

Résolution approchée de Processus Décisionnels Markoviens de grande taille

Approximately solving large Markov Decision Processes

L. Cucala

F. Garcia

R. Sabbadin

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

Résumé

Le cadre des Processus Décisionnels de Markov (PDM) s'est généralisé comme cadre de représentation et de résolution de problèmes de décision séquentielle, dans la communauté IA, ces dernières années. Toutefois, il reste difficile de traiter dans le cadre des PDMs, des problèmes réalistes, dont les espace d'états et de décisions sont multidimensionnels et donc de très grande taille ($> 10^6$ états). Pourtant, ces problèmes peuvent souvent être représentés de manière plus concise et décomposés en sous-problèmes relativement indépendants (ils sont alors dits "faiblement couplés"). L'objectif de cet article est de recenser les différents types de méthodes qui ont été proposées récemment dans la communauté, pour traiter ces "grands" problèmes faiblement couplés, et de les illustrer sur un problème concret de gestion forestière "simplifiée".

Mots Clef

Processus Décisionnels de Markov, problèmes faiblement couplés, apprentissage par renforcement.

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 (simplified) real-world forest management problem.

Keywords

Markov Decision Processes, weakly coupled problems, Reinforcement Learning.

1 Introduction

Le cadre des Processus Décisionnels de Markov (PDM) [21] s'est généralisé comme cadre de représentation et de résolution de problèmes de décision séquentielle, dans la communauté IA, ces dernières années. Toutefois, il reste difficile de traiter dans le cadre des PDMs, des problèmes réalistes, dont les espace d'états et d'actions sont multidimensionnels et donc de très grande taille ($> 10^6$ états). Pourtant, ces problèmes peuvent souvent être représentés de manière plus concise et décomposés en sous-problèmes relativement indépendants (ils sont alors dits "faiblement couplés").

Parmi les approches proposées ces dernières années pour résoudre de tels problèmes, on peut distinguer trois grandes catégories :

- Les méthodes à base d'agrégation d'états.
- Les méthodes de type décomposition des espaces d'états et décisions.
- Les méthodes de type apprentissage par renforcement multi-agents.

Le principe des méthodes de type "agrégation d'états" consiste à regrouper certains états et décisions par caractéristiques communes, et ainsi de réduire la taille des espaces d'états et de décisions du problème [4, 7, 10]. Souvent, ces méthodes sont couplées avec des représentations factorisées (par variables) des transitions entre états, ou des fonctions récompenses [14, 3], dans des structures de type Réseaux Bayésiens [19, 8]. Dans la même catégorie, on peut classer les approches de type "macro-décisions", agrégeant cette fois les décisions en politiques complexes [12, 20]. Les méthodes de type "décomposition", visent à réduire la complexité du PDM étudié, en le décomposant en sous-problèmes résolus séparément, les solutions élémentaires étant ensuite recombinaées. Ces méthodes peuvent elles mêmes être séparées en :

- Méthodes *sérielles*, lorsque l'espace d'états est la réunion de sous-espaces d'états, faiblement communicants

(pour toute politique, la probabilité de sortir d'un sous-espace donné est faible) [9, 12, 18]

- Méthodes *parallèles*, lorsque l'espace d'états est un produit cartésien de sous-espaces [16, 23].

Les méthodes de type Apprentissage par Renforcement, surtout lorsqu'elles sont "directes", sont déjà aptes à traiter des problèmes de plus grande taille que les méthodes classiques de résolution des PDMs, puisqu'elles ne nécessitent pas de stocker les probabilités de transition. Elles peuvent également être conjuguées avec des méthodes de type "agrégation", ou "décomposition" [2].

Dans cet article, nous allons tout d'abord rappeler brièvement quelques notions sur les PDMs (Section §2). Ensuite, nous décrivons le modèle simplifié de gestion forestière, qui nous servira d'illustration, et nous montrerons comment il peut être modélisé et résolu dans le cadre des PDMs (Section §3). Nous constaterons que les méthodes classiques sont rapidement limitées lorsque la taille du problème, représentée par le paramètre N (nombre de parcelles) augmente. Enfin, dans la Section §4 nous décrivons 3 méthodes de résolution approchée, issues des trois domaines décrits ci-dessus, et nous montrerons les améliorations successives qu'elles apportent à la résolution du problème modélisé.

2 Processus Décisionnels Markoviens

Les processus décisionnels de Markov (PDM) forment un modèle de la dynamique d'un agent en interaction avec un environnement stochastique à travers une séquence d'étapes de décision. Le modèle standard que nous considérons ici [21] est décrit par un espace d'états S de taille $\#S$ et un espace de décisions D de taille $\#D$, par une dynamique markovienne dans S caractérisée par des probabilités de transition $P(s' | s, d)$ de passer de l'état s à l'état s' après avoir exécuté la décision a à l'instant $n \in \mathbb{N}$, et par les récompenses locales $r(s, d, s')$ associées à chaque transition (s, d, s') .

On définit une politique comme une fonction π de S dans D qui à tout état s associe une décision à exécuter $a = \pi(s)$. Etant donné un état initial s_0 , suivre une politique π détermine un ensemble de trajectoires possibles $s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n \rightarrow \dots$ selon les probabilités de transition $P(s_{i+1} | s_i, d_i)$. A chacune de ces trajectoires est associée une séquence de récompenses $r_0 \rightarrow r_1 \rightarrow \dots \rightarrow r_n \rightarrow \dots$, avec $r_i = r(s_i, d_i, s_{i+1})$.

2.1 Optimalité et fonction de valeur

Le problème d'optimisation associé à un PDM consiste à déterminer une politique π qui maximise pour tout état initial une fonction de valeur définie comme une mesure de la somme espérée des récompenses le long des trajectoires parcourues selon π . Le critère d'optimalité le plus rencontré, dit critère γ -pondéré, correspond à la fonction de valeur

de S dans \mathbb{R} suivante :

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

Le paramètre $0 \leq \gamma < 1$ est un facteur d'actualisation qui permet de faire porter un plus grand poids aux récompenses présentes ou proches dans le futur.

