Solving Large Weakly Coupled Markov Decision Processes : Application to Forest Management

F. Garcia and R. Sabbadin

INRA-Unité Biométrie et Intelligence Artificielle, Chemin de Borde-Rouge, BP 27, 31326 Castanet-Tolosan Cedex, France (sabbadin@toulouse.inra.fr)

Abstract: The formal MDP framework (Markov Decision Process) has become the model of choice for modeling and solving sequential decision problems in the AI community. However, realistic problems are generally difficult to treat in this framework : the state and the decision spaces are generally multi-dimensional so that their sizes are huge ($> 10^6$ states). Nevertheless these problems may often be represented in a compact way and be decomposed into relatively independent subproblems (they are "weakly coupled"). The purpose of this paper is to survey different methods that have been recently proposed by the AI community to address these "large" weakly coupled problems. This is illustrated over a toy-forest management problem. We hope to be able to apply the proposed methods to a real-case study.

Keywords: Markov Decision Processes; Weakly coupled problems; Reinforcement Learning.

1. INTRODUCTION

It is difficult, in the Markov Decision Processes framework (MDP) [Puterman, 1994] to deal with realistic problems with multidimensional state and action spaces, due to the induced problem size ($> 10^6$ states). However, these problems are often expressed simply and compactly as a collection of more or less independent subproblems. When this is the case, the initial problem is said to be weakly coupled, and several families of methods have been recently proposed, that allow to solve such problems, for larger and larger state spaces. Namely, three families can be inventored :

- The state aggregation methods group states in subsets sharing the same features, thus reducing the size of the MDP [Dearden and Boutillier, 1997]. In the same family, actions are sometimes rather aggregated in macro-actions [Precup et al., 1998].

- The decomposition methods aim at decreasing the complexity of the MDP by splitting the original problem into smaller subproblems that are then solved independently. The elementary solutions are then combined in order to provide an approximately optimal solution to the global problem. The decomposition methods can be either serial [Dean and Lin, 1995] when the global state space is the union of smaller subspaces without much intercommunication, or parallel [Singh and Cohn, 1998] when it is a product of elementary spaces.

- Multi-agent reinforcement learning methods [Littman, 2001] combine Reinforcement Learning [Sutton and Barto, 1998] with multi-agent methods, used as a means of decomposing the initial problem.

In this paper, we first briefly recall some notions about MDPs (Section 2). Then we describe the simplified forest management model that we use as an illustration, and we show how it can be modeled and solved in the MDP framework (Section 3). It will become clear that classical methods do not perform well when the size of the problem (parameter N : number of stands) increases. In Section 4 we describe how three methods issued from the families just described, can be used to improve the resolution of the initial problem. Finally, in Section 5 we show numerical comparison in terms of size of the problems solved by the methods, time needed to compute solutions, and quality of the approximate solutions.

2. MARKOV DECISION PROCESSES

Markov Decision Processes model the dynamics of an agent interacting with a stochastic environment, through a sequence of decisions. The standard model [Puterman, 1994] consists of a state space S of size $\#S$, of a decision space D ($\#D$), of a Markovian dynamics described by transition probabilities $P_t(s'|s,d)$ of going from state s to s' when d has been performed at time $t \in I\!N$, and of local rewards $r_t(s,d,s')$ associated to each transition (s,d,s').

A policy π is defined as mapping from S to D, assigning to every state s an action $d = \pi(s)$. An initial state s_0 and a policy π determine a set of pos-

sible trajectories $s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_n \rightarrow \cdots$ to which are assigned probabilities $\Pi_i P(s_{i+1}|s_i, d_i)$. To each trajectory is assigned a sequence of rewards (r_i), with $r_i = r_t(s_i, d_i, s_{i+1})$. The optimization problem associated to a MDP consists in finding a policy π which maximizes (for every initial state) a value function defined as a measure of the expected sum of the rewards obtained throughout the execution of π. The most commonly encountered value function is the discounted criterion defined as: $\forall s \in S$,

$$V^{\pi}(s) = E\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, \pi(s_i), s_{i+1}) | s_0 = s\right].$$

where $0 \leq \gamma < 1$ is a discounting factor, allowing to increase the importance of present rewards, relatively to future ones. In general, finding $\pi^* = \text{argmax}_{\pi} V^{\pi}$ is closely linked to the computation of $V^* = \max_{\pi} V^{\pi} = V^{\pi^*}$.

