J_*^δ to a piecewise constant function on S (e.g. $e(J_*^{\delta_k})(x) = J_*^{\delta_k}(\frac{2^i}{2^k})$ if $x \in (2^{i-1}, 2^i]$, and $e(J_*^{\delta_k})(0) = J_*^{\delta_k}(0)$).

3 Multigrid Successive Approximation

In this section, we derive upper bounds on the complexity of computing an ϵ-optimal cost function. We are only interested in the dependence on α and ϵ, for α bounded away from 0 and in the limit $\epsilon \downarrow 0$. As we will see, the diameter of discretization δ must also depend on ϵ, so we will keep track of δ as well. We will use the order of magnitude notation defined as follows: If f and g are non-negative functions of α and ϵ (or δ), and there exists some $\epsilon_0 > 0$ and constant $c > 0$ such that $f(\alpha,\epsilon) \leq cg(\alpha,\epsilon)$, for all $\epsilon < \epsilon_0$, we write $f = \mathrm{O}(g)$ or equivalenly, $g = \Omega(f)$. For example, from what we have shown, $\|J_*^\delta\|_Q = \mathrm{O}(1)$ and $\|J_* - e(J_*^\delta)\|_\infty = \mathrm{O}(\frac{\delta}{1-\alpha})$. Throughout this section **Assumptions 2.1** and **2.2** are in effect.

3.1 Complexity of Single-Grid Successive Approximation

To compute an ϵ-optimal cost function, we need to pick a proper grid size so that

$$\|J_* - e(J_*^\delta)\|_\infty \leq \epsilon/2,$$

and on that grid we use successive approximation for k iterations to find J_k^δ where

$$\|J_*^\delta - J_k^\delta\|_\infty \leq \epsilon/2.$$

Then it follows from the Triangle inequality that

$$\begin{aligned} \|J_* - e(J_k^\delta)\|_\infty &\leq \|J_* - e(J_*^\delta)\|_\infty + \|e(J_*^\delta) - e(J_k^\delta)\|_\infty \\ &= \|J_* - e(J_*^\delta)\|_\infty + \|J_*^\delta - J_k^\delta\|_\infty \\ &\leq \epsilon, \end{aligned}$$

and $e(J_k^\delta)$ is an ϵ-optimal cost function.

From the discretization error bound (3), it is necessary and sufficient to have $\frac{\delta}{1-\alpha} = k\epsilon$, for some constant k. Hence, $\frac{1}{\delta} = \mathrm{O}(\frac{1}{(1-\alpha)\epsilon})$.

With this discretization, the number of grid points, $|S^\delta|$ is $\frac{1}{\delta}$. So, for each fixed $\tilde{x} \in S^\delta$, in order to compute the expression

$$\min_{u \in C}\{\tilde{g}(\tilde{x}, u) + \alpha \sum_{\tilde{z} \in S^\delta} \tilde{P}(\tilde{z}|\tilde{x}, u) J^\delta(\tilde{z})\}, \tag{4}$$

the number of arithmetic operations required is $\mathrm{O}(\frac{1}{\delta})$. Here the computation of the summation term dominates the total operation count.

To perform one iteration of the successive approximation algorithm, we need to compute (4) for $|S^\delta|$ points. So, the number of arithmetic operations per iteration is $\mathrm{O}((\frac{1}{\delta})^2) = \mathrm{O}((\frac{1}{(1-\alpha)\epsilon})^2)$.

If we do k iterations of the successive approximation algorithm, then using the error bound (2), in order to have $\|J_*^\delta - J_k^\delta\|_\infty \leq \frac{\epsilon}{2}$, it suffices to choose

$$k = \mathrm{O}(\log(\tfrac{1}{(1-\alpha)\epsilon})/|\log(1-\rho)|).$$

Since ρ is a constant, the total number of iterations needed is $\mathrm{O}(\log(\frac{1}{(1-\alpha)\epsilon}))$. Hence in order to compute an ϵ-optimal cost function using the usual single-grid successive approximation, the total number of arithmetic operations is $\mathrm{O}((\log \frac{1}{(1-\alpha)\epsilon})(\frac{1}{(1-\alpha)\epsilon})^2)$. We will show below that by using a multigrid method, the total number of arithmetic operations can be reduced to $\mathrm{O}((\frac{1}{(1-\alpha)\epsilon})^2)$.

