Discounted Markov Decision Processes with Constrained Costs: the decomposition approach

Abdellatif Semmouri[1*], *Mostafa* Jourhmane[1], and *Bahaa Eddine* Elbaghazaoui[2]

[1]Laboratory TIAD, Faculty of Sciences and Techniques, Sultan Moulay Slimane University, Campus Mghilla, Beni Mellal, Morocco
[2]Laboratory of Computer Sciences, Faculty of Sciences Kenitra, IbnTofail University, Morocco

Abstract. In this paper we consider a constrained optimization of discrete time Markov Decision Processes (MDPs) with finite state and action spaces, which accumulate both a reward and costs at each decision epoch. We will study the problem of finding a policy that maximizes the expected total discounted reward subject to the constraints that the expected total discounted costs are not greater than given values. Thus, we will investigate the decomposition method of the state space into the strongly communicating classes for computing an optimal or a nearly optimal stationary policy. The discounted criterion has many applications in several areas such that the Forest Management, the Management of Energy Consumption, the finance, the Communication System (Mobile Networks) and the artificial intelligence.

1 Introduction

The decomposition method consists in dividing the space of states into subsets which are weakly coupled. This technique was first introduced by Bather [1]. In his context, the decomposition of the state space is described and based on the accessibility between the states. The state space is divided into several Levels. Following Ross and Varadarajan [5] have presented a similar decomposition method to solve the constrained problem of the long-time average Markov Decision Processes. In this decomposition, the state space is partitioned into Strongly Communicating Classes and a set (perhaps empty) of transient states. Next, Baykal-Gursoy and Ross [6], Daoui and Abbad [7] investigated the same decomposition to solve the unconstrained problem of the long-time average.

* Corresponding author: abd_semmouri@yahoo.fr

We model in this work the environment as a Constrained Markov Decision Processes, defined by a tuple

$$(S, A = \bigcup_{x \in S} A_x, T, r, \{c_k\}_{k=1,\dots,K}, \{\alpha_k\}_{k=1,\dots,K}, \gamma, x_0)$$

where S is the set of states x, A is the set of actions a, $T(y/x,a) = Prob(y/x,a)$ is the transition probability, $r(x,a) \in \mathbb{R}$ is the reward function which denotes immediate reward incurred by taking action a in state x, $c_k(x,a) \in \mathbb{R}$ is the k^{th} cost function upper bounded by , $\alpha_k \in \mathbb{R}$ of k^{th} cost constraint, $\gamma \in [0,1)$ is the discount factor and x_0 is the initial fixed state. The goal is to compute an optimal policy u^* that maximizes the expected cumulative discounted rewards earned at state x_0 while expected cumulative discounted costs are bounded:

$$max_u \mathcal{V}_{x_0}(u) := \mathbb{E}_u^{x_0}\left[\sum_{n=1}^{\infty} \gamma^{n-1} r(X_n, A_n)\right]$$

s. t.

$$\mathcal{C}_{x_0}^{k}(u) := \mathbb{E}_u^{x_0}\left[\sum_{n=1}^{\infty} \gamma^{n-1} c_k(X_n, A_n)\right] \leq \alpha_k, \qquad \forall \mathrm{k} = 1, \dots, \mathrm{K}$$

Define the random variable R by

$$R = \sum_{n=1}^{\infty} \gamma^{n-1} r(X_n, A_n)$$

Here, $\{X_n\}_{n\geq 1}$ is the state process taking values in the finite space S and $\{A_n\}_{n\geq 1}$ is the action process taking values in the finite action space A. The notation $1_{(.,.)}$ represents the indicator function.

Set

$$\mathcal{V}_{\mathrm{x_0}}^{*} := \sup_{\mathrm{u} \in \mathcal{U}^{\mathrm{MS}}} \mathcal{V}_{\mathrm{x_0}}(\mathrm{u})$$

where $\mathcal{U}^{\mathrm{MS}}$ denotes the set of all stationary policies.

A stationary policy u is called optimal (ε-optimal) if $\mathcal{V}_{\mathrm{x_0}}(\mathrm{u}) = \mathcal{V}_{\mathrm{x_0}}^{*}$ ($\mathcal{V}_{\mathrm{x_0}}(\mathrm{u}) > \mathcal{V}_{\mathrm{x_0}}^{*} - \varepsilon$).

