Résolution de processus décisionnels de Markov à espace d'état et d'action factorisés - Application en agroécologie

Julia Radoszycki

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le *09/10/2015* **par :**

JULIA RADOSZYCKI

Résolution de processus décisionnels de Markov à espace d'état et d'action factorisés
Application en agroécologie

JURY

PIERRE-OLIVIER CHEPTOU	Directeur de recherche	Président du Jury
ALAIN DUTECH	Chargé de recherche	Rapporteur
SABRINA GABA	Chargée de Recherche	Co-directrice de thèse
HERVÉ MONOD	Directeur de Recherche	Rapporteur
NATHALIE PEYRARD	Chargée de Recherche	Directrice de thèse
RÉGIS SABBADIN	Directeur de Recherche	Co-directeur de thèse

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle

Unité de Recherche :
unité MIAT (UR 875) - INRA Toulouse

Directeur(s) de Thèse :
Sabrina GABA, *Nathalie PEYRARD* et *Régis SABBADIN*

Rapporteurs :
Alain DUTECH et *Hervé MONOD*

Résumé

Cette thèse porte sur la résolution de problèmes de décision séquentielle sous incertitude, modélisés sous forme de processus décisionnels de Markov (PDM) dont l'espace d'état et d'action sont tous les deux de grande dimension. La résolution de ces problèmes avec un bon compromis entre qualité de l'approximation et passage à l'échelle est encore un challenge. Les algorithmes de résolution dédiés à ce type de problèmes sont rares quand la dimension des deux espaces excède 30, et imposent certaines limites sur la nature des problèmes représentables.

Nous avons proposé un nouveau cadre, appelé PDMF^3, ainsi que des algorithmes de résolution approchée associés. Un PDMF^3 est un processus décisionnel de Markov à espace d'état et d'action factorisés (PDMF-AF) dont non seulement l'espace d'état et d'action sont factorisés mais aussi dont les politiques solutions sont contraintes à une certaine forme factorisée, et peuvent être stochastiques. Les algorithmes que nous avons proposés appartiennent à la famille des algorithmes de type itération de la politique et exploitent des techniques d'optimisation continue et des méthodes d'inférence dans les modèles graphiques.

Ces algorithmes de type itération de la politique ont été validés sur un grand nombre d'expériences numériques. Pour de petits PDMF^3, pour lesquels la politique globale optimale est disponible, ils fournissent des politiques solutions proches de la politique globale optimale. Pour des problèmes plus grands de la sous-classe des processus décisionnels de Markov sur graphe (PDMG), ils sont compétitifs avec des algorithmes de résolution de l'état de l'art en termes de qualité. Nous montrons aussi que nos algorithmes permettent de traiter des PDMF^3 de très grande taille en dehors de la sous-classe des PDMG, sur des problèmes jouets inspirés de problèmes réels en agronomie ou écologie. L'espace d'état et d'action sont alors tous les deux de dimension 100, et de taille 2^{100}. Dans ce cas, nous comparons la qualité des politiques retournées à celle de politiques expertes.

Dans la seconde partie de la thèse, nous avons appliqué le cadre et les algorithmes proposés pour déterminer des stratégies de gestion des services écosystémiques dans un paysage agricole. Les adventices, plantes sauvages des milieux agricoles, présentent des fonctions antagonistes, étant à la fois en compétition pour les ressources avec la culture et à la base de réseaux trophiques dans les agroécosystèmes. Nous cherchons à explorer quelles organisations du paysage (ici composé de colza, blé et prairie) dans l'espace et dans le temps permettent de fournir en même temps des services de production (rendement en céréales, fourrage et miel), des services de régulation (régulation des populations d'espèces adventices et de pollinisateurs sauvages) et des services culturels (conservation d'espèces adventices et de pollinisateurs sauvages). Pour cela, nous avons développé un modèle de la dynamique des adventices et des pollinisateurs et de la fonction de récompense pour différents objectifs (production, maintien de la biodiversité ou compromis entre les services). L'espace d'état de ce PDMF^3 est de taille 32^{100}, et l'espace d'action de taille 3^{100}, ce qui en fait un problème de taille conséquente. La résolution de ce PDMF^3 a conduit à identifier différentes organisations du paysage permettant d'atteindre différents bouquets de services écosystémiques, qui diffèrent dans la magnitude de chacune des trois classes de services écosystémiques.

Mots-clefs : processus décisionnel de Markov, optimisation continue, méthodes de gradient, inférence dans les modèles graphiques, écoinformatique, modélisation mathématique, paysage, services écosystémiques, pollinisation, adventices

Abstract

This PhD thesis focuses on the resolution of problems of sequential decision making under uncertainty, modelled as Markov decision processes (MDP) whose state and action spaces are both of high dimension. Resolution of these problems with a good compromise between quality of approximation and scaling is still a challenge. Algorithms for solving this type of problems are rare when the dimension of both spaces exceed 30, and impose certain limits on the nature of the problems that can be represented.

We proposed a new framework, called F^3MDP, as well as associated approximate resolution algorithms. A F^3MDP is a Markov decision process with factored state and action spaces (FA-FMDP) whose solution policies are constrained to be in a certain factored form, and can be stochastic. The algorithms we proposed belong to the family of approximate policy iteration algorithms and make use of continuous optimisation techniques, and inference methods for graphical models.

These policy iteration algorithms have been validated on a large number of numerical experiments. For small F^3MDPs, for which the optimal global policy is available, they provide policy solutions that are close to the optimal global policy. For larger problems from the graph-based Markov decision processes (GMDP) subclass, they are competitive with state-of-the-art algorithms in terms of quality. We also show that our algorithms allow to deal with F^3MDPs of very large size outside the GMDP subclass, on toy problems inspired by real problems in agronomy or ecology. The state and action spaces are then both of dimension 100, and of size 2^{100}. In this case, we compare the quality of the returned policies with the one of expert policies.

In the second part of the thesis, we applied the framework and the proposed algorithms to determine ecosystem services management strategies in an agricultural landscape. Weed species, *ie* wild plants of agricultural environments, have antagonistic functions, being at the same time in competition with the crop for resources and keystone species in trophic networks of agroecosystems. We seek to explore which organizations of the landscape (here composed of oilseed rape, wheat and pasture) in space and time allow to provide at the same time production services (production of cereals, fodder and honey), regulation services (regulation of weed populations and wild pollinators) and cultural services (conservation of weed species and wild pollinators). We developed a model for weeds and pollinators dynamics and for reward functions modelling different objectives (production, conservation of biodiversity or trade-off between services). The state space of this F^3MDP is of size 32^{100}, and the action space of size 3^{100}, which means this F^3MDP has substantial size. By solving this F^3MDP, we identified various landscape organizations that allow to provide different sets of ecosystem services which differ in the magnitude of each of the three classes of ecosystem services.

Keywords : Markov decision process, continuous optimisation, gradient methods, inference in graphical models, ecoinformatics, mathematical modelling, landscape, ecosystem services, pollination, weeds

Remerciements

Ca y est, le bout du tunnel est là ! Je tiens à remercier en premier lieu mes directeurs de thèse, sans qui cette thèse n'aurait certainement pas eu lieu. Régis, pour son expertise sur les PDM, Nathalie sur les méthodes variationnelles et Sabrina sur les modèles en écologie ! Merci à tous les trois pour avoir su me guider et vous adapter à moi tout au long de la thèse, et pour votre très grande disponibilité. Je crois que j'ai eu des directeurs en or et je mesure ma chance !

J'adresse mes profonds remerciements à Hervé Monod et Alain Dutech pour avoir accepté de rapporter ma thèse, ce qui n'est jamais une tâche facile. Merci d'avoir su pointer certains points sensibles. Je remercie également Pierre-Olivier Cheptou d'avoir accepté de participer au jury. Merci à tous de vous être intéressé à mes travaux.

Je remercie également ceux qui ont bien voulu me donner des conseils lors de deux comités de thèse : Florent Teichteil, François Massol, Victor Picheny, Nicolas Munier-Jolain, Benoît Ricci. Sans vos conseils, cette thèse n'aurait pas été la même.

Je tiens à remercier bien sûr ceux qui m'ont entourée au quotidien dans le laboratoire MIAT : merci à Nathalie, Fabienne et Alain pour leur efficacité et leur gentillesse. Merci à Romain A. avec qui il a été très agréable de travailler, toujours dans la bonne humeur (et merci pour les cafés !). Merci à Damien B., Mickaël et Marie-Jo pour leur disponibilité... et pour leur patience dans mon utilisation laborieuse des serveurs ! Merci à Sylvain, Damien L. et David S. pour m'avoir fait rire en toutes circonstances (il faut savoir que je suis bon public !). Merci à Robert de m'avoir appris à conduire avec un régulateur ! Merci à David R., Marion Sautier, Céline N., Romain L. et Nathalie V2 pour leurs conseils. Merci à mes compagnons de galère, les thésards : les vétérans Mathieu et Jimmy, Magali et Hiep, et enfin Charlotte, Franck et Clément ! Je vous souhaite le meilleur pour votre thèse, et j'espère que vous m'inviterez à votre soutenance, et qu'on restera en contact ! Merci à Anaïs pour son soutien par le rire, et à Mohammed mon compagnon de bureau. Charlotte tu vas me manquer... Comme le dit Magali, 'la thèse n'est pas un long fleuve tranquille', mais quand on est aussi bien entouré, on ne peut que s'en sortir !

Merci à toute l'UMR Agroécologie de l'INRA de Dijon pour m'avoir accueillie à plusieurs reprises, et en particulier à Martin et Rémi qui m'ont donné plein de conseils et de données, et à Anthony qui m'a accueillie très gentiment le premier jour ! Merci à tous les membres du projet AgrobioSE, et en particulier à V. Bretagnolle pour ses conseils. Merci à Emmanuel Dubois et tous les doctorants de l'IRIT, et en particulier à Anke Brock. Merci à P. Besse, B. Laurent, S. Scott, C. Maugis, A. Joulin et O. Mazet de l'INSA. Merci à ceux qui m'ont initié à la recherche : A. Allauzen, F. Yvon et T. Lavergne au LIMSI, M. Chavance à l'INSERM. Merci à toute l'équipe du LIMSI pour son accueil plus que chaleureux.

Un certain nombre de personnes ont été là pour m'entourer en dehors du cercle de l'INRA, et je tiens à les remercier chaleureusement. Toute ma famille proche : ma sœur Lise, qui m'impressionnera toujours, mes parents, mon tonton préféré et mes super

grand-parents. Mes super colocs, qui ont été là au quotidien pour me soutenir quand j'en avais besoin : Matthias, Natalia (René on t'aime !), Magali (alias Magouille), Niko, Clémentine (vive le lindy et BsAs) et Cyrille. Un merci en particulier à Magali et Niko qui étaient là dans la période la plus difficile, et qui ont été plus qu'au top ! Et à Matthias, qui m'a montré la voie et qui m'impressionne par sa capacité de travail. Spéciale dédicace à ton association qui fait rêver !

Merci à Vincent et Ezequiel pour les bons moments passés en Argentine, et pour m'avoir ouvert les yeux sur plein de choses. Spéciale dédicace à ceux qui m'ont permis de faire des parenthèses très sympas hors territoire national (ou pas) : David à Londres, Bayrem à Brême, Fabrice à Héraklion, Charlotte à Argelès, Marie en Ardèche, Matthias à Aix, Nadi et Vincent à Paris, Estelle à Perpignan, Marion à Cahors, Ali et Cécile à Crest, Jon a San Sebastian. Spéciale dédicace à la fine équipe : Magouille, Marion, Guillaume, Marianne, Olivier, Thomas, Lucie et Nico. Merci d'avoir toujours de nouvelles idées pour ne pas s'installer dans la routine. Merci à Marie pour avoir refait le monde avec moi en Crète. Merci à Manon pour les ciné-sushis, à Anne pour la course et les petits repas, à Julie pour les sorties, à Camille, ma compatriote montpelliéro-aveyronnaise ! Merci à Fanny et Estelle pour le Vara Groupe et les trop rares retrouvailles. Merci à Fabien, Amandine et toute l'équipe pour les concerts. Merci à Djeyda et Jon d'être aussi chouettes, j'espère qu'on se verra bientôt. Merci à Ibou pour les soirées jeux et la guitare. Merci aux camarades insaïens : Ronan, Thomas, Marion, Aline et Soulivanh et à ceux de l'ONERA : Adrien, Rémi, Nicolas. Merci à Seb, Luc et Benjamin pour leurs conseils et leur soutien pendant ma recherche d'emploi. Merci à mon équipe des Doctoriales, aux organisateurs et à Hugo pour ces conseils. Merci à Harold et Franck d'organiser les soirées pub. Longue vie à l'association des jeunes scientifiques du campus d'Auzeville !

A ma petite soeur

Qu'offre soudain la vie, lorsqu'on tourne certaines pages lala lalalala
(Les Vieilles Pies)

Table des matières

Introduction générale

L'agroécologie vise à réduire l'utilisation des intrants chimiques en agriculture en valorisant les services écosystémiques [Ass05] fournis par la biodiversité (pollinisation, production agricole, régulation des pathogènes...). La complexité de ces processus et de leurs interactions nécessite l'utilisation de modèles mathématiques et d'outils informatiques, développés notamment dans la communauté scientifique appelée *computational sustainability* [Gom09]. Concevoir des systèmes agricoles permettant de fournir plusieurs services écosystémiques, et pas seulement des services de production, nécessite de repenser les stratégies de gestion actuelles. Cela demande tout d'abord de bien comprendre les liens entre biodiversité et services écosystémiques, et les compromis qui existent entre services écosystémiques (on parle de compromis entre services lorsque l'augmentation de la fourniture d'un service entraîne la diminution de la fourniture d'un autre service). Pour cela, la modélisation peut être une alternative complémentaire de l'approche expérimentale. Les modèles existants pour décrire les liens entre biodiversité et services écosystémiques et les compromis entre services écosystémiques dans les agroécosystèmes sont pour l'instant relativement rares [TBCR13].

Lorsqu'il s'agit de prendre des décisions dans un paysage agricole, les décisions se prennent de manière séquentielle (chaque jour, chaque mois ou chaque année). De plus, l'effet des actions, c'est-à-dire des opérations agricoles, est incertain (l'incertitude peut être liée à la météo par exemple), de même que la dynamique des espèces (à cause du vent par exemple). Répondre à la question *'Comment agencer les cultures ou les pratiques dans l'espace et dans le temps pour parvenir à des compromis satisfaisants entre services écosystémiques ?'* demande donc de résoudre un problème de décision séquentielle sous incertitude. Une simple approche d'évaluation-comparaison de stratégies 'expertes' ne permet pas d'avoir la garantie qu'il n'existe pas de meilleure stratégie. De plus, dans un paysage, le nombre de stratégies possibles répondant aux consignes d'un expert peut devenir rapidement trop grand pour être construit et analysé 'manuellement'. C'est pourquoi il est nécessaire de proposer des outils mathématiques d'optimisation pour aider à répondre à ces questions.

Le cadre des processus décisionnels de Markov [Put94] est le cadre naturel pour la modélisation de problèmes de décision séquentielle dans l'incertain. Les processus décisionnels de Markov sont utilisés dans de nombreux domaines, de la robotique [CCTKL13] au domaine médical [SBSR04, BPH$^+$05] en passant par le dialogue artificiel [WY07] ou le domaine de l'agriculture et de l'environnement [Ken86, SSR07]. Cependant, dans ces

applications à des domaines réels, les espaces d'état et d'action sont souvent de grande taille. C'est le cas en particulier lorsqu'il s'agit de prendre des décisions sur chaque parcelle d'un paysage. Une résolution exacte est alors difficile, voire impossible, et il est nécessaire de proposer des méthodes de résolution approchée.

Au-delà d'une motivation d'application en agroécologie, l'objectif de cette thèse est de proposer un cadre et des algorithmes génériques pour résoudre des processus décisionnels de Markov dont l'espace d'état et d'action sont factorisés et de grande taille (de l'ordre de 100 variables d'état et d'action, pas forcément binaires), et dont les informations disponibles pour prendre les décisions font partie des contraintes du problème. La résolution de tels problèmes est un thème actif de recherche (voir par exemple [RJF+12, OWS13]), et les méthodes existantes présentent des limites soit dans les capacités de représentation soit dans la taille des problèmes pouvant être résolus.

Cette thèse se rattache au domaine de l'intelligence artificielle, mais propose aussi des contributions en modélisation dans le domaine de l'écologie théorique.

Principales contributions de la thèse

Intelligence artificielle :

Nous avons proposé un nouveau cadre de représentation pour les problèmes de décision séquentielle sous incertitude, appelé PDMF^3 (processus décisionnel de Markov factorisé trois fois : les fonctions de transition et de récompense ainsi que les politiques sont factorisées). Nous avons montré que le problème d'optimisation dans ce cadre est de complexité intermédiaire entre l'optimisation dans le cadre des PDM et l'optimisation dans le cadre des POMDP (PDM partiellement observables, [KLC98]). Nous avons également proposé une famille d'algorithmes de résolution approchée pour les problèmes modélisés dans le cadre PDMF^3. Cette famille d'algorithmes, de type 'itération de la politique', repose sur l'utilisation de méthodes d'inférence dans les modèles graphiques pour la phase d'évaluation de la politique, et de méthodes d'optimisation continue pour la phase d'amélioration de la politique.

