

Aggregation in Stochastic Dynamic Programming

Theodore J. Lambert III
Marina A. Epelman
Robert L. Smith

Technical Report number 04-07
July 29, 2004

University of Michigan
Industrial and Operations Engineering
1205 Beal Avenue
Ann Arbor, MI 48109

Aggregation in Stochastic Dynamic Programming

Theodore J. Lambert III, Marina A. Epelman, Robert L. Smith ^{*†‡}

July 29, 2004

Abstract

We present a general aggregation method applicable to all finite-horizon Markov decision problems. States of the MDP are aggregated into macro-states based on a pre-selected collection of “distinguished” states which serve as entry points into macro-states. The resulting macro-problem is also an MDP, whose solution approximates an optimal solution to the original problem. The aggregation scheme also provides a method to incorporate inter-period action space constraints without loss of the Markov property.

1 Introduction

One of the methods for mitigating the difficulty of solving large and complex optimization problems, including large-scale dynamic programming problems, is model aggregation, where a problem is replaced by a smaller, more tractable model obtained by techniques such as combining data, merging variables, limiting the amount of information relied on in the decision-making, etc. For some of these approaches, solution to the aggregated problem needs to be disaggregated to obtain a solution to the original problem. See [10] for a broad survey of aggregation methods in optimization.

State aggregation in Markov decision processes (MDPs) is an approach involving partitioning the state space into larger “macro-states.” Aggregation of states can result in a significant reduction in the size of the state space. However, it also results in loss of information, and a tradeoff exists between the quality of the resulting solution and the reduction in the computational and storage requirements. Also, aggregation in general does not preserve the Markov property of the underlying process. Some of the work in applying aggregation to MDPs utilizes iterative aggregation and disaggregation techniques where the methods are used as accelerators for existing iterative procedures ([10]). For example, in [2] aggregation is used within successive approximation steps of a policy iteration algorithm to reduce the computation time for these iterations, while [12]

^{*}This research is supported in part by the NSF under Grants DMI-9713723, DMI-9900267, and DMI-0217283; MURI Grant ARO DAAH04-96-1-0377; ASSERT Grant ARO DAAG55-98-1-0155; GM Collaborative Research Laboratory in Advanced Vehicle Manufacturing at UM Grant; and by the Horace H. Rackham School of Graduate Studies Faculty Grant.

[†]T.J. Lambert III is with Department of Mathematics, Truckee Meadows Community College, Reno, NV (email: TLambert@tmcc.edu)

[‡]M.A. Epelman and R.L. Smith are with department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA (email: mepelman@umich.edu and rlsmith@umich.edu)

investigates the calculation of error bounds on the optimal expected cost function of a suboptimal design for a specific class of homogeneous MDPs. The latter paper focuses on MDPs where each element of the state vector behaves independently of the other elements and develops a procedure which determines a suboptimal design that improves the calculated bounds.

In this note we present a state aggregation scheme for finite-horizon stochastic Dynamic Programs (DPs), which we represent as homogeneous infinite-horizon MDPs that reach a trapping state within a pre-determined number of transitions. The states of the MDP are aggregated into macro-states as follows: a subset of the states is designated “distinguished states,” and each distinguished state i is associated with a macro-state containing state i and all non-distinguished states that can be reached from i without passing through another distinguished state. Note that macro-states may overlap as a result of such aggregation.

Our approach is in the same spirit as [1], which considers deterministic shortest path problems in acyclic networks. In that work, aggregation is performed by arbitrarily partitioning the nodes of the network into macro-nodes and computing costs on the macro-arcs connecting the macro-nodes by a forward-reaching algorithm. The aggregated network is acyclic by construction, and each macro-node has a unique micro-node “entry point,” similar to a distinguished state in our setting.