D'une manière générale, la recherche de $\pi^* = \operatorname{argmax}_\pi V^\pi$ est étroitement liée au calcul de $V^* = \max_\pi V^\pi = V^{\pi^*}$. En effet, un résultat fondamental des PDM est l'existence d'une équation d'optimalité caractérisant entièrement cette fonction de valeur optimale, que l'on nomme aussi *équation de Bellman* [1]. Dans le cadre du critère γ -pondéré, cette équation prend la forme : $\forall s \in S$

$$V^*(s) = \max_{d \in D} \sum_{s' \in S} p(s' | s, d) \{r(s, d, s') + \gamma V^*(s')\} \quad (1)$$

On montre alors que la solution V^* de cette équation est unique, et surtout que la connaissance de cette fonction de valeur optimale V^* permet de définir une politique optimale $\pi^* : \forall s \in S$

$$\pi^*(s) = \operatorname{argmax}_{d \in D} \sum_{s' \in S} p(s' | s, d) \{r(s, d, s') + \gamma V^*(s')\}$$

2.2 Algorithmes de résolution

L'approche la plus classique se base sur la résolution directe de l'équation d'optimalité de Bellman, en utilisant pour cela une méthode itérative de type point fixe, d'où son nom anglais de *value iteration* [1].

La solution de l'équation 1 est obtenue comme limite de la suite : $\forall s \in S$

$$V_{n+1}(s) = \max_{d \in D} \sum_{s' \in S} p(s' | s, d) \{r(s, d, s') + \gamma V_n(s')\}$$

avec $V_0()$ initialisée à 0 par exemple. Il est établi que cet algorithme itératif converge en un nombre maximum d'itérations polynomial en $\#S, \#D, 1/(1-\gamma) \log(1/(1-\gamma))$, chaque itération étant de complexité $O(\#D\#S^2)$ [17].

La seconde approche de résolution, nommée *policy iteration*, consiste à itérer directement sur la politique. Soit la politique π_n à l'itération n . Dans une première étape on calcule la fonction de valeur $V_n = V^{\pi_n}$ en résolvant le système d'équations linéaires : $\forall s \in S$

$$V_n(s) = \sum_{s' \in S} p(s' | s, \pi_n(s)) \{r(s, \pi_n(s), s') + \gamma V_n(s')\}.$$

Dans un second temps on améliore la politique courante en posant $\forall s \in S$

$$\pi_{n+1}(s) = \operatorname{argmax}_{d \in D} \sum_{s' \in S} p(s' | s, d) \{r(s, d, s') + \gamma V_n(s')\}$$

On alors montre que la politique π_{n+1} domine la politique π_n , c'est à dire que $\forall s \quad V^{\pi_{n+1}}(s) \geq V^{\pi_n}(s)$, l'égalité n'étant obtenue que si $V^{\pi_{n+1}} = V^{\pi_n} = V^*$, soit lorsque π_n est une politique optimale.

La complexité de l'algorithme d'itération de la politique est en $O(\#D\#S^2) + O(\#S^3)$ par itération, avec un nombre maximum d'itérations polynomial en $\#D$, $\#S$ à γ constant. Expérimentalement, l'approche *policy iteration* est plus efficace que la *value iteration*. Pour les deux algorithmes, les limitations liées à l'espace mémoire nécessaire proviennent du stockage des matrices de probabilités de transition.

3 Modèle simplifié du problème de gestion forestière

Nous présentons maintenant le problème agronomique sur lequel s'appuie notre étude. Il s'agit d'un problème de gestion de forêt à plusieurs parcelles dont on veut optimiser les revenus obtenus et dont on modélise le fonctionnement sous la forme d'un PDM.

Pour cela, nous nous inspirons d'études précédentes appliquant le même cadre mathématique à des problèmes agronomiques semblables [22] [13].

3.1 Etats du système

On considère que la forêt est constituée de N parcelles homogènes, c'est à dire que chacune des parcelles ne contient que des arbres qui ont le même âge et qui appartiennent à la même espèce. Les N parcelles peuvent être de tailles différentes et ne possèdent pas forcément les mêmes caractéristiques géophysiques.

Sur chaque parcelle n , on définit a_t^n , la classe d'âge des arbres au temps t avec $a_t^n \in \mathcal{A} = \{1, \dots, A\}$ (à partir d'un certain âge, les arbres gardent les mêmes propriétés, ils restent dans la classe d'âge A). Le vecteur d'état du système à l'instant t est donc :

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

3.2 Décisions

Après avoir observé l'état du système à l'instant t , on décide quelles parcelles couper sur la période $[t, t + 1]$.

On définit pour cela le vecteur

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

où $d_t^n = 0$ si on coupe les arbres de la parcelle n , 1 sinon. La coupe de la parcelle prend effet sur la période $[t, t + 1]$, et peut être contrariée par un éventuel incendie sur cette période.

On choisit également quel niveau de budget e_t consacrer à la prévention des incendies dans la forêt pour la période $[t, t + 1]$

$$e_t \in \mathcal{E} = \{1, \dots, E\}.$$

Ce budget peut être alloué à la construction de routes pare-feu, de miradors, à l'achat de canadais, etc. Ces décisions profitent à l'ensemble de la forêt et non pas à une parcelle en particulier. Ce budget e_t modifie les probabilités d'incendie sur chaque parcelle dans l'intervalle de temps $[t, t + 1]$. Notons que ce niveau de prévention incendie est le seul facteur qui lie les N parcelles.

3.3 Transitions

Une fois les choix effectués en t , on s'intéresse aux transitions vers les autres états du système. Les probabilités d'incendie $P_{incendie}(n, a_t^n, e_t)$ sont définies pour chaque parcelle en fonction de l'âge des arbres de la parcelle et du budget consacré à la prévention. Notons que notre modèle ne comporte pas d'aspect spatial: l'occurrence d'un incendie sur une parcelle n'augmente pas la probabilité d'incendie sur les parcelles voisines.

Le passage de t à $t + 1$ au niveau de la parcelle n s'effectue alors de la manière suivante :

- si on coupe ($d_t^n = 1$) alors $a_{t+1}^n = 1$ (qu'il y ait ou non incendie sur $[t, t + 1]$);
- si on ne coupe pas :
 - si incendie (avec probabilité $P_{incendie}(n, a_t^n, e_t)$) alors $a_{t+1}^n = 1$;
 - si pas d'incendie alors $a_{t+1}^n = \min(a_t^n + 1, A)$ (passage dans la classe d'âge supérieure).