Modélisation pour l'écologie théorique :

Nous avons modélisé un problème théorique d'interactions entre cultures (blé, colza, luzerne), adventices et pollinisateurs (sauvages et domestiques) dans le cadre PDMF^3. Les algorithmes de résolution proposés nous ont conduit à identifier des organisations du paysage permettant de maintenir à la fois le rendement agricole et la biodiversité dans toutes les exploitations agricoles du paysage.

Plan du manuscrit

Nous décrirons d'abord, dans le chapitre 1, l'état de l'art sur les processus décisionnels de Markov pour problèmes factorisés. Nous verrons que les algorithmes existants,

exacts ou approchés, soit ne permettent pas de représenter tout type de factorisation, soit ne passent pas à l'échelle pour la résolution. Dans le chapitre 2, nous décrirons en détail les contributions méthodologiques de cette thèse pour la résolution de processus décisionnels de Markov factorisés. Enfin, dans le chapitre 3, nous décrirons les contributions appliquées de la thèse, à savoir la modélisation et la résolution du problème d'allocation des cultures dans l'espace et le temps pour maintenir plusieurs services écosystémiques.

Chapitre 1

Processus décisionnels de Markov et algorithmes de résolution pour problèmes factorisés

Nous présentons d'abord les processus décisionnels de Markov (PDM, [Bel57, Put94, MK12, SB08]) et les méthodes de résolution exactes associées (section 1.1). La plupart de ces méthodes sont connues depuis longtemps. Nous présentons également une approche de résolution plus récente, celle de la planification par inférence, qui apporte une nouvelle vision des PDM qui s'avère utile pour proposer des méthodes de résolution approchée de problèmes factorisés. Nous décrivons ensuite les modes de représentation et les méthodes de résolution des PDM lorsque l'espace d'état est factorisé (section 1.2), ou lorsque l'espace d'action est factorisé (cadre multiagent collaboratif, section 1.3) [1]. Enfin, dans la section 1.4, nous décrivons plus en détail les modes de représentation et les méthodes de résolution de PDM lorsqu'à la fois l'espace d'état et l'espace d'action sont factorisés, ce qui est l'objet de cette thèse. Nous verrons que la résolution de ces derniers problèmes avec un bon compromis entre qualité de l'approximation et passage à l'échelle est encore un challenge, et qu'il reste de la place pour la proposition d'algorithmes de résolution approchée.

1.1 Processus décisionnels de Markov (PDM)

1.1.1 Le cadre

Nous noterons $S^t \in \mathcal{S}$ la variable aléatoire correspondant à l'état du système au temps t, et s^t sa réalisation. De même, nous noterons $A^t \in \mathcal{A}$ la variable aléatoire correspondant à l'action mise en œuvre au temps t, et a^t sa réalisation. Enfin, nous noterons $H^t = (S^0, A^0, ..., S^{t-1}, A^{t-1}, S^t)$ le vecteur aléatoire correspondant à l'historique à la date t du processus, et

1. Nous parlerons d'espace d'état (respectivement d'action) factorisé lorsque l'état du système (respectivement l'action permettant de contrôler le système) sont décrits par plusieurs variables.

$h^t = (s^0, a^0, ..., s^{t-1}, a^{t-1}, s^t)$ sa réalisation. L'hypothèse principale dans les PDM est que la probabilité d'atteindre un état s^{t+1} suite à l'exécution de l'action a^t n'est fonction que de a^t et de l'état courant s^t (propriété de Markov) :

$$\forall h^t, a^t, s^{t+1},\ \mathbb{P}(S^{t+1} = s^{t+1} | H^t = h^t, A^t = a^t) = P^t(s^{t+1}, s^t, a^t).$$

A chaque pas de temps, une fonction réelle, appelée fonction de récompense, permet de modéliser la satisfaction vis-à-vis de l'état du système et de l'action exécutée. Un PDM se définit plus formellement ainsi :

Définition 1 (PDM)**.** *Un processus décisionnel de Markov est un tuple* $(\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R)$ *où :*

- $\mathcal{S}$ *est l'espace des états du système*
- $\mathcal{A}$ *est l'espace des actions qui permettent d'agir sur le système*
- $\mathcal{T}$ *représente l'espace des temps (discret)*
- $P : \mathcal{T} \times \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0; 1]$ *est la fonction définissant les distributions de probabilité décrivant les transitions entre les états ;* $P^t(s', s, a)$ *représente la probabilité de passer de l'état* $s \in \mathcal{S}$ *au temps* t *à l'état* $s' \in \mathcal{S}$ *au temps* $t + 1$ *sous l'effet de l'action* $a \in \mathcal{A}$
- $R : \mathcal{T} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *est la fonction de récompense ;* $R^t(s, a) \in \mathbb{R}$ *représente la récompense associée au fait d'être dans l'état* s *et d'avoir choisi l'action* a *à la date* t*.*

Nous supposerons que les domaines $\mathcal{S}$ et $\mathcal{A}$ sont finis. L'horizon de temps peut être fini ($\mathcal{T} = \{0, 1, ..., T\}, T \in \mathbb{N}^*$) ou infini ($\mathcal{T} = \mathbb{N}$). Dans le cas d'un horizon infini, nous ferons l'hypothèse que les fonctions de transition P et de récompense R sont stationnaires, c'est-à-dire qu'elles ne dépendent pas du temps ($\forall t \in \mathcal{T}, P^t = P$ et $R^t = R$). On parlera de trajectoire plutôt que d'historique pour décrire une réalisation à horizon infini des variables d'état et d'action au cours du temps : $(s^0, a^0, ..., s^t, a^t, ...)$.

Exemple 1. *Imaginons un sujet qui, pendant l'hiver, souhaite décider chaque jour de prendre ou non un médicament pour lutter contre le rhume. On considère un horizon de temps infini (*$\mathcal{T} = \mathbb{N}$*). L'état du sujet, chaque matin, est soit malade (1) soit en bonne santé (0) :* $\mathcal{S} = \{0, 1\}$*. Et il peut choisir de prendre un traitement (1) ou non (0) :* $\mathcal{A} = \{0, 1\}$*. On suppose que l'état du sujet au jour* $j + 1$ *dépend uniquement de l'état du sujet le jour* j *et du fait qu'il ait pris ou non un traitement le jour* j*. L'évolution de la maladie est aléatoire et le traitement n'est pas systématiquement efficace. On suppose que la fonction de transition est connue et donnée par la matrice :*

$$\mathbb{P}(S^{t+1} = 1 | S^t = s, A^t = a) = P(1, s, a) =$$

s \ a	0	1
0	0.3	0.1
1	0.9	0.2

Chaque jour, le sujet obtient une récompense réelle modélisant son niveau de satisfaction

de son état de santé, et prenant en compte le coût du traitement :

$R(s,a) =$

s \ a	0	1
0	1	0.9
1	0	-0.1

1.1.2 Les politiques

Pour contrôler un PDM, on peut définir des politiques, qui vont indiquer comment choisir les actions à chaque pas de temps t en fonction de l'historique h^t que l'on a observé.

Définition 2 (politique stochastique)**.** *Une politique stochastique est une famille de distributions de probabilité selon laquelle une action a doit être sélectionnée pour chaque historique observé h^t. On la note $\pi = \{\pi^t(a|h^t), \forall t, \forall h^t, \forall a\}$. $\pi^t(a|h^t)$ est la probabilité de choisir l'action a au pas de temps t si on a observé l'historique h^t.*

Définition 3 (politique déterministe)**.** *Une politique déterministe définit précisément l'action qui doit être sélectionnée en se basant sur l'historique h^t. On la note $\pi = \{\pi^t(h^t), \forall t, \forall h^t\}$. $\pi^t(h^t) \in \mathcal{A}$ représente l'action qui doit être sélectionnée au pas de temps t si on a observé l'historique h^t.*

Définition 4 (politique markovienne)**.** *On parle de politique (stochastique ou déterministe) markovienne lorsque la politique ne dépend que de l'état courant s^t et non de l'ensemble de l'historique h^t. Dans le cas contraire, on parle de politique histoire-dépendante.*

Définition 5 (politique (markovienne) stationnaire)**.** *On appelle politique (stochastique ou déterministe) stationnaire une politique markovienne qui ne dépend pas du temps ($\forall (t,t') \in \mathcal{T}^2, \pi^t = \pi^{t'}$).*

Nous noterons Π l'ensemble le plus général des politiques (histoire-dépendantes stochastiques), Π^{MS} l'ensemble des politiques markoviennes stationnaires, Π^D l'ensemble des politiques déterministes et Π^{MSD} l'ensemble des politiques markoviennes stationnaires déterministes ($\Pi^{MSD} \subset \Pi^{MS} \subset \Pi$ et $\Pi^{MSD} \subset \Pi^D \subset \Pi$). Notons que pour une politique $\pi \in \Pi^{MS}$ donnée, l'état du système suit une chaîne de Markov de probabilités de transition
$P_\pi(s^{t+1}|s^t) = \sum_{a \in \mathcal{A}} P^t(s^{t+1}, s^t, a)\pi(a|s^t)$.

Exemple 2. *Dans l'exemple du traitement contre le rhume (voir exemple 1), une politique markovienne stationnaire déterministe possible est de choisir de prendre le traitement uniquement si on est malade : $\forall t \in \mathcal{T}, \pi^t(s^t) = 0$ si $s^t = 0$, $\pi^t(s^t) = 1$ si $s^t = 1$. L'état de santé suit alors une chaîne de Markov de matrice de transition :*

$P_\pi(s'|s) =$

s' \ s	0	1
0	0.7	0.8
1	0.3	0.2

La figure 1.1 donne une représentation graphique d'un PDM à horizon fini sous forme de diagramme d'influence [Sha86] pour une politique markovienne. Les cercles correspondent à des variables d'état, les carrés à des variables de décision, et les losanges à des noeuds d'utilité.

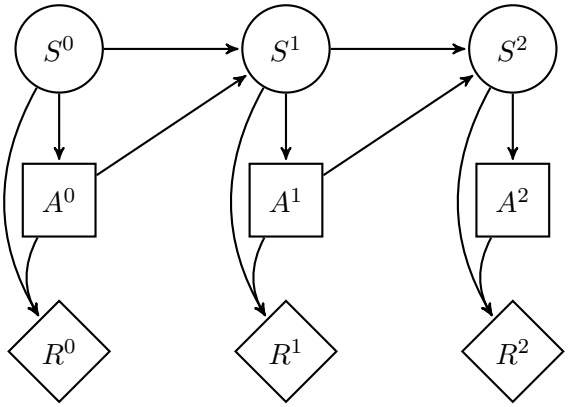


FIGURE 1.1 – Représentation d'un PDM d'horizon $T = 2$ sous forme de diagramme d'influence (pour une politique markovienne)

1.1.3 Évaluation des politiques

Évaluer et comparer les politiques entre elles demande de définir un critère. Ce critère est appelé fonction de valeur (c'est une fonction de l'état initial).

Évaluation à horizon fini

Définition 6 (fonction de valeur à horizon fini)**.** *Pour un PDM à horizon fini ($\mathcal{T} = \{0, 1, ..., T\}$), on appelle fonction de valeur de la politique $\pi \in \Pi$ la fonction $V_\pi : \mathcal{S} \to \mathbb{R}$ qui associe à chaque état initial $s^0 \in \mathcal{S}$ l'espérance de la somme des récompenses obtenues en appliquant π à partir de s^0 jusqu'à l'horizon T :*

$$\forall s^0 \in \mathcal{S},\ V_\pi(s^0) = \mathbb{E}\left[\sum_{t=0}^{T-1} R^t(S^t, A^t) + R^T(S^T)\middle| s^0, \pi\right]$$

où $R^T(S^T)$ est la récompense terminale, fonction seulement de S^T. L'espérance est prise sur l'ensemble des historiques $(s^0, a^0, s^1, a^1 ... s^T)$ possibles en exécutant la politique π à partir de l'état initial s^0.

Théorème 1. *Soit $\pi \in \Pi^D$. Soit $U_\pi^t(h^t)$ la valeur obtenue du temps t au temps T en suivant la politique π, fonction de l'historique h^t :*

$$\forall t \in \mathcal{T},\ U_\pi^t(h^t) = \mathbb{E}\left[\sum_{t'=t}^{T-1} R^{t'}(S^{t'}, A^{t'}) + R^T(S^T)\middle| h^t, \pi\right]$$

On a :

$$\begin{aligned} U_\pi^T(h^T) &= R^T(s^T) \\ \forall t \in \{T-1, ..., 0\},\ U_\pi^t(h^t) &= R^t(s^t, \pi^t(h^t)) + \sum_{s' \in \mathcal{S}} P^t(s', s^t, a^t) U_\pi^{t+1}(h^t, s', \pi^t(h^t)) \\ \forall s^0 \in \mathcal{S},\ V_\pi(s^0) &= U_\pi^0(s^0) \end{aligned}$$

L'évaluation d'une politique à horizon fini se fait donc 'en partant de la fin', en résolvant des problèmes à un pas de temps, ce qui est à la base de la programmation dynamique [Put94].

Évaluation à horizon infini

Définition 7 (fonction de valeur à horizon infini, critère γ-pondéré)**.** *Pour un PDM à horizon infini ($\mathcal{T} = \mathbb{N}$), dans le cas du critère γ-pondéré, on appelle fonction de valeur de la politique $\pi \in \Pi$ la fonction $V_\pi : \mathcal{S} \to \mathbb{R}$ qui associe à chaque état initial $s^0 \in \mathcal{S}$ l'espérance de la somme pondérée des récompenses obtenues en appliquant π à partir de s^0 :*

$$\forall s^0 \in \mathcal{S},\ V_\pi(s^0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(S^t, A^t) \middle| s^0, \pi\right]$$

où $\gamma \in]0;1[$ est appelé facteur d'amortissement. L'espérance est prise sur l'ensemble des trajectoires $(s^0, a^0, s^1, ...s^t, a^t, ...)$ possibles en exécutant la politique π à partir de l'état initial s^0.

Le facteur d'amortissement a un sens économique, et permet également de garantir que la somme soit convergente. C'est ce critère que l'on considérera dans toute la thèse mais il existe d'autres critères à horizon infini, comme le critère moyen :

Définition 8 (fonction de valeur à horizon infini, critère moyen)**.** *Pour un PDM à horizon infini ($\mathcal{T} = \mathbb{N}$), dans le cas du critère moyen, on appelle fonction de valeur de la politique $\pi \in \Pi$ la fonction $V_\pi : \mathcal{S} \to \mathbb{R}$ qui associe à chaque état initial $s^0 \in \mathcal{S}$ l'espérance du gain moyen par étape obtenu en appliquant π à partir de s^0 :*

$$\forall s^0 \in \mathcal{S},\ V_\pi(s^0) = \lim_{T \to \infty} \mathbb{E}\left[\frac{1}{T} \sum_{t=0}^{T-1} R(S^t, A^t) \middle| s^0, \pi\right]$$

L'espérance est prise sur l'ensemble des trajectoires $(s^0, a^0, s^1, ...s^t, a^t, ...)$ possibles en exécutant la politique π à partir de l'état initial s^0.

Dans le cas de l'horizon infini, il est utile de définir aussi ce qu'on appelle la Q-fonction, ou Q-valeur :

Définition 9 (Q-fonction)**.** *Dans le cas de l'horizon infini, on appelle Q-fonction, ou Q-valeur, associée à la politique π la fonction $Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ définie par :*

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A},\ Q_\pi(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V_\pi(s')$$

$Q_\pi(s,a)$ représente la valeur obtenue en partant de s, en appliquant l'action a, puis en suivant la politique π.

Théorème 2 (équation de Bellman)**.** *Soit $\pi \in \Pi^{MS}$. Dans le cas de l'horizon infini, V_π est l'unique solution du système*

$$\forall s \in \mathcal{S},\ V_\pi(s) = Q_\pi(s,\pi(s)) = R(s,\pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s',s,\pi(s)) V_\pi(s') = R(s,\pi(s)) + \gamma \mathbb{E}[V_\pi(S')|s,\pi]$$

où, si π est stochastique, $R(s,\pi(s)) = \sum_{a \in \mathcal{A}} \pi(a|s) R(s,a)$.

Exemple 3. *Dans l'exemple 1, en prenant $\gamma = 0.9$, la valeur de la politique π qui consiste à prendre le traitement uniquement si on est malade est donnée par le système suivant :*

$$\begin{cases} 0.37 V_\pi(0) - 0.27 V_\pi(1) = 1 \\ -0.72 V_\pi(0) + 0.82 V_\pi(1) = -0.1 \end{cases}$$

dont la solution est :

$$\begin{cases} V_\pi(0) = 7.2752 \\ V_\pi(1) = 6.2661 \end{cases}$$

Dans la suite, on notera $\mathcal{V}$ l'ensemble des fonctions de valeur, c'est-à-dire l'ensemble des fonctions de $\mathcal{S}$ dans $\mathbb{R}$.

