Approximating Spatial Markov Decision Processes For Environmental Management

R. Sabbadin

Unité de Biométrie et Intelligence Artificielle, INRA, Toulouse, France
E-mail: fgarcia@toulouse.inra.fr

Abstract: In this paper we will focus on spatialized decision problems which we propose to model in the framework of (highly) multidimensional Markov Decision Processes (MDPs) which exhibit only local dependencies between variables. We propose to approximate a Markov chain on a multidimensional random variable by a Markov chain on a set of weakly dependent random variables. This allows to (approximately) solve multidimensional MDPs with hundreds of variables, to the price of a loss of exactness of the process model. The method is mostly empirical yet, however it allows to deal with decision problems far larger than the one usually dealt with in the MDP framework.

Keywords: *Markov Chains; Markov Decision Processes; Approximation; spatial multistage decision problems; environmental management.*

1. INTRODUCTION

Markov Decision Processes (e.g. Puterman, 1994) are commonly used for modeling and solving sequential decision problems under uncertainty in Artificial Intelligence. However, environmental management problems can not be easily modeled and solved in this framework, due to the high dimensionality of their state and action spaces which put them out of reach from the usual enumerative Dynamic Programming algorithms. This dimensionality problem is especially present when spatial features of the underlying processes are taken into account in the management problem, which is often the case in environmental management problems such as weeds dispersal control, fire protection or animal populations dispersal control.

Several ways to deal with the dimensionality problem in MDPs have been proposed in the past. They have in common the fact that they exploit "independence" or "weak dependence" between state variables of the process. Among these methods we can point out *State aggregation methods* (Dearden and Boutilier, 1997), *State space decomposition methods* (Dean and Lin, 1995), *Multi-agents Reinforcement Learning* (Litman, 2001) and *Bayesian Networks* (Pearl, 1988). See (Garcia and Sabbadin, 2001) for pointers to references.

In this paper we propose an approximation method for multidimensional Markov chains which can be used for approximately solving Markov Decision Processes. Although different from the approaches we have just quoted, it is also a kind of decomposition method. We will propose to approximate the Markov chain on a multidimensional random variable by a Markov chain on a set of "weakly dependent" random variables. This will allow to tackle problems with hundreds of variables, to the price of a loss of exactness in the process model. In the next Section we will briefly introduce Markov Chains and Markov Decision Processes. In Section 3 we will describe Multidimensional Markov chains and the dimensionality problem on a simple fire propagation example. In Section 4 we will describe the approximation method we propose as well as an empirical validation method. Finally, in Section 5 we will show how our Markov Chain approximation scheme can be used in order to represent and solve spatialized decision problems.

2. MARKOV CHAINS AND MARKOV DECISION PROCESSES

2.1 Discrete Time Markov Chains (DTMC)

Let us give some basic definitions concerning Markov Chains:

Definition 1 (Discrete Time Markov Chains) *Let S be a set of states and H the horizon (H is a finite or countable set of time steps). A discrete time Markov chain is a set* $(X_t)_{t \in H}$ *of random variables such that* $X_t \in S$, $\forall t$.

Definition 2 (Homogeneous Markov Chains) *An Homogeneous Markov Chain is a Discrete Time Markov Chain which verifies the following property:*

$\forall t, \forall x_0,...,x_{t+1} \in S,$

$P(X_{t+1}=x_{t+1} | X_0=x_0,...,X_t=x_t) = P(X_{t+1}=x_{t+1} | X_t=x_t).$

$P(X_{t+1}=x_{t+1} | X_t=x_t)$ is denoted p_{ij}^t if $x_t=i$ and $x_{t+1}=j$. The Markov Chain is homogeneous if p_{ij}^t does not depend on t.

The transition matrix of an homogeneous Markov chain is the matrix $P=(p_{ij})_{i,j \in S}$.

Definition 3 (accessibility, recurrent states, recurrent class) *A state j is* accessible *from a state i iff there exists a finite n, and a sequence of states $i_1=i,...,i_n=j$ such that every transition $i_k \rightarrow i_{k+1}$ has a positive probability. A state i is* recurrent *iff for every j accessible from i, i is also accessible from j. If i is recurrent, A(i), the set of states which are accessible from i forms a* recurrent class*.*

Definition 4 (periodicity) *A recurrent class R of a Markov chain is* periodic *iff there exists a partition $S_1,...,S_m$ (m>1) of R such that all transitions from S_k lead to S_{k+1} if $k \neq m$ and to S_1 if k=m. The class is* aperiodic *iff it is not periodic. A Markov chain is periodic (resp. aperiodic) iff it contains at least one (resp. no) periodic class.*

We may also be interested in the long-term behavior of a Markov Chain, i.e. X_∞ (in this case H=∞).