We will solve the problem of the constrained discounted Markov Decision Processes exploiting the decomposition of the state space S into the strongly communicating classes by steps. First, we solve the restricted MDPs in subsection 3.1. We introduce a new MDP called intermediate MDP in subsection 3.2. We find a corresponding optimal policy. In section 4, we combine the results in subsections 3.1 and 3.2 in order to construct a nearly optimal policy for the original problem.

2 Preliminaries

2.1 Sample space, policies and measures

The finite state and action spaces are denoted by S and A, respectively. The sample space is given by $\Omega = \{S \times A\}^{\infty}$, so that the typical realization ω can be represented as $\omega = (x_1, a_1, x_1, a_1, \dots)$.

The state and action random variables X_n, $\mathrm{A_n}$ for $n = 1,2,\dots$ are then defined as the coordinate mappings $X_n(\omega) = x_n$ and $A_n(\omega) = a_n$.

The sample space Ω will be equipped with the σ-algebra $\mathcal{B}$ generated by the random variables $(X_n, A_n, n = 1,2,\dots)$.

In order to give a formal definition of a policy, first let $\mathcal{Q}$ be the set of all probability measures on the action space A, i.e:

$$\mathcal{Q} \coloneqq \{(q_1, q_2, \dots, q_{|A|}) : q_1 + q_2 + \cdots + q_{|A|} = 1, \\ q_i \geq 0, 1 \leq i \leq |A|\}$$

where $|A|$ is the cardinality of A. Then a policy u is defined to be a sequence $u = (u^1, u^2, \dots)$ where u^k is a mapping from $\{S \times A\}^{k-1} \times S$ to $\mathcal{Q}$. We write u_i^m, $1 \leq i \leq |A|$, for the i^{th} component of u^m.

For a fixed policy u and initial state x, we can now construct the probability measure P_u^x for the measurable space $(\Omega, \mathcal{B})$. The finite-dimensional distributions of the probability measure P_u^x are defined as follows:

$$P_u^x(X_1 = x) = 1 \quad (1)$$

$$P_u^x(A_m = a/X_1 = x_1, A_1 = a_1, \dots, X_{m-1} = x_{m-1}, \\ A_{m-1} = a_{m-1}, X_m = x_m) \\ = u_a^m(x_1, a_1, \dots, x_{m-1}, a_{m-1}, x_m) \quad (2)$$

$$P_u^x(X_{m+1} = y/X_1 = x_1, A_1 = a_1, \dots, X_{m-1} = x_{m-1}, \\ A_{m-1} = a_{m-1}, X_m = x, A_m = a) = p_{xav} \quad (3)$$

where p_{xay} is the law of motion, which is given and determined from the physical of the problem. From a stantard application of the Kolmogorov consistency theorem, we know there exists a unique probability measure P_u^x, on $(\Omega, \mathcal{B})$ such that (1)-(3) hold for all possible histories and all $m \geq 1$. Thus, for each policy u and initial state x, we have constructed a probability space $(\Omega, \mathcal{B}, P_u^x)$.

A policy f is said to be stationary if the same decision rule is used in every epoch. It is determined by a nonnegative function f on $S \times A$ such that

$$\sum_{a \in A} f(x, a) = 1$$

for every $x \in S$.

A stationary policy f is called deterministic (non-randomized) if for every$x \in S$, we have $f(x, a) = 1$ for exactly one action $a \in A$. The set of deterministic policies is denoted by $\mathcal{U}^{MD}$. For a stationary policy u the corresponding transition matrix $P(f) = \left[P_{x,y}(f)\right]_{x,y \in S}$ is defined by

$$P_{x,y}(f) = \sum_{a \in A} p_{xay} f(x, a)$$

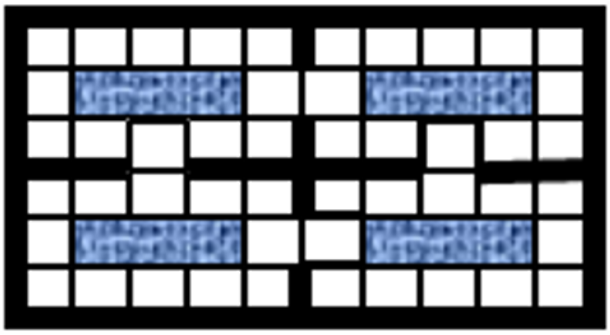

Fig. 1. The movement of a robot

2.2 Decomposition theory

The state space S has a naturel partition into strongly communicating classes $\mathcal{C}_1$,..., $\mathcal{C}_p$ and a set of states $\mathcal{T}$. This decomposition has the following properties:
i) The states in $\mathcal{T}$ are transient under all stationary policies.
ii) If $\mathcal{R}$ is a recurrent class associated with some stationary policy, then $\mathcal{R}$ is contained in one of the strongly communicating classes.
iii) There exists a stationary policy whose associated recurrent classes exactly correspond to the strongly communicating classes.
iv) Under any stationary policy u and initial state x given, we have