1.1.4 Le problème d'optimisation (ou de résolution) d'un PDM

L'objectif d'un problème décisionnel de Markov est de trouver une des **politiques optimales** π^* telles que $\forall \pi \in \Pi, V_{\pi^*} \geq V_\pi$, c'est-à-dire telles que

$$\forall \pi \in \Pi, \forall s \in \mathcal{S},\ V_{\pi^*}(s) \geq V_\pi(s).$$

On appelle **fonction de valeur optimale** V^* la fonction de valeur de toute politique optimale ($V^* = V_{\pi^*}$). Les théorèmes qui suivent démontrent qu'aussi bien dans le cas de l'horizon fini que dans le cas de l'horizon infini, ce problème d'optimisation admet au moins une solution, et caractérisent les politiques optimales.

Théorème 3 ([PT87])**.** *Résoudre un PDM, pour le critère fini, γ-pondéré ou moyen, est un problème P-complet.*

Politiques optimales à horizon fini

Théorème 4 ([Put94])**.** *Dans le cas de l'horizon fini, il existe une politique optimale markovienne et déterministe, mais pas forcément stationnaire. La fonction de valeur optimale V^* d'un PDM d'horizon T vérifie :*

$$\begin{aligned} U^{*T}(s^T) &= R^T(s^T) \\ \forall t \in \{T-1, ..., 0\},\ U^{*t}(s^t) &= \max_{a \in \mathcal{A}} R^t(s^t,a) + \sum_{s' \in \mathcal{S}} P^t(s',s^t,a) U^{*t+1}(s') \\ \forall s^0 \in \mathcal{S},\ V^*(s^0) &= U^{*0}(s^0) \end{aligned}$$

Une politique optimale markovienne déterministe $\pi^* = (\pi^{*0}, ..., \pi^{*T-1})$*, pas forcément unique, est déterminée par :*

$$\forall s \in \mathcal{S}, \forall t \in \{0, ..., T-1\},\ \pi^{*t}(s) \in \underset{a \in \mathcal{A}}{\operatorname{argmax}} \left\{ R^t(s,a) + \sum_{s' \in \mathcal{S}} P^t(s', s, a) U^{*t+1}(s') \right\}$$

Politiques optimales à horizon infini

Dans le cas de l'horizon infini, on rappelle que les fonctions de transition et de récompense sont stationnaires. Les deux théorèmes qui suivent permettent de montrer qu'il existe une politique optimale markovienne déterministe et stationnaire. On appelle **Q-fonction optimale**, notée Q^*, la Q-fonction de toute politique optimale ($Q^* = Q_{\pi^*}$).

Théorème 5 (équation d'optimalité de Bellman, [Put94])**.** *V^* est l'unique solution du système*

$$\forall s \in \mathcal{S},\ V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a) = \max_{a \in \mathcal{A}} \left\{ R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V^*(s') \right\}$$

Théorème 6 ([Put94])**.** *Toute politique $\pi^* \in \Pi^{MSD}$ définie par :*

$$\forall s \in \mathcal{S},\ \pi^*(s) \in \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q^*(s,a) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \left\{ R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V^*(s') \right\}$$

est une politique optimale.

Un troisième théorème est utile pour proposer des algorithmes de résolution de PDM à horizon infini : il montre comment améliorer une politique markovienne stationnaire déterministe donnée.

Théorème 7 ([Put94])**.** *Soit $\pi \in \Pi^{MSD}$. Toute politique π^+ définie par :*

$$\forall s \in \mathcal{S},\ \pi^+(s) \in \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q_\pi(s,a) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \left\{ R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V_\pi(s') \right\}$$

vérifie $V_{\pi^+} \geq V_\pi$ et ($V_{\pi^+} = V_\pi \Leftrightarrow V_\pi = V_{\pi^+} = V_{\pi^*}$). Autrement dit, π^+ est au moins aussi bonne que π et, dans le cas où π et π^+ sont de même valeur, c'est qu'elles sont optimales.

1.1.5 Méthodes exactes de résolution

Dans la suite, nous décrivons les méthodes classiques de résolution exactes des PDM à horizon fini et infini. Nous décrivons aussi l'approche plus récente de planification par inférence, qui voit la résolution d'un PDM comme un problème d'inférence dans un modèle graphique.

Programmation dynamique à horizon fini

Le théorème 4 permet de calculer récursivement la fonction de valeur et la politique optimales. L'algorithme 1 correspondant est de complexité $O(T|\mathcal{S}|^2|\mathcal{A}|)$.

Data: $\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R$
Result: V^*, π^*
$U^{*T}(s^T) \leftarrow R^T(s^T),\ \forall s^T \in \mathcal{S}$;
for $t \leftarrow T-1$ **to** 0 **do**
  for $s \in \mathcal{S}$ **do**
    $U^{*t}(s) = \max_{a\in\mathcal{A}} \{R^t(s,a) + \sum_{s'\in\mathcal{S}} P^t(s',s,a)U^{*t+1}(s')\}$;
    $\pi^*_t(s) \in \text{argmax}_{a\in\mathcal{A}} \{R^t(s,a) + \sum_{s'\in\mathcal{S}} P^t(s',s,a)U^{*t+1}(s')\}$;
  end
end
$V^* \leftarrow U^{*0}$;

Algorithme 1: Programmation dynamique à horizon fini

Il existe aussi des méthodes spécifiques aux PDM orientés par les buts (dont l'horizon est fini mais non déterminé), basées sur une recherche guidée par une heuristique, comme par exemple l'algorithme LAO* [HZ01]. Cet algorithme cherche une solution optimale pour un état initial donné. Contrairement à la programmation dynamique, il n'évalue pas tout l'espace d'états (les états qui ne peuvent être atteints à partir de l'état initial par une solution optimale ne sont pas explorés). Cet algorithme est donc intéressant dans le cas de grands espaces d'états, où il peut être plus efficace que les algorithmes de programmation dynamique (surtout si la solution optimale visite seulement une partie de l'espace d'états).

Programmation dynamique à horizon infini

L'algorithme d'itération de la valeur [Bel57] se base sur la résolution directe de l'équation d'optimalité de Bellman (théorème 5), en utilisant une méthode itérative de point fixe. Soit V_n la fonction de valeur courante à l'itération n. Un critère d'arrêt couramment utilisé est : $\| V_{n+1} - V_n \|_\infty \leq \frac{\epsilon(1-\gamma)}{2\gamma}$. Cela garantit que $V_{n+1}(s)$ soit éloignée d'au plus $\frac{\epsilon}{2}$ de $V^*(s)$ pour tout état s, et que la politique, obtenue grâce au théorème 6, soit ϵ-optimale (c'est-à-dire que sa valeur soit à moins de ϵ de V^* pour tout état) [Put94]. Chaque itération de cet algorithme est de complexité $\mathcal{O}(|\mathcal{A}||\mathcal{S}|^2)$ [LDK95]. De plus, le nombre maximal d'itérations est polynomial en $\mathcal{S}$, $\mathcal{A}$ et $\frac{1}{1-\gamma}$ [LDK95].

Data: $\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R, \gamma, \epsilon$
Result: V^*, π^*
initialiser $V_0 \in \mathcal{V}$;
$n \leftarrow 0$;
while $\| V_{n+1} - V_n \|_\infty > \frac{\epsilon(1-\gamma)}{2\gamma}$ **do**
  for $s \in \mathcal{S}$ **do**
    $V_{n+1}(s) = \max_{a \in \mathcal{A}} \{R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V_n(s')\}$
  end
  $n \leftarrow n + 1$;
end
for $s \in \mathcal{S}$ **do**
  $\pi(s) \in \text{argmax}_{a \in \mathcal{A}} \{R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V_n(s')\}$;
end
$V^* \leftarrow V_n$;
$\pi^* \leftarrow \pi$;

Algorithme 2: Algorithme d'itération de la valeur

En s'appuyant sur le théorème 7, on peut proposer un algorithme d'itération sur la politique (algorithme 3). En pratique, cet algorithme converge souvent plus rapidement que l'itération de la valeur [Put94]. Cependant, chaque étape d'évaluation est de complexité $\mathcal{O}(|\mathcal{S}|^3)$ (en utilisant une méthode naïve de résolution du système d'équations linéaires), et chaque étape d'amélioration est de complexité $\mathcal{O}(|\mathcal{A}||\mathcal{S}|^2)$.

L'algorithme d'itération de la politique modifié [PS78] consiste, dans la phase d'évaluation, à résoudre approximativement le système par un petit nombre d'itérations successives, ce qui est plus rapide quand l'espace d'états est de grande taille. Les algorithmes d'itération de la politique et d'itération de la valeur peuvent être vus comme des cas particuliers de l'algorithme d'itération de la politique modifié.

Exemple 4. *Dans l'exemple du traitement contre le rhume (voir exemple 1), une politique optimale à horizon infini pour $\gamma = 0.9$, calculée par la MDP toolbox [CCC+14] avec l'algorithme d'itération de la politique, est la politique qui consiste à prendre le traitement tous les jours : $\forall t \in \mathcal{T}, \pi^t(0) = \pi^t(1) = 1$. Sa valeur est :*

$$\begin{cases} V_\pi(0) = 8.011 \\ V_\pi(1) = 6.9121 \end{cases}$$

Programmation linéaire

D'après le théorème 5, on peut obtenir la fonction de valeur optimale d'un processus décisionnel de Markov à horizon infini en résolvant un programme linéaire. L'algorithme 4 renvoie la politique optimale en un temps polynomial en $|\mathcal{S}|$ et $|\mathcal{A}|$.

Exemple 5. *Dans l'exemple du traitement contre le rhume (voir exemple 1), le pro-*

Data: $\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R, \gamma$
Result: V^*, π^*
initialiser $\pi_0 \in \Pi^{MSD}$;
$n \leftarrow 0$;
repeat
  1. Évaluation : résoudre le système
  $V_n(s) = R(s, \pi_n(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, \pi_n(s)) V_n(s'),\ \forall s \in \mathcal{S}$;
  2. Amélioration :
  for $s \in \mathcal{S}$ **do**
    $\pi_{n+1}(s) \in \text{argmax}_{a \in \mathcal{A}} \{R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V_n(s')\}$;
  end
  $n \leftarrow n + 1$;
until $\pi_{n+1} = \pi_n$;
$V^* \leftarrow V_{n-1}$;
$\pi^* \leftarrow \pi_n$;

Algorithme 3: Algorithme d'itération de la politique

Data: $\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R, \gamma$
Result: V^*, π^*
résoudre $V^* = \min_{V \in \mathcal{V}} \sum_{s \in \mathcal{S}} V(s)$ sous les contraintes :
$V(s) \geq R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V(s') \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$;
for $s \in \mathcal{S}$ **do**
  $\pi^*(s) \in \text{argmax}_{a \in \mathcal{A}} \{R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s', s, a) V^*(s')\}$
end

Algorithme 4: Programmation linéaire

gramme linéaire est le suivant :

$$\begin{array}{ll} \underset{V \in \mathcal{V}}{min} & \sum_{s \in \mathcal{S}} V(s) \\ s.c & V(0) \geq 1 + \gamma(0.7V(0) + 0.3V(1)) \\ & V(0) \geq 0.9 + \gamma(0.9V(0) + 0.1V(1)) \\ & V(1) \geq \gamma(0.1V(0) + 0.9V(1)) \\ & V(1) \geq -0.1 + \gamma(0.8V(0) + 0.2V(1)) \end{array}$$

Les variables du programme linéaire sont les $V(s), s \in \mathcal{S}$ *et il y a* $|\mathcal{S}| \times |\mathcal{A}|$ *contraintes.*

Planification par inférence

Plus récemment, certains auteurs ont montré que l'on pouvait voir le problème de résolution d'un PDM comme un problème d'inférence dans un modèle graphique (voir annexe A pour une présentation détaillée des modèles graphiques), donnant lieu à une nouvelle approche appelée planification par inférence (*planning as inference* [Att03, BT12]). Ainsi, [THS06] montre que la résolution d'un PDM à horizon infini est équivalente à un problème de maximisation de vraisemblance dans un mélange de PDM à horizon fini représentés par des réseaux bayésiens dynamiques [Mur02]. La politique est considérée comme stochastique et les paramètres du modèle de mélange sont les valeurs de cette distribution de probabilité. Les auteurs proposent un algorithme EM (voir [DLR77]) pour résoudre ce problème de maximum de vraisemblance. Cet algorithme (avec une inférence exacte dans l'étape E et la version *greedy* de l'étape M) est équivalent à l'algorithme d'itération de la politique. Cela prouve qu'il converge vers un optimum global (les algorithmes EM n'ont la garantie de converger que vers un optimum local [DLR77]).

[FB09] propose une formulation de l'algorithme EM légèrement différente, qui évite l'utilisation d'une variable auxiliaire et prend en compte plus rigoureusement le cas de l'horizon fini. Cet article traite aussi le cas où l'on recherche une politique optimale déterministe, et où la probabilité de transition est quasiment déterministe (*antifreeze EM*).

Plus récemment, [KP14a] montre que la résolution d'un PDM est équivalente à une maximisation de vraisemblance dans un seul réseau bayésien dynamique, et non un mélange de réseaux bayésiens dynamiques. Par ailleurs, [KP14b] transforme le problème d'optimisation d'un PDM en un problème de marginal-*maximum a posteriori* [LI13] dans un réseau bayésien dynamique[1].

De manière générale, formuler la résolution d'un PDM comme un problème d'inférence dans un modèle graphique permet d'utiliser des algorithmes de résolution existants pour les problèmes d'inférence (voir annexe A). Cette approche n'a pas d'intérêt pratique pour les PDM simples mais devient intéressante dans le cas de problèmes plus complexes, factorisés ou à espace d'état et/ou d'action continu, qui doivent être résolus de manière approchée (voir par exemple [FB10] ou [RTV12]).

1. Ces travaux sont en fait décrits dans le cadre POMDP [KLC98], qui est une généralisation du cadre PDM.

1.1.6 Méthodes de résolution approchées

Les méthodes exactes décrites dans la section 1.1.5 ne sont applicables que si les fonctions de transition et de récompense sont connues *a priori*. Lorsque ce n'est pas le cas, des méthodes issues de l'apprentissage par renforcement [SB98, KS06], qui exploitent des simulations du processus, peuvent être utilisées. Dans ce manuscrit, nous nous intéresserons peu aux méthodes d'apprentissage par renforcement. En effet, nous pensons qu'utiliser un modèle explicite, lorsqu'il est disponible, peut apporter une information utile dans la recherche de solutions proches de l'optimalité.

Les méthodes exactes deviennent coûteuses dès que la taille de l'espace d'états $\mathcal{S}$ excède quelques milliers. C'est pourquoi il peut être intéressant de se limiter à la recherche d'une politique de bonne qualité (de fonction de valeur proche de la valeur optimale) mais plus facile à stocker et interpréter que la politique optimale. Pour cela, on peut choisir de paramétrer la fonction de valeur ou la politique, c'est-à-dire les représenter par un vecteur de paramètres de faible dimension [2]. Plusieurs paramétrisations sont possibles : décomposition dans une base de fonctions, réseau de neurones...

Dans cette section, nous décrivons deux méthodes de résolution approchée [Pow11] en particulier : la programmation linéaire approchée, dans laquelle la fonction de valeur est paramétrée, et la recherche de politiques paramétrées. Dans la suite du chapitre 1 nous en décrirons dans le cadre des PDM factorisés, et dans le chapitre 2 nous décrirons celle que nous proposons.

Programmation linéaire approchée

Dans la programmation linéaire approchée [SS85], on restreint la recherche aux fonctions de valeur qui sont des combinaisons linéaires de fonctions de base $h_k : \mathcal{S} \to \mathbb{R}, k = 1...K$:

$$\forall s \in \mathcal{S},\ V(s) = \sum_{k=1}^{K} w_k h_k(s)$$

Il s'agit alors de trouver les coefficients $w_k, k = 1...K$ solutions du problème :

$$\begin{aligned} \underset{w_k, k=1...K}{\text{minimize}} \quad & \sum_{s\in\mathcal{S}} \sum_{k=1}^{K} w_k h_k(s) \\ \text{subject to} \quad & \sum_{k=1}^{K} w_k h_k(s) \geq R(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s',s,a) \sum_{k=1}^{K} w_k h_k(s') \ \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \end{aligned}$$

puis d'utiliser le théorème 7 pour obtenir la politique. Il y a en pratique moins de variables à déterminer que dans le programme linéaire de l'algorithme 4 (K au lieu de $|\mathcal{S}|$) mais le nombre de contraintes est toujours de $|\mathcal{S}| \times |\mathcal{A}|$ (même si certaines peuvent devenir inactives). Il existe toujours une solution à ce problème si une des fonctions de base est constante (en général on prend $h_1(s) = 1$) [SS85]. Le choix des fonctions de

2. Il existe une troisième possibilité, qui est de paramétrer à la fois la fonction de valeur et la politique. On parle alors de méthodes acteur-critique [SB98].

base influe sur la qualité de l'approximation [dFVR03]. La solution du problème n'est donc pas unique. [dFVR03] donne une borne de l'erreur commise par la programmation linéaire approchée sur la fonction de valeur et des conseils pour le choix des fonctions de base. Plusieurs auteurs ont proposé des méthodes pour réduire le nombre de contraintes (voir par exemple [dFVR04, KH08]).