3.2 Complexity of Multigrid Successive Approximation

To use the multigrid method, a series of successively finer grids $S^{\delta_1}, S^{\delta_2}, \ldots, S^{\delta_k}$ is used, where S^{δ_k} is the first grid level which satisfies $\|J_* - e(J_*^{\delta_k})\|_\infty \leq \frac{\epsilon}{2}$. The multigrid successive approximation algorithm is described below:

1. Start at δ_1, and obtain some J^{δ_1}. The work done on this grid level is assumed negligible.

2. Having computed J^{δ_i}, if $i = k$ then stop. Else extrapolate J^{δ_i} to the next finer grid $S^{\delta_{i+1}}$ and perform $O(\frac{1}{|\log(1-\alpha)\rho|})$ iterations of the successive approximation algorithm.

It remains to verify that by doing $O(\frac{1}{|\log(1-\alpha)\rho|})$ iterations on each grid level, the function J^{δ_k} produced by the algorithm satisfies $\|J_* - e(J^{\delta_k})\|_\infty \leq \epsilon$. (Recall that ρ is a constant, so we are saying that it suffices to do a constant number of iterations on each grid level.) But here is an informal argument: Observe that the discretization error (between J_* and J_*^δ) depends linearly on δ (see (2)). So by going from δ_i to δ_{i+1} the discretization error is reduced by half. But, by iterating $O(\log 0.5/|\log(\alpha(1-\rho))|)$ times we also reduce the error $\|J_*^{\delta_{i+1}} - J^{\delta_{i+1}}\|_\infty$ by half. (Because of extrapolation errors, a few more iterations may be needed.) So we go to the next grid level when it is no longer effective to iterate on the same grid, and we do that after some constant number of iterations.

To analyze the complexity of the multigrid algorithm, recall from earlier analysis that the total number of arithmetic operations on grid level δ_r is $O((\frac{1}{\delta_r})^2)$. Hence

$$\begin{aligned}
\text{Total number of arithmetic operations} &= O([(\tfrac{1}{\delta_k})^2 + (\tfrac{1}{\delta_{k-1}})^2 + (\tfrac{1}{\delta_{k-2}})^2 \cdots]) \\
&= O((\tfrac{1}{\delta_k})^2[1 + (\tfrac{\delta_k}{\delta_{k-1}})^2 + (\tfrac{\delta_k}{\delta_{k-2}})^2 \cdots]) \\
&= O((\tfrac{1}{\delta_k})^2[1 + (\tfrac{1}{2})^2 + (\tfrac{1}{4})^2 + (\tfrac{1}{8})^2 \cdots]) \\
&= O((\tfrac{1}{\delta_k})^2) \\
&= O((\tfrac{1}{(1-\alpha)\epsilon})^2).
\end{aligned}$$

4 Lower Bound Results

In this section we show that no algorithm exists which produces an ϵ-optimal function J with fewer than $O((\frac{1}{(1-\alpha)\epsilon})^2)$ operations.

For a fixed S and C, an instance of the discounted MDP is a tuple (α, ϵ, g, P) where g and P satisfy **Assumptions 2.1** and **2.2**. The problem is given any instance compute a J such that $\|J - J_*\|_\infty \leq \epsilon$. The model of computation is the real number computer discussed earlier. The computer is given the values of α and ϵ, but has to sample (determine) the values of g and P by asking an oracle. For example, when given the tuple (x, u), the oracle will return to the computer the value $g(x, u)$, and when given (y, x, u) the oracle will return $P(y|x, u)$.