$$\sum_{i=1}^{p} P_u^x(X_n \in \mathcal{C}_i \ \ a.a.) = 1 \qquad (4)$$

where ***a.a.*** abbreviates "almost always". Hence, for all policy u, the state process eventually enters one of the strongly communicating classes and remains forever.
v) The partition $\{\mathcal{C}_1, \dots, \mathcal{C}_p, \mathcal{T}\}$ can be obtained by an efficient polynomial-time algorithm (KW. Ross and R. Varadarajan). It is based on a depth-first procedure of graph theory.

In this subsection, any restricted MDP of the given MDP has the same laws of motion as the original MDP. Given an MDP M, with state space S, we define the state-dependant action spaces to be $A_x = A$, for all $x \in S$. Then, we invoke the recursive procedure FIND-CLASSES to find the strongly communicating classes of M.

Algorithm (Ross-Varadarajan)

Procedure FIND-CLASSES(Input: M; Output: ξ,T)
(Given a MDP M, with state space S and action spaces $A_x, x \in S$, returns the set ξ of strongly communicating classes of M and the set T of states that are transient under all stationary policies)
1. Partition de state space of M into communicating classes $D_1, D_2, \dots, D_q$.
2. If $q = 1$, set $\xi \leftarrow \{D_1\}$, $T \leftarrow \emptyset$: STOP.
3. Otherwise, set $T \leftarrow \emptyset$, $\xi \leftarrow \emptyset$; DO for $k = 1,2, \dots, q$
4. For each $x \in D_k$, set $B_x \leftarrow \{a \in A_x : p_{xay} = 0, \ for\ all\ y \notin D_k\}$ and $\bar{B}_x \leftarrow A_x - B_x$.
5. If $\bar{B}_x = \emptyset$ for all $x \in D_k$, set $\xi_k \leftarrow \{D_k\}$, $T_k \leftarrow \emptyset$, GO to STEP 9.
6. Otherwise, call FIND-RESTRICTED-MDP(D_k,$(B_x)_{x \in D_k}$; $\bar{\bar{M}}_k$, $\bar{\bar{T}}_k$).
7. If $D_k = \bar{\bar{T}}_k$, set $\xi_k \leftarrow \emptyset$, $T_k \leftarrow \bar{\bar{T}}_k$, GO to STEP 9.
8. Otherwise, call FIND-CLASSES(M_k; ξ_k,$\bar{\bar{T}}$); set $T_k \leftarrow \bar{T}_k \cup \bar{\bar{T}}_k$.
9. Set $\xi \leftarrow \xi \cup \xi_k$, $T \leftarrow T \cup T_k$.

Procedure FIND-RESTRICTED-MDP(Input: D,B_x, $x \in D$; Output: $\bar{\bar{M}}$,$\bar{\bar{T}}$)
(Given a set of states D and the action sets B_x,$x \in D$, returns a MDP $\bar{\bar{M}}$, restricted to $D - \bar{\bar{T}}$, where all the states in $\bar{\bar{T}} \subset T$ are transient under all stationary policies for M)
1. Set $\bar{\bar{T}} \leftarrow \emptyset$, $\bar{\bar{S}} \leftarrow D$, $\bar{\bar{A}}_x \leftarrow B_x$ for all $x \in D$.
2. While $\bar{\bar{A}}_x = \emptyset$ for some $x \in \bar{\bar{S}}$ DO
3. $T^* \leftarrow \{x \in \bar{\bar{S}} : \ \bar{\bar{A}}_x = \emptyset\}$.
4. $\bar{\bar{S}} \leftarrow \bar{\bar{S}} - T^*$; $\bar{\bar{T}} \leftarrow \bar{\bar{T}} \cup T^*$.
5. For all $x \in \bar{\bar{S}}$, set $\bar{\bar{A}}_x \leftarrow \bar{\bar{A}}_x - \{a : p_{xay} > 0 \ for\ some\ y \in T^*\}$.
6. $\bar{\bar{M}}$ is the MDP with state space $\bar{\bar{S}}$ and action spaces $\bar{\bar{A}}_x, x \in \bar{\bar{S}}$.