Recherche de politiques paramétrées

Lorsqu'il s'agit de rechercher des politiques paramétrées, le problème d'optimisation devient :

$$\underset{\theta \in \mathbb{R}^N}{\operatorname{argsup}}\, V_\theta$$

où $\theta \in \mathbb{R}^N$ est le paramètre qui permet de représenter la politique (stochastique). Les méthodes couramment utilisées sont alors des méthodes de type montée de gradient [NW06]. Cet algorithme d'optimisation, sous certaines conditions, a la propriété de converger vers un maximum local (global si la fonction objectif est concave). Les méthodes de montée de gradient sont beaucoup utilisées dans le cadre de l'apprentissage par renforcement [SB98, Wil92]. Le gradient est alors estimé par simulation [MT01], et des méthodes de réduction de la variance de son estimation ont été proposées [GBB04].

D'autres approches utilisent un algorithme EM pour la recherche de politiques paramétrées, comme par exemple [KP11], qui montre les liens entre algorithme EM et méthode de gradient de plus profonde montée. Plus récemment, les auteurs de [FB12] ont montré que pour un PDM la montée de gradient naturelle [Kak02] et l'algorithme EM [THS06] ont des directions de recherche dans l'espace des paramètres proches de celles d'une méthode de Newton approchée [3]. Ils ont donc proposé une nouvelle méthode de résolution approchée des PDM de type méthode de Newton approchée, qui permet de trouver de meilleures solutions que les algorithmes EM ou de montée de gradient.

Dans la suite, nous nous intéressons aux cadres proposés pour décrire des PDM à espace d'état et/ou d'action factorisés (décrits par plusieurs variables). La figure 1.2 représente les liens entre ces différents cadres, et renvoie à la section dans laquelle ils sont décrits, avec les algorithmes de résolution associés. Nous avons vu que résoudre un PDM était de complexité polynomiale en la taille de l'espace d'état et en la taille de l'espace d'action. Dans le cas de PDM factorisés, la taille des espaces d'état et/ou d'action est exponentielle en le nombre de variables, et seule une résolution approchée est envisageable.

1.2 PDM à espace d'état factorisé (PDMF)

Nous nous intéressons tout d'abord au cas des PDM à espace d'état factorisé, ou PDM factorisés, décrits pour la première fois dans [BDG95]. Cela signifie que l'état du système est décrit par plusieurs variables aléatoires, et qu'il y a des indépendances

3. La méthode de Newton est une méthode d'optimisation continue dont la direction de recherche fait intervenir l'inverse de la hessienne.

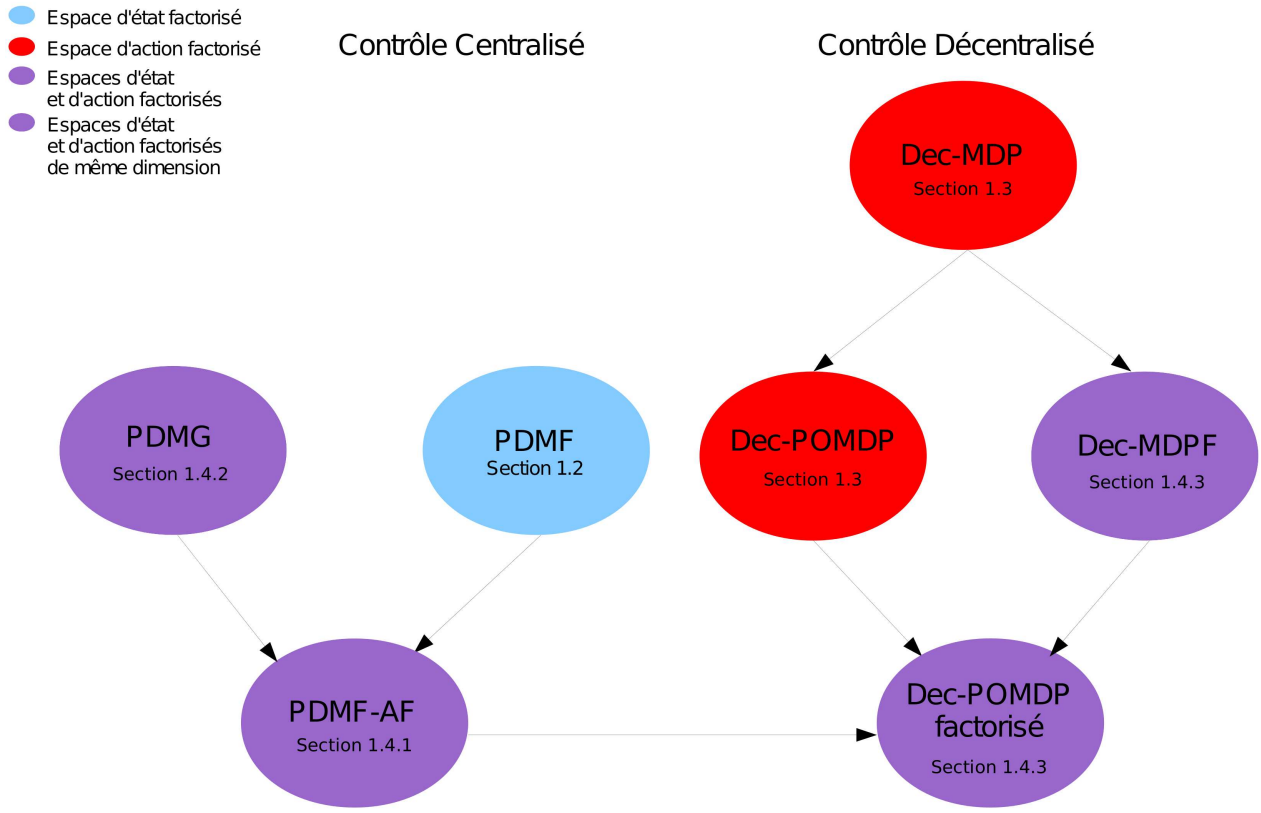


FIGURE 1.2 – Représentation des différents cadres de PDM factorisés ; une flèche d'un cadre a vers un cadre b signifie que le cadre b peut être vu comme une extension du cadre a

dans les fonctions de transition et/ou de récompense. Nous commençons par décrire un exemple simple (section 1.2.1), puis nous donnons une définition formelle du cadre PDMF (section 1.2.2). Les premières méthodes historiques de résolution des PDMFs sont décrites dans la section 1.2.3 puis nous parlons d'approches plus récentes (section 1.2.4). Enfin, nous faisons un bilan de ce qu'apportent les méthodes de résolution développées dans ce cadre à la question qui nous intéresse : la résolution de PDM à espace d'état et d'action factorisés (section 1.2.5).

1.2.1 L'exemple *Coffee Robot*

Nous décrivons ici un exemple tiré de [BDG00], appelé *Coffee Robot*. Un robot doit aller acheter un café pour sa propriétaire, une employée de bureau. Quand il pleut, le robot doit se munir d'un parapluie avant de sortir, pour éviter d'être mouillé. Pour décrire l'état du système, six variables aléatoires binaires sont utilisées :

1. $\mathcal{HCO}$: la propriétaire a-t-elle un café ?
2. $\mathcal{HCR}$: le robot a-t-il un café ?
3. $\mathcal{W}$: le robot est-il mouillé ?
4. $\mathcal{R}$: est-ce qu'il pleut ?
5. $\mathcal{U}$: le robot a-t-il un parapluie ?
6. $\mathcal{O}$: le robot est-il au bureau ?

On a donc $S^t = (\mathcal{HCO}^t, \mathcal{HCR}^t, \mathcal{W}^t, \mathcal{R}^t, \mathcal{U}^t, \mathcal{O}^t)$ et $|\mathcal{S}| = 2^6 = 64$. Le robot dispose de quatre actions possibles ($|\mathcal{A}| = 4$) :

1. $\mathcal{G}o$: se déplacer vers l'autre lieu (le bureau s'il est au café et inversement)
2. $\mathcal{B}uyC$: acheter un café, qu'il obtiendra seulement s'il est au café
3. $\mathcal{D}elC$: donner le café à sa propriétaire (cette action ne pourra avoir l'effet voulu que si le robot a un café et est au bureau)
4. $\mathcal{G}etU$: prendre un parapluie, ce qui n'est possible qu'au bureau.

L'effet des actions est incertain. Par exemple, si le robot tente de donner le café à sa propriétaire au bureau, il se peut que le café se renverse. La fonction de transition est dite factorisée car il y a des indépendances entre les variables d'état. Par exemple, lorsque l'action exécutée par le robot est $\mathcal{D}elC$, on a :

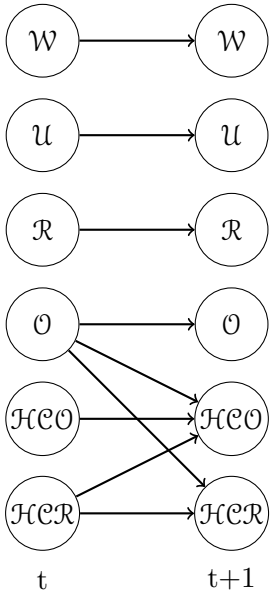
$$\begin{aligned} P(S^{t+1}|S^t, A^t = \mathcal{D}elC) \quad = \quad & P_1(\mathcal{W}^{t+1}|\mathcal{W}^t, A^t = \mathcal{D}elC) \times P_2(\mathcal{U}^{t+1}|\mathcal{U}^t, A^t = \mathcal{D}elC) \\ & \times P_3(\mathcal{R}^{t+1}|\mathcal{R}^t, A^t = \mathcal{D}elC) \times P_4(\mathcal{O}^{t+1}|\mathcal{O}^t, A^t = \mathcal{D}elC) \\ & \times P_5(\mathcal{HCO}^{t+1}|\mathcal{HCO}^t, \mathcal{O}^t, \mathcal{HCR}^t, A^t = \mathcal{D}elC) \\ & \times P_6(\mathcal{HCR}^{t+1}|O^t, \mathcal{HCR}^t, A^t = \mathcal{D}elC) \end{aligned}$$

La figure 1.3 montre une représentation graphique de ces indépendances dans la fonction de transition sous forme de réseau bayésien dynamique [DK89].

Le robot reçoit une récompense de 0.9 lorsque la propriétaire a un café (0 sinon) ajoutée à 0.1 lorsqu'il est sec (0 s'il est mouillé). La récompense ne dépend pas de l'action effectuée par le robot. On a donc :

$$R(S^t, A^t) = R(\mathcal{HCO}^t, \mathcal{W}^t) = R_1(\mathcal{HCO}^t) + R_2(\mathcal{W}^t)$$

La figure 1.4 donne une représentation graphique de la fonction de récompense.



$\mathcal{O}^t$	$\mathcal{HCR}^t$	$\mathcal{HCO}^t$	$\mathcal{HCO}^{t+1}$
1	1	1	1.0
0	1	1	1.0
1	0	1	1.0
0	0	1	1.0
1	1	0	0.8
0	1	0	0.0
1	0	0	0.0
0	0	0	0.0

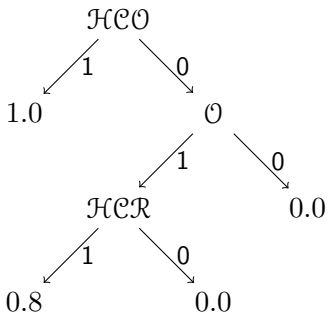
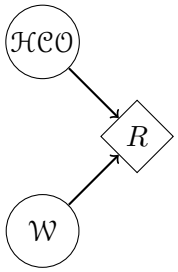


FIGURE 1.3 – Représentation partielle de la fonction de transition P pour le problème *Coffee Robot*. Le diagramme représente les dépendances entre les variables d'état pour l'action $\mathcal{D}elC$ sous forme de réseau bayésien dynamique. Le tableau du milieu décrit la probabilité conditionnelle $P(\mathcal{HCO}^{t+1} = 1 | \mathcal{HCO}^t, \mathcal{O}^t, \mathcal{HCR}^t, A^t = \mathcal{D}elC)$. Celle-ci peut être représentée de manière plus compacte par un arbre de décision (à droite).

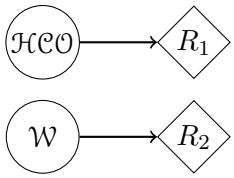
1.2.2 Définition

Supposons que l'état du système à un temps donné t est caractérisé par n variables aléatoires $S_1^t, S_2^t, ..., S_n^t$ où S_i^t prend ses valeurs dans $\mathcal{S}_i$, espace fini. Les $S_i^t, i = 1...n$ sont appelées variables d'état (en anglais *fluents*). On note $S^t = (S_1^t, ..., S_n^t)$. Dans le cas le plus simple où toutes les variables d'état sont binaires ($|\mathcal{S}_i| = 2 \; \forall i = 1...n$), on a $|\mathcal{S}| = 2^n$. La taille de l'espace d'états croît exponentiellement avec le nombre n de variables d'état. C'est pourquoi il est fait l'hypothèse que la transition et la récompense peuvent être représentées de manière compacte (ce qui se justifie dans beaucoup d'applications). Dans le cadre PDMF, la fonction de transition est souvent représentée sous forme de réseau bayésien dynamique incluant les variables d'action (voir exemple figure 1.5) :

Définition 10 (PDMF)**.** *Un PDM factorisé (PDMF) est un tuple* $(\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R)$ *où :*



$\mathcal{HCO}$	$\mathcal{W}$	R
1	0	1.0
1	1	0.9
0	0	0.1
0	1	0.0



$\mathcal{HCO}$	R_1
0	0.0
1	0.9

$\mathcal{W}$	R_2
0	0.1
1	0.0

FIGURE 1.4 – Représentation de la fonction de récompense R du problème *Coffee Robot* sous deux formes équivalentes. La première, avec le diagramme et le tableau de gauche, montre de quelles variables d'état dépend la récompense et énumère toutes les combinaisons possibles de ces deux variables dans une table. La seconde représente la décomposition additive de la fonction de récompense, plus compacte : $R(S^t, A^t) = R_1(\mathcal{HCO}^t) + R_2(\mathcal{W}^t)$.

— *L'espace d'états est un produit cartésien d'espaces finis :* $\mathcal{S} = \prod_{i=1}^{n} \mathcal{S}_i$.
— *L'espace d'actions (fini) est noté* $\mathcal{A}$.
— *L'espace des temps (discret) est noté* $\mathcal{T}$.
— *La fonction de transition P est un réseau bayésien dynamique :*

$$\begin{aligned} \forall s^t, s^{t+1}, a^t,\ P(s^{t+1}, s^t, a^t) &= \mathbb{P}(S^{t+1} = s^{t+1} | S^t = s^t, A^t = a^t) \\ &= \prod_{i=1}^{n} \mathbb{P}(S_i^{t+1} = s_i^{t+1} | pa(S_i^{t+1}) = pa(s_i^{t+1})) \\ &= \prod_{i=1}^{n} P_i(s_i^{t+1} | pa(s_i^{t+1})) \end{aligned}$$

où $pa(S_i^{t+1}) \subset \{S_1^{t+1}, ..., S_n^{t+1}, S_1^t, ..., S_n^t, A^t\}$ représente l'ensemble des parents de S_i^{t+1} dans le réseau bayésien dynamique et $pa(s_i^{t+1})$ représente la réalisation de ce vecteur aléatoire.

— $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *est la fonction de récompense, qui peut être ou non structurée.*

La notation $pa(S_i^{t+1})$ représente l'ensemble des variables aléatoires parentes de S_i^{t+1} dans le réseau bayésien dynamique. Dans l'exemple simple de la figure 1.5 où $n = 3$, on a $pa(S_1^{t+1}) = \{S_1^t, S_2^t, A^t\}$, $pa(S_2^{t+1}) = \{S_1^t, S_3^t, S_1^{t+1}, A^t\}$ et $pa(S_3^{t+1}) = \{S_3^t, A^t\}$. La seule condition pour que ce soit un réseau bayésien dynamique est qu'il n'y ait pas de cycle, ni d'arcs de $t+1$ vers t. Pour la dépendance de S_2^{t+1} en S_1^{t+1} on parle d'*arc synchrone.*

Comme expliqué dans [DS08] , plusieurs types d'indépendances peuvent exister dans un PDMF (dans la fonction de transition et la fonction de récompense) :

1. **indépendance fonctionnelle** : On parle d'indépendance fonctionnelle lorsque la transition d'au moins une variable d'état est indépendante de la valeur de certaines variables d'état au pas de temps précédent (ce qui est toujours le cas dans un PDMF puisque la fonction de transition est représentée par un réseau bayésien dynamique), ou lorsque la fonction de récompense est indépendante de

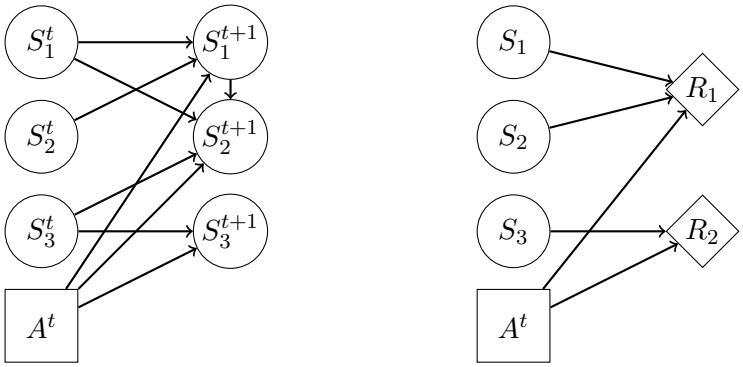


FIGURE 1.5 – Exemple de PDMF où $n = 3$. A gauche le réseau bayésien dynamique représentant les indépendances dans la fonction de transition (par analogie avec les diagrammes d'influence [Sha86], on utilise un carré pour représenter la variable de décision). A droite la représentation graphique de la décomposition additive de la fonction de récompense ($r = 2$).

certaines variables d'état. Dans l'exemple de la figure 1.5, il y a indépendance fonctionnelle car S_1^{t+1} est par exemple indépendante de S_3^t. Dans l'exemple *Coffee Robot* (voir section 1.2.1), la variable d'action n'est pas représentée dans le réseau bayésien dynamique car les dépendances entre les variables sont fonction de la valeur prise par l'action. Pour représenter l'exemple *Coffee Robot*, on utilise donc un réseau bayésien dynamique pour chaque action possible. Mais on pourrait aussi utiliser un seul réseau bayésien dynamique incluant la variable d'action. Selon le problème considéré, l'une ou l'autre des représentations sera la plus compacte. Dans le problème *Coffee Robot*, il y a également une indépendance fonctionnelle dans la récompense, puisque celle-ci ne dépend que des variables d'état $\mathcal{HCO}$ et $\mathcal{W}$.