Proposition 1 (limit behavior of an aperiodic Markov Chain) *If $(X_t)_{t \in N}$ is aperiodic, $X_\infty = \lim_{n \rightarrow \infty} X_n$ exists and $X_\infty = \lim_{n \rightarrow \infty} P^n.X_0$.*

Furthermore if the chain possesses a unique recurrent class*, X_∞ is independent from X_0.*

2.2 Markov Decision Processes (MDP)

The standard *discounted MDP model* (Puterman, 1994) is defined by a tuple (S, A, P, R). The horizon H is either finite or infinite, S is the finite set of possible states, A is the finite set of available actions, $P : S \times A \times S \rightarrow [0,1]$ the transition probability function (P(i,a,j) is the probability that j results from i when a is applied), $R : S \times A \rightarrow \Re$ is a reward function. A *deterministic* policy $\pi : S \times \{0..H\} \rightarrow A$ is a mapping from states and time steps to actions. Policies may be *stationary*, in which case π is independent of *t*.

The *discounted value of a policy* in a given state s_0 is defined by :

$$V^d_\pi(s_0) = E[\Sigma_{t=0..H} \gamma^t.R(s_t, \pi(s_t))] \quad (1)$$

where $0<\gamma\leq1$ is the discounting factor (when H=∞, $\gamma<1$ so that the sum converges).

The average value of a policy is defined as :

$$V^a_\pi(s_0) = \lim_{T \rightarrow H} (1/T).E[\Sigma_{t=0..T} R(s_t, \pi(s_t))] \quad (2)$$

In terms of discrete Markov Chains, let P_π be the transition matrix associated to a policy π ($P_\pi(i,j) = P(i, \pi(i),j)$). Then, to π we can associate a Markov chain $(X_t^\pi)_{t \in N}$ such that $X_t^\pi = P_\pi^t.X_0$, $\forall t \geq 1$. It can be shown that $V^a_\pi(s) = \lim_{t \rightarrow H} \Sigma_s X_t(s).R(s, \pi(s)) = \Sigma_s X_H(s).R(s, \pi(s))$ (see, e.g. (Altman, 1999)). $V^d_\pi(s)$ can also be expressed as limit of the integral of R with respect to a random variable, but in this case, the Markov process is not stationary.

Now that we have recalled that the solution of a MDP can be obtained from the limit of a Markov chain, we will focus for a while on the approximation of multidimensional Markov chain limits, before we come back to MDPs in Section 5. But first, let us briefly recall why dimensionality is a problem through an example.

3. MULTIDIMENSIONAL MARKOV CHAINS

3.1 Example of multidimensional Markov chain

We will model an example of fire spread on an area with different soil occupancies (forest, grass, lakes...) through the use of a multidimensional Markov chain. The area is represented by a grid (Figure 1), and to every cells are associated soil occupancies, which have different probabilities of: fire ignition, fire extinction and complete burning. In addition, fire diffusion probabilities are assigned to the different directions (north, east...), reflecting the effect of the wind. Actions are not considered in the example, but could be through their effects on the different probabilities and coefficients.

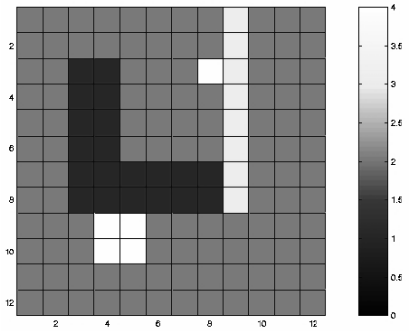

Figure 1. Soil occupancies, dark to clear, forest, grass, water and hazard zone.

The four soil occupancies are: forest, grass, water and fire hazard zones. α, β, and γ are respectively the ignition, extinction and complete burning probabilities (Table 1).