Complexity

The time complexity of this algorithm is $\mathcal{O}(n^3 a)$, where $n = |S|$ and $a = |A|$.

Example 1. Consider an MDP with the following data:

$$S = \{1,2,3,4\};\ A_1 = \{a_1\};\ A_2 = \{a_1, a_2\};\ A_3 = \{a_1, a_2\};\ A_4 = \{a_1\};$$

$$p_{1a_12} = 1;\ p_{2a_12} = 0.4;\ \ p_{2a_13} = 0.6;\ \ p_{2a_22} = 1;\ \ p_{3a_12} = 0.5;$$

$$p_{3a_13} = 0.5;\ p_{3a_23} = 0.3;\ \ p_{3a_24} = 0.7.$$

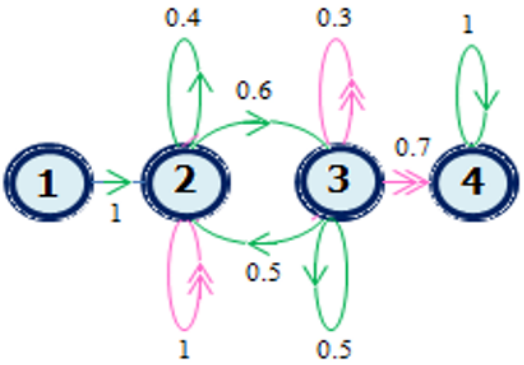

Fig.2. The state transition diagram

Ross-Varadarajan algorithm provides de strongly communicating classes: $\mathcal{C}_1 = \{2,3\}$; $\mathcal{C}_2 = \{4\}$ and the set of transient states: $\mathcal{T} = \{1\}$.

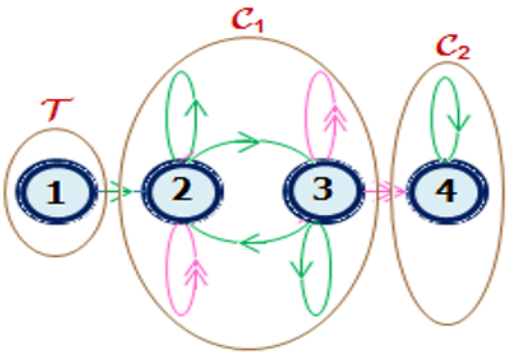

Fig.3. The decomposition method

We set for each state $x \in \mathcal{C}_i$, $i = 1, \dots, p$:

$$\mathcal{F}_x = \{a \in A :\ p_{xay} = 0, \forall y \notin \mathcal{C}_i\}$$

By starting from a state $x \in \mathcal{C}_i$, the set $\mathcal{F}_x$ contains the actions which guarantee that the state process will remain in the strongly communicating class $\mathcal{C}_i$.

Proposition 1 (see [5]). For all policy u and all initial state $x \in S$,

$$\sum_{i=1}^{p} P_u^x(X_n \in \mathcal{C}_i\ \ a.a.) = 1 \qquad (5)$$

$$P_u^x(A_n \in \mathcal{F}_{X_n}\ \ a.a.) = 1 \qquad (6)$$

where ***a.a.*** abbreviates "almost always".

For $i = 1, \dots, p$, we set

$$\Phi_i := (X_n \in \mathcal{C}_i \ \ a.a.)$$

3 New MDP$_s$

3.1 Restricted MDPs

For each $i = 1, \dots, p$ we define a new MDP, called MDP$_i$ as follows:
1) The state space is $\mathcal{C}_i$;
2) For each $x \in \mathcal{C}_i$, the set of available actions is given by the state-dependent action spaces $\mathcal{F}_x$;
3) The laws of motion, cost and reward functions are the same as for the original MDP but restricted to the state-dependent action spaces $\mathcal{F}_x$.

Proposition 2 (see [5]). For all $i = 1, \dots, p$, we have:
i) $\mathcal{F}_x$ is nonempty for all $x \in \mathcal{C}_i$;
ii) $\sum_{y \in \mathcal{C}_i} p_{xay} = 1$ for all $a \in \mathcal{F}_x$, $x \in \mathcal{C}_i$.
iii) Each MDP$_i$ is a communicating MDP.