2. **indépendance contextuelle** : indépendance par rapport à un contexte, c'est-à-dire par rapport à l'instanciation d'un sous-ensemble de variables. Pour tirer profit de ce type d'indépendance, il faut utiliser une représentation compacte plutôt qu'une table pour représenter les probabilités de transition conditionnelles et la fonction de récompense. Par exemple, dans la figure 1.3, un arbre de décision est utilisé pour représenter la probabilité conditionnelle $P(\mathcal{HCO}^{t+1} = 1|\mathcal{HCO}^t, \mathcal{O}^t, \mathcal{HCR}^t, A^t = \mathcal{D}elC)$, ce qui permet de tirer parti du fait que $P(\mathcal{HCO}^{t+1} = 1|\mathcal{HCO}^t = 1) = 1$ par exemple.
3. **fonction identique dans plusieurs contextes disjoints** : certaines représentations, comme les diagrammes de décision algébriques [DB98], permettent de tirer parti du fait que la fonction de transition ou de récompense soit identique dans plusieurs contextes disjoints. Par exemple, la distribution conditionnelle $P(\mathcal{HCO}^{t+1}|\mathcal{HCO}^t, \mathcal{O}^t, \mathcal{HCR}^t, A^t = \mathcal{D}elC)$, est la même dans les contextes $(\mathcal{HCO} = 0 \cap \mathcal{O} = 0)$ et $(\mathcal{HCO} = 0 \cap \mathcal{O} = 1 \cap \mathcal{HCR} = 0)$.
4. **décomposition additive de la fonction de récompense** : on parle de décom-

position additive de la fonction de récompense lorsque la récompense associée au fait d'exécuter l'action a dans l'état s vérifie :

$$R(s,a) = \sum_{\alpha=1}^{r} R_\alpha(pa_R(R_\alpha), a)$$

où $pa_R(R_\alpha) \subset \{s_1, ..., s_n\}$ (dans la représentation graphique, les variables d'état qui interviennent dans la fonction locale de récompense R_α sont ses parents). Dans l'exemple *Coffee Robot*, on a $r = 2$, $pa_R(R_1) = \mathcal{HCO}$ et $pa_R(R_2) = \mathcal{W}$ (voir figure 1.4). Dans l'exemple de la figure 1.5, on a $r = 2$, $pa_R(R_1) = \{S_1, S_2\}$ et $pa_R(R_2) = \{S_3\}$.

1.2.3 Principales méthodes de résolution

Pour traiter les PDMFs, il existe principalement trois types de méthodes, qui permettent d'exploiter tout ou partie de ces indépendances (voir table 1.1) :

1. **programmation linéaire approchée**
2. **SVI (*structured value iteration*) et SPI (*structured policy iteration*)** : algorithmes de programmation dynamique utilisant des représentations sous forme d'arbres de décision
3. **SPUDD (*stochastic planning using decision diagrams*)** : algorithme de programmation dynamique utilisant des représentations sous forme de diagrammes de décision algébriques

La méthode 1 est approchée. Les méthodes 2 et 3 sont des méthodes exactes qui existent aussi sous des versions approchées.

	ALP	SVI et SPI	SPUDD
indépendance fonctionnelle	x	x	x
indépendance contextuelle	x*	x	x
fonction identique dans des contextes disjoints			x
décomposition additive de la récompense	x		

TABLE 1.1 – Caractéristiques des PDMFs prises en compte par les principaux algorithmes de résolution
$*$: lorsqu'une représentation structurée est utilisée (par exemple représentation par règles [GKPV03])

Programmation linéaire approchée

Un PDMF peut être vu comme un PDM avec un grand espace d'état. De ce fait, il peut être résolu par programmation linéaire approchée, en supposant une décomposition linéaire de la fonction de valeur (voir section 1.1.6). Les fonctions de base pour la représentation de la fonction de valeur peuvent être locales (ne dépendre que de certaines

variables d'état). [PBPS02] propose une méthode pour le choix des fonctions de base. Pour réduire le nombre de contraintes, [DD06] propose une approche primal-duale (il y a alors K contraintes).

[GKPV03] propose un algorithme de type itération de la politique (utilisant la programmation linéaire pour la phase d'évaluation) et un algorithme de type programmation linéaire approchée. Tous les deux utilisent une décomposition linéaire de la fonction de valeur et un algorithme de décomposition des contraintes (qui exploite les indépendances fonctionnelles pour réduire le nombre de contraintes nécessaires au calcul de la solution). Ils utilisent une représentation par règles, ce qui permet de prendre en compte (en plus de l'indépendance fonctionnelle et de la décomposition additive) l'indépendance contextuelle.

Algorithmes de programmation dynamique structurée (SPI, SVI, SPUDD)

L'algorithme SPI [BDG95, BDG00] est un algorithme de type itération de la politique modifiée (voir section 8) qui utilise des arbres de décision pour représenter le problème (fonction de transition et fonction de récompense) mais également la fonction de valeur et la politique. Cela permet de réduire à la fois l'espace mémoire nécessaire et le temps de calcul (car au lieu de mettre à jour chaque état l'algorithme met à jour chaque feuille de l'arbre). Cependant, ce n'est pas parce que la représentation du problème est compacte que la politique optimale l'est aussi. L'arbre représentant la politique peut donc devenir plus grand que l'espace d'état. L'algorithme SVI [BDG00] fonctionne également avec des arbres de décision mais est de type itération de la valeur (voir section 8).

Enfin, l'algorithme SPUDD [HSAHB99] est aussi de type itération de la valeur mais utilise des diagrammes de décision algébriques. Dans un diagramme de décision algébrique [DB98], les noeuds peuvent avoir plusieurs parents, contrairement aux arbres de décision. Cela permet donc de prendre en compte des indépendances contextuelles dans la transition et la récompense, et offre généralement une représentation plus compacte qu'un arbre de décision. Un inconvénient de cette représentation est qu'elle nécessite que les variables soient binaires ; il est toujours possible de se ramener à ce cas en scindant le domaine des variables, mais cela augmente rapidement la taille de l'espace d'état. Une version approchée de SPUDD a également été proposée : APRICODD [SAHB00].

1.2.4 Autres méthodes

Plusieurs approches ont été proposées pour les PDMFs dans le domaine de l'apprentissage par renforcement, où le modèle de transition et de récompense ne sont pas forcément connus [Deg07, KSM10]. Dans [KDM00], les modèles de dynamique et de récompense sont supposés connus et exploités. Un algorithme de descente de gradient stochastique est utilisé pour rechercher une politique représentée sous forme de contrôleur à états finis :

Définition 11 ([MPKK99])**.** *Un contrôleur (stochastique) à états finis est un tuple* $(\Sigma, \eta^0, \eta, \psi)$ *où :*

— Σ *est l'ensemble des états internes possibles*
— $\eta^0 : \Sigma \to [0;1]$ *est la distribution initiale sur les états internes*
— $\eta : \Sigma \times \Omega \times \Sigma \to [0;1]$ *est la fonction de transition sur les états internes (*$\forall(\sigma, \sigma') \in \Sigma^2, \forall o \in \Omega,\ \eta(\sigma, o, \sigma')$ *représente la probabilité de passer de l'état interne* σ *à l'état interne* σ' *sachant que l'agent a observé* o
— $\psi : \Sigma \times \mathcal{A} \to [0;1]$ *représente la politique (*$\forall \sigma \in \Sigma, \forall a \in \mathcal{A},\ \psi(\sigma, a)$ *est la probabilité de choisir l'action* a *quand l'agent est dans l'état interne* σ*).*

Dans [KDM00], l'espace d'observation Ω est l'espace $\mathcal{S}_i$ associé à la variable d'état S_i (le choix de $i \in \{1, ..., n\}$ dépend de l'état interne σ).

En dehors de l'apprentissage par renforcement, on peut citer l'algorithme VISA [JB06], qui utilise des représentations sous forme de graphes causaux, ou MADCAP [SUD10], équivalent d'APRICODD [SAHB00] utilisant les diagrammes de décision algébriques affines [SM05]. Enfin, dans le domaine de la recherche heuristique, sLAO* [FH02] est l'équivalent de LAO* (voir section 1.1.5) pour les PDMFs utilisant une représentation sous forme de diagrammes de décision algébriques. Deux algorithmes de ce type ont remporté la compétition IPCC 2011 : PROST [KE12], basé sur l'algorithme UCT [KS06] et Glutton [KMW12], basé sur une extension de l'algorithme LRTDP [BG03] pour les problèmes à horizon fini.

1.2.5 Bilan

De nombreuses méthodes exactes ou approchées ont été proposées pour les PDMFs, qui peuvent être résolus de manière efficace pour des tailles d'espace d'état allant jusqu'à $|\mathcal{S}| = 10^{40}$ (voir table 1.2). Cependant, si l'espace d'action est également factorisé ou de grande taille (ce qui est l'objet de cette thèse), ces méthodes ne sont pas efficaces. Mais avant de nous intéresser à ce cas, nous allons considérer les travaux effectués pour les PDMs dont l'espace d'action seulement est factorisé.

1.3 PDM à espace d'action factorisé : le cadre multiagent décentralisé (Dec-POMDP)

1.3.1 Définition

On se place ici dans le cadre des problèmes multiagents coopératifs : plusieurs agents peuvent agir sur le système, mais la fonction de récompense est commune à tous les agents. Les agents n'ont qu'une vision partielle (et éventuellement différente) du système, via une variable d'observation. L'espace d'action est factorisé, puisqu'il y a une variable d'action par agent. Le système est dit :

— **totalement observable** (*fully observable*) si chaque agent connaît l'état du système à partir de ses observations
— **collectivement totalement observable** (*jointly fully observable*) si on peut connaître l'état du système à partir des observations de tous les agents

algorithmes	[BDG00] SVI et SPI	[HSAHB99] SPUDD	[SAHB00] APRICODD	[SUD10] MADCAP
mode de représentation	DBN+arbre	DBN+ADD	DBN+ADD	DBN+AADD
méthode	exacte	exacte	approchée	approchée
horizon	infini	infini	infini	infini
forme de récompense	struct.	struct.	struct.	struct. et add.
taille maximale	$\lvert\mathcal{S}\rvert = 1.8 \times 10^6$	$n = 24$ $\lvert\mathcal{S}\rvert = 6.3 \times 10^7$	$n = 35$ $\lvert\mathcal{S}\rvert = 3.4 \times 10^{10}$	$n = 24$ $\lvert\mathcal{S}\rvert = 1.7 \times 10^7$
hypothèses, limites		vars bin	vars bin	vars bin

(a) Algorithmes de programmation dynamique

algorithmes	[GKPV03]	[DD06]
mode de représentation		
méthode	approchée	approchée
horizon	infini	infini
forme de récompense	add.	add.
taille maximale	$\lvert\mathcal{S}\rvert \approx 10^{40}$	$n = 20$ $\lvert\mathcal{S}\rvert \approx 10^6$
hypothèses, limites	approximation linéaire de la FV	approximation linéaire de la FV

(b) Algorithmes de programmation linéaire approchée

algorithmes	[KDM00]	[KSM10]
mode de représentation	DBN+arbre	DBN+arbre
méthode	approchée	approchée
horizon	infini	infini
forme de récompense	struct.	struct.
taille maximale	$\lvert\mathcal{S}\rvert = 5632$	$n = 20$ $\lvert\mathcal{S}\rvert \approx 10^6$
hypothèses, limites	recherche d'un contrôleur à états finis	

(c) Algorithmes d'apprentissage par renforcement

algorithmes	[FH02] sLAO*	[KE12] PROST	[KDMW12] Glutton
mode de représentation	ADD	DBN	DBN
méthode	approchée	approchée	approchée
horizon	indéfini ou infini	indéfini	indéfini
forme de récompense	struct.	struct.	struct.
taille maximale	$n = 40$ $\lvert\mathcal{S}\rvert \approx 10^{12}$	$n = 50$ $\lvert\mathcal{S}\rvert \approx 10^{15}$	$n = 50$ $\lvert\mathcal{S}\rvert \approx 10^{15}$
hypothèses, limites			

(d) Algorithmes de recherche heuristique

TABLE 1.2 – Principaux algorithmes pour PDMFs
struct. : structurée, add. : additive, vars bin : variables binaires, FV : fonction de valeur
Les tailles maximales de problèmes pouvant être traités sont issues des expériences présentées dans les publications et sont à prendre comme des ordres de grandeur plus que comme des données précises. Dans tous ces travaux, $|\mathcal{A}|$ ne dépasse jamais 10 ou 20.

— **collectivement partiellement observable** s'il n'est pas possible de déduire l'état du système des observations faites par tous les agents (cas le plus général).

Le cadre permettant de décrire de tels problèmes est celui des *Decentralized Partially Observable Markov Decision Processes* (Dec-POMDPs, voir [BGIZ02, GZ04, BCSM08, Oli10]) :

Définition 12 (Dec-POMDP)**.** *Un Dec-POMDP à m agents est un tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R, \Omega, O, P^0)$ où :*

- *$\mathcal{S}$ est l'espace des états (fini)*
- *$\mathcal{A} = \prod_{i=1}^{m} \mathcal{A}_i$ est l'ensemble (fini) des actions jointes. $\mathcal{A}_i$ représente l'ensemble des actions pouvant êtres prises par l'agent i.*
- *$\mathcal{T}$ représente l'espace des temps (discret)*
- *$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0; 1]$ est la fonction de transition*
- *$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ est la fonction de récompense*
- *$\Omega = \prod_{i=1}^{m} \Omega_i$ est l'ensemble (fini) des observations jointes. Ω_i représente l'ensemble des observations pour l'agent i.*
- *$O : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \Omega \to \mathbb{R}$ est la fonction d'observation. $O(o|s, a, s')$ représente la probabilité que chaque agent observe o_i lorsqu'ils exécutent l'action jointe a à partir de l'état s et que le système arrive dans l'état s'*
- *P^0 est la distribution de probabilité initiale sur les états.*

Dans le cas où $m = 1$, on retrouve le cadre **POMDP** [KLC98]. Si le système est totalement observable, il suffit de définir le Dec-POMDP par $(\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R)$ (on parle alors de **MMDP** pour *multiagent Markov decision process* [Bou99]). Dans la suite, nous nous intéressons au cas non totalement observable, car le cadre que nous proposons au chapitre 2 peut être vu comme un cas particulier de celui-ci. Si le système est collectivement totalement observable (si il existe une application $J : \Omega \to \mathcal{S}$ telle que si $O(o|s, a, s') > 0$ alors $J(o) = s'$), on parle de **Dec-MDP** [BGIZ02].

A cause du fait que la fonction d'observation est jointe et non factorisée, on ne peut pas représenter facilement la fonction de transition d'un Dec-POMDP sous forme de réseau bayésien dynamique.

Théorème 8 ([BGIZ02])**.** *A horizon fini, trouver la solution optimale d'un Dec-MDP ou Dec-POMDP avec $n \geq 2$ est un problème NEXP-complet.*

1.3.2 Méthodes de résolution

La notion de fonction de valeur d'une politique π (voir définitions 6 et 7) est remplacée par la notion de valeur pour la distribution initiale P^0, notée $V_\pi(P^0)$. Pour un Dec-MDP ou un Dec-POMDP, il existe à horizon fini une politique optimale jointe déterministe $\pi = (\pi_1, ..., \pi_m)$ composée d'un ensemble de politiques individuelles (ou locales) π_i qui associent à chaque historique d'observations $(o_i^1, ..., o_i^t)$ une action a_i. En dehors des méthodes de résolution exactes [HBZ04, SCZ05, ADC07], peu efficaces lorsque le nombre d'agents devient grand, certains ont proposé des méthodes qui font une recherche exhaustive dans un espace de politiques restreint (voir par exemple [SZ07]), ou

qui font une recherche approchée dans l'espace complet des politiques (voir par exemple [NTY+03, OKV08]). Récemment, [DABC13] a montré que résoudre un Dec-POMDP était équivalent à résoudre un MDP à état continu, ce qui permet d'utiliser des méthodes de résolution développées pour les POMDPs ou les MDPs à état continu et de résoudre de manière exacte des problèmes à horizon plus grand que les méthodes existantes.