	Forest	Grass	Water	Fire hazard

α	0	0	0	0.1
β	0.1	0.4	1	0.2
γ	0.1	0.3	0	0.2

Table 1. Soil occupancies parameters

When describing the fire status of the grid, each of the 144 cells can take one of the following states: 1=no fire, 2=burning, 3=burnt. So, the global status is described by the global variable $x=\{x_1,...,x_N\}$ (N=144). Since the fire evolution is random, it will be represented through the global random variable $X_t : x_1,...,x_N \rightarrow [0,1]$ which has 3^N components. We want to determine Γ, the matrix of the Markov process governing X_t. In order to do this, we first determine the relation $X_{t+1}=f(X_t)$ and then express it in matrix form $X_{t+1}=\Gamma.X_t$, following the steps :

$X_t \rightarrow$ (local) $Y_t = \gamma(X_t) \rightarrow$ (diffusion) $X_{t+1} = \delta(Y_t)$.

The fire diffusion probabilities, corresponding to a wind coming from the west, are : East = 0.35, North = South = 0.2, West = 0.1.

3.2 Local evolution

The starting parameters are : ignition probability $\alpha_i=P(Y_t^i=2|X_t^i=1)$, extinction probability $\beta_i=P(Y_t^i=1|X_t^i=2)$ and complete burning probability $\gamma_i = P(Y_t^i=3|X_t^i=2)$.

Thus, local transitions can be represented by $P_i = \begin{bmatrix} 1-\alpha_i & \beta_i & 0 \\ \alpha_i & 1-\beta_i-\gamma_i & 0 \\ 0 & \gamma_i & 1 \end{bmatrix}$ and $Y_t^i = P_i.X_t^i$

The transition probability from $(x_1,...,x_N)$ to $(y_1,...,y_N)$ is

$$Y_t(y_1,...,y_N) = \gamma(\{x_1,...,x_N\})_{(y1,...,yN)} = P_1(y_1|x_1) . P_2(y_2|x_2) ... P_N(y_N|x_N).$$

More generally,

$$Y^t(y_1,..,y_N) = (\gamma(X_t))_{(y1,..,yN)} = \sum_{(x_1,...,x_N)\in\{0,1,2\}^N} X_t(x_1,...,x_N).\prod_{i=1}^{N} P_i(x_i,y_i) \quad (4)$$

3.3 Diffusion

The parameters are the fire diffusion probabilities from cell i to j : $(d_{ij})_{(ij)\in\{1..N\}^2}$,

where $d_{ij}=P(x_j^{t+1}=2|y_i^t=2, y_j^t=1)$.

Let $I_j = \{i_1,...,i_{pj}\} = \{i / d_{ij}>0\}$ be the set of cells from which there can be a diffusion to cell j. It will be assumed that $|I_j| << N$. We have $P(x_j^{t+1}=2|y_1,...,y_N) = P(x_j^{t+1}=2|y_{i1},...,y_{ipj}, y_j)$. If we let $d_j= \Pi_{i\in I,yi=2} (1- d_{ij})$ ($d_j=1$ if $\{i\in I, y_i=2\}$ is empty), then:

$P(x_j^{t+1}|y_{i1},...,y_{ipj}, y_j)$ can be expressed as a matrix : $Q_j = \begin{bmatrix} d_j & 0 & 0 \\ 1-d_j & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Finally,

$X_{t+1}=\delta(Y_t)=\sum_{(y1...yN)} Y_t(y_1...y_N).\Pi_{j=1..N} Q_j(x_j^{t+1},y_j)$ (5)

3.4 Monodimensional Markov Process

From now on, we have the following relation on multidimensional (dimension N) random variables : $X_{t+1}=\delta(\gamma(X_t))$. It is possible to express it as a stationary Markov Process over monodimensionnal random variables over a finite state space of cardinal 3^N ($X'_{t+1} = P'.X'_t$) thanks to the following bijection φ : $\{0,1,2\}^N \rightarrow \{0,1,2,...,3^N\}$, $(x_1,...,x_N) \rightarrow x'=\sum_{i=1..N} x_i. 3^i$.

However, the matrix P' is of dimension $3^N \times 3^N$, which makes the process useless for N greater than a dozen, far from what we would like to be able to handle (several hundreds)! This motivates the approximation method that we suggest next.

4. APPROXIMATING MULTIDIMENSIONAL MARKOV CHAINS

4.1 Multidimensional approximation model

As we have seen, the size of the random variable involved in the N-dimensional Markov process is 3^N. As this prevents us from modeling realistic problems, we propose to approximate the N-dimensional variable X_t by a product $X'_t=\delta(Y_t)=\Pi_{i=1..N}X_t^i$ of "independent" variables (the approximation process is described in Figure 2).