Using an adversary type argument ([Traub 80] and [Traub 83]), we find a lower bound complexity on the discounted MDP by lower-bounding the number of questions the computer must ask the oracle. In particular, we will show that no matter what algorithm is used, the computer must sample $\Omega((\frac{1}{(1-\alpha)\epsilon})^2)$ points of P. The proof is based on the "adversary" technique. Here is an outline of the proof: Consider an instance of the problem (α, ϵ, g, P). Let X be the set of points of P sampled by the computer. We will show that unless $|X| = \Omega((\frac{1}{(1-\alpha)\epsilon})^2)$, an adversary can construct another instance $(\alpha, \epsilon, g, P')$ (as a function of X) for which P' agrees on X with P but such that the optimal cost functions of the two problems differ by more than 2ϵ at some point $x \in S$. Based on the points sampled, the computer cannot possibly differentiate the two problems and so whatever J the computer computes, the adversary can pick the instance for which J is not its ϵ-optimal cost function.

The computer may sample g and P adaptively or non-adaptively. In the former, the choice of the current point sampled may depend on the past values of previous samples, in the latter, the choice of the current point sampled doesn't depend on the values obtained earlier. The lower bound results below apply to an adaptive computer (and, in particular a non-adaptive computer as well).

Notice that our multigrid algorithm is non-adaptive because the points at which g and P are sampled are predetermined. Thus our results establish that, for the problem considered, adaptation does not help.

4.1 Lower Bound Example

Let $S = [0, 1]$ and $C = \{1\}$ (so the control can be ignored). Consider the following problem instance (α, ϵ, g, P), where for all $x, y \in S$, $g(y) = y$ and $P(y|x) = 1$. Let its optimal cost function be denoted by J_*. (Note that **Assumptions 2.1** and **2.2** are trivially satisfied.) At the onset, we let the computer know g for free (e.g. it can sample g without penalty) and whenever the computer samples P it obtains the value 1.

Let X be the set of points at which P is sampled by the computer while it solves the above described instance. We will show unless $|X| = \Omega((\frac{1}{(1-\alpha)\epsilon})^2)$, we can construct another instance $(\alpha, \epsilon, g, P')$ with optimal cost function J_*', such that $\|J_*' - J_*\|_\infty \geq 2\epsilon$, and such that $P(y|x) = P'(y|x)$, $\forall (x, y) \in X$.

Let $P'(y|x) = P(y|x) + E(x, y)$, where $E(x, y)$ is the "perturbative" term that depends on X. One may view P, P', and E as real-valued functions on the unit square $\{(x, y) : 0 \leq x, y \leq 1\}$ and X is a set of points on this square. For reference, let the horizontal axis be the x-axis and the vertical axis be the y-axis. We now construct $E(x, y)$.

Let $\delta \in (0, 1)$ be a small constant to be determined later. Without loss of generality, assume that $\frac{1}{\delta}$ is an integer. Partition the square into $(\frac{1}{\delta})^2$ cells of dimensions $\delta \times \delta$. If a cell contains one or more points of X, it is said to be sampled, otherwise it is unsampled.

On the sampled cells, E takes the value 0, whereas on the unsampled cells the value of E is assigned as follows:

Consider a column of cells, i.e. those with the same x coordinates. (We will refer to cells by the coordinates of their centers.) We shall focus on the unsampled cells in this column. Divide the unsampled cells into two equal portions according to their y coordinates (ignoring any leftover cell). The first portion consists of those with y coordinates smaller than the y coordinates of those in the second portion. To each cell in the first portion attach a pyramid of height $\frac{\delta}{2}$ with base $\delta \times \delta$ fitted exactly onto the cell. For the second portion attach similar pyramids but of height $-\frac{\delta}{2}$, i.e. an inverted pyramid. We do the same for every column. On an unsampled cell, the value of E at the point (x, y) is given by the height of the face of the pyramid at point (x, y), where $E(x, y)$ takes on a positive (or negative value) if the corresponding pyramind is upright (or inverted).