Dans le cas de l'horizon infini, le problème est indécidable [MHC99]. On utilise le plus souvent pour représenter les politiques un contrôleur à états finis par agent (voir définition 11).
Certains auteurs recherchent des solutions approchées (voir par exemple [BHZ05, PKMK00]), d'autres recherchent des solutions optimales pour une certaine taille de contrôleurs donnée (voir par exemple [ABZ07]).

1.3.3 Bilan

La résolution exacte d'un Dec-POMDP à plus de deux agents est donc un problème difficile en général. Les méthodes existantes sont résumées dans la table 1.3. On peut constater dans le cas d'un horizon fini un compromis entre la taille des problèmes pouvant être traités (taille des espaces d'état, d'action et d'observation) et la longueur de l'horizon. Que ce soit à horizon fini ou infini, le nombre d'agents dans les problèmes que peuvent traiter ces méthodes ne dépasse jamais 2 ou 3 (et les tailles d'espace d'état et d'action sont de l'ordre de la centaine). Le problème peut être simplifié si on considère des Dec-MDPs factorisés, où chaque agent a son propre espace d'état (voir section 1.4.3).

1.4 PDM à espaces d'état et d'action factorisés

Nous nous intéressons maintenant au cas des PDM dont l'espace d'état et l'espace d'action sont factorisés (décrits par plusieurs variables discrètes). Plusieurs cadres existent pour représenter ces problèmes. Le cadre des PDMF-AFs (voir section 1.4.1) est une extension de celui des PDMFs au cas d'un espace d'action factorisé. Le cadre des PDMGs (voir section 1.4.2) est un cas particulier de PDMF-AF adapté aux problèmes de décision spatialisée. Enfin, le cadre très général des Dec-POMDPs à espace d'état factorisé (voir section 1.4.3) est une extension de celui des Dec-POMDPs.

1.4.1 PDMFs à espace d'actions factorisé (PDMF-AF)

Définition

De même que pour les PDMFs (voir section 1.2.2), l'état du système à un temps donné t est caractérisé par n variables aléatoires $S_1^t, S_2^t, ..., S_n^t$. Par contre, la représentation de l'action est plus générale puisque l'action que peut exécuter l'agent pour modifier le système est décrite par m variables aléatoires $A_1^t, A_2^t, ..., A_m^t$ où A_j^t prend ses valeurs dans $\mathcal{A}_j$, espace fini. Les A_j sont appelées variables d'action, et l'action jointe au temps

algorithmes	[SZ07]	[NTY+03] JESP	[OKV08] DICE(-A)	[DABC13] FB-HSVI	
mode de représentation					
méthode	approchée	approchée	approchée	exacte	
horizon	fini	fini	fini	fini	
type	PD	PD	RP	RH	
forme de récompense					
taille maximale	$m = 2$ $\lvert\mathcal{S}\rvert = 100$ $\lvert\mathcal{A}\rvert = 16$ $\lvert\Omega\rvert = 25$ $T = 100$	$m = 2$ $\lvert\mathcal{S}\rvert = 2$ $\lvert\mathcal{A}\rvert = 9$ $\lvert\Omega\rvert = 4$ $T = 7$	$m = 3$ $\lvert\mathcal{S}\rvert = 81$ $\lvert\mathcal{A}\rvert = 8$ $\lvert\Omega\rvert = 8$ $T = 8$	$\lvert\mathcal{S}\rvert = 4$ $\lvert\mathcal{A}\rvert = 9$ $\lvert\Omega\rvert = 4$ $T = 100$	$\lvert\mathcal{S}\rvert = 256$ $\lvert\mathcal{A}\rvert = 36$ $\lvert\Omega\rvert = 81$ $T = 10$
hypothèses, limites					

(a) Algorithmes à horizon fini

algorithmes	[BHZ05]	[PKMK00]	[ABZ07]
mode de représentation			
méthode	approchée	approchée	approchée
horizon	infini	infini	infini
type	PD	RP	ONL
forme de récompense			
taille maximale	$m = 2$ $\lvert\mathcal{S}\rvert = 16$ $\lvert\mathcal{A}\rvert = 25$ $\lvert\Omega\rvert = 16$		$m = 2$ $\lvert\mathcal{S}\rvert = 4$ $\lvert\mathcal{A}\rvert = 4$ $\lvert\Omega\rvert = 25$
hypothèses, limites	recherche de contrôleurs à états finis		

(b) Algorithmes à horizon infini

TABLE 1.3 – Principaux algorithmes pour Dec-PODMPs
struct. : structurée, add. : additive
PD : programmation dynamique
RP : recherche de politiques
RH : recherche heuristique
ONL : optimisation non linéaire
Les tailles maximales de problèmes pouvant être traités sont issues des expériences présentées dans les publications et sont à prendre comme des ordres de grandeur plus que comme des données précises.

t est notée $A^t = (A_1^t, ..., A_m^t)$ [4]. De même que dans le cadre PDMF, la fonction de transition d'un PDMF-AF est représentée par un réseau bayésien dynamique, qui fait intervenir les différentes variables d'action (voir exemple figure 1.6) :

Définition 13 (PDMF-AF [KD02])**.** *Un PDMF à espace d'actions factorisé (PDMF-AF) est un tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R)$ où :*

- *L'espace d'états est un produit cartésien d'espaces finis : $\mathcal{S} = \prod_{i=1}^{n} \mathcal{S}_i$.*
- *L'espace d'actions est un produit cartésien d'espaces finis : $\mathcal{A} = \prod_{j=1}^{m} \mathcal{A}_j$.*
- *L'espace des temps (discret) est noté $\mathcal{T}$.*
- *La fonction de transition P est un réseau bayésien dynamique :*

$$\begin{aligned} \forall s^t, s^{t+1}, a^t,\ P(s^{t+1}, s^t, a^t) &= \mathbb{P}(S^{t+1} = s^{t+1} | S^t = s^t, A^t = a^t) \\ &= \prod_{i=1}^{n} \mathbb{P}(S_i^{t+1} = s_i^{t+1} | pa(S_i^{t+1}) = pa(s_i^{t+1})) \\ &= \prod_{i=1}^{n} P_i(s_i^{t+1} | pa(s_i^{t+1})) \end{aligned}$$

  où $pa(S_i^{t+1}) \subset \{S_1^t, ..., S_n^t, A_1^t, ..., A_m^t\}$ représente l'ensemble des parents de S_i^{t+1} dans le réseau bayésien dynamique et $pa(s_i^{t+1})$ représente la réalisation de ce vecteur aléatoire.
- *$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ est la fonction de récompense.*

Dans l'exemple simple de la figure 1.6, tiré de [KD02], où $n = 3$ et $m = 2$, on a $pa(S_1^{t+1}) = \{A_1^t, S_1^t\}$, $pa(S_2^{t+1}) = \{A_2^t, S_1^t, S_2^t\}$ et $pa(S_3^{t+1}) = \{A_2^t, S_2^t, S_3^t\}$.

De même que pour les PDMFs, des arbres ou des diagrammes de décision algébriques peuvent être utilisés pour représenter de manière compacte les probabilités conditionnelles $P_i(s_i^{t+1} | pa(s_i^{t+1}))$. Lorsque n et m sont grands, une hypothèse de récompense additive est souvent faite pour représenter de manière compacte la fonction de récompense :

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A},\ R(s, a) = \sum_{\alpha=1}^{r} R_\alpha(pa_R(R_\alpha))$$

où $pa_R(R_\alpha) \subset \{s_i, i = 1...n, a_j, j = 1...m\}$. Dans l'exemple de la figure 1.6, on a $r = 3$, $pa_R(R_1) = \{S_3, A_1\}$, $pa_R(R_2) = \{S_1, S_2, A_1\}$ et $pa_R(R_3) = \{S_3, A_2\}$.

Méthodes de résolution

Pour résoudre les PDMF-AFs, [KD02], qui utilise des représentations sous forme d'arbres de décision ou de diagrammes de décision algébriques (ADDs), propose une approche de type minimisation de modèle. Cela consiste à se ramener à un PDM simple dont l'espace d'état et d'action sont de petite taille et qui a la même fonction de valeur optimale que le PDMF-AF. Récemment, [RJF+12] a proposé les algorithmes exacts de

4. On peut aussi considérer selon la nature du problème qu'il y a m agents plutôt que m variables d'action pour un agent.

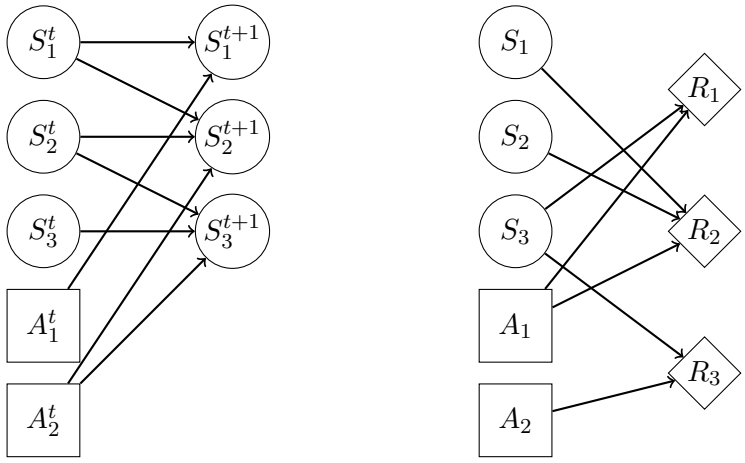


FIGURE 1.6 – Exemple de PDMF-AF où $n = 3$ et $m = 2$. A gauche le réseau bayésien dynamique représentant les indépendances dans la fonction de transition (tiré de [KD02]). A droite la représentation graphique de la décomposition additive de la fonction de récompense ($r = 3$).

programmation dynamique symbolique FAR et MBFAR, qui sont des améliorations de SPUDD [HSAHB99], ainsi qu'une méthode approchée (*sequential hindsight*).

En approchant la fonction de valeur par une somme pondérée de fonctions de base, [GKP01] a pu proposer une méthode de type programmation linéaire approchée. Cependant, le nombre de contraintes est exponentiel en la largeur d'arbre [5] correspondant au graphe de coordination. Des méthodes de programmation linéaire approchée existent aussi pour les PDMF-AFs hybrides, dont les variables d'état et d'action peuvent être continues ou discrètes (voir par exemple [KHG06]).

Dans le domaine de l'apprentissage par renforcement, la fonction de transition du PDMF-AF n'est pas forcément connue. Elle doit simplement pouvoir être simulée. Plusieurs travaux ont été effectués dans ce cadre. [GLP02] approche la fonction de valeur par une somme pondérée de fonctions de base. [SH04] utilise des modèles graphiques non orientés, appelés produits d'experts, pour approcher la fonction de valeur. [BA09] recherche des politiques factorisées et paramétrées par une méthode de montée de gradient (voir section 2.5 pour une comparaison de cet algorithme avec notre approche).

Bilan

Les principaux algorithmes existant pour la résolution de PDMF-AFs sont comparés dans la table 1.4. Les algorithmes symboliques ne peuvent traiter que des problèmes structurés (dont la transition et la récompense peuvent être représentées de manière compacte sous forme de diagrammes de décision algébriques) et à variables binaires.

5. La largeur d'arbre (*treewidth* en anglais) d'un graphe est un nombre qui caractérise la complexité de mise en œuvre de certaines procédures (par exemple l'élimination de variables) sur ce graphe. Ce nombre vaut 1 pour un arbre et n pour un graphe complet à n sommets.

La seule approche qui permet de traiter des problèmes dont l'espace d'état et l'espace d'action sont à la fois très grands est l'algorithme FPG [BA09] ($|\mathcal{S}| \approx 10^{75}, |\mathcal{A}| \approx 10^{150}$). Comme nous le verrons dans la section 2.5, FPG considère un critère moyen et non γ-pondéré et recherche des politiques paramétrées avec peu de paramètres en comparaison avec notre approche.

1.4.2 Le cadre des PDM sur graphe (PDMG)

Définition

Le cadre PDMG a été proposé pour résoudre des problèmes de décision séquentielle spatialisés [FS06, PS06, SPF12]. Il s'agit de prendre des décisions en chaque noeud d'un graphe G (un ensemble de parcelles agricoles par exemple). Dans le cadre PDMG, il y a donc une variable d'action associée à chaque variable d'état, elle-même associée à un nœud du graphe G. Chaque variable d'état S_i^{t+1} à l'instant $t+1$ n'est influencée que par la variable d'action A_i^t associée au même nœud et par un sous-ensemble des variables d'état à l'instant t (les parents de S_i^{t+1}, associées aux nœuds voisins dans le graphe orienté G, voir figure 1.7). De plus, la fonction de récompense se décompose en une somme de fonctions de récompenses locales associées à chaque nœud et qui ne dépendent que de la variable d'action du nœud et des variables d'état associées aux nœuds parents. Le cadre PDMG peut donc être vu comme un cas particulier du cadre PDMF-AF, comme le montre la définition suivante :

Définition 14 (PDMG [SPF12])**.** *Un PDM sur graphe (PDMG) est un tuple* $M = (n, \mathcal{S}, \mathcal{A}, \mathcal{T}, P, R)$ *où :*

- *n est le nombre de variables d'états, mais aussi le nombre de variables d'action*
- *L'espace d'états est un produit cartésien d'espaces finis :* $\mathcal{S} = \prod_{i=1}^{n} \mathcal{S}_i$.
- *L'espace d'actions est un produit cartésien d'espaces finis :* $\mathcal{A} = \prod_{j=1}^{n} \mathcal{A}_j$.
- *L'espace des temps (discret) est noté* $\mathcal{T}$.
- *La fonction de transition P est un réseau bayésien dynamique :*

$$\begin{aligned} \forall s^t, s^{t+1}, a^t,\ P(s^{t+1}, s^t, a^t) &= \mathbb{P}(S^{t+1} = s^{t+1} | S^t = s^t, A^t = a^t) \\ &= \prod_{i=1}^{n} \mathbb{P}(S_i^{t+1} = s_i^{t+1} | pa(S_i^{t+1}) = pa(s_i^{t+1})) \\ &= \prod_{i=1}^{n} P_i(s_i^{t+1} | pa(s_i^{t+1})) \end{aligned}$$

  où $pa(S_i^{t+1}) \subset \{S_1^t, ..., S_n^t\} \cup \{A_i^t\}$ *représente l'ensemble des parents de* S_i^{t+1} *dans le réseau bayésien dynamique et* $pa(s_i^{t+1})$ *représente la réalisation de ce vecteur aléatoire.*

- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$*, la fonction de récompense, se décompose de la manière suivante :*

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A},\ R(s,a) = \sum_{i=1}^{n} R_i(pa_R(R_i))$$

  où $pa_R(R_i) = pa(s_i^{t+1})$.

algorithmes	[KD02]		[RJF+12] FAR et MBFAR		[RJF+12] seq. hindsight	
mode de représentation	DBN+ADD		DBN+ADD/AADD		DBN+ADD/AADD	
méthode	exacte		exacte		approchée	
horizon	infini		infini		infini	
forme de récompense	struct.		struct.		struct.	
taille maximale	$m = 17$ $n = 17$ $\|\mathcal{S}\| = 10^6$ $\|\mathcal{A}\| = 10^5$	$m = 60$ $n = 16$ $\|\mathcal{S}\| = 10^5$ $\|\mathcal{A}\| = 10^{18}$	$m = 3$ $n = 6$ $\|\mathcal{S}\| = 10^9$ $\|\mathcal{A}\| = 27$	$m = 12$ $n = 12$ $\|\mathcal{S}\| = 4096$ $\|\mathcal{A}\| = 4096$	$m = 3$ $n = 6$ $\|\mathcal{S}\| = 10^9$ $\|\mathcal{A}\| = 27$	$m = 12$ $n = 12$ $\|\mathcal{S}\| = 4096$ $\|\mathcal{A}\| = 4096$
hypothèses, limites	vars bin		vars bin		vars bin	

(a) Algorithmes symboliques

algorithmes	[GLP02]	[SH04]	[BA09] FPG
mode de représentation			
méthode	approchée	approchée	approchée
horizon	infini	infini	infini (critère moyen)
forme de récompense			
taille maximale	$m = 15$ $n = 15$ $\|\mathcal{S}\| = 10^{14}$ $\|\mathcal{A}\| = 32000$	$m = 40$ $n = 12$ $\|\mathcal{S}\| = 4096$ $\|\mathcal{A}\| \approx 10^{12}$	$m = 500$ $n = 250$ $\|\mathcal{S}\| \approx 10^{75}$ $\|\mathcal{A}\| \approx 10^{150}$
hypothèses, limites	approximation linéaire de la FV	approximation de la FV avec produits d'experts	recherche de politiques factorisées paramétrées

(b) Algorithmes d'apprentissage par renforcement

algorithmes	[GKP01]
mode de représentation	DBN
méthode	approchée
horizon	infini
forme de récompense	add.
taille maximale	$m = 30$ $n = 30$ $\|\mathcal{S}\| \approx 10^{28}$ $\|\mathcal{A}\| \approx 10^9$
hypothèses, limites	approximation linéaire de la FV

(c) Algorithme de programmation linéaire approchée

TABLE 1.4 – Principaux algorithmes pour PDMF-AFs
struct. : structurée, add. : additive, vars bin : variables binaires, FV : fonction de valeur
Les tailles maximales de problèmes pouvant être traités sont issues des expériences présentées dans les publications et sont à prendre comme des ordres de grandeur plus que comme des données précises.