Indeed, a fundamental result concerning MDPs is the existence of an optimality equation, known as Bellman's equation which fully characterizes the optimal value function [Bellman, 1957]. In the case of the discounted criterion, this equation takes the form: $\forall s \in S$

$$V^*(s) = \max_{d \in D} \sum_{s' \in S} p(s' \mid s, d)\{r(s, d, s') + \gamma V^*(s')\}$$

It can be shown that the solution of this equation is unique, and that knowing this solution allows to determine an optimal policy $\pi^* : \forall s \in S, \pi^*(s) =$

$$\underset{d \in D}{\text{argmax}} \sum_{s' \in S} p(s' \mid s, d)\{r(s, d, s') + \gamma V^*(s')\}$$

The two most classical algorithms for solving MDPs are the iterative algorithms Value Iteration and Policy Iteration. For a fixed γ, both algorithms converge in a polynomial (in $\#S$ and $\#D$) number of iterations, with respective complexity of $O(\#D\#S^2)$ and $O(\#D\#S^2 + \#S^3)$ per iteration. Experimentally, Policy Iteration is found to converge faster than Value Iteration.

3. FOREST MANAGEMENT MODEL

The problem that will illustrate the various approaches to solving "large" MDPs is a multi-stand forest management problem pervaded with uncertainty, in which we want to maximize the long-term expected revenue from timber sales. We model the problem in a MDP framework, following similar works from [Rapaport et al., 2001] and [Kennedy, 1998].

3.1 States of the System

The forest is composed of N homogeneous stands (same tree species), that can be of different sizes and shapes. On each stand n, $a_t^n \in \mathcal{A} = \{1, \cdots, A\}$ represents the age of trees[1], in years, or in tenth of years, depending on the species considered.

The state vector at time period t is:

$$s_t = (a_t^1, \cdots, a_t^N) \in S = \mathcal{A}^N$$

3.2 Decisions

At time t, we decide which stands will be clearcut within the next time period. The decision vector is:

$$d_t = (d_t^1, \cdots, d_t^N) \in \{0, 1\}^N$$

where $d_t^n = 0$ if stand n is clearcut, 1 if not. The cutting decision may not have the desired effect if a fire occurs within the $[t, t+1]$ period.

Another decision to be taken, is the fire protection expenditure level, $e_t \in \mathcal{E} = \{1, \cdots, E\}$, applied globally for the forest for the current period. e_t may consist of funds allocated to the fire tower network, road maintenance... It shall be noticed that e_t is the only factor that links the dynamics of the different stands, and thus prevents us from considering them as independent.

3.3 Transitions

Once a decision vector d_t has been chosen, it determines a transition function on the ages of the different stands. This transition function is stochastic, due to the stochastic nature of the fire event, modeled by the probability table $P_{fire}(n, a_t^n, e_t)$ indexed by stand number, age, and fire protection level. So, the dynamics for each stand is defined as follows:

- If $d_t^n = 0$ (clearcut), $a_{t+1}^n = 1$.
- If $d_t^n = 1$, the age of trees at the next time period depends on the fire event: if there is a fire (with probability $P_{fire}(n, a_t^n, e_t)$), $a_{t+1}^n = 1$ and if not, $a_{t+1}^n = min(a_t^n + 1, A)$ (trees grow older).