In a Markov setting our approach is similar to the models described in [6], [3], and [4]. In [6], authors consider hierarchical control of an infinite horizon Markov chain (e.g., state of a plant) by a “regulator” who controls the plant at each time epoch, and a “supervisor” who intervenes only in distinguished states. They describe the system dynamics at the supervisor’s level (i.e., the macro-system), provide characterizations of the supervisor’s optimal policy, and propose a method for finding it, but do not focus on how the regulator’s strategies are chosen. [4] also discusses an inherently hierarchical infinite horizon MDP, in which decisions on the upper level are made every T epochs, and between the upper level decisions, T -horizon lower-level policies are applied. The system dynamics and rewards are impacted by both upper and lower level decisions. The goal is to find an optimal combination of an upper-level policy and a lower-level policy in the average and discounted cost settings. In [3], the ideas of [6] are applied to infinite horizon average cost MDPs. The aggregation scheme also works by observing the stochastic process underlying the MDP in pre-selected “distinguished” states only (this procedure is referred to as “time-aggregation,” since the resulting stochastic process proceeds, as it were, at a slower pace than the underlying process). Assuming that there is only one action available in each of the non-distinguished states, the authors define the transition probabilities and the cost structure for the aggregated problem, and discuss policy iteration procedures for finding optimal actions to be used in the distinguished states. The aggregation also takes on a hierarchical flavor: non-distinguished states in which actions cannot be chosen, or deemed unimportant for the purposes of the aggregation, are considered to be lower-level. In all of the papers above it is assumed that the underlying process is strongly ergodic, i.e., for every control policy, the induced Markov chain has a single ergodic class.

Although the state aggregation scheme we employ is similar to those above, our definition of actions available in each macro-state, or “macro-actions,” is quite different: a macro-action consists of an action to be undertaken in the distinguished state, as well as decision rules to be followed in the non-distinguished states until another distinguished state (and hence another macro-state) is reached. This definition of macro-actions allows us to show that with transition probabilities and costs appropriately defined, the aggregated problem is an MDP, and if any and all deterministic

Markov decision rules for non-distinguished states can be used in defining the macro-actions, the aggregated problem has the same optimal cost as the original problem. Moreover, by construction, no disaggregation step is required to obtain a solution of the original problem.

Another advantage of the aggregation scheme proposed here is its inherent ability to incorporate inter-period constraints on actions into the problem formulation, while maintaining the Markov properties of the model. There may be two distinct reasons for introducing such constraints into the problem. On the one hand, the constraints may be intrinsic to the problem. On the other hand, in many instances the size of the state space or the action space presents difficulties in the solution process, and the inter-period restriction of policies can be viewed as a way to reduce the computational burden, making the problem tractable. The aggregate problem can provide the solution over a restricted action space, requiring less work than solving the original problem; moreover, this solution is a feasible solution to the original problem.

The idea of policy constraints has been discussed in a short communication [8] in which a maximal gain feasible policy is found by applying a bounded enumeration procedure on a policy ranking list. The first step of the procedure involves solving the unconstrained problem using value iteration, and thus it is clear that no computational savings can be gained over the original problem. We would like to point out that this is not the same concept as a constrained Markov decision problem (see [11]) in which the constraint is with respect to the expected long-run average of a secondary cost function.

The rest of this note is organized as follows: in Section 2 we review the standard MDP notation and assumptions, and specify assumptions on the problems we will be considering here. In Section 3, we describe the aggregation scheme we employ, and prove the Markov property of the resulting aggregated problem. We also establish that the aggregated problem has the same optimal solution as the original problem if no policy constraints are imposed. In Section 4 we discuss two conceptual problem examples in which our aggregation scheme can be used in the solution process.

2 Preliminaries: Markov decision problems

Consider a Markov decision process (MDP) with decisions occurring at epochs $n = 1, 2, 3, \dots$. At each epoch n , a state $i \in \{1, \dots, m\}$ is observed, and a decision k is selected. We assume that the set $A(i)$ of decisions available in each state i is finite.

If at epoch n the system is in state i and decision $k \in A(i)$ is chosen, then the probability that state $j \in \{1, \dots, m\}$ will be observed at epoch $n + 1$ is P_{ij}^k , and an additive reward r_i^k is accrued. Note that the transition probabilities and rewards depend only on the system state and chosen decision, and do not depend on n . (We will later show how to incorporate a time-dependent stochastic dynamic program into this form by appending time as part of the state information.)