For fixed $i = 1, \dots, p$, consider the evolution of the state and action processes for MDP$_i$. For all $n = 1,2, \dots$ we have

$$X_n \in \mathcal{C}_i \quad \text{and} \quad A_n \in \mathcal{F}_x$$

Each policy u determines a probability measure $P_{u,i}^{x_0}$ on the sample space associated with MDP$_i$. The corresponding expected total discounted reward and costs for MDP$_i$ are given by

$$\mathcal{V}_{x_0}^{i}(u) := \mathbb{E}_{u,i}^{x_0}\left[\sum_{n=1}^{\infty} \gamma^{n-1} r(X_n, A_n)\right]$$

and

$$\mathcal{C}_{x_0}^{i,k}(u) := \mathbb{E}_{u,i}^{x_0}\left[\sum_{n=1}^{\infty} \gamma^{n-1} c_k(X_n, A_n)\right], \qquad k = 1, \dots, K$$

Definition 1. A policy u is said to be feasible for MDP$_i$ if $\mathcal{C}_{x_0}^{i,k}(u) \le \alpha_k$, for all $k = 1, \dots, K$. $\mathcal{U}_{x_0}$ denotes the set of all feasible policies.

For each MDP$_i$, we also need to introduce an associated linear program (LP_i) with decision variables $\{z_{x,a}^{x_0} : x \in \mathcal{C}_i, a \in \mathcal{F}_a\}$.

Linear Program (LP_i)

$$m_i^{x_0} = max \sum_{x \in \mathcal{C}_i} \sum_{a \in \mathcal{F}_x} r(x,a) z_{x,a}^{x_0}$$

Subject to

$$\sum_{x \in \mathcal{C}_i} \sum_{a \in \mathcal{F}_x} (\delta_{x,y} - \gamma p_{xay}) z_{x,a}^{x_0} = \delta_{x_0,y}, \qquad y \in \mathcal{C}_i$$

$$\sum_{x \in \mathcal{C}_i} \sum_{a \in \mathcal{F}_x} c_k(x,a) z_{x,a}^{x_0} \le \alpha_k, \qquad k = 1, \dots, K$$

$$z_{x,a}^{x_0} \ge 0, \qquad x \in \mathcal{C}_i, a \in \mathcal{F}_x$$

where $z_{x,a}^{x_0}$ can be interpreted as a discounted occupancy measure of (x, a), and $\delta_{x,y} = 1$ if $x = y$ and 0 otherwise.

Denote

$$G \coloneqq \{i\colon\ 1 \le i \le p, LP_i\ is\ feasible\}$$

and set $m_i^{x_i} = -\infty$ for each $i \notin G$. The relationship between MDP_i and LP_i is expressed in the following theorem.

Theorem 1. There exists a feasible policy for MDP_i if and only if $i \in G$, then there exists an optimal stationary policy $f_{x_0}^i$ for MDP_i.

Proof. The proof follows from the results which are established by Altman [8].

Let $\{z_{x,a}^{x_0}\}$ be an optimal extreme point for LP_i. An optimal stationary policy and the corresponding optimal value are computed as

$$f_{x_0}^i(x,a) = \begin{cases} \dfrac{z_{x,a}^{x_0}}{\sum_{a' \in A_x} z_{x,a'}^{x_0}}, if \sum\limits_{a' \in A_x} z_{x,a'}^{x_0} \neq 0 \\ \delta_{a,a(x)}, Otherwise \end{cases}$$

where $a(x) \in \mathcal{F}_x$ are arbitrary and $x \in \mathcal{C}_i$.
and

$$\mathcal{V}_{x_0}^{\dot{}}\left(f_{x_0}^i\right) = Sup_{u \in \mathcal{U}^{MS}} \mathcal{V}_{x_0}^{\dot{}}(u) = \sum_{x \in \mathcal{C}_i} \sum_{a \in \mathcal{F}_x} r(x,a) z_{x,a}^{x_0} = m_i^{x_0}$$

Theorem 2. For all policy $u \in \mathcal{U}_{x_0}$ and $i = 1, \ldots, p$, we have

$$P_u^{x_0}\left(R \le m_i^{x_0},\ \Phi_i\right) = P_u^{x_0}(\ \Phi_i)$$

Proof. If $P_u^{x_0}(\Phi_i) = 0$, then it follows $P_u^{x_0}\left(R \le m_i^{x_0},\ \Phi_i\right) = 0$.
Thus, $P_u^{x_0}\left(R \le m_i^{x_0},\ \Phi_i\right) = P_u^{x_0}(\ \Phi_i)$
Unless then,

$$P_u^{x_0}\left(R \le m_i^{x_0},\ \Phi_i\right) = P_u^{x_0}\left(R \le m_i^{x_0} / \Phi_i\right).P_u^{x_0}(\ \Phi_i) = P_u^{x_0}(\ \Phi_i)$$

(See Rosenthal [9]).