We then define a transition probability $P_{d_t^n, e_t}(a_t^n, a_{t+1}^n)$ for every stand. At the global level, due to the independence of the various stands, e_t being fixed:

$$P(s_{t+1}|s_t, e_t, d_t) = \prod_{n=1}^{N} P_{d_t^n, e_t}(a_t^n, a_{t+1}^n)$$

3.4 Outcomes from Timber Sales

Immediate outcomes originate from timber sales and are a function of the cutting decision on every stand,

1. We consider that all trees older than A keep the same properties, and thus need not be distinguished

and of the global fire protection level.

$$r_t = r(s_t, e_t, d_t, s_{t+1}) = -k(e_t) - k'(d_t) + \sum_{n=1}^{N} r_n(d_t^n, a_t^n, a_{t+1}^n, n),$$

with $k(e_t)$ the cost of protection, $k'(d_t)$ cutting costs, and $r_n(d_t^n, a_t^n, a_{t+1}^n, n)$ price of the whole stand timber stock if $d_t^n = 1$, and of the salvaged timber if there is a fire. $r_n(d_t^n, a_t^n, a_{t+1}^n) = 0$ if we choose not to cut and there is no fire.

4. SOLVING LARGE FOREST MANAGEMENT PROBLEMS

Our objective is to be able to deal with problems of about 50 stands and 6 age classes. This is yet from from real-world application needs, but as se will see in Section §5 it already challenges exact solving methods for MDP, therefore we explore approximation methods.

4.1 State Aggregation Methods

The idea is to decrease the sizes of the states and actions spaces by modifying the representation of these states and actions. Following [Kennedy, 1998] and [Rapaport et al., 2001], we assume that the forest is composed of N identical stands. This simplifying assumption allows to group states by age class and to adopt, for states and actions, the following representations:

4.1.1 States of the System

The state vector is now $s_t = (n_t^1, \cdots, n_t^A) \in S'$, with $n_t^a =$ number of stands aged a at time t. Note that we have $\forall t, \sum_{a=1}^{A} n_t^a = N$.

This is an aggregation technique, grouping all stands of age a in a unique state variable n^a, with the underlying assumption that differences between stands do not matter for the problem. It can be shown that the size of the modified state space is $\#S' = C_{N+A-1}^{N}$ [Cucala, 2001]. This size, in $O(N^{A-1})$, should be compared to the size of the original state space $\#S = A^N$. For instance, for $A = 5$ and $N = 6$, the size is reduced from 15625 to 210, and for $A = N = 10$, from 10^{10} to less than 10^5.

4.1.2 Decisions

Similarly, the size of the original decision space is $\#D = 2^N \times E$. We can also define an aggregated representation of the form $d'(s) = \{c^1, \ldots, c^A, e\}$, with $0 \leq c^a \leq n^a$, the number of stands aged a that will be cut.

With this representation, the size $\#D(s)$ of the decision space depends on the current state s:

$$\#D(s) = (n^1 + 1) \times \ldots \times (n^A + 1) \times E.$$

It can be shown that $\forall s, \#D(s) \leq (N/A+2)^A \times E$, that is $\#D(s) = O(N^A \times E)$.

4.1.3 Transitions

Transitions from t to $t+1$ are expressed differently in the aggregated model, for stands aged $a \leq A-2$ and for those aged $A-1$ or A.

- for $a \leq A-2$, if c_t^a stands are cut, there remains $n = n_t^a - c_t^a$ stands before fire events. Considering that each remaining stand has an identical probability of burning $P_{fire}(a, e_t)$ and that fire events are independent, a probability distribution over n_{t+1}^{a+1} can be defined : $\forall z \in 0 \ldots n$,

$$P(n_{t+1}^{a+1} = z | n_t^a - c_t^a = n, e_t) = C_n^z \times [P_{fire}(a, e_t)]^{n-z} \times [1 - P_{incendie}(a, e_t)]^z$$