Functions $d(n, i)$ prescribe a choice of decision at state i at epoch n ; we refer to these functions as (Markovian deterministic) *decision rules*. For each i , function $d(n, i)$ returns a decision in $A(i)$. Sequences of decision rules, $\pi = \{d(n, \cdot), n = 1, 2, \dots\}$ which specify the choice of decisions over time, are called *policies*. A policy is called *stationary* if $d(n, \cdot) = d(\cdot) \forall n$. Notice that under any stationary policy, the system we are considering is a homogeneous Markov chain with rewards. We will be concerned with systems that satisfy the following assumptions (all policies, including

nonstationary ones, are included in the assumptions below):

1. At epoch $n = 1$, state $i = 1$ is observed.
2. Only a null decision is available in state $i = m$. Moreover, $P_{mm} = 1$ and $r_m = 0$, i.e., m is a trapping state and no further reward is accrued once this state is reached.
3. State m is reachable from every other state under any policy.
4. For any two distinct states i and j , if j is reachable from i under some policy, then i is not reachable from j under any policy.

Assumption 4 can be easily interpreted via a graph representation of the MDP. Consider a directed graph with the set of nodes $\{1, \dots, m\}$. An arc (i, j) is included in the graph if $P_{ij}^k > 0$ for some $k \in A(i)$. Assumption 4 is equivalent to the assumption that this graph contains no directed cycles. Equivalently, the states can be numbered in such a way that j is reachable from i only if $i < j$. Moreover, observe that when assumptions 3 and 4 are satisfied, state m will be observed after at most $m - 1$ transitions.

An important class of problems satisfying assumptions 1 – 4 are finite horizon stochastic DPs, which can be recast in this framework by incorporating the time index into the description of the state and introducing an artificial terminal state (see 4.2 for details).

We will be concerned with finding policies that minimize (or maximize) the expected total reward. Since the state and decision sets of the resulting MDP are finite, there exists a stationary deterministic policy which is optimal with respect to the above criterion (see, for example, [9, Thm. 7.1.9]). Therefore, we can restrict our attention to stationary policies without loss of generality. (Note also that in our setting each state will be visited at most once.)

We will refer to the MDP just described as a *micro-system*, its states as *micro-states*, and decisions as *micro-decisions*, to distinguish them from those arising in the decision process resulting from aggregation.

3 Aggregation in acyclic Markov decision problems

3.1 Aggregation: Constructing Macro-States

We begin by describing the process of constructing “macro-” states of the aggregated problem, along with defining several useful concepts.

Path: Consider two distinct micro-states i and j . A *path* from i to j is a sequence of states $\omega = (i, i^1, \dots, i^l, j)$ such that there exists a sequence of decisions (k, k^1, \dots, k^l) which satisfy

$$P_{ii^1}^k > 0, P_{i^1i^2}^{k^1} > 0, \dots, P_{i^li^l}^{k^l} > 0.$$

We will say that j is *reachable* from i if there exists a path from i to j . The *length* of the above path is $l + 1$.

Distinguished states: Let S be an arbitrary subset of $\{1, \dots, m\}$ containing states 1 and m , i.e., $S = \{i_1 = 1, i_2, i_3, \dots, i_{M-1}, i_M = m\} \subseteq \{1, \dots, m\}$. We refer to the elements of S as *distinguished*

states. The choice of distinguished states will define the aggregation as described later on in this section.

Macro-states: With every element $i \in S$, we associate the set $I = I^i \subseteq \{1, \dots, m\}$ defined as the set of all states j reachable from i without having to pass through another distinguished state $i' \in S$, $i' \neq i$, i.e.,

$$I^i = \{i\} \cup \{j : \exists \omega \text{ a path from } i \text{ to } j \text{ s.t. } \omega \cap S = \{i\}\}.$$

(When taking the set intersection in the above definition, ω should be treated as an unordered set.) We refer to the set I^i as the *macro-state* associated with i . Note that, by construction, each distinguished state $i \in S$ is contained in exactly one macro-state I^i , and each macro-state I defined as above contains exactly one distinguished state. Note also that these collections of states are not necessarily disjoint.