A noter que nous avons utilisé dans cette définition les mêmes notations que pour les PDMF-AFs (définition 13) afin de rendre la comparaison plus facile. Un PDMG est donc un PDMF-AF à récompense additive qui vérifie $m = n = r$ et $\forall i = 1...n,\ pa(s_i^{t+1}) = pa_R(R_i) \subset \{S_1^t, ..., S_n^t\} \cup \{A_i^t\}$.

Dans [SPF12] les notations sont différentes et la notion de graphe est plus apparente. Si N représente la fonction de voisinage liée au graphe G (*ie* $j \in N(i)$ signifie que le nœud j est parent/voisin du nœud i) et que $s^t_{N(i)}$ est une notation pour $\{s_j^t, j \in N(i)\}$, la fonction de transition est notée :

$$\forall s^t, s^{t+1}, a^t,\ P(s^{t+1}, s^t, a^t) = \prod_{i=1}^{n} P_i(s_i^{t+1} | s^t_{N(i)}, a_i^t)$$

et la fonction de récompense est notée :

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A},\ R(s,a) = \sum_{i=1}^{n} R_i(s_{N(i)}, a_i)$$

Dans l'exemple de la figure 1.7, où $n = 3$, on a : $N(1) = \{1,3\}, N(2) = \{1,2\}$ et $N(3) = \{2\}$. Cela signifie que : $pa(S_1^{t+1}) = \{S_1^t, S_3^t, A_1^t\}$, $pa(S_2^{t+1}) = \{S_1^t, S_2^t, A_2^t\}$, $pa(S_3^{t+1}) = \{S_2^t, A_3^t\}, pa(R_1) = \{S_1, S_3, A_1\}, pa(R_2) = \{S_1, S_2, A_2\}$, et $pa(R_3) = \{S_2, A_3\}$.

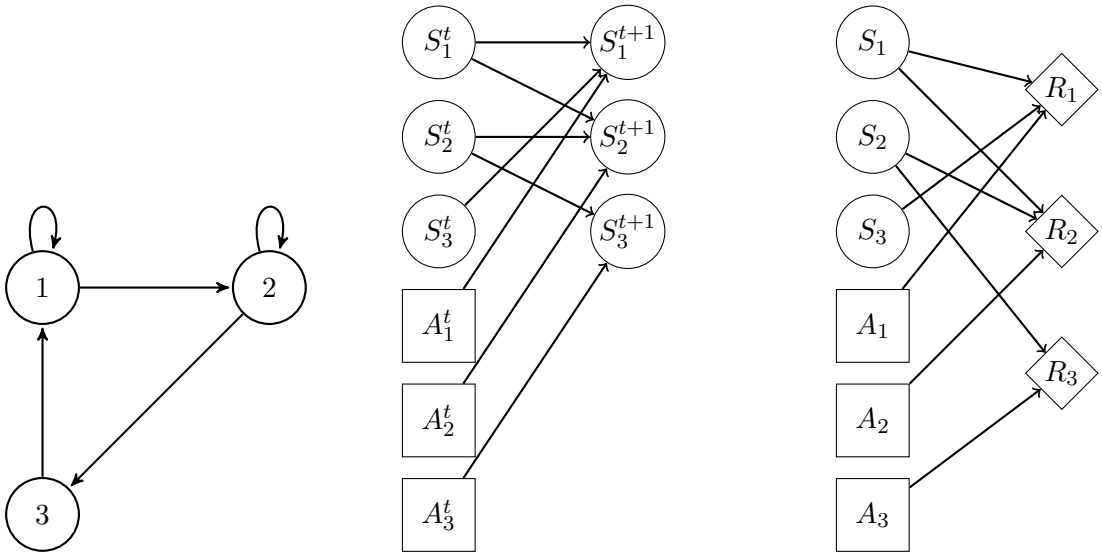


FIGURE 1.7 – Exemple de PDMG à $n = 3$ noeuds. A gauche, le graphe G représentant les dépendances entre les variables d'état : $N(1) = \{1,3\}, N(2) = \{1,2\}, N(3) = \{2\}$. Au milieu, le réseau bayésien dynamique représentant les indépendances dans la fonction de transition. A droite, la représentation graphique des indépendances dans la fonction de récompense. Notons que $\forall i = 1...n,\ pa_R(R_i) = pa(S_i^{t+1})$.

Dans le cadre des PDMGs, la recherche se limite aux politiques stationnaires déterministes factorisées (ou locales) basées sur la structure du graphe, c'est-à-dire qu'on

recherche une politique $\pi = (\pi_1, ..., \pi_n)$ constituée de politiques locales π_i se basant sur l'état des nœuds voisins ($\pi_i : \mathcal{S}_{N(i)} \rightarrow \mathcal{A}_i$). En se restreignant à la recherche des politiques factorisées déterministes, il n'existe plus de garantie de trouver une politique de fonction de valeur optimale (un contre-exemple est donné dans [SPF12]).

Méthodes de résolution

[FS06] a proposé un algorithme de type programmation linéaire approchée pour résoudre les PDMGs. L'hypothèse sur la fonction de valeur est la suivante :

$$\forall s \in \mathcal{S},\ V(s) \approx \sum_{i=1}^{n} V_i(s_i)$$

La complexité de l'algorithme est linéaire en le nombre de variables d'état n et exponentielle en la largeur du graphe G (c'est-à-dire en la taille du plus grand voisinage : $\max_{i=1...n} |N(i)|$).

[PS06, SPF12] a également proposé un algorithme de type itération de la politique approchée que nous allons décrire plus en détail. Notre approche de résolution approchée de PDMF-AF, présentée dans le chapitre 2, sera en effet comparée à cet algorithme, appelé MF-API (*mean-field approximate policy iteration*), lorsque le PDMF-AF est de type PDMG. Dans cet algorithme, les étapes d'évaluation et d'amélioration de la politique sont approchées. L'évaluation approchée est dite en champ moyen car la chaîne de Markov associée à l'exécution de la politique à évaluer est approchée par n chaînes de Markov unidimensionnelles indépendantes mais non stationnaires. Cette approximation implique une décomposition additive de la fonction de valeur selon :

$$\forall s \in \mathcal{S},\ V(s) \approx \sum_{i=1}^{n} V_i(pa(s_i))$$

La fonction de valeur, qui porte sur les variables d'état, est une somme de fonctions de faible arité, sur ces mêmes variables. L'étape d'amélioration de l'algorithme est également approchée de manière à ce que, après amélioration, la politique reste locale. Cependant, il n'y a pas de garantie que cette étape entraîne effectivement une amélioration de la politique pour tout état initial $s \in \mathcal{S}$ (une politique locale qui soit meilleure pour tout état initial n'existe pas forcément de toute façon). Une comparaison expérimentale [SPF12] montre que l'algorithme de programmation linéaire approchée [FS06] est plus rapide mais que dans certains cas MF-API donne de meilleures politiques et de manière générale une meilleure approximation de la fonction de valeur.

[FGS09] a aussi proposé des méthodes de type apprentissage par renforcement adaptées de méthodes existantes. Plus récemment, [CLCI13] a proposé pour les PDMGs un algorithme de type itération de la valeur approchée. La fonction de valeur est approchée par un produit de fonctions locales et calculée par minimisation d'une divergence de Kullback avec un algorithme de type *belief propagation* (voir section A.2). Comme le montre la table 1.5, les expériences présentées dans cet article ne vont pas aussi loin en taille que les autres approches, et les temps de calcul ne sont pas donnés, mais les

résultats obtenus sont compétitifs avec les algorithmes ALP et MF-API (supérieurs sur certains problèmes, inférieurs sur d'autres).

Bilan

Le cadre de représentation PDMG ne permet pas de représenter tous les PDM à espace d'état et d'action factorisés, puisque espace d'état et d'action doivent avoir la même factorisation, de même que transition et récompense. Les algorithmes de résolution pour PDMGs sont comparés dans la table 1.5. Ils peuvent traiter des problèmes de taille très grande, allant jusqu'à $|\mathcal{S}| = |\mathcal{A}| = 10^{150}$. Mais ils recherchent des politiques factorisées déterministes dont la factorisation est la même que celle de la transition et de la récompense.

algorithmes	[FS06] ALP	[FGS09]	[SPF12] MF-API	[CLCI13] FVI
mode de représentation	graphe orienté	graphe orienté	graphe orienté	graphe orienté
méthode	approchée	approchée	approchée	approchée
horizon	infini	infini	infini	infini
type	PLA	AR	IPA	IVA
forme de récompense	add.	add.	add.	add.
taille maximale	$n = 36$ $\|\mathcal{S}\| \approx 10^{21}$ $\|\mathcal{A}\| \approx 10^{11}$	$n = 100$ $\|\mathcal{S}\| \approx 10^{60}$ $\|\mathcal{A}\| \approx 10^{30}$	$n = 500$ $\|\mathcal{S}\| \approx 10^{150}$ $\|\mathcal{A}\| \approx 10^{150}$	$n = 20$ $\|\mathcal{S}\| \approx 10^{9}$ $\|\mathcal{A}\| \approx 10^{6}$
hypothèses liées au cadre	même factorisation pour l'espace d'état et l'espace d'action même factorisation pour transition, récompense et politique recherche de politiques factorisées déterministes			
approximation de la FV	linéaire	linéaire	linéaire	multiplicative

TABLE 1.5 – Principaux algorithmes pour PDMGs
add. : additive
PLA : programmation linéaire approchée
AR : apprentissage par renforcement
IPA : itération de la politique approchée
IVA : itération de la valeur approchée
FV : fonction de valeur
Les tailles maximales de problèmes pouvant être traités sont issues des expériences présentées dans les publications et sont à prendre comme des ordres de grandeur plus que comme des données précises.

1.4.3 Dec-(PO)MDPs à espace d'état factorisé

Nous considérons ici les travaux effectués dans le cadre multiagent décentralisé (Dec-POMDP, voir section 1.3) où l'espace d'état est également factorisé (décrit par plusieurs

variables). Tout d'abord, nous décrivons les travaux effectués dans le cas où l'espace d'état a la même factorisation que l'espace d'action, puis les travaux effectués dans le cas le plus général.

Le cadre des Dec-MDPFs

Dans cette section, nous supposons un horizon fini $\mathcal{T} = \{0, 1, ..., T-1\}$. Nous considérons les Dec-MDPs dont l'espace d'état est factorisé en un produit cartésien d'espaces d'état locaux associés à chaque agent. Sous certaines hypothèses, la complexité du problème est moins grande que dans le cas général des Dec-POMDPs (où le problème est NEXP-complet, voir [BGIZ02]), et des algorithmes de résolution efficaces ont été proposés.

Définition 15 (DEC-MDPF,[BCSM08])**.** *Un DEC-MDP factoré (DEC-MDPF) à m agents est un Dec-MDP à m agents tel que l'état du système peut être décomposé en $m+1$ composantes : $\mathcal{S} = \mathcal{S}_0 \times \mathcal{S}_1 \times ... \times \mathcal{S}_m$, où pour tout $i \in \{1, 2, ..., m\}$, $\mathcal{S}_i$ correspond aux composantes de l'état de l'agent i qui sont observées et affectées par au moins un des agents du système. $\mathcal{S}_0$ décrit l'ensemble des composantes de l'état du système qui peuvent éventuellement être observées par les agents mais qui ne sont pas affectées par les actions des agents. Dans un DEC-MDP factoré, l'état S_i d'un agent i appartient donc à l'ensemble $\mathcal{S}_0 \times \mathcal{S}_i$.*

Définition 16. *Un DEC-MDPF est dit à transitions et observations indépendantes si les $\mathcal{S}_i$ sont disjoints et si la fonction de transition et la fonction d'observation peuvent être factorisées en un produit de probabilités :*

$$P(s', s, a) = \prod_{i=1}^{m} P_i(s'_i | s_i, a_i)$$

$$O(o|s, a, s') = \prod_{i=1}^{m} O_i(o_i | s, a, s', o_{\setminus i})$$

où $o_{\setminus i} = (o_1, ..., o_{i-1}, o_{i+1}, ..., o_m)$.

Définition 17. *Un DEC-MDPF est dit localement totalement observable si il existe une application $J_i : \Omega_i \to \mathcal{S}_i$ pour chaque agent $i \in \{1, ..., m\}$ telle que si $O(o|s, a, s') > 0$ alors $\forall i \in \{1, ..., m\},\ J_i(o_i) = s'_i$.*

Propriété 1 ([GZ04])**.** *Dans le cas d'un DEC-MDPF à transitions et observations indépendantes, l'état courant partiel observé par l'agent i s_i^t est une statistique suffisante pour l'historique de ses observations $(o_i^1, ..., o_i^t)$. Dans le cas d'un DEC-MDPF à transitions et observations indépendantes, une politique locale est donc une application $\pi_i : \mathcal{S}_i \times \mathcal{T} \to \mathcal{A}_i$.*

Propriété 2 ([BZLG04])**.** *Le problème de décision associé à la résolution d'un DEC-MDPF à transitions et observations indépendantes qui est localement totalement observable est NP-complet.*

Dans le cadre Dec-POMDP (sans hypothèse de totale observabilité collective), les hypothèses de transition et observation indépendantes ne permettent pas de réduire la complexité du problème [BCSM08].

[BZLG04] a proposé le premier algorithme permettant de résoudre de manière exacte les DEC-MDPFs à transitions et observations indépendantes : l'algorithme *coverage-set*. Dans le cas des DEC-MDPs généraux, cet algorithme ne peut en pratique résoudre que de petits problèmes à deux agents et très peu de dépendances. [PZ09] a ensuite proposé un algorithme plus efficace, basé sur une reformulation du problème en un programme bilinéaire. Plus récemment, [DAD12] se ramène à la résolution d'un MDP à états continus correspondant à des distributions de probabilité sur les états du DEC-MDP originel (*state occupancy distributions*). Les auteurs proposent un algorithme de recherche heuristique plus efficace que les précédentes approches, permettant de traiter des problèmes où $m = T = 10$.

Autres sous-cadres de Dec-POMDPs à espace d'état factorisé comme l'espace d'action

Dans cette section, nous nous intéressons toujours au cas où l'espace d'état a la même factorisation que l'espace d'action (une variable d'état est associée à chaque agent).

De nombreuses sous-catégories de Dec-MDPs ont été proposées avec des algorithmes de résolution spécifiques, comme les Dec-MDPs dirigés par les évènements [BZL04, ML11] ou ceux où une communication entre agents est possible mais coûteuse [GZ08]. [MV11] propose le cadre Dec-SIMDP (*decentralized sparse-interaction Markov decision process*), qui est plus général que le cadre des Dec-MDPFs à transitions et observations indépendantes.

Des sous-cadres de Dec-POMDPs ont également été proposés. [NVTY05] définit les ND-POMDPs, qui ne sont autres que des Dec-POMDPs à transition et observation indépendantes et à fonction de récompense additive basée sur un graphe d'interaction. La résolution des ND-POMDPs à horizon fini reste un problème NEXP-complet. L'algorithme de recherche heuristique de [DABC14] permet de résoudre exactement des ND-POMDPs ayant jusqu'à 15 agents, mais pour un horizon limité (de l'ordre de $T = 7$). [KZT11] propose un algorithme EM pour Dec-POMDPs à espace d'état factorisé permettant de chercher des politiques sous forme de contrôleur à états finis. Cet algorithme s'applique que l'horizon considéré soit fini ou infini. Il s'appuie sur une hypothèse de fonction de valeur additive ($V_\pi(s^0)$ doit s'écrire comme une somme de fonctions de faible arité sur les variables décrivant s^0), qui est vérifiée pour un certain nombre de sous-classes de Dec-POMDPs, comme les Dec-MDPFs à transition et observation indépendantes, les ND-POMDPs [NVTY05] ou les TD-POMDPs [WD10]. Enfin, [CM12] propose des algorithmes de résolution pour le modèle DyLIM (*dynamic local interaction model*), dans lequel les interactions entre agents peuvent être dynamiques, c'est-à-dire évoluer au cours du temps.

Le cadre général des Dec-POMDPs à espace d'état factorisé

Certains travaux récents [OWS13] s'attachent à la résolution de Dec-POMDPs dont l'espace d'état est factorisé (décrit par plusieurs variables) mais où on n'a pas forcément une variable d'état associée à chaque variable d'action (ou agent). On parle alors de Dec-POMDP factorisé :

Définition 18 (Dec-POMDP factorisé [Oli10])**.** *Un Dec-POMDP factorisé à m agents est un Dec-POMDP à m agents $(\mathcal{S}, \mathcal{A}, \mathcal{T}, P, R, \Omega, O, P^0)$ dont l'espace d'états est un produit cartésien d'espaces finis : $\mathcal{S} = \prod_{i=1}^{n} \mathcal{S}_i$. Dans la plupart des travaux, la fonction de transition et la fonction d'observation sont représentées par un réseau bayésien dynamique. De plus, la fonction de récompense est supposée additive :*

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A}, R(s,a) = \sum_{\alpha=1}^{r} R_\alpha(pa_R(R_\alpha))$$

où $pa_R(R_\alpha) \subset \{s_i, i = 1...n, a_j, j = 1...m\}$.