- for older trees, functioning is a bit different, since at time $t+1$ stands aged A may come from stands previously aged $A-1$ as well as A. It can be shown that $\forall z \in 0 \ldots n$,

$$P(n_{t+1}^A = z | n_t^{A-1} - c_t^{A-1} = n_1, n_t^A - c_t^A = n_2, e_t) = P'(n_{t+1}^A = z_1 | n_t^{A-1} - c_t^{A-1} = n_1, e_t) \times P'(n_{t+1}^A = z - z_1 | n_t^A - c_t^A = n_2, e_t),$$

where $P'(n_{t+1}^A = z_1 | n_t^{A-1} - c_t^{A-1} = n_1, e_t) = C_{n_1}^{z_1} \times [P_{fire}(A-1, e_t)]^{n_1 - z_1} \times [1 - P_{fire}(A-1, e_t)]^{z_1}$ and $P'(n_{t+1}^A = z - z_1 | n_t^A - c_t^A = n_2, e_t) = C_{n_2}^{z-z_1} \times [P_{fire}(A, e_t)]^{n_2 - z + z_1} \times [1 - P_{fire}(A, e_t)]^{z - z_1}$.

Finally, the overall transition probability is defined as:

$$P(s_{t+1} | s_t, e_t, c_t) = \prod_{a=1}^{A-2} P(n_{t+1}^{a+1} | n_t^a - c_t^a, e_t) \times P(n_{t+1}^A | n_t^{A-1} - c_t^{A-1}, n_t^A - c_t^A, e_t).$$

4.1.4 Outcomes

Outcomes are also aggregated by age class, which leads to the following overall outcome formula:

$$r_t = r(s_t, e_t, d_t, s_{t+1}) = -k(e_t) - k'(d_t) + \sum_{a=1}^{A} r(c_t^a, n_t^a, n_{t+1}^{a+1}, a).$$

For every age class, $r(c_t^a, n_t^a, n_{t+1}^{a+1}, a)$ is the sum of the revenue from the timber cut, and the timber salvaged from fire. We get, for $a \leq A-2$:

$$r(c_t^a, n_t^a, n_{t+1}^{a+1}, a)$$

$$= price_1(a) \times c_t^a + price_2(a) \times (n_t^a - n_{t+1}^{a+1} - c_t^a).$$

The first term corresponding to the timber cut, and the second to the timber salvaged (where $price_1$ and $price_2$ are the respective timber prices).

For the two oldest age classes, the situation is a bit more difficult, since among the stands that burn at time t, it is impossible from the simple observation of c_t, n_t and n_{t+1} to determine which part was aged $A-1$ and which part was aged A. We get:

$$r(c_t^{A-1}, n_t^{A-1}, n_{t+1}^A, A-1) + r(c_t^A, n_t^A, n_{t+1}^A, A)$$

$$= price_1(A-1) \times c_t^{A-1} + price_1(A) \times c_t^A$$

$$+price_2'(A, A-1) \times (n_t^{A-1} - n_{t+1}^A - c_t^{A-1} - c_t^A).$$

Where the two first terms correspond to the return from timber cut, and the last term from the salvaged timber. $price_2'(A, A-1)$ is a combination of the timber prices for ages $A-1$ and A, with weights corresponding to the a priori probability that a burnt stand comes from age category $A-1$ or A: this is the only approximation of the model, which becomes then exact when $price_2(A-1) = price_2(A) = price_2'(A, A-1)$.

4.2 Decomposition Methods

The idea of decomposition methods is to divide the initial MDP into smaller problems that will be solved independently, the elementary solutions being combined to give an approximate solution to the initial problem. In the literature, decomposition methods can be split into serial decomposition and parallel decomposition methods.

Serial decomposition methods [Dean and Lin, 1995] are used when the state space can be considered as the union ($S = S_1 \bigcup S_2 \bigcup ... \bigcup S_N$) of elementary state spaces which are "weakly communicating". Each subset S_i is divided into $S_i = R_i \cup U_i$, where R_i is a set of inner states (from which by any action the system remains in S_i) and U_i is a set of frontier states (which do not posses the inner states property). Weakly communicating means that the U_i are small with regards to the S_i. Methods based on serial decomposition for solving MDPs generally consist in assigning arbitrary values to the frontier states and then iteratively solving the sub-MDPs on the S_is and updating the values on the U_is.