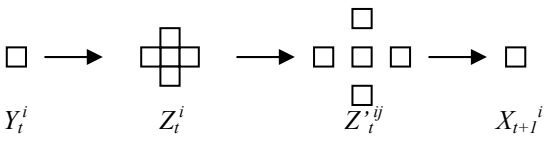

Figure 2. Multidimensional Markov chain approximation.

The diffusion equation can be written (with a renumbering of the neighbor cells from 1 to 5) :

$Z_t^i(z_1...z_5) = f(Y_t^i) = \sum_{y=1..3} Y_t^i(y).\Pi_{j=1..5} P^i(z_j|\ y)$ with

$$P^i(z_j|\ y) = \begin{pmatrix} 1 & 1-d_{ij} & 1-\delta(j,1) \\ 0 & d_{ij} & \delta(j,1) \\ 0 & 0 & 0 \end{pmatrix}. \quad (6)$$

$\delta(j,1)=1$ if $j=1$, 0 else.

The principle of the approximation that we use is the following. We write $Z'^i_t(z_1...z_5)$ as a product of independent probabilities.

$$Z'^i_t(z_1...z_5) = \Pi_{j=1..5} Z'^{ij}_t(z_j)$$
$$= \Pi_{j=1..5} \sum_{y=1..3} Y_t^i(y).P^{i'}(z_j|\ y) \quad (7)$$

Where $P^{i'}(z_j|\ y) = P^i(z_j|\ y)$ with d'_{ij} replacing d_{ij}.

4.2 Simple diffusion approximation

As a first approximation, we just let $d'_{ij} = d_{ij}$. In order to test the result, we compared on 3×3 spaces (N=9) the results of the approximate and exact processes (the exact process is simulated through a Monte-Carlo (MC) method) over an horizon of T=300. What we compared was, for different values of α, β, γ and d_{ij}, the maximal difference of probability of a cell being burnt as computed by the exact and approximate processes. The result was not very good: the maximal difference in probability was around 0.3, which led us to look for another definition of $d'_{ij}=g(d_{ij})$. The result on the example of Figure 1 is given on Figure 3 (second grid).

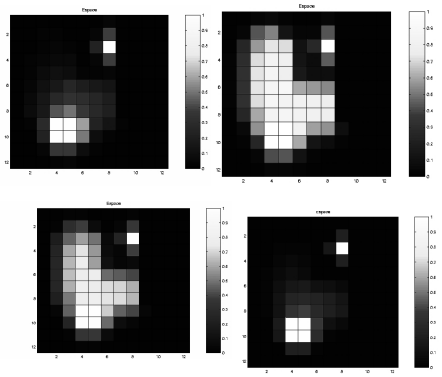

Figure 3. Probabilities of cells being burnt (after 150 time steps). From left to right, top to down : Exact probabilities (Monte-Carlo simulation), approximate probabilities (no correction for d_{ij}), simple approximation and burning speed adjustment.

4.3 Correction of the diffusion parameter

The first correction we proposed was to take $d'_{ij}=g(d_{ij})$ such that $||Z'^i_t(z_1...z_5) - Z^i_t(z_1...z_5)||$ is minimal for a β, γ combination for which the simple approximation gave the worst results (β=0.1, γ=0.1). This gave for example, g(0.1)=0.091 ; g(0.2)=0.164 ; g(0.3)=0.234.

Unfortunately, this method did not greatly improved the result on the 144 cells grid (Figure 3, third grid). This is why we proposed another method of correction, based on burning speed adjustment. In this method, we plot the ES(t) and ES'(t), expectations of the surface burnt on a on 3×3 space for the forest parameters (for which the simple approximation gave the worst results), in the exact and approximate case ($ES(t)\in[0,9]$ and $ES'(t)\in[0,9]$). In both cases, the curves are linear for the early steps. Thus, g is built such that the two initial burning speeds correspond. This gives, on the example, g(0.1)=0.055 ; g(0.2)=0.084 ; g(0.3)=0.126. On the 144 cells example, the result is quite good, as shown in Figure 3, fourth grid.