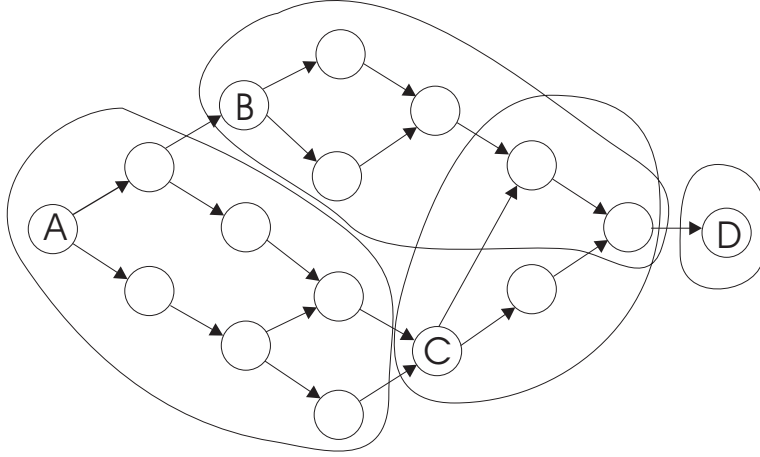


Figure 1: Example: macro-state construction

Figure 1 demonstrates the macro-state construction. In this figure, $S = \{A, B, C, D\}$, and the four macro-states resulting from this selection are shown.

Let $i, j \in S$ be two distinct distinguished states. We define

$$\Omega_{(i,j)} = \{\omega : \text{a path from } i \text{ to } j \text{ such that } \omega \cap S = \{i, j\}\}, \quad (1)$$

i.e., $\Omega_{(i,j)}$ is the collection of all paths from i to j which do not pass through any other distinguished states, and so, by definition of a macro-state, remain in the macro-state I^i until they reach micro-state j .

3.2 Aggregated Problem

The macro-system is a stochastic decision process with observed states $I = I^i$, $i \in S$. We refer to the transition times of this process as *macro-epochs*, denoted $N = 1, 2, \dots$

At each macro-epoch N , with the system occupying macro-state $I = I^i$, a (macro-)decision K is chosen. Let Δn^i be the number of states visited by the longest path $\omega \in \Omega_{(i,\cdot)}$ which is entirely contained in the set I , i.e.,

$$\Delta n^i = \max_{j \in S, \omega \in \Omega_{(i,j)}} \{t : \exists \omega \in \Omega_{(i,j)} \text{ of length } t\}.$$

A decision K is then chosen in the form

$$K = K^i = (k, d(1, \cdot), d(2, \cdot), \dots, d(\Delta n^i - 1, \cdot)), \quad (2)$$

where $k \in A(i)$, and $d(1, \cdot), d(2, \cdot), \dots, d(\Delta n^i - 1, \cdot)$ are (Markovian deterministic) decision rules. That is, a macro-decision consists of a decision chosen in the distinguished state i , followed by a sequence of decision rules to be used in each of the consecutive micro-states the system might reach while in the macro-state I . If the system “leaves” the current macro-state in fewer than $\Delta n^i - 1$ transitions, the decision rules specified for the remaining micro-epochs are ignored.

We refer to functions $D(N, I)$, prescribing a choice of macro-decisions in the above form for each macro-state, as (Markovian deterministic) decision rules for the macro-problem. Sequences $\Pi = \{D(N, \cdot), N = 1, 2, \dots\}$ of decision rules constitute policies for the macro problem; a policy is stationary if $D(N, \cdot) = D(\cdot), \forall N$.

Transition probabilities in the macro-system: If at macro-epoch N macro-state I is observed, and decision K is undertaken, the probability that macro-state J will be observed at epoch $N + 1$ is denoted by P_{IJ}^K . Recall that by construction, each macro-state contains precisely one distinguished state. Let $i \in S$ be the distinguished micro-state contained in I , and $j \in S$ — in J . P_{IJ}^K is defined to be the probability of the micro-system reaching distinguished state j from distinguished state i without visiting any other distinguished states, when the micro-decisions defined by the decision rules that constitute the macro-decision K are used (we say that micro-decisions are *induced* by K). That is, P_{IJ}^K is equal to the probability that the micro-system follows *some* path contained in the set $\Omega_{(i,j)}$. The probability that a particular path $(i, i^1, i^2, \dots, i^l, j) = \omega \in \Omega_{(i,j)}$ will be followed, denoted by P_ω^K , can be computed as

$$P_\omega^K = P_{ii^1}^k \cdot \prod_{p=1}^{l-1} P_{i^p i^{p+1}}^{d(p, i^p)} \cdot P_{i^l j}^{d(l, i^l)}, \quad (3)$$