On définit ainsi une probabilité de transition $P_{d_t^n, e_t}(a_t^n, a_{t+1}^n)$ entre les classes d'âge a_t^n et a_{t+1}^n . On a alors au niveau de la forêt entière, compte tenu de l'indépendance des transitions entre les parcelles :

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

3.4 Revenus

Les revenus immédiats proviennent uniquement de la coupe du bois et sont fonction de la parcelle d'origine du bois, qui détermine la quantité et la qualité, ainsi que de l'âge auquel le bois a été coupé. Il faut soustraire à ces revenus les dépenses liées à la coupe, qui dépendent également de la parcelle, et celles liées à la prévention d'incendie.

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

avec $k(e_t)$ le coût lié à la prévention incendie, $k'(d_t)$ le coût lié à la coupe, et $r_n(d_t^n, a_t^n, a_{t+1}^n, n)$ le prix moyen du stock de bois si $d_t^n = 1$, le prix du bois sauvé si $d_t^n = 0$ et si incendie, et 0 sinon.

4 Résolution approchée du problème de gestion forestière

Notre objectif est de pouvoir traité une forêts d'environ 50 parcelles et 6 classes d'âge. Comme nous le verrons section §5, les approches exactes ne permettent pas de traiter de telles dimensions, et nous devons donc considérer des méthodes approximatives.

4.1 Méthode de type "agrégation d'états"

L'idée est ici de réduire la taille des espaces d'états et éventuellement de décisions, en modifiant la représentation initiale de ces états ($s = \{a_1, \dots, a_N\} \in \mathcal{A}^N$) et décisions ($d = \{d_1, \dots, d_N, e\} \in \{0, 1\}^N \times \mathcal{E}$).

Ainsi, en suivant [13] ou [22], on considère que la forêt est constituée de N parcelles homogènes et on estime de plus que les N parcelles sont de tailles identiques et possèdent les mêmes caractéristiques géophysiques. Cette hypothèse simplificatrice nous permet de regrouper les états par classe d'âge et d'adopter, pour les états et décisions, les représentations suivantes :

États du système. Le vecteur d'état du système à l'instant t est représenté par :

$$s_t = (n_t^1, \dots, n_t^A) \in S'$$

avec n_t^a = nombre de parcelles d'âge a à l'instant t . On a alors, $\forall t, \sum_{a=1}^A n_t^a = N$.

En fait, il s'agit ni plus ni moins d'une technique d'agrégation, regroupant toutes les parcelles de même âge a dans la même variable n^a . En occultant les différences spécifiques à chaque parcelle, on ne s'intéresse plus alors qu'aux classes d'âge des arbres (et non plus à la parcelle sur laquelle ils se trouvent) et on peut se permettre d'agréger certains états.

On montre (voir [6]) que la taille de l'espace d'états modifié est $\#S' = C_{N+A-1}^N$. Cette taille, qui est en $O(N^{A-1})$, est à rapprocher de la taille de l'espace d'états initial S , $\#S = A^N$. Par exemple, pour $A=5$ et $N=6$, on passe de 15625 états possibles à 210, et pour $A = N = 10$, de 10^{10} à moins de 10^5 .

Décisions. La taille de l'espace de décisions initial est $\#D = 2^N \times E$. De la même manière que pour les états, on peut définir une représentation agrégée des décisions, sous la forme $d'(s) = \{c^1, \dots, c^N, e\}$, avec $0 \leq c^a \leq n^a$, le nombre de parcelles d'âge a qui vont être coupées. Cette représentation des décisions introduit une difficulté supplémentaire, puisque l'ensemble $D(s)$ des décisions disponibles dépend de l'état courant s (car c^a est borné par n^a , pour tout âge a). Cependant les algorithmes classiques de résolution de PDMs prennent en compte le fait que l'espace des décisions soit dépendant de l'état courant.

Avec cette représentation des décisions, la taille $\#D(s)$ dé-

pend de s :

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

On montre toutefois aisément que

$$\forall s, \#D(s) \leq (N/A + 2)^A \times E,$$

soit $\#D(s) = O(N^A \times E)$.

Transitions. Une fois les choix effectués en t , on s'intéresse aux transitions vers les autres états du système. A noter que le budget e_t consacré au temps t pour la prévention modifie les probabilités d'incendie sur les parcelles dans l'intervalle de temps $[t, t+1]$. On définit la probabilité $P_{\text{incendie}}(a, e_t)$ d'incendie en fonction de la classe d'âge et du budget consacré à la prévention (elle est indépendante de la parcelle par hypothèse d'homogénéité des parcelles).

Le passage des parcelles de la classe d'âge a à $a+1$ ($a \in \{1, \dots, A-2\}$) sur l'intervalle de temps $[t, t+1]$ s'effectue de la manière suivante :

on choisit $c_t^a \leq n_t^a$, nombre de parcelles à couper sur la classe d'âge a .

- c_t^a parcelles sont coupées (et rejoignent la classe 1 au temps $t+1$),
- restent $r = n_t^a - c_t^a$ parcelles.

Chacune de ces r parcelles restantes a une probabilité $P_{\text{incendie}}(a, e_t)$ d'incendie, ce qui définit une distribution de probabilités sur n_{t+1}^{a+1} . Pour $z \in \{0, \dots, r\}$, $P(n_{t+1}^{a+1} = z | n_t^a - c_t^a = r) = C_r^z \times [P_{\text{incendie}}(a, e_t)]^{r-z} \times [1 - P_{\text{incendie}}(a, e_t)]^z$: z parcelles passent dans la classe d'âge supérieure et les $r-z$ autres parcelles brûlent et vont dans la classe 1.

Pour la classe d'âge A , le fonctionnement est légèrement différent: en effet, les parcelles d'âge A à l'instant t qui ne sont pas coupées et qui échappent aux incendies restent dans la même classe d'âge à l'instant $t+1$.

Les parcelles d'âge A à $t+1$ viennent donc des parcelles présentes dans les classes A et $A-1$ au temps t . Le passage des parcelles dans la classe d'âge A sur l'intervalle de temps $[t, t+1]$ s'effectue alors de la manière suivante :

on choisit $c_t^{A-1} \leq n_t^{A-1}$ et $c_t^A \leq n_t^A$.

- $c_t^{A-1} + c_t^A$ parcelles sont coupées,
- restent $r = n_t^{A-1} - c_t^{A-1}$ et $r' = n_t^A - c_t^A$ parcelles.