4.4 Large neighbourhoods diffusion approximation

Up to now, we assumed that the neighbourhoods considered in the diffusion process were small enough in order to express and compute the value of $Z_t^i(z_1...z_5)$ for all possible combinations of $z_1...z_5$. However, this may not be possible for larger neighbourhoods, even for ones with only 9 or 16 elements!

For these larger neighbourhoods it would be more convenient to compute Z'^{ij}_t directly from Y_t^i by simulation, without computing the intermediate value Z_t^i. In order to do this, we can follow the following procedure :

```
For each i=1..N do
  For k=1..K  % K is a huge constant;
      Draw y_t^i from Y_t^i;
      Draw every z_j from y_t^i and P^i(z_j| y_t^i);
      Update Z'_t^ij;
  End;
End.
```

In this way, there are no more limitations on the size of the neighborhoods we can handle for approximating the diffusion process. Considering the time and space complexities of the approximation, We have :

- Time complexity of computing every Z'^{ij}_t : O(N.K.I) (where I is the size of the biggest neighborhood)
- Space complexity : O(N.I).

Once again, the approximation Z'^{ij}_t obtained through simulation can be improved by comparing

the resulting approximate process with a Monte Carlo simulation of the exact model.

5. DECISIONS

Up to now we limited our study to the approximation of Markov chains evolution. However, our aim is to use this approximation scheme for solving decision problems.

It is clear that once a policy (i.e. a mapping from global states to actions) is fixed, a Markov chain results which we can study in order to evaluate the policy. To be more precise, we will restrict ourselves to finite-horizon problems with horizon H, and additive terminal-state reward functions :

$$R(x_t^1,...,x_t^N)=\sum_{i=1..N} R_i(x_t^i) \text{ if } t=H \text{ and } 0 \text{ else} \quad (8)$$

Then, with this definition,

$$V_\pi(X_0)=E[R(x_H^1,...,x_H^N)|\pi] =\sum_{i=1..N} {}^tX_H^i.R_i \quad (9)$$

which is easy to compute.

The trouble is that although we have managed to overcome (to the price of drastic simplifications) the problem of state space dimensionality, we have not yet solved the one of action space dimensionality : to every cell of the state space representation may correspond several actions, which means that the number of global actions available at each time step is exponential in the number of cells N. This is not to say about the policy space, i.e. the number of states to actions mapping, of size $|S|^{|A|}$, doubly exponential in N (in our simplified case!). Clearly, it is unrealistic to perform policy optimization in the whole policy space, and in what follows we will restrict our study to two limited subsets of available policies, which we illustrate on the fire example : *static decisions* and *dynamic unconditional decisions*.

5.1 Static decisions

In this case, in order to limit the size of the policy space to explore, we limit ourselves to studying the set of static policies, that is action choices which depend neither on the current state of the world, nor on the time step. In the fire example, this would correspond to the problem of choosing the implantation of fire towers : this is done once and for all and the choice is not modified depending on the current status (burning, burnt...) of the different cells.

With this assumption, the policy space is equal to the action space, of size $|A|^{|N|}$ (for each cell i, we have to choose among |A| alternatives). Furthermore, the static decision problem is more often posed in the following terms : you have k fire towers to locate among N cells, which generates an action space of size N!/(k!(N-k)!). This still makes a lot of policies to evaluate. However, we are now in front of a classical combinatorial optimization problem : to each possible configuration of fire towers is associated a value, the expected surface burnt at time step H (more generally, $E[R(x_H^1,...,x_H^N)|\pi]$), and we want to maximize or minimize this value. Just any discrete optimization algorithm (such as genetic algorithms) can be used in theory.

Practically still, due to the big amount of time needed to perform a single policy evaluation, discrete optimization algorithms may not be efficient by themselves, and further means of limiting the action space are needed. These means of decreasing the search space are certainly problem dependent, and for the fire problem for example, one such way would be to divide the N cells into k equal sets within which a single fire tower is to be placed. Then a possible way of finding an optimal location is to successively optimize the location of each fire tower in its subset in turns, the other towers location being fixed. Once the k towers locations have been optimized, we come back to the first one, and so on, until a locally optimal configuration is found. This may need a number of evaluations in the order of several Ns, which seems to be acceptable.

5.2 Dynamic unconditional decisions

Another decision problem that could be solved in the spatialized framework is that of finding dynamic but "unconditional" policies, i.e. independent of the current fire status of the cells, in our fire example.