where decisions k and $d(p, i^p)$, $p = 1, \dots, l$ are induced by K . Using (3), we can express the probability P_{IJ}^K as

$$P_{IJ}^K = \sum_{\omega \in \Omega_{(i,j)}} P_\omega^K. \quad (4)$$

Rewards in the macro-system: If at macro-epoch N the system is in the macro-state I , and decision K is chosen, an additive reward of R_I^K is accrued. R_I^K is defined to be the expected total reward accrued by the micro-system, starting in distinguished state i contained in I , in all transitions until the next distinguished state is reached, when the micro-decisions are induced by K . Consider two distinct distinguished states $i, j \in S$ and the associated macro-states I and J . The

reward accrued by the micro-system if a path $(i, i^1, i^2, \dots, i^l, j) = \omega \in \Omega_{(i,j)}$ is followed, denoted by r_ω^K , is

$$r_\omega^K = r_i^k + \sum_{p=1}^l r_{i^p}^{d(p, i^p)}, \quad (5)$$

where decisions k and $d(p, i^p)$, $p = 1, \dots, l$ are induced by K , and the probability that this path will be followed is given by (3). Using this definition, R_{IJ}^K can be computed as

$$R_I^K = \sum_{j \in S} \sum_{\omega \in \Omega_{(i,j)}} P_\omega^K r_\omega^K. \quad (6)$$

We will be concerned with minimizing (or maximizing) the expected total macro-reward in the macro-system. We refer to the resulting optimization problem as the aggregated, or macro-, problem (by comparison, the original optimization problem will be referred to as the micro-problem). The next theorem demonstrates that the macro-system defined above is an MDP, and that a stationary policy is optimal for the aggregated problem.

Theorem 1 *Suppose the micro-system satisfies assumptions 1–4, and the macro-system is constructed as described above. Then the macro-system constitutes a Markov decision process and there exists a stationary policy which is optimal.*

Proof: To show that the macro-system constitutes an MDP, we must show that the transition probabilities and rewards depend on the past only through the current state and the action selected in that state (see [9, Chapter 2]).

Suppose the macro-system is in state I at epoch N , and the macro-action K is chosen. Then, according to the definitions (4) and (6), P_{IJ}^K and R_I^K depend on the past only through the current macro-state and macro-action, i.e., through I and K . We conclude that the macro-system indeed constitutes an MDP.

By construction, the set of possible states of the macro-system, and the set of macro-actions available in each macro-state is finite. (The latter follows since we restricted our attention to *Markovian deterministic* decision rules in (2).) Therefore, by Theorem 7.1.9 of [9], there exists a deterministic stationary policy for the aggregated problem which is optimal, establishing the second claim of the theorem. ■

By construction, and using assumption 2, micro-state m is a distinguished state, with $I^m = \{m\}$. Moreover, only a null macro-action is available in this state, with $P_{I^m I^m} = 1$ and $R_{I^m} = 0$, so that once this macro-state is reached, no further reward is accrued. By assumption 1, the macro-process starts in state I^1 . Finally, under assumption 3, macro-state I^m is reachable from every other macro-state, while every other macro-state is visited at most once (assumption 4). Therefore, I^m will be observed after at most $|S| - 1$ transitions in the macro-system.

Recall that, as discussed in Section 2, the states of the micro-problem can be numbered in such a way that, for any i and j and for any decision k , $P_{ij}^k > 0$ only if $i < j$. In the macro-system, this structure will be preserved, with $P_{IJ}^K > 0$ only if $i < j$ for $i, j \in S$.

Let $\pi = \{d(n, \cdot) = d(\cdot), \forall n\}$ be an arbitrary stationary policy for the micro-problem. The value of this policy can be evaluated via backward induction (see, for example, [7, Section 4.6]) by setting

$u^\pi(m) = 0$, and computing, for $i = m - 1, m - 2, \dots, 2, 1$,

$$u^\pi(i) = r_i^{d(i)} + \sum_{j=i+1}^m P_{ij}^{d(i)} u^\pi(j). \quad (7)$$

$u^\pi(i)$ is the value of the state i under policy π , and the value of the policy is $u^\pi(1)$ — expected total reward accrued under policy π .