Chacune des r premières parcelles restantes a une probabilité $P_{\text{incendie}}(A-1, e_t)$ d'incendie, les r' autres, une probabilité $P_{\text{incendie}}(A, e_t)$, ce qui laisse entrevoir $r+r'+1$ possibilités pour n_{t+1}^A . Pour $z \in \{0, \dots, r+r'\}$, $P(n_{t+1}^A = z | r, r') = \sum_{k=0}^z \{C_r^k \times [P_{\text{incendie}}(A-1, e_t)]^{r-k} \times [1 - P_{\text{incendie}}(A-1, e_t)]^k\} \times \{C_{r'}^{z-k} \times [P_{\text{incendie}}(A, e_t)]^{r'+k-z} \times [1 - P_{\text{incendie}}(A, e_t)]^{z-k}\}$: z parcelles sont dans la classe d'âge A et les $r+r'-z$ autres parcelles brûlent et vont dans la classe 1.

Le nombre de parcelles dans la classe d'âge 1 au temps $t + 1$ vaudra le nombre de parcelles coupées ou brûlées dans toutes les autres classes d'âge sur la période $[t, t + 1]$. Les matrices de transition des parcelles de la classe d'âge a à $a + 1$ ($a \in \{1, \dots, A - 1\}$) sur l'intervalle de temps $[t, t + 1]$ sont donc de la forme :

$$P_{a \rightarrow a+1}(i, j) = P(n_{t+1}^{a+1} = j | n_t^a - c_t^a = i) \\ = \begin{cases} C_i^j \times [P_{incendie}(a, e_t)]^{j-i} \times [1 - P_{incendie}(a, e_t)]^i \\ si \ i \geq j \\ 0 \ si \ i < j \end{cases}$$

De plus, la matrice de transition des parcelles de la classe d'âge A vers A sur l'intervalle de temps $[t, t + 1]$ est de la forme : $P_{A \rightarrow A}(i, j) = P(n_{t+1}^A = j | n_t^A - c_t^A = i)$

$$= \begin{cases} C_i^j \times [P_{incendie}(A, e_t)]^{j-i} \times [1 - P_{incendie}(A, e_t)]^i \\ si \ i \geq j \\ 0 \ si \ i < j \end{cases}$$

On a enfin, au niveau de la forêt entière, compte tenu de l'indépendance des classes d'âge :

$$P(s_{t+1} | s_t, e_t, c_t) = \prod_{a=1}^{A-2} P_{a \rightarrow a+1}(n_t^a - c_t^a, n_{t+1}^{a+1}) \\ \times \sum_{k=0}^{n_A^{t+1}} \{P_{A-1 \rightarrow A}(n_t^{A-1} - c_t^{A-1}, k) \times P_{A \rightarrow A}(n_t^A - c_t^A, n_{t+1}^A - k)\}$$

A partir des matrices stockées et de la formule ci-dessus, nous pouvons calculer les probabilités globales de transition.

Revenus. Les revenus sont calculés de la même manière que dans le modèle initial, en omettant seulement les spécifications de chaque parcelle :

$$r_t = r(s_t, e_t, c_t, s_{t+1}) = -coût(e_t) - coût(c_t) + \\ \sum_{a=1}^{A-2} r_a(c_t^a, n_t^a, n_{t+1}^{a+1}, a) \\ + r_{(A-1, A)}(c_t^{A-1}, c_t^A, n_t^{A-1}, n_t^A, n_{t+1}^A, A - 1, A)$$

avec

- $coût(e_t)$ coût lié à la prévention incendie,
- $coût(c_t)$ coût lié à la coupe,
- $r_a(c_t^a, n_t^a, n_{t+1}^{a+1}, a)$, $a \in \{1, \dots, A - 2\}$
 $= \{\# \text{parcelles de classe } a \text{ coupées}\} \times \{\text{prix } a\}$
 $+ \{\# \text{parcelles de classe } a \text{ non coupées et brûlées}\}$
 $\times \{\text{prix bois sauvé } a\}$.
- $r_{(A-1, A)}(c_t^{A-1}, c_t^A, n_t^{A-1}, n_t^A, n_{t+1}^A, A - 1, A)$
 $= \{\# \text{parcelles de classe } A-1 \text{ coupées}\} \times \{\text{prix } A-1\}$
 $+ \{\# \text{parcelles de classe } A \text{ coupées}\} \times \{\text{prix } A\}$
 $+ \{\# \text{parcelles de classe } A-1 \text{ ou } A \text{ non coupées et brûlées}\} \times \{\text{prix bois sauvé } (A-1, A)\}$.

N.B.: Pour le *Prix bois sauvé* $(A-1, A)$, on pondère en fonction des parcelles non coupées dans $A-1$ et A et en fonction

des probabilités respectives d'incendie car on ne connaît pas la répartition du nombre de parcelles brûlées dans les classes $A-1$ et A . Cela représente la seule approximation que comporte le modèle agrégé pour ce problème. Si, en plus de l'hypothèse d'homogénéité des parcelles, on fait l'hypothèse (raisonnable), que le revenu lié au bois "sauvé" est identique pour les âges $A-1$ et A et que les probabilités d'incendie sont identiques, alors ce modèle est exact.

4.2 Méthode de type "décomposition"

Les méthodes de type "décomposition" forment une deuxième catégorie de méthodes de résolution approchée. Elles consistent à diviser le PDM global en sous-PDM que l'on résout classiquement avant d'utiliser différentes techniques pour trouver une solution globale (optimale ou approchée) à partir des solutions locales.

Suivant le type du PDM global, on distingue les méthodes de *décomposition en série* [9, 18], et de *décomposition parallèle* [23, 16].

La décomposition en série. Elle est utilisée lorsque l'on peut décomposer l'espace d'états en union de sous-espaces ($S = S_1 \cup S_2 \cup \dots \cup S_n$).

[9] suggèrent d'effectuer une double partition de l'espace d'états:

- Dans un premier temps, on cherche à partager l'espace en n régions faiblement couplées R_1, R_2, \dots, R_n ,
- ensuite, on sépare chaque région R_i en deux:
 - l'état s appartient à U_i s'il existe une décision qui permette de passer de cet état à un état extérieur à R_i (probabilité de transition non nulle).
 - l'état s appartient à $K_i = \text{Noyau}(R_i)$ sinon.