Corollary 1. If $u \epsilon \mathcal{U}_{x_0}$ and $i \notin G$, then we have

$$P_u^{x_0}(\ \Phi_i) = 0$$

Proof. If $i \notin G$, then $m_i^{x_0} = -\infty$. Since the reward function $r(.,.)$ is bounded below (due to the finite state and action spaces), the result then directly follows from theorem 2.

Remark 1. Corollary 1, combined with the fact that $\{\Phi_i\}_{i=1}^p$ form a partition of the sample space Ω implies that G is nonempty whenever $\mathcal{U}_{x_0}$ is nonempty.

3.2 Intermediate MDP

Over the original sample, define for each policy u the following expected time-average reward

$$\beta_x(u) \coloneqq E_u^x\left[\lim_{n \to \infty} inf \frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{p} m_i^{x_0}.1_{(X_m \in \mathcal{C}_i)}\right], \qquad x \in S$$

In this subsection we consider the unconstrained problem of maximizing $\beta_x(u)$ over all stationary policy u and $x \in S$. From the proposition 1 and the Lebesgue's dominated convergence theorem we have for all policy u

$$\beta_{x_0}(u) = \sum_{i=1}^{p} m_i^{x_0} E_u^{x_0}\left[\lim_{n\to\infty} inf \frac{1}{n}\sum_{m=1}^{n} 1_{(X_m \in \mathcal{C}_i)}\right] = \sum_{i=1}^{p} m_i^{x_0} P_u^{x_0}(\Phi_i) \qquad (7)$$

It is well known that there exists an optimal pure policy **g** for this problem which can be found by standard tools of the dynamic programming such that policy improvement, value iteration algorithm or linear programming approach (see Ross and Varadarajan [5], Baykal-Gursoy and Ross [6], Puterman [11], Bertsekas [16]). The following lemma gives an upper bound for the supremum of the original discounted reward.

Lemma 1. Let $\beta_{x_0} = Sup_{u\in\mathcal{U}^{MD}}\beta_{x_0}(u)$ and $\mathcal{V}_{x_0}^{*} = Sup_{u\in\mathcal{U}^{MD}}\mathcal{V}_{x_0}(u)$. Then, $\mathcal{V}_{x_0}^{*} \le \beta_{x_0}$.

Proof. Let u a stationary policy. From theorem 2, corollary 1 and Rosenthal [9] it follows

$$\mathcal{V}_{x_0}(u) := E_u^{x_0}[R] = \sum_{i\in G'} E_u^{x_0}[R/\Phi_i].P_u^{x_0}[\Phi_i] \le \sum_{i=1}^{p} m_i^{x_0} P_u^{x_0}(\Phi_i)$$

where $G' = \{i \in G : P_u^{x_0}(\Phi_i)\}$.
By (7), we have $\mathcal{V}_{x_0}(u) \le \beta_{x_0}$ for all stationary policy u. Thus, we conclude that $\mathcal{V}_{x_0}^{*} \le \beta_{x_0}$.

Let **g** be an optimal policy for the intermediate MDP associated with the supremum value β_{x_0} and let H be the subset of G defined as

$$H := \{i \in G: \mathcal{C}_i \textit{ contains a recurrent class under } \mathbf{g}\}$$

Without loss of generality, we may assume that each $\mathcal{C}_i, i \in H$ is closed under **g**, (Otherwise, modify **g** so that $\mathbf{g}(x) \in \mathcal{F}_x$ for all $x \in \mathcal{C}_i, i \in H$. Clearly the modified policy has the desired property, and is not difficult to show that it continues to maximize $\beta_{x_0}(u)$).

3.3 Aggregated MDP

For solving the intermediate MDP problem, we use the well-known technique that is called aggregated MDP method.