Let $\Pi = \{D(N, \cdot) = D(\cdot), \forall N\}$ be an arbitrary stationary policy for the macro-problem. In light of Theorem 1, the value of this policy can also be evaluated via backward induction. In particular, suppose that the cardinality of S is $M \leq m$, and $S = \{i_1 = 1, i_2, i_3, \dots, i_{M-1}, i_M = m\} \subseteq \{1, \dots, m\}$. Then the macro-system has states $I \in \{I^i, i \in S\}$. To compute the value of policy Π , set $U^\Pi(I^M) = U^\Pi(I^m) = 0$, and compute for $i = i_{M-1}, \dots, i_2, i_1$,

$$U^\Pi(I^i) = R_{I^i}^{D(I^i)} + \sum_{j>i, j \in S} P_{I^i I^j}^{D(I^i)} U^\Pi(I^j). \quad (8)$$

Again, we refer to $U^\Pi(I^i)$ as the value of the state I^i under policy Π . The value of the policy is precisely $U^\Pi(I^{i_1}) = U^\Pi(I^1)$.

The next theorem shows that the solution to the macro-problem obtained by this aggregation scheme is equivalent in value to the solution of the original problem if no further restrictions on the macro-actions are imposed.

Theorem 2 *The optimal value of the macro-problem is equal to the optimal value of the original problem when no restrictions on macro-actions are imposed.*

Proof: Without loss of generality we are assuming a minimization problem. Let V^* and v^* denote the optimal value of the macro-problem and the micro-problem, respectively.

We first argue that $V^* \leq v^*$. Given an optimal policy $\pi = \{d(n, \cdot) = d(\cdot) \forall n\}$ for the micro-problem, let us define a stationary macro-policy $\Pi = \{D(N, \cdot) = D(\cdot) \forall N\}$ as follows: for every macro-state $I = I^i$ with $i \in S$, the policy Π will prescribe the macro-action

$$D(I^i) = (d(i), d(1, \cdot), d(2, \cdot), \dots, d(\Delta n^{i_p} - 1, \cdot)),$$

using the decision rules $d(n, \cdot) = d(\cdot)$ for all n .

We argue that for any distinguished state $i \in S$, $U^\Pi(I^i) = u^\pi(i)$. The argument is inductive. Since $i_M = m \in S$, $U^\Pi(I^m) = u^\pi(m) = 0$. Proceeding inductively, suppose the claim is true for all distinguished states $j \in S$ with $j > i$. Consider distinguished state $i \in S$. Note that the value of the micro-state i under policy π , $u^\pi(i)$, can be interpreted as the expected total return accrued by the micro-system in the transitions in future epochs conditional of the system occupying state i at the current epoch. By construction, every sequence of transitions of the micro-system starting in state i will encounter another distinguished state at some future epoch. Therefore, the value of i

under the policy π can be computed as

$$\begin{aligned}
u^\pi(i) &= \sum_{j>i, j \in S} \left(\sum_{\omega \in \Omega_{(i,j)}} P_\omega^{D(I^i)} \left(u^\pi(j) + r_\omega^{D(I^i)} \right) \right) \\
&= \sum_{j>i, j \in S} \left(\sum_{\omega \in \Omega_{(i,j)}} P_\omega^{D(I^i)} r_\omega^{D(I^i)} \right) \\
&\quad + \sum_{j>i, j \in S} \left(u^\pi(j) \sum_{\omega \in \Omega_{(i,j)}} P_\omega^{D(I^i)} \right) \\
&= R_{I^i}^{D(I^i)} + \sum_{j>i, j \in S} P_{I^i I^j}^{D(I^i)} U^\Pi(I^j) = U^\Pi(I^i).
\end{aligned} \tag{9}$$

Here, the third equality follows from (6) and the induction assumption, and the last equality follows from (8).

Since $i_1 = 1$ is a distinguished state, it follows that $U^\Pi(I^1) = u^\pi(1)$. Since π was selected to be an optimal policy in the micro-problem, we conclude that $V^* \leq v^* = u^\pi(1)$.

It remains to show that $v^* \leq V^*$. Every policy Π for the macro-problem can be implemented as a history-dependent policy for the micro-problem. By an argument similar to the previous one it can be shown that the value obtained under Π in the macro-problem is the same as that obtained using the corresponding history-dependent policy for the original problem. Since the optimal value in an MDP cannot be improved by considering history-dependent policies (e.g., [9]), $v^* \leq U^\Pi(I^1)$ for every policy Π , and therefore $v^* \leq V^*$.