On construit alors des sous-PDM en prenant pour espace d'états ceux appartenant à R_i et en reprenant les transitions et récompenses du PDM global. On attribue à chaque état appartenant à $U = \cup U_i$, extérieur au noyau, un coût représentant une pénalité associée au fait de sortir de la région R_i par cet état-là. Par une méthode itérative, on peut tour à tour résoudre les PDM locaux et modifier les pénalités en fonction des politiques optimales obtenues dans chaque région. Il y a rapidement convergence et la politique optimale globale est alors la réunion de toutes les politiques locales. Parr [18] s'appuie sur ce travail mais préfère, dans chaque région, établir une liste de politiques locales dont l'une au moins sera epsilon-optimale quel que soit le vecteur des pénalités associées. Puis il propose de résoudre le PDM dont l'espace d'états est U , l'ensemble des états "communicants", et dont l'ensemble des décisions possibles est la réunion des listes de politiques locales.

Ces deux méthodes sont très performantes lorsque l'on doit résoudre des PDM faiblement couplés, à partir desquels il est possible d'avoir un petit nombre d'états communicants.

La décomposition parallèle. Elle est utilisée lorsqu'il est possible de voir l'espace d'états comme un produit car-

tésien de sous-espaces d'états: $S = S_1 \times S_2 \times \dots \times S_n$. Généralement, on peut également décomposer l'espace de décisions de la même manière: $D = D_1 \times D_2 \times \dots \times D_n$, ou plus généralement $D \subseteq D_1 \times D_2 \times \dots \times D_n$. On peut ainsi définir des sous-PDM d'espace d'états S_i et d'espace de décisions D_i .

Cela inclut les problèmes d'allocation de ressources, où le seul lien entre les sous-PDM est que le choix d'une décision dans un sous-PDM réduit le nombre de décisions possibles dans les autres. Les récompenses sont par contre simplement additives: la récompense totale est la somme des récompenses locales.

[23] proposent un algorithme facile d'implémentation permettant d'obtenir la solution optimale du problème global à partir des solutions optimales (ou de bornes inf et sup sur ces solutions) des sous-PDM. Il consiste à éliminer progressivement les décisions qui ne sont pas compétitives jusqu'à obtenir une seule décision valable pour chaque état, ce qui définit la politique optimale globale. Cette méthode nécessite de parcourir l'espace d'états en totalité, ce qui limite l'algorithme aux PDM de taille modérée.

Dans [16], les différents sous-PDM sont placés dans un état de concurrence: ils se "battent" entre eux pour obtenir le plus de ressources possible et ce sont les plus compétitifs (ceux qui amènent les plus grandes récompenses) qui sont satisfaits. Ici, les sous-PDM sont résolus itérativement de manière exacte avec un partage de ressources fixe, puis le partage est réétabli en fonction de l'apport de chaque sous-PDM à la récompense globale. La solution globale n'est toutefois pas garantie d'être optimale.

Dans notre problème, l'espace d'états pouvant se décomposer en produit cartésien, il apparaît logique de se tourner vers la décomposition parallèle. Malheureusement, les méthodes citées plus haut s'adaptent aux problèmes couplés uniquement par les ressources, et non par les transitions, comme notre problème de gestion forestière. Nous proposons donc une nouvelle version, itérative de méthode de décomposition, adaptée à notre problème, faiblement couplé par les modes de transition, et non les ressources.

Décomposition du problème. Dans notre modèle initial, le seul lien unissant les différentes parcelles est le niveau de protection global, ce qui empêche de résoudre le problème localement, parcelle par parcelle. L'idée est donc de créer un indice de protection local e_t^n sur chaque parcelle n , avec $e_t^n \in \{1, \dots, E\}$. Nous pouvons ensuite décomposer notre PDM en N sous-PDM, définis sur chaque parcelle n de la manière suivante:

- Etats: $a_t^n \in \{1, \dots, A\}$ est l'âge des arbres de la parcelle n au temps t .
- Décisions: $\{d_t^n, e_t^n\} \in \{0, 1\}^N \times \mathcal{E}$
- **Transitions et Revenus:** P^n et R^n sont définies de la même manière que les matrices de transitions et de récompenses au niveau parcellaire dans le modèle initial, le niveau de protection local e_t^n se substituant au niveau de protection global e_t .

On peut résoudre chaque sous-PDM par une méthode classique de programmation dynamique, puisqu'ils sont de taille réduite: le nombre d'états est A et le nombre de décisions $2 \times E$, pour chaque sous-PDM. Nous disposons maintenant sur chaque parcelle n d'une politique locale Π^n , indiquant pour chaque âge de la parcelle la décision (couper ou non) et le niveau de protection local à choisir, ainsi que sa fonction de valeur V^n .

Si les niveaux de préventions étaient réellement indépendants, la réunion des politiques Π^n serait optimale au niveau globale. Malheureusement, le niveau de prévention doit être identique sur toute la forêt. Comment savoir quelle politique globale Π appliquer pour chaque état global?

Méthode directe. Une première méthode consiste à former une politique globale approchée:

$$\Pi_{app}(s_t) = \{\Pi^1(a_t^1), \dots, \Pi^N(a_t^N)\} \cup e_t$$

en choisissant les décisions locales optimales $\Pi^i(a_t^i)$, et en opérant, pour e_t un compromis:

$$e_t(s_t) = \underset{e \in E}{\operatorname{argmax}} \sum_{n=1}^N V^n(a_t^n) \times 1_{e_t^n(a_t^n)=e},$$

où $e_t^n(a_t^n)$ est le niveau de protection optimal localement sur la parcelle n , et $V^n(a_t^n)$ la fonction de valeur correspondant.

Ainsi, le niveau de protection global choisi pour un état donné est celui qui contribue le plus, localement, à la fonction d'utilité globale. Cette méthode est bien entendu heuristique, et rien ne garanti que le niveau de prévention global n'induisse pas une perte d'utilité importante pour les parcelles pour lesquelles il n'est pas optimal.

Nous suggérons deux variantes à notre algorithme initial, pour traiter ce problème:

Variante 1: Modification de la politique de coupe. Cette variante consiste, après avoir choisi l'indice de protection global e_t comme précédemment, à recalculer pour chaque parcelle la politique de coupe la mieux adaptée à e_t . Pour cela, on utilise les fonctions de valeur locales $V^n(a_t^n)$ calculées précédemment comme approximations de la fonction de valeur de notre politique optimale-approchée. La nouvelle politique locale de coupe sera recalculée de manière "gloutonne" sur chaque parcelle:

$$d_t^n(a_t^n) = \underset{d \in \{1,2\}}{\operatorname{argmax}} \{R^n(a_t^n, d, e_t) + \gamma \sum_{a_{t+1}^n \in \{1, \dots, A\}} P^n(a_t^n, a_{t+1}^n, d, e_t) \times V^n(a_{t+1}^n)\}$$

Bien sûr, il est inutile de recalculer d_t^n sur les parcelles pour lesquelles le niveau de protection local optimal e_t^n est égal à e_t .