The aggregated MDP is defined as follows:
1) The state space is $\tilde{S} = \{1,2,\dots,p+t\}$, where $t = |T|$.
2) The state-dependent action space $\tilde{B}_i$, $i \in \tilde{S}$, are

$$\tilde{B}_i = \{\theta\} \cup \{(x,a) : x \in \mathcal{C}_i, a \notin \mathcal{F}_x\},\ 1 \le i \le p.$$
$$\tilde{B}_i = A_I, \qquad p+1 \le i \le p+t$$

3) For $i = 1,2,\dots,p$, the law of motion is given by
$\tilde{p}_{i\theta i} = 1$
$\tilde{p}_{i(x,a)j} = \sum_{y\in\mathcal{C}_j} p_{xay}$ for all $1 \le j \le p$, $(x,a) \in \tilde{B}_i$
$\tilde{p}_{i(x,a)j} = p_{xaj}$ for all $p+1 \le j \le p+t$, $(x,a) \in \tilde{B}_i$
4) For $i = p+1,\dots,p+t$, the law of motion is given by
$\tilde{p}_{iaj} = \sum_{y\in\mathcal{C}_j} p_{iay}$ for all $1 \le j \le p$, $a \in \tilde{B}_i$
$\tilde{p}_{iaj} = p_{iaj}$ for all $p+1 \le j \le p+t$, $a \in \tilde{B}_i$
4) The rewards are:
$\tilde{r}(i,a) = r(i,a)$ for all $p+1 \le i \le p+t$ and $a \in A_i$

$\tilde{r}(i,\theta) = m_i^{x_0}$ for all $1 \le i \le p$
$\tilde{r}(i,(x,a)) = r(x,a)$ for all $1 \le i \le p$

By considering the evolution of the aggregated MDP, the corresponding average reward is defined as follows:

$$\tilde{\beta}_i(u) \coloneqq E_u^i\left[\lim_{n\to\infty} inf \frac{1}{n}\sum_{m=1}^{n}\sum_{k=1}^{p} m_i^{x_0}.1_{(X_m=k)}\right]$$

and denote $\tilde{\beta}_i \coloneqq Sup_{u\in\mathcal{U}^{MS}}\tilde{\beta}_i(u),\ i \in \tilde{S}$.

4 An optimal policy for the original MDP

In this section we construct a stationary policy f^* as follows:

1) For each $i \in G$ let $f_{x_i}^i$ be the optimal stationary policy for the MDP$_i$ as given in subsection 3.1;
2) Let g be the optimal policy as given in subsection 3.2. Let H the set of $i \in G$ such that $\mathcal{C}_i$ is closed under g;
3) Define a stationary policy f^* as follows: when in state $x \in \mathcal{C}_i$ with $i \in H$, apply the policy $f_{x_i}^i$, otherwise apply g.

Theorem 3. The stationary policy f^* as constructed above is optimal for the original problem.

Proof. Since f^* is identical to g outside of $\bigcup_{i\in H}\mathcal{C}_i$, and since $\mathcal{C}_i, i \in H$ is closed under both f^* and g, we get

$$P_{f^*}^{x_0}(\Phi_i) = P_g^{x_0}(\Phi_i) \text{ for all } i \qquad (8)$$

From Rosenthal [9] and the fact that f^* is identical to g over $\mathcal{C}_i$ for each $i \in H$, we have for $k = 1, \dots, K$

$$\mathcal{C}_{x_0}^k(f^*) = \sum_{i\in H} P_{f^*}^{x_0}(\Phi_i).\mathcal{C}_{x_0}^{i,k}\left(f_{x_0}^i\right) \le \alpha \sum_{i\in H} P_{f^*}^{x_0}(\Phi_i) = \alpha \qquad (9)$$

Thus, f^* is a feasible policy for the original problem (i.e. $f^* \in \mathcal{U}_{x_0}$).

In the other hand, we have

$$\mathcal{V}_{x_0}(f^*) = \sum_{i\in H} P_{f^*}^{x_0}(\Phi_i).\ \mathcal{V}_{x_0}^i\left(f_{x_0}^i\right) = \sum_{i\in H} P_{f^*}^{x_0}(\Phi_i).m_i^{x_0}$$

By combining (5), (7), (8) and lemma 1, we get

$$\mathcal{V}_{x_0}(f^*) = \sum_{i\in H} P_{f^*}^{x_0}(\Phi_i).m_i^{x_0} = \beta_{x_0} \ge \mathcal{V}_{x_0}^*$$

Hence, the stationary policy f^* is optimal for the original problem.