Combining the two inequalities, we conclude that $v^* = V^*$, establishing the claim of the theorem. ■

4 Examples of aggregation

4.1 Incorporating inter-period constraints on actions

As a high-level motivating example, consider a problem arising in designing procedures for controlling a chemical process/reaction over a finite time horizon. For the purpose of this example, suppose that the temperature of the material undergoing the reaction is an important characteristic of the process, and that the temperature of the material at time t effects the controls that may be selected at time t . Assume that the temperature at time $t + 1$ is random with a distribution that depends on the temperature and control selected at time t . A natural mathematical model of this process would be an MDP that at each time t has as its state the temperature at time t . However, if we were to incorporate the restriction that, in order to keep the reaction process stable, one component of the control vector can be changed only when the temperature of the material is near freezing, the actions become history dependent: if at time t the temperature is not near freezing, this control component is restricted to have the same value as it had at the time $t - 1$. Hence the current model is no longer Markovian.

To preserve the Markov property of the model under this condition, one could augment the state description at each time period to include the value of the control used at the preceding time instance — however, doing so will greatly extend the state space. Instead, one can use the aggregation scheme of Section 3 as follows: define the set S of distinguished states to be all the states of the process in which the control can be changed, i.e., in which the temperature is near freezing. The macro-actions will be prescribed in the form $D(I^i) = (k, k, \dots, k)$, ensuring a constant control until the next time a distinguished state is reached.

Problems with limited observations, in which the state of the system is observed only when one of the distinguished states is reached, can also be approached with this aggregation scheme. Many hierarchical control models fit into this framework as well (in the terminology of [3], consider the “supervisor” making a decision in a distinguished state and leaving instructions for the “regulator” to follow).

4.2 Time-based distinguished state selection

In this subsection we describe another method for selecting distinguished states for the aggregation scheme. This method is applicable for finite horizon stochastic DPs. We further restrict our attention to problems in which the set of available actions is independent of state, and, for simplicity of presentation, we assume that at every stage the state space of the MDP is the finite set X . An MDP with horizon T can be recast in the framework of Section 2 by incorporating the time index into the description of the state. Here $P_{ij}^k > 0$ only if $i = (x, n)$ and $j = (x', n + 1)$ for some $x, x' \in X$. Also, we introduce an artificial terminal state m , such that $P_{im}^k = 1$ for all $i = (x, T)$ and for all k . It can be easily verified that assumptions 1 – 4 are satisfied.

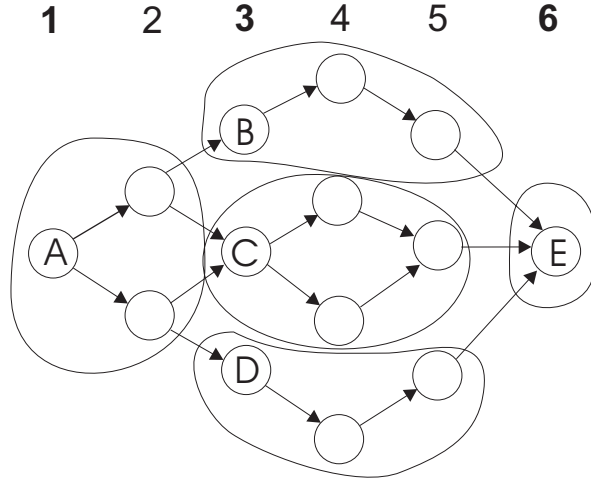


Figure 2: Time-Based Distinguished State Selection

Let \bar{T} be a subset of $\{1, \dots, T\}$ which contains 1 and T , and choose all the states $i = (x, t)$ with $x \in X$ and $t \in \bar{T}$ as the distinguished states. Figure 2 demonstrates this time-based selection. Here $\bar{T} = \{1, 3, 6\}$. Note that in the figure the macro-states partition the state space of the problem.

This is not true in general, but may be the case in some instances; consider, for example, the MDP model of the hierarchical decision model of [4].