Cette méthode améliore la politique globale trouvée précédemment, au prix d'une itération supplémentaire sur tous les états de A^N pour lesquels la politique locale optimale diffère de la politique globale.

Variante 2: Choix d'une politique aléatoire. Plutôt que de choisir un niveau de protection global e_t fixe pour chaque état global s_t , on peut décider de choisir un niveau de prévention stochastique, défini par une distribution de probabilité sur cet indice, en fonction des valeurs seuils $V_e = \sum_{n=1}^N V^n(a_t^n) \times 1_{e_t^n(a_t^n)=e}$:

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

La comparaison expérimentale de ces méthodes de décomposition figure dans la section §5.

4.3 Apprentissage par Renforcement

L'apprentissage par renforcement consiste à apprendre un comportement optimal au travers d'une séquence d'expériences au sein d'un environnement, ces expériences consistant à agir dans un état donné, et à observer le nouvel état résultant et la récompense ou punition instantanée associée (voir [25]). D'un point de vue pratique, l'apprentissage par renforcement peut être considéré comme une approche permettant de dépasser les techniques classiques de résolution des PDM selon deux directions :

- L'emploi de simulations de la dynamique du processus à contrôler, afin d'orienter l'exploration de l'espace des fonctions de valeurs ou des politiques. Cela est traduit en pratique par l'emploi d'algorithmes itératifs stochastiques caractéristiques de l'apprentissage par renforcement.
- L'emploi de représentations structurées et compactes des fonctions de valeur et des politiques, permettant d'aborder ainsi la résolution de problèmes décisionnels de très grande taille impossible à traiter en programmation dynamique classique.

L'algorithme Q-learning. L'algorithme Q-learning est une méthode d'apprentissage par renforcement permettant de résoudre l'équation de Bellman pour le critère γ -pondéré. Il est le plus utilisé en pratique, du fait de sa simplicité. Son principe consiste à mettre à jour itérativement les valeurs de la fonction V^* recherchée, sur la base de l'observation des transitions instantanées et de leur revenu associé.

À une politique π fixée de fonction de valeur V^π , on associe la nouvelle fonction : $\forall s \in S, d \in D$

$$Q^\pi(s, d) = \sum_{s'} p(s' | s, d) \{r(s, d, s') + \gamma V^\pi(s')\}.$$

L'interprétation de la valeur $Q^\pi(s, d)$ est la suivante : c' est la valeur espérée du critère pour le processus partant de s , exécutant la décision d , puis suivant la politique π par la suite. Il est clair que $V^\pi(s) = Q^\pi(s, \pi(s))$, et l'équation de Bellman vérifiée par la fonction Q^* devient :

$$Q^*(s, d) = \sum_{s'} p(s' | s, d) \{r(s, d, s') + \gamma \max_b Q^*(s', b)\},$$

$\forall s \in S, d \in D$. On a alors $\forall s \in S, V^*(s) = \max_d Q^*(s, d)$, $\pi^*(s) = \operatorname{argmax}_d Q^*(s, d)$.

Le principe de l'algorithme Q-learning est de mettre à jour à la suite de chaque transition (s_n, d_n, s_{n+1}, r_n) la fonction de valeur courante Q_n pour le couple (s_n, d_n) , où s_n représente l'état courant, d_n la décision sélectionnée et réalisée, s'_n l'état résultant et r_n la récompense immédiate, selon la règle de mise à jour suivante :

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

où le taux d'apprentissage α_n décroît vers 0 avec n . Le choix de la décision à exécuter est effectué en tenant compte de l'estimation Q_n afin de favoriser les décisions les plus prometteuses tout en maintenant un taux non nul d'exploration. Il est immédiat d'observer que l'algorithme Q-learning est une formulation stochastique de l'algorithme de *value iteration* pour les PDM. La convergence de cet algorithme a été bien étudiée et est maintenant établie sous les hypothèses assez générales [2].

Apprentissage par renforcement multi-agents. Le premier avantage immédiat de l'application d'un algorithme de type Q-learning au problème de gestion forestière est d'éviter le stockage en mémoire des matrices de transition. Il suffit en effet de pouvoir simuler à partir d'un état courant s_t et d'une décision d_t un nouvel état aléatoire s_{t+1} , ce qui peut être fait parcelle par parcelle. Le stockage de la fonction Q , de taille $E(2A)^N$, reste toutefois problématique, et une approche multi-agents s'impose. L'idée de la technique multi-agents peut se résumer ainsi: on suppose que chaque parcelle est gérée indépendamment, et N agents cherchent donc à apprendre simultanément une politique optimale, en terme de décision de coupe d_t^n de la parcelle dont ils ont la charge. La Décision de chaque agent influe bien sûr sur l'état global du système, et donc sur le revenu r communs à tous les agents. Enfin, un dernier agent s'occupe uniquement du niveau e_t de prévention incendie.

Cette approche est motivée par la réduction de l'espace mémoire nécessaire pour stocker les $N + 1$ fonctions de Q-valeur. Il est donc indispensable de limiter l'information disponible au niveau de chaque agent. Une première analyse nous a permis de retenir les facteurs importants suivants :

- l'âge a_t^n des arbres de la parcelle n ,
- l'âge moyen \bar{a}_t des arbres de la forêt,
- le nombre de parcelles coupées à la période précédente $Coupe_{t-1}$,
- le niveau de prévention courant e_{t-1} .

Quant à l'agent chargé du choix de la prévention incendie e_t , on estime également qu'il n'est pas nécessaire de lui indiquer l'état global, mais qu'une répartition des parcelles par classe d'âge est suffisante. Les critères de choix retenus sont donc, au temps t :

- la répartition des parcelles $\{n_t^1, \dots, n_t^A\}$,

– le niveau de prévention sur la période précédente e_{t-1} .