Parallel decomposition [Singh and Cohn, 1998] is used when the state space is the Cartesian product of sub-spaces: $S = S_1 \times S_2 \times ... \times S_N$. Generally, the decision space can be decomposed in the same way: $D = D_1 \times D_2 \times ... \times D_N$, or more generally $D \subseteq D_1 \times D_2 \times ... \times D_N$.

Our forest management problem belongs to the second category of problems, for which parallel decomposition methods may be adapted. Unfortunately, existing methods are only adapted for problems for which the sub-MDPs are only linked by the use of a common, limited resource (which restricts D), whereas in our problem the dynamics of the sub-MDPs are not independent, since they are all affected by the choice of the decision variable e_t, the global protection level. So we propose new parallel decomposition methods that adapts to the specificity of our problem.

4.2.1 Problem decomposition

In our forest management model, the only link between the stands is the global protection level e_t. The idea that we develop in this section is to break this link by allowing, as a first approximation, to use different levels of protection, $e_t^n \in \{1, \cdots, E\}$ in the different stands. Then the new global MDP $< S, D', P, R >$ can be solved exactly by solving the independent sub-MDPs $< S_n, D_n', P_n, R_n >$, where $S_n = \{1 \ldots A\}$, $D_n' = \{0,1\} \times \{1 \ldots E\}$ and P_n and R_n are defined as before. We get now local optimal policies $\pi^n : A \rightarrow \{0,1\} \times E$, and associated value functions $V^n : A \rightarrow I\!R$.

The question is now how to build a global policy from these local policies? If the protection levels were really independent, the simple union of the π^n would be optimal, unfortunately this is not the case, since a global policy shall have a unique prevention level. Several solutions may be explored:

4.2.2 Direct method

The first method consists in building an approximate global policy:

$$\pi_{app}(s_t) = \{\pi^1(a_t^1)^{\downarrow\{0,1\}}, \cdots, \pi^N(a_t^N)^{\downarrow\{0,1\}}\} \cup e_t$$

where e_t is chosen as

$$e_t(s_t) = argmax_{e \in E} \sum_{n=1}^{N} V^n(a_t^n) \times 1_{e_t^n(a_t^n)=e}.$$

where $e_t^n(a_t^n) = \pi^n(a_t^n)^{\downarrow E}$ is the local optimal protection level. Thus, $e_t(s_t)$ is a compromise between the different stands that contributes for the most to the global value function $V(s_t) = \sum_{n=1}^{N} V^n(a_t^n)$. This method is only heuristic, and only takes into account, for evaluating protection level e_t, the stands where it is found to be locally optimal.

A first way to improve this first choice is to adapt, in the stands in which e_t is not locally optimal, the current cutting decision.

4.2.3 Updating the cutting decision

Once the global protection level e_t is chosen as before, we may use the local value functions $V^n(a_t^n)$ just computed, in order to recompute "greedily" (using one computation step only) the local cutting decisions. This is done as follows:

$$d_t^n(a_t^n) = \underset{d\in\{1,2\}}{\text{argmax}} \sum_{a_{t+1}^n \in \{1,\cdots,A\}} \{r_n(d, e_t, a_t^n, a_{t+1}^n) + \gamma P_{d,e_t}(a_t^n, a_{t+1}^n) \times V^n(a_{t+1}^n)\}$$

Of course this new computation of d_t^n is done only for stands in which e_t^n is different from e_t.

This method improves the policy computed before, for an additional cost limited to the above argmax computation, limited to the local states (at most $N \times A$), for which the local fire protection level is different from the optimal.