A finite-horizon version of the model of [4] provides an example of a problem in which this aggregation scheme can be applied to incorporate the dependence of the lower-level transition probabilities and rewards on the upper-level decision deployed at the distinguished states. Such dependencies on history are inherent in (4) and (6) in our aggregation scheme, with distinguished states as described in this subsection. (Also, this model provides a natural possibility for extending our aggregation scheme to infinite-horizon settings.)

Although in some examples history-dependence and inter-period constraints are inherent in the problem, below we provide an example where the constraints can also be imposed as a way of obtaining an approximate solution to an MDP that is computationally intractable.

The following adaptive radiation treatment design problem was first described in [5]. In radiation treatment therapy, radiation is applied to cancerous tissue. In the process healthy cells may also be damaged by the radiation. Thus, radiation fields must be designed to deliver sufficient dose to the tumor, while at the same time avoiding excessive radiation to nearby healthy tissue. The treatment is performed over n days (typically about 30). In most cases, treatment plans are designed by setting a target dose for each voxel (pixel, point) in the patient, calculating the overall intensities of the radiation beams that will deliver a dose close to the target, and delivering identical treatments by scaling these intensities by n .

However, each time radiation is administered, because of random patient movement and/or setup error, delivery error occurs, manifesting itself in the shift of the patient’s body with respect to the radiation field. These random shifts are assumed to be independent since each treatment is delivered separately. At the end of the treatment process, a cost of the treatment can be calculated as a function of the difference between the target doses and the actual doses received. Equipment currently in development allows radiologists to collect information (via CT imaging and other techniques) about the position of the patient and the radiation actually delivered after each daily treatment. Therefore, treatment planning can be performed adaptively, adjusting the goals of each of the remaining treatments. An MDP model of adaptive treatment, however, is computationally intractable even for extremely simplified patient representations. Moreover, taking CT images and recomputing beam intensities at each treatment is prohibitively costly. A reasonable way to address both of these issues is to pre-select the days on which CT images will be taken and the treatment adjusted (for example, first day of each week, or with frequency increasing towards the end of the treatment), and use a constant policy in the intervening days. Time-based distinguished state selection we described allows to formulate and solve this policy-constrained problem as an MDP.

5 Conclusions

In this note we presented an aggregation scheme for finite-horizon MDPs that preserves the Markov property of the problem and produces solutions that are immediately applicable to the original problem, without a disaggregation step. The scheme allows for incorporation of inter-period constraints on the actions, imposed to ensure feasibility or to reduce the computational demands of obtaining an approximate solution to the problem.

References

- [1] J.C. Bean, J.R. Birge, and R.L. Smith. Aggregation in dynamic programming. *Oper. Res.*, 35:215–220, 1987.
- [2] D.P. Bertsekas and D.A. Castanon. Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE Trans. Automat. Contr.*, 34:589–598, 1989.
- [3] Xi-Ren Cao, Zhiyuan Ren, Shalabh Bhatnagar, Michael Fu, and Steven Marcus. A time aggregation approach to Markov decision processes. *Automatica*, 28:929–943, 2002.
- [4] H.S. Chang, P.J. Fard, S.I. Marcus, and M. Shayman. Multitime scale Markov decision processes. *IEEE Trans. Automat. Contr.*, pages 976–987, 2003.
- [5] Michael C. Ferris and Meta M. Voelker. Fractionation in radiation treatment planning. *Math. Prog.*, 2004. to appear.
- [6] J. Forestier and P. Varaiya. Multilayer control of large markov chains. *IEEE Trans. Automat. Contr.*, 23:298–304, 1978.
- [7] R.G. Gallager. *Discrete Stochastic Processes*. Kluwer, Boston, 1996.
- [8] N.A.J. Hastings and D. Sadjadi. Markov programming with policy constraints. *European J. Oper. Res.*, 3:253–255, 1979.
- [9] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.
- [10] D.F. Rogers, R.D. Plante, R.T. Wong, and J.R. Evans. Aggregation and disaggregation techniques and methodology in optimization. *Oper. Res.*, 38:553–582, 1991.
- [11] Linn Sennott. Computing average optimal constrained policies in stochastic dynamic programming. *Probab. Engrg. Inform. Sci.*, 15:103–133, 2001.
- [12] C.C. White and K. Schlussel. Suboptimal design for large scale, multimodule systems. *Oper. Res.*, 29:865–875, 1981.