L'algorithme Q-learning multi-agent que nous avons appliqué est décrit dans [15, 5]. Pour chaque agent i un tableau Q_i associant la valeur $Q(s, d)$ au choix de la décision d dans l'état s . Partant d'un état global donné, chaque agent parcellaire observe l'environnement qui l'entoure, à savoir $\{a_t^n, \bar{a}_t, Coupe_{t-1}, e_{t-1}\}$ et en déduit la décision optimale d_t^n adéquate. Ensuite, on simule l'âge de la parcelle au temps $t + 1$ à l'aide des probabilités de transition $P_{d_t^n}(a_t^n, a_{t+1}^n)$ et on calcule la récompense associée à la parcelle. Lorsque tous les agents parcellaires ont décidé, le dernier agent choisit le niveau de prévention incendie pour la prochaine période, en fonction de l'ancien niveau de prévention, et de la répartition des parcelles après simulation. Pour finir, les tableaux Q_i sont tous réactualisés en prenant pour récompense r_n la récompense totale, somme des revenus obtenus sur chaque parcelle.

5 Comparaisons numériques

Le PDM global comporte A^N états et $2^N \times E$ décisions. Comme il a été dit, les méthodes classiques ne peuvent être utilisées que pour des espaces d'états et de décisions de dimensions réduites. Ici, ces dimensions augmentent exponentiellement avec N et A .

Nous avons effectué deux séries de tests en faisant varier le nombre de parcelles N , pour les cas $A = 3, E = 2$ et $A = 6, E = 3$. Nous avons pour cela attribué des valeurs réalistes aux paramètres du modèle, et retenu un coefficient $\gamma = 0.9$. Quand cela était possible, les méthodes d'agrégation, de décomposition et d'apprentissage ont été comparées à l'approche exacte. Les critères retenus ont été le temps de calcul, les algorithmes étant codés en MATLAB sous linux, et la qualité des politiques π générées, définie par le critère moyen normalisé

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

Les valeurs V^π et ρ^π ont été calculées exactement lorsque cela était possible, ou estimées par simulation selon l'algorithme d'apprentissage ATD [11].

5.1 Approche exacte

L'algorithme d'itération de la politique a été retenu pour son efficacité. Voici pour le cas $A = 3, E = 2$ les temps T de résolution exacte en secondes, et le critère ρ associé, obtenu par résolution d'un système linéaire de taille A^N .

N	1	2	3	4	5
T	0.04	0.22	2.87	59.09	1277.5
ρ	22.39	22.07	21.93	21.90	21.89

A partir de 6 parcelles, le nombre d'états est trop important pour que soient stockées les matrices globales de transition et de revenus.

Pour le cas $A = 6, E = 3$, les temps de calcul sont les suivants :

N	1	2	3
T	0.09	2.85	244.17

La limite mémoire est ici atteinte pour $N = 4$ parcelles. On s'aperçoit donc des limitations fortes de cette approche exacte qui interdit d'aborder des problèmes de dimension raisonnable.

5.2 Méthodes d'agrégation

Nous effectuons les deux mêmes séries de tests, sachant que les politiques obtenues sont ici identiques aux politiques optimales, du fait des hypothèses d'homogénéité entre parcelles et sur le prix du bois sauvé vérifiées au sein du modèle numérique.

Dans le cas agrégé, ρ_π peut être obtenu directement à partir de la fonction de valeur du problème agrégée V , selon

$$\rho_\pi = \frac{1}{N} \frac{1}{\#S} \sum_{s' \in S'} V(s') n(s')$$

où $n(s')$ est le nombre d'états de S ayant même représentation s' dans S' , facilement calculable.

Voici les valeurs ρ_π et les temps de calculs mesurés pour le cas $A = 3$ et $E = 2$. On distingue les temps de préparation T_P (calcul des nouvelles matrices de transition et de revenus) des temps T_S d'appel au solveur de PDM.

N	1	2	3	4	5
T_P	0.03	0.15	0.63	2.66	10.28
T_S	0.06	0.14	0.24	0.4	0.93
ρ	22.39	22.07	21.93	21.90	21.89

N	6	7	8	9	10
T_P	36.43	120.62	376.8	1129.6	3280
T_S	2.12	6.53	15.45	37.62	57.96
ρ	21.89	21.89	21.89	21.89	21.89

Notons que $\rho(N)$ semble être constant pour N élevé. Enfin, à partir de $N = 11$ parcelles, la limite mémoire est atteinte.

Pour le cas $A = 6$ et $E = 3$, les temps de calculs sont les suivants :

N	1	2	3	4	5
T_P	0.15	2.24	28.69	286.26	2276.4
T_S	0.19	0.69	3.49	10.31	41.05

Le calcul pour 6 parcelles est très long (plus d'une journée) et à partir de 7 parcelles, le nombre d'états est trop important pour que les matrices de transition et de revenus soient stockées.

La méthode d'agrégation par âges permet ainsi d'augmenter quelque peu la taille des problèmes traités.

5.3 Méthodes de décomposition

Nous réutilisons les mêmes valeurs numériques et nous résolvons le problème selon la méthode directe et ses deux variantes. La valeur ρ_π est maintenant estimé par simulation avec l'algorithme ATD. Nous donnons ci-dessous des résultats numériques que pour la première variante qui s'est montrée la plus efficace.

Méthode directe. Les temps de calculs sont globalement bien plus faibles qu'avec la méthode d'agrégation. Par exemple pour $A = 3$ et $E = 2$, $N = 10$ nécessite 21.1 s. Toutefois, à partir de 14 parcelles, le nombre d'états est trop important pour que soit stocké le vecteur des indices de protection optimaux. Même remarque pour $A = 6$ et $E = 3$, où $N = 5$ nécessite 2.85 s. La limite mémoire apparaît ici à partir de 9 parcelles.

Variante: Modification de la politique de coupe. Cette variante est sensiblement plus longue, la phase de modification de politique de coupe étant coûteuse en temps de calcul.

Voici pour $A = 3$ et $E = 2$ les temps de résolution T en secondes, et la valeur moyenne des politiques.

N	2	3	4	5	10	13
T	0.06	0.12	0.24	0.67	256.02	8952.5
ρ	19.72	21.22	20.76	21.28	0.0	0.0

À partir de 14 parcelles, le nombre d'états est trop important pour que soit stocké le vecteur des indices de protection optimaux.

On constate une perte peu importante relativement à la méthode exacte. Cette perte reste toutefois difficile à estimer pour N élevé.