For this simplified version of the global MDP resolution problem, the point is to find a sequence of actions $\pi=(a_1,...,a_H)$ (i.e. a policy dependent on the time step, but independent of the state of the system), which allows to optimize the value function $V_\pi(X_0)$.

In the fire example, the problem could be to find a sequence of moves for a fire squad initially placed in cell i (moves are between adjacent cells), which helps to minimize the expected surface burnt in the long run (or at a finite horizon H). We can suppose that the presence of the fire squad in a cell improves the probability of extinction of cells in a given neighborhood. In this case, we can define an approximation $X_{t+1} = f_a(X_t)$ for the multidimensional Markov chain which is similar to that of Section 4, except that the transition function f_a now depends on the current action a (i.e.

position of the fire squad after its move). Then, we can represent the problem of finding an unconditional sequence of actions minimizing the expected surface burnt at time horizon H as a tree-search problem :

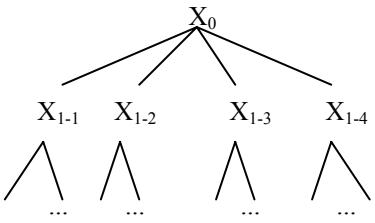

Figure 4. Tree-search problem

The branching factor of the tree is the number of available actions. Branches are labeled by actions and nodes are the belief states X_{t-i} resulting from the application of decision i at time step t-1. To each node X_{t-i} can be associated a value $v(X_{t-i})= \sum_{k=1..N} X_{t-i}^{k}(2).s_k$ where s_k is the surface of cell k. In other terms, $v(X_{t-i})$ is the expected surface burnt.

Equivalently, this node-valued tree representation can be replaced with a branch-valued tree representation : let $k(X_{(t-1)-j},X_{t-i})= v(X_{t-i})- v(X_{(t-1)-j})$. $k(X_{(t-1)-j},X_{t-i})$ is simply the expected area that becomes burnt between t-1 and t, applying i. Clearly, $k\geq 0$ since burnt cells remain burnt whatever happens. So, we are left with the problem of finding a minimum cost path in a tree, where the cost of a path is the sum of local costs which are all positive or zero. Classical search methods, such as the A* algorithm can be used.

5.3 General case

The general case for decision handling is when decisions are both dynamic and "state dependent". The problem with this case is that even before optimizing policies (mapping from states to actions), the representation of a single policy is problematic! In this case, it should be advantageous to use parameterized representations of policies (e.g. taking the area and center of gravity of burning surface as input parameters) considered as stationary, and to perform optimization of the parameters (e.g. using simulation)...

6. CONCLUDING REMARKS

We proposed an approximation method for highly dimensional Markov chains representing growth-diffusion processes. The method was illustrated on a fire propagation example, but can be applied to many other spatial environmental examples: weeds propagation, animal population dynamics, soil erosion...

We have proposed ways of handling decisions and decision optimization in this approximate framework, however, we have not tested the various decision optimization techniques of Section 5 yet.

One weak point of the method is its lack of accuracy in the diffusion model, however we proposed an improvement through parameters adjustments which gave good results. More generally, the parameters can be empirically fine-tuned on small sub-parts of the global process which is modeled.

The good point, on the other hand, is the ability of the method to tackle very large problems: in the paper we dealt with a 144 cells problem in order to evaluate the results through a Monte Carlo method, but we although dealt with problems of 4900 cells (70×70), which took about eight hours of CPU time to be solved (convergence of the chain was obtained after 130 steps). Furthermore, the space complexity of the approximate method is linear in the number of cell, where an exact method is exponential!

Next, we will extend the method to spatially explicit population dynamics models, which will not bring new theoretical difficulties. Our practical objective is to assess the impact of land use change (deforestation, reforestation) on the dynamics of birds populations in the south of France (regional project). The main difference with our current fire toy example is in the increased size of the neighborhoods in the diffusion process : it requires the use of simulation in the approximation scheme.

7. REFERENCES

Altman, E., 1999. *Constrained Markov Decision Processes*. Chapman & Hall/CRC.

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

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

Garcia, F. and Sabbadin, R., 2001. Solving large, weakly-coupled Markov Decision Processes: Application to forest management. International Congress on Modelisation and Simulation (MODSIM'01), vol. 4, pp. 1707-1712, Canberra, Australia.

Pearl, J., 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Palo Alto.

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