It is clear that with this construction of E, P' satisfies **Assumption 2.2**, because $P'(y|x)$ is bounded below by a positive constant. To verify that P' satisfies **Assumption 2.1** it is sufficient to show that E satisfies that assumption. It is clear that for all x, $|E(x, y) - E(x, y')| \leq |y - y'|$. This is because by construction, for any fixed x, the slope of $E(x, y)$ as a function of y is one of -1, 0, or 1. (Observe that each face of a pyramid makes a 45 degree angle with the base plane.) And, it is clear that for all y, $|E(x, y) - E(x', y)| \leq |x - x'|$. So $\int_0^1 |E(x, y) - E(x', y)|dy \leq |x - x'|$. It now remains to show that $\|J_* - J_*'\|_\infty$ is sufficiently large.

Let $\beta \in (0, \frac{1}{2})$, and suppose that $|X| \leq \beta(\frac{1}{\delta})^2$. Then there must exist at least $\frac{1}{2\delta}$ columns with no more than $\frac{2\beta}{\delta}$ sampled cells in each column. This means that there are at least $\frac{1}{2\delta}$ columns each with at least $\frac{(1-2\beta)}{\delta}$ unsampled cells . Let Z be the set of all x coordinates of these columns. Let

$$g_1(x) = \int_{y=0}^{1} E(x, y)g(y)dy.$$

It is clear that $g_1(x) \leq 0$, for all x. Moreover, $g_1(x) \leq -k'\delta$, for all $x \in Z$, where k' is some positive constant.

Writing out J_* and J_*' explicitly,

$$\begin{aligned} J_*(x) &= g(x) + \alpha\frac{1}{2} + \alpha^2\frac{1}{2} + \cdots \\ J_*'(x) &= g(x) + \alpha\{\frac{1}{2} + g_1(x)\} + \alpha^2\{\frac{1}{2} + \int_z P'(z|x)g_1(z)dz\} + \cdots \end{aligned}$$

And using the property of g_1 noted earlier, it can be shown that

$$\begin{aligned} \int_z P'(z|x)g_1(z)dz &= \int_z P(z|x)g_1(z)dz + \mathrm{O}(\delta^2) \\ &\leq -k''\delta + \mathrm{O}(\delta^2), \end{aligned}$$

for some positive constant k''. Doing a term by term comparison between J_* and J_*', for all x, we obtain

$$J_*(x) - J_*'(x) \geq \alpha g_1(x) + \alpha^2\{k''\delta - \mathrm{O}(\delta^2)\} + \alpha^3\{k''\delta - \mathrm{O}(\delta^2)\} + \cdots$$

Ignoring $\mathrm{O}(\delta^2)$ terms, we have

$$\begin{aligned} \|J_* - J_*'\|_\infty &\geq \alpha k'\delta + \alpha^2 k''\delta + \alpha^3 k''\delta + \cdots \\ &\geq k''\frac{\alpha\delta}{1-\alpha} \\ &\geq k\frac{\delta}{1-\alpha}, \end{aligned}$$

where k is a positive constant. By choosing $\delta = \frac{2(1-\alpha)\epsilon}{k}$, we note that if $|X|$ is not more than $\beta(\frac{1}{\delta})^2 = \Omega((\frac{1}{(1-\alpha)\epsilon})^2)$, then $\|J_* - J_*'\|_\infty \geq 2\epsilon$. Hence the lower bound is established and we conclude that the multigrid algorithm is optimal.

5 Extensions and Summary

The above described results can be generalized in numerous directions. Here are some extensions:

1. The state space S can be any bounded subset of $\mathbf{R}^n$, and the control space C any bounded subset of $\mathbf{R}^m$. Moreover, for each state there may be constraints on the allowed actions, i.e. $u \in C_x \subset C$.