Les temps de résolution, en secondes pour $A = 6$ et $E = 3$ sont les suivants :

N	1	2	3	4	5	6
T	0.08	0.21	0.89	4.2	40.55	939.29

À partir de 7 parcelles, les calculs sont vraiment très longs (plusieurs jours), et à partir de 9, le nombre d'états est trop important pour que soit stocké le vecteur des indices de protection optimaux.

Variante: Choix d'une politique aléatoire. Les temps de résolution sont quasiment les mêmes que pour la méthode directe. Quant à la précision, pour notre exemple de 4 parcelles, 3 classes d'âge et 2 niveaux de prévention, nous obtenons une perte relative d'environ 6,7%, ce qui est moins performant que la méthode directe. Néanmoins, nous pensons que sur certains exemples cette méthode peut donner de meilleurs résultats.

5.4 Apprentissage par renforcement

Les temps de calcul et la qualité des politiques apprises dépend bien sûr du critère d'arrêt choisi pour l'algorithme d'apprentissage. Pour $A = 3$ et $E = 2$ nous effectuons

10000 itérations car la politique obtenue ne varie pas sensiblement au au-delà.

Voici dans ce cas les temps d'apprentissage en secondes, et la qualité des politiques apprises pour les premiers N :

N	3	5	8	10	50	100
T	12.85	19.8	30.04	36.7	173.62	347.87
ρ	20.77	20.95	19.33	19.01		

Avec cette méthode, on ne rencontre de problèmes de stockage des résultats que pour un nombre très élevé de parcelles. Les calculs sont eux relativement rapides, et surtout n'augmentent pas exponentiellement avec le nombre de parcelles: il n'y a pas d'explosion combinatoire en N des temps de calcul. Pour N petit, la perte de qualité des politiques apprises reste faible. Il reste à étudier plus précisément ce qu'il en est pour $N = 100$.

Dans le cas $A = 6$ et $E = 3$, nous effectuons 20000 itérations. Les temps d'apprentissage sont alors les suivants :

N	2	3	5	8	10	12
T	19.07	25.61	39.4	63.38	84.33	147.78

À partir de 13 parcelles, le nombre d'états est trop important pour que soit stocké le vecteur donnant la politique optimale.

6 Conclusion

Les résultats de cette étude sont encore préliminaires, mais certaines conclusions peuvent d'ores et déjà être tirées. Il semble ainsi que seuls les méthodes approchées par décomposition et par renforcement puissent nous permettre de traiter des problèmes pour une taille N supérieure à 50. La méthode par décomposition nécessite pour cela encore quelques développements concernant le calcul de la politique optimale approchée de protection incendie.

La prochaine question qu'il va être nécessaire de traiter concerne l'évaluation des politiques approchées. Ainsi, si l'apprentissage permet de traiter des cas à N très élevé, il est nécessaire de pouvoir dire quelque chose sur la qualité des politiques obtenues, ce qui reste difficile actuellement.

Enfin, à plus long terme, il nous faudra coupler les travaux que nous venons de présenter avec ceux que nous développons dans le cadre des processus décisionnels de Markov avec critères non classiques [24].

Références

- [1] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [2] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont (MA), 1996.

- [3] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.
- [4] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1104–1111, Montreal, Canada, 1995. Morgan Kaufman.
- [5] R. Crites and A. Barto. Improving elevator performance using reinforcement learning. In *Neural Information Processing Systems (NIPS)*, 1996.
- [6] L. Cucala. Résolution de processus décisionnels de markov de grande taille faiblement couplés. INRA-Rapport de stage ingénieur INSA, 2001.
- [7] T. Dean and R. Givan. Model minimization in markov decision processes. In *Proc. of the 14th National Conf. on Artificial Intelligence (AAAI'97)*, pages 106–111, Providence, RI, 1997. AAAI Press.
- [8] T. Dean and Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.
- [9] T. Dean and S. H. Lin. Decomposition techniques for planning in stochastic domains. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1121–1127, Montreal, Canada, 1995. Morgan Kaufman.
- [10] R. Dearden and C. Boutilier. Abstraction and approximate decision theoretic planning. *Artificial Intelligence*, 89:219–283, 1997.
- [11] F. Garcia and F. Serre. Efficient asymptotic approximation in temporal difference learning. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, Berlin, 2000.
- [12] M. Hauskrecht, N. Meuleau, L.P. Kaelbling, T. Dean, and C. Boutilier. Hierarchical solution of markov decision processes using macro-actions. In *UAI'98*, 1998.
- [13] J.O.S. Kennedy. Optimal strategies for protection of forests providing timber and non timber outputs. In *First world Congress on environmental and resources economists*, Venice, Italie, 25-27 juin 1998.
- [14] D. Koller and R. Parr. Computing factored value functions for factored mdps. In *IJCAI'99*, 1999.
- [15] M. L. Littman. Value-function reinforcement learning in markov games. *Journal of Cognitive Systems Research*, 2:55–66, 2001.
- [16] N. Meuleau, M. Hauskrecht, K.E. Kim, L. Peshkin, L.P. Kaelbling, T. Dean, and C. Boutilier. Solving very large weakly coupled markov decision processes. In *AAAI'98*, 1998.
- [17] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov chain decision processes. *Journal of Mathematics of Operations Research*, 12(3):441–450, 1987.
- [18] R. Parr. Flexible decomposition algorithms for weakly coupled mdps. In *UAI'98*, 1998.
- [19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, 1988.
- [20] D. Precup, R. Sutton, and S. Singh. Theoretical results on reinforcement learning with temporally abstract behaviors. In *Proc. 10th European Conference on Machine Learning (ECML'98)*, pages 382–393, Chemnitz, Allemagne, 1998.
- [21] M.L. Puterman. *Markov Decision Processes*. John Wiley and Sons, New York, 1994.
- [22] A. Rapaport, L. Doyen, and J.P. Terreaux. Sustainability analysis for a forestry management model. In *3rd european conference EFITA*, Montpellier, France, 18-20 juin 2001.
- [23] S. Singh and D. Cohn. How to dynamically merge markov decision processes. In *Advances in Neural Information Processing Systems*, volume 10, pages 1057–1063, Cambridge, 1998. MIT Press.
- [24] F. Sol. Résolution de processus décisionnels de markov avec critères non classiques. INRA-Rapport de stage ingénieur INSA, 2001.
- [25] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, 1998.