4.2.4 Stochastic policy

Rather than choosing a fixed global protection level e_t for each global state s_t, we can choose a stochastic protection level obtained, for instance through $V_e = \sum_{n=1}^{N} V^n(a_t^n) \times 1_{e_t^n(a_t^n)=e}$:

$$P(e_t = e) = \frac{V_e}{\sum_{e'=1}^{E} V_{e'}}, \quad \forall e \in \{1, \cdots, E\}.$$

4.3 Reinforcement Learning

Reinforcement Learning (RL) consists in learning an optimal behavior through repeated experiences within an environment [Sutton and Barto, 1998]. It can also be seen as a convenient way to overcome some limitations of MDP, in two different directions:

- The use of simulation of the dynamic process to control, in order to direct the exploration of the state and action spaces. This translates into the use of iterative stochastic algorithms, typically used in RL.
- The use of structured or compact representations of value functions and policies, thus allowing to tackle with large, multi-dimensional problems.

4.3.1 The Q-learning algorithm

The value function V^π of a policy π can be directly learned from observed trajectories by using the TD(λ) algorithm, without maintaining an estimation of the transition probabilities $p(s'|s, \pi(s))$ [Sutton and Barto, 1998]. It is also possible to learn directly an estimation of the optimal value function of the problem with the algorithm Q-learning . Q-learning regularly updates an estimation Q_n of the optimal Q-value function denoted by Q^*: $\forall s, d$,

$$Q^*(s,d) = \sum_{s'\in S} p(s'|s,d)\left(r(s,d,s') + \gamma V^{\pi^*}(s')\right),$$

which is characterized by the optimality equations: $\forall s, d, Q^*(s,d) =$

$$\sum_{s'\in S} p(s'|s,d)\left(r(s,d,s') + \gamma \max_{d'\in A} Q^*(s',d')\right).$$

An optimal policy can then be directly derived from Q^* by $\pi^*(s) = \text{argmax}_{d\in D}\, Q^*(s,s)$, $\forall s$.

The principle of the Q-learning algorithm is to update, after every transition (s_n, d_n, s_{n+1}, r_n) the estimated value function Q_n for (s_n, d_n), accordingly to the update rule:

$$\begin{aligned} Q_{n+1}(s_n,d_n) \leftarrow & \quad (1-\alpha_n)Q_n(s_n,d_n) \\ & + \alpha_n\{r_n + \gamma \max_b Q_n(s'_n, b)\} \end{aligned}$$

where the learning rate α_n decreases to 0 when n increases. The convergence of the Q-learning algorithm to the optimal policy is established under general hypotheses.

4.3.2 Multi-agent Reinforcement Learning

As we have seen, the Q-learning algorithm may avoid in the forest management problem to store explicitly the global transition matrix P. Nevertheless, it needs to store the Q function, of size $E(2A)^N$, which can not be done for large values of N.

This motivates the use of a multi agent algorithm, as advocated by [Littman, 2001]. The idea underlining the multi agent approach is to consider that each stand is managed independently by an agent, the fire protection level being managed by a $N+1^{th}$ agent. Of course, the motivation of this approach is to limit the size of the memory needed for storing the $N+1$ $Q-$functions. This implies limiting the information available to each agent.

After a first analysis, it appeared that the following factors were important for the stand agents: age a_t^n of trees in stand n, average age $\overline{a}_t$ of trees in the forest, number of stands cut in the preceding period Cut_{t-1} and current prevention level e_{t-1}. Concerning the prevention agent, an aggregated representation of the ages of trees ($\{n_t^1, \cdots, n_t^A\}$) would be an important factor, as well as the current prevention level e_{t-1}.

The global size needed to store the Q-functions is then in $O(E \times (A^2 \times N^2 + N^{A-1}))$. The limiting factor being the protection agent's need for the

age repartition profile. We used the multi-agent Q-learning algorithm [Littman, 2001] to solve the problem. In its spirit it is very close to the original Q-learning algorithm, each stand agent choosing an action, updating the global state, then the fire protection agent chooses an new protection level. Then, all Q-functions are updated.

5. NUMERICAL COMPARISONS

The global MDP model of the forest has A^N states and $2^N \times E$ decisions. As was already mentioned, the size of the problem grows exponentially with N and A. This prevents us from comparing experimentally the above methods for too large values of A.