Remark 2. If there exists some $i \in H$ such that the policy $f_{x_0}^i$ is nearly optimal for the MDP$_i$, then f^* is a nearly optimal for the original problem.

5 Conclusion

The theoretical framework of Markov decision processes gives the semantic foundation for a wide range of problems involving planning under uncertainty. For solving large-scale

MDPs, the decomposition method is required. This approach consists to lead a naturel decomposition of state space into subsets that are weekly coupled. Hence, each small MDP is solved separately via the linear programming. In artificial intelligence with many rooms, this literature will be well applied. Thus, the decomposition method allows reducing the complexity of computing an optimal or a nearly optimal policy for the Constrained Markov Decision Processes Problems using the intermediate MDP technique.

In the future work, we will study the possibility to apply dynamic programming tools for solving the constrained Markov decision processes in the discounted case.

Acknowledgements

The authors would like to thank the following people. Firstly, Professor Dr. C. Daoui of Sultan Moulay Slimane University, Beni Mellal, Morocco for his help and encouraging during the period of research. Secondly, Mr. Lekbir Tansaoui, ELT teacher, co-author and textbook designer in Mokhtar Essoussi High School, Oued Zem, Morocco for proofreading this paper. We also wish to express our sincere thanks to all members of the organizing committee of the Conference ICCSRE'2020 and referees for careful reading of the manuscript, valuable suggestions and of a number of helpful remarks..

References

1. J. Bather. (1973). Optimal decision procedures for finite Markov chains. Part II: Communicating systems. Advances in Applied Probability, 5(3), 521-540. https://doi.org/10.2307/1425832
2. J. Bather. (1973). Optimal decision procedures for finite Markov chains. Part III: General convex systems. Advances in Applied Probability, 5(3), 541-553. https://doi.org/10.2307/1425833
3. Hou, Z., Filar, J. A., & Chen, A. (Eds.). (2013). Markov processes and controlled Markov chains. Springer Science & Business Media.
4. KW., Ross, & R., Varadarajan. (1988). Markov decision processes with sample path constraints: the communicating case. Mathematics of Operations Research, 37(5), 780-790. https://doi.org/10.1287/opre.37.5.780
5. KW., Ross, & R., Varadarajan. (1991). Multichain Markov decision processes with a sample path constraint: a decomposition approach. Mathematics of Operations Research, 16 (1), 195-207. https://doi.org/10.1287/moor.16.1.195
6. M., Baykal-Gursoy, & KW., Ross. (1992). Variability sensitive Markov decision processes. Operations Research, 17(3), 558-571. https://doi.org/10.1287/moor.17.3.558
7. C., Daoui, & M., Abbad. (2007). On some algorithms for limiting average Markov decision processes. Operations Research, 35(2), 261-266. https://doi.org/10.1016/j.orl.2006.03.006
8. E., Altman. (1999). Constrained Markov Decision Processes. Chapman and Hall: London, U.K.
9. J.S., Rosenthal. (2006). A first look at rigorous probability theory, A. World Scientific Publishing: Singapore.
10. RA., Howard. (1960). Dynamic programming and Markov processes. MIT Press: Cambridge.

11. M.L., Puterman. (1994). Markov decision processes discrete stochastic dynamic programming. John Wiley & Sons: New York.
12. N., Bauerle, & U., Rieder. (2011). Markov decision processes with applications to finance. Springer Science & Business Media: Berlin Heidelberg.
13. C., Derman. (1970). Finite state Markovian decision processes. Academic Press: New York.
14. D.P., Bertsekas. (1976). Dynamic programming and stochastic control. Academic Press: New York.
15. D.P., Bertsekas, & S.E., Shreve. (1978). Stochastic optimal control. Academic Press: New York.
16. D.P., Bertsekas. (1995). Dynamic programming and optimal control I. Athena Scientific: Belmont, Massachusetts.
17. D.P., Bertsekas. (1995). Dynamic programming and optimal control II. Athena Scientific: Belmont, Massachusetts.
18. M., Herzberg, & U., Yechiali. (1994). Accelerating procedures of the value iteration algorithm for discounted Markov decision processes, based on a one-step look-ahead analysis. Operations Research, 42(5), 940-946. http://www.jstor.org/stable/171550