2. The Lipschitz Continuity assumptions on P can be relaxed to handle piecewise continuous probability densities.

3. General discretization procedures (based on [Whitt 78]) may be incorporated into the multigrid algorithm, so grids may be non-uniform.

4. Similar results can be obtained for the case when accessibility condition is not assumed.

The table below summarizes the complexity results for computing an ϵ-optimal cost function (under Lipschitz continuity assumptions of g and P in x, u, and y variables). Complexity results for single-grid and multigrid successive approximations, with and without accessibility condition are tabulated. Let $d = 2n+m$ (where n and m are the dimensions of S and C respectively).

	With Accessibility	No Accessibility
Single Grid	$\log(\frac{1}{(1-\alpha)\epsilon})(\frac{1}{(1-\alpha)\epsilon})^d$	$\frac{\log(\frac{1}{(1-\alpha)\epsilon})}{\lvert\log\alpha\rvert}(\frac{1}{(1-\alpha)^2\epsilon})^d$
Multi Grid	$(\frac{1}{(1-\alpha)\epsilon})^d$	$\frac{1}{\lvert\log\alpha\rvert}(\frac{1}{(1-\alpha)^2\epsilon})^d$
Lower Bound	$(\frac{1}{(1-\alpha)\epsilon})^d$	$(\frac{1}{(1-\alpha)^2\epsilon})^d$

For computing an ϵ-optimal cost function, multigrid successive approximation is in general nearly optimal (within a factor of $|\log\alpha|^{-1}$), so any other algorithms, e.g. policy iteration, linear programming, etc. cannot have much better complexity.

Can we have faster algorithms if g and P are *smooth* (C^∞ functions) so that algorithms can use directional information from the derivatives? The answer is no. Eventhough the lower bound results are constructed using functions with sharp edges and corners, these singularities can be rounded to give smooth functions without significantly affecting the lower bounds. However, faster algorithms may be possible, if there are smoothness bounds on higher derivatives.

Other extensions we are considering are relaxing the accessibility condition to some form of ergodicity condition (we have obtained similar results for a k-stage accessibility condition instead of the 1-stage condition discussed), different formulations of the problem so that deterministic dynamic programming programming problems can be included, and different models of computation.

6 References

[Hartl 80] Hartley, R. , L. C. Thomas, and D. J. White (eds.) 1980 *Recent Developments in Markov Decision Processes* Academic Press.

[Denar 67] Denardo, E. V. 1967 Contraction Mappings in The Theory Underlying Dynamic Programming *SIAM Review* vol 9, 165 – 177.

[Whitt 78] Whitt, W. 1978 Approximations of Dynamic Programs I *Mathematics of Operations Research* vol. 3, 231 –243.

[Whitt 79] Whitt, W. 1979 Approximations of Dynamic Programs II *Mathematics of Operations Research* vol. 4, 179 –185.

[Traub 80] Traub, J. F. and H. Wozniakowski 1980 *A General Theory of Optimal Algorithm* Academic Press.

[Traub 83] Traub, J. F. and Wasilowski, G. W. and Wozniakowski, H. 1983 *Information, Uncertainty, Complexity* Addison-Wesley Publishing Company.

[Hackb 81] Hackbusch, W. and U. Trottenberg (eds.): Multigrid Methods (Proceedings, Koln-Proz 1981). *Lecture Notes in Math.* **960**, Springer-Verlag.

[Hackb 85] Hackbusch, W. 1985 *Multi-Grid Methods and Applications* Springer-Verlag.

[Brand 86] Brandt, A. 1986 Multi-level approaches to large scale problems. *Proceedings of International Congress of Mathematicians* (Berkeley, California, August 1986)

[Berts 87] Bertsekas, D. P. 1987 *Dynamic Programming: Deterministic and Stochastic Models* Prentice-Hall, Englewood Cliffs, New Jersey.

[Odoni 69] Odoni, A. R. 1969 On finding the maximal gain for Markov decision processes *Operation Res.* vol 17, p. 857–860.