So, we made two series of tests, varying the N parameter, for the two following configurations: **C1** $A = 3$; $E = 2$, and $A = 6$; $E = 3$. Then, for the various algorithms we tested the limit number of stands (N_{max}) for which a solution could be found. Furthermore, for $N = 5$ we computed the time needed to obtain the solutions, for cases **C1** and **C2**. For the case **C1** for which an estimation of the real value function could be computed, we computed the quality of the solutions returned by the approximate algorithms. The quality criterion was the average value per stand of the policies over S.

$$\rho^\pi = \frac{1}{N}\frac{1}{\#S}\sum_{s\in S} V^\pi(s).$$

The values V^π and ρ^π where computed exactly when possible, or estimated [Garcia and Serre, 2000] by the ATD algorithm. Computation was performed using MATLAB (Mathwork, Inc) in a Linux environment. Results are listed in Table 1.

Table 1: Comparison results.

Method	N_{max} **C1**	N_{max} **C2**	T **C1**	T **C2**	ρ **C1**
Exact	5	4	1277.5	-	100%
Agreg.	10	6	11.21	2317.45	100%
Direct	13	8	0.21	2.85	95%
Updated	13	8	0.67	40.55	97%
RL	> 100	13	19.8	39.4	96%

6. CONCLUSIONS

The results of this study are still preliminary, but conclusions can already be drawn at this stage. First of all, aggregation methods may not be sufficient by themselves for solving multidimensional MDPs with a large dimension. At that point, decomposition methods, eventually coupled with RL methods may be preferred. Concerning the decomposition methods, which are the most efficient in terms of computation time, the space limitation (which limits the number of stands) comes from the representation of the protection vector, of size $O(A^N)$. For the RL methods, the space limitation also comes from the representation of the policy of the prevention agent, which is in $O(N^A \times E)$. For these two kinds of method, more care should be given to the representation of the fire protection policy, and maybe less information could be sufficient for providing a satisfying approximate policy.

Another difficulty comes from the evaluation of the RL policies that are obtained for large N values. At the moment it is difficult to conclude anything on the quality of the obtained policies.

7. REFERENCES

Bellman, R. E., *Dynamic Programming*, Princeton University Press, 1957.

Cucala, L. Résolution de processus décisionnels de Markov de grande taille faiblement couplés, INRA-Rapport de stage ingénieur INSA, 2001.

Dean, T., and S. H. Lin, Decomposition techniques for planning in stochastic domains, In: Proc. IJCAI'95, Montreal, Canada: Morgan Kaufman, 1121–1127, 1995.

Dearden, R., and C. Boutilier, Abstraction and approximate decision theoretic planning, *Artificial Intelligence* 89:219–283. 1997.

Garcia, F., and F. Serre, Efficient asymptotic approximation in temporal difference learning, In: Proceedings ECAI'2000, 296–300, 2000.

Kennedy, J. O. S. Optimal strategies for protection of forests providing timber and non timber outputs, In: First world Congress on environmental and resources economists, 1998.

Littman, M. L. Value-function reinforcement learning in markov games, *Journal of Cognitive Systems Research* 2:55–66, 2001.

Precup, D., R. Sutton, and S. Singh, Theoretical results on reinforcement learning with temporally abstract behaviors, In: Proc. ECML'98, 382–393, 1998.

Puterman, M. *Markov Decision Processes*, New York: John Wiley and Sons, 1994.

Rapaport, A., L. Doyen, and J. Terreaux, Sustainability analysis for a foretry management model, In: Proc. 3rd European Conference EFITA, volume 2,385-390, 2001.

Singh, S., and D. Cohn, How to dynamically merge markov decision processes. In: Advances in Neural Inform. Process. Systems, Cambridge, MIT Press, volume 10, 1057–1063, 1998.

Sutton, R. S., and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, Massachusetts: MIT Press, 1998.