Dans le cas de l'horizon fini, [OWS13] propose une approche basée sur les jeux bayésiens collaboratifs graphiques pour résoudre de manière approchée des problèmes avec un grand nombre d'agents mais dont l'horizon est limité (par exemple, $m = 1000$ pour $T = 3$, $m = 750$ pour $T = 4$, $m = 100$ pour $T = 6$).

[PP11], sans hypothèse sur la fonction de valeur, propose un algorithme EM approché pour le cas de l'horizon infini. L'algorithme proposé a une complexité polynomiale en le nombre d'agents et de variables d'états. Des expériences sont présentées pour des problèmes ayant jusqu'à 10 agents.

Bilan

Les approches les plus récentes et compétitives pour la résolution de Dec-POMDPs à espace d'état factorisé sont comparées dans la table 1.6. Pour les approches à horizon fini, on constate un compromis entre l'horizon et la taille des problèmes qui peuvent être traités. Ainsi, un problème à deux agents pourra être résolu pour un horizon de 1000, mais un problème à 1000 agents ne pourra être résolu que pour un horizon de 3.

Pour les approches à horizon infini, la résolution de problèmes de taille importante ($|\mathcal{S}| \approx 10^{17}, |\mathcal{A}| \approx 10^{12}, |\Omega| \approx 10^{14}$) demande à ce que la fonction de valeur puisse s'écrire comme une somme de fonctions faisant intervenir un petit nombre d'agents et de variables d'état [KZT11].

algorithmes	[DAD12]		[DABC14] FB-HVSI (ext)		[KZT11]	[PP11]	[OWS13] FFSPC	
mode de représentation						DBN	DBN	
méthode	exacte		exacte		approchée	approchée	approchée	
horizon	fini		fini		fini ou infini	infini	fini	
type	RH		RH		EM	EM approché	AJB	
forme de récompense						additive	additive	
taille maximale	$m = 2$ $\|\mathcal{A}\| = 25$ $\|\Omega\| = 4096$ $T = 100$	$m = 2$ $\|\mathcal{A}\| = 25$ $\|\Omega\| = 81$ $T = 1000$	$m = 7$ $\|\mathcal{S}\| = 12$ $\|\mathcal{A}\| = 2187$ $\|\Omega\| = 128$ $T = 8$	$m = 15$ $\|\mathcal{S}\| = 60$ $\|\mathcal{A}\| \approx 10^9$ $\|\Omega\| = 32768$ $T = 7$	$m = 20$ $\|\mathcal{S}\| \approx 10^{17}$ $\|\mathcal{A}\| \approx 10^{12}$ $\|\Omega\| \approx 10^{14}$	$m = 10$ $\|\mathcal{S}\| = 10^9$ $\|\mathcal{A}\| = 1024$ $\|\Omega\| \approx 6 \times 10^7$	$m = 1000$ $\|\mathcal{S}\| \approx 10^{301}$ $\|\mathcal{A}\| \approx 10^{301}$ $\|\Omega\| \approx 10^{301}$ $T = 3$	$m = 100$ $\|\mathcal{S}\| \approx 10^{30}$ $\|\mathcal{A}\| \approx 10^{30}$ $\|\Omega\| \approx 10^{30}$ $T = 6$
hypothèses, limites	Dec-MDPF à transition et observation indépendantes		localité d'interaction (ex : ND-POMDP)		FV additive recherche de contrôleurs à états finis			

TABLE 1.6 – Principaux algorithmes pour Dec-PODMPs à espace d'état factorisé
struct. : structurée, add. : additive
RH : recherche heuristique
EM : algorithme *Expectation Maximization*
AJB : approximation par des jeux bayésiens
FV : fonction de valeur
Les tailles maximales de problèmes pouvant être traités sont issues des expériences présentées dans les publications et sont à prendre comme des ordres de grandeur plus que comme des données précises.

1.4.4 Bilan

Les approches développées pour les PDM à espace d'état et d'action factorisés diffèrent dans leur façon de représenter les politiques. Certaines les représentent sous forme d'arbres ou de diagrammes de décision, ce sont les approches symboliques (voir par exemple [RJF+12]). L'inconvénient est qu'une telle politique, globale mais représentée sous forme d'arbre, peut potentiellement prendre en mémoire un espace exponentiel.

D'autres approches utilisent des politiques factorisées. C'est le cas des approches développées dans le cadre des Dec-POMDPs à espace d'état factorisé [DAD12, KZT11, OWS13] et dans le cadre des PDMGs [SPF12, CLCI13]. C'est le cas aussi de [SH04] ou [BA09] en apprentissage par renforcement, où chaque politique locale est en plus paramétrée.

Finalement, rares sont les algorithmes existants pour PDM à espace d'état et d'action factorisés qui s'attaquent à des problèmes de la taille de ceux qui nous intéressent (typiquement, de l'ordre de 100 variables d'état et d'action pas forcément binaires, c'est-à-dire $|\mathcal{S}| = |\mathcal{A}| \geq 10^{30}$). La principale exception est celle des algorithmes proposés dans le cadre PDMG (voir section 1.4.2). Cependant, ce cadre présente différentes restrictions que nous aimerions lever (même factorisation de l'espace d'état et d'action, même structure pour la transition, la récompense et la politique). Il existe d'autres exceptions dans des cadres plus généraux, notamment les algorithmes FPG [BA09] et FFSPC [OWS13]. Nous les comparerons plus en détail avec notre approche dans la section 2.5. Mais l'algorithme FPG suppose des politiques factorisées paramétrées avec peu de paramètres, ce qui ne permet pas d'explorer une grande partie de l'espace des politiques factorisées. Et l'algorithme FFSPC, à horizon fini, recherche des politiques basées sur l'historique des observations, et ne peut résoudre des problèmes de grande taille que pour un horizon limité ($T = 6$).

Dans le chapitre qui suit, nous proposons un nouveau cadre de PDM à espace d'état et d'action factorisés, ainsi que des algorithmes de résolution associés permettant de traiter des problèmes de grande taille.

Chapitre 2

Contributions à la résolution de PDM à espace d'état et d'action factorisés

Dans ce chapitre, nous présentons nos contributions à la résolution de PDM à espace d'état et d'action factorisés. Tout d'abord, nous présentons un nouveau cadre, le cadre PDMF^3, dans lequel la politique est stochastique et factorisée et sa structure fait partie des données du problème (voir section 2.1). Dans la section 2.2, nous décrivons la méthode que nous proposons pour l'évaluation de politiques stochastiques factorisées, basée sur l'utilisation d'algorithmes d'inférence dans les modèles graphiques. Puis dans la section 2.3, nous analysons les caractéristiques du problème d'optimisation de politiques stochastiques factorisées dans les PDMF^3, et nous proposons différents algorithmes de type itération de la politique. Nous évaluons ces algorithmes dans la section 2.4, sur des problèmes aléatoires ou 'jouets' de complexité croissante. Enfin, nous discutons les similitudes et différences avec des travaux proches dans la section 2.5, et envisageons des perspectives à ce travail dans la section 2.6.

2.1 Un nouveau cadre de PDM à espaces d'état et d'action factorisés

2.1.1 Le cadre PDMF^3

Afin de résoudre des problèmes de décision à l'échelle du paysage plus généraux que ceux qui ont pu être résolus jusqu'à présent, nous proposons une généralisation des PDMGs. Nous nous plaçons en fait dans le cadre PDMF-AF à récompense additive avec une structure de politique donnée. Nous appelons ce nouveau cadre PDMF^3, puisqu'à la fois les fonctions de transition et de récompense *et* les politiques sont factorisées, mais avec leur factorisation propre (contrairement aux PDMGs où transition, récompense et politiques sont factorisées de la même manière). Voici la définition formelle d'un PDMF^3 :

Définition 19 (PDMF^3)**.** *Un PDMF^3 est un tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, P, pa_\delta, R, P^0)$ où :*

- *L'espace d'états est un produit cartésien d'espaces finis : $\mathcal{S} = \prod_{i=1}^{n} \mathcal{S}_i$.*
- *L'espace d'actions est un produit cartésien d'espaces finis : $\mathcal{A} = \prod_{j=1}^{m} \mathcal{A}_j$, où $\forall j = 1...m,\ \mathcal{A}_j = \{1, ..., |\mathcal{A}_j|\}$.*
- *L'espace des temps (discret) est noté $\mathcal{T}$.*
- *La fonction de transition P vérifie :*

$$\begin{aligned} \forall s^t, s^{t+1}, a^t,\ P(s^{t+1}, s^t, a^t) &= \mathbb{P}(S^{t+1} = s^{t+1} | S^t = s^t, A^t = a^t) \\ &= \prod_{i=1}^{n} \mathbb{P}(S_i^{t+1} = s_i^{t+1} | pa_P(S_i^{t+1}) = pa_P(s_i^{t+1})) \\ &= \prod_{i=1}^{n} P_i(s_i^{t+1} | pa_P(s_i^{t+1})) \end{aligned}$$

  $pa_P(S_i^{t+1}) = pa_P^S(S_i^{t+1}) \cup pa_P^A(S_i^{t+1})$ représente l'ensemble des variables influençant S_i^{t+1}, avec $pa_P^S(S_i^{t+1}) \subset \{S_j^t, j = 1...n, S_{j'}^{t+1}, j' = 1...n, j' \neq i\}$ et $pa_P^A(S_i^{t+1}) \subset \{A_k^t, k = 1...m\}$. $pa_P(s_i^{t+1})$ représente une réalisation du vecteur aléatoire $pa_P(S_i^{t+1})$.
- *La structure des politiques recherchées est définie par m listes de variables $\{pa_\delta(A_j), j = 1...m\}$ sur lesquelles on peut se baser pour décider de chaque variable d'action (voir exemple figure 2.1) :*

$$\forall j \in \{1, ..., m\},\ pa_\delta(A_j) = pa_\delta^S(A_j) \cup pa_\delta^A(A_j)$$

  où $pa_\delta^S(A_j) \subset \{S_i, i = 1...n\}, pa_\delta^A(A_j) \subset \{A_k, k = 1...m, k \neq j\}$. Ces variables sont l'équivalent des variables d'observation dans les Dec-POMDPs (voir section 1.4.3). On notera $pa_\delta(a_j)$ une réalisation du vecteur aléatoire $pa_\delta(A_j)$.
- *Le modèle graphique représentant les dépendances entre variables d'état et d'action, issu de pa_P et pa_δ, ne contient pas de circuit (ce modèle graphique est en fait un réseau bayésien dynamique).*
- *La fonction de récompense R se décompose de la manière suivante :*

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A},\ R(s, a) = \sum_{\alpha=1}^{r} R_\alpha(pa_R(R_\alpha))$$

  où $pa_R(R_\alpha) = pa_R^S(R_\alpha) \cup pa_R^A(R_\alpha)$ avec $pa_R^S(R_\alpha) \subset \{s_i, i = 1...n\}$ et $pa_R^A(R_\alpha) \subset \{a_j, j = 1...m\}$. R est supposée positive et bornée par R_{max}.
- *La distribution initiale sur les états, P^0, est factorisée et représentée par un réseau bayésien.*

Nous nous intéressons aussi bien aux problèmes à horizon infini (avec facteur d'amortissement) qu'aux problèmes à horizon fini. Cependant, dans le cas de l'horizon infini, nous ferons une approximation de la valeur en utilisant un horizon fini 'suffisamment grand' (voir section 2.2.4).

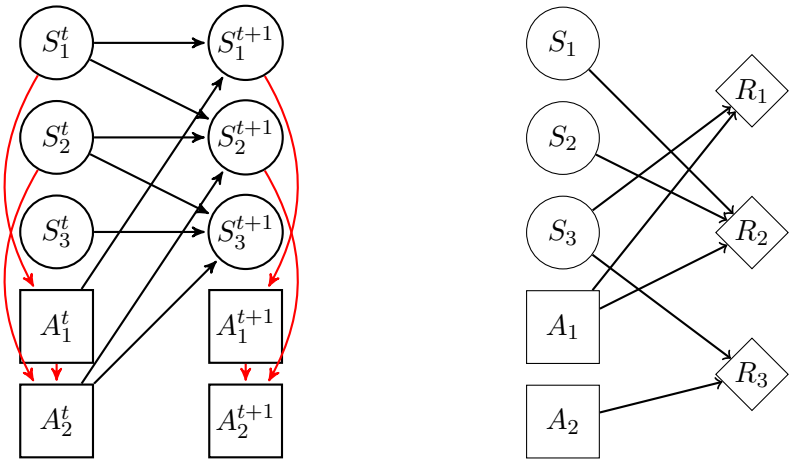


FIGURE 2.1 – Exemple de PDMF^3 correspondant au PDMF-AF à récompense additive de la figure 1.6 auquel ont été ajoutées en rouge les flèches correspondant à une structure de politique donnée : $pa_\delta(A_1) = \{S_1\}, pa_\delta(A_2) = \{S_2, A_1\}$; à gauche le réseau bayésien dynamique représentant la structure de la transition et des politiques, à droite la représentation graphique de la structure de la récompense.

2.1.2 Politiques recherchées

Nous recherchons, pour cette structure de politique donnée, la meilleure politique stochastique factorisée (stationnaire). En effet, comme nous le montrerons dans la section 2.1.5, pour une structure de politique donnée la meilleure politique factorisée n'est pas forcément déterministe. Une recherche parmi les politiques factorisées stochastiques peut donc conduire à une meilleure politique qu'une recherche parmi les politiques factorisées déterministes, comme celle qui est faite dans tous les travaux sur les PDMGs [SPF12]. De plus, une telle recherche donne accès aux nombreux algorithmes d'optimisation continue.

La meilleure politique stochastique factorisée n'a aucune raison d'être stationnaire. Cependant, nous limitons notre recherche, y compris dans le cas d'un horizon fini, aux politiques stationnaires car elles sont plus faciles à interpréter, calculer et stocker en mémoire.

Définition 20 (politique stochastique factorisée)**.** *Une politique stochastique factorisée (PSF) δ de structure pa_δ est une politique stationnaire stochastique (voir définition 5) vérifiant :*

$$\forall a^t \in \mathcal{A}, \forall s^t \in \mathcal{S}, \delta(a^t|s^t) = \prod_{j=1}^{m} \delta_j(a_j^t|pa_\delta(a_j^t))$$

où $\delta_j(a_j^t|pa_\delta(a_j^t))$ représente la probabilité de choisir l'action a_j^t pour la variable d'action A_j au temps t étant donné que les variables parentes de A_j sont dans l'état joint $pa_\delta(a_j^t)$.

Dans l'exemple de la figure 2.1, on a $pa_\delta(A_1) = \{S_1\}, pa_\delta(A_2) = \{S_2, A_1\}$. Cela signifie que, à chaque pas de temps, pour décider de A_1 on se base sur la valeur de la

variable S_1 puis on décide de A_2 à partir de la valeur choisie pour A_1 et de la valeur de la variable S_2.

En toute généralité, les structures des politiques factorisées dans le cadre PDMF^3 peuvent être plus complexes, comme nous le verrons dans l'application en agroécologie au chapitre 3. Elles peuvent par exemple faire intervenir des variables d'actions du pas de temps précédent :

$$\forall a^t \in \mathcal{A}, \forall s^t \in \mathcal{S}, \delta(a^t|s^t, a^{t-1}) = \prod_{j=1}^{m} \delta_j(a_j^t|pa_\delta(a_j^t))$$

où $pa_\delta(a_j^t) \subset \{s_i^t, i = 1...n, a_k^t, k = 1...m, k \neq j, a_{k'}^{t-1}, k' = 1...m\}$, ou même des variables d'action de plusieurs pas de temps en arrière. La seule condition est qu'il n'y ait pas de cycle dans le réseau bayésien dynamique sur les variables d'état et d'action (voir figure 2.1, schéma de gauche). Mais pour des raisons de simplicité des notations nous nous limitons dans ce chapitre à des structures de politique vérifiant :

$$\forall j = 1...m,\ pa_\delta^S(A_j) \subset \{S_i, i = 1...n\} \text{ et } pa_\delta^A(A_j) \subset \{A_k, k = 1...m, k \neq j\}.$$

2.1.3 Lien avec les autres cadres

Le cadre PDMG (voir définition 14) peut être vu comme un cas particulier du cadre PDMF^3 où $m = r = n$ et :

$$\forall i \in \{1, ..., n\}, pa_R(R_i) = pa_P(S_i) = \{S_{N(i)}, A_i\} \text{ et } pa_\delta(A_i) = \{S_{N(i)}\}$$

Mais, contrairement, aux approches de résolution des PDMG, nous cherchons des politiques factorisées stochastiques, pas forcément déterministes.

Le cadre PDMF^3 peut être vu à son tour comme un cas particulier de Dec-POMDP factorisé (voir définition 18) où :

— $\forall j = 1...m, \Omega_j = \prod_{k/k \in pa_\delta^S(A_j)} \mathcal{S}_k \times \prod_{k/k \in pa_\delta^A(A_j)} \mathcal{A}_k$: les observations des agents sont un sous-ensemble des variables d'état (et éventuellement des actions prises par d'autres agents)

— la fonction d'observation O est déterministe :

$$\forall o \in \Omega, \forall (s, s') \in \mathcal{S}^2, \forall a \in \mathcal{A}, O(o|s, a, s') = \prod_{j=1}^{m} O_j(o_j|pa_\delta(a_j)) = \prod_{j=1}^{m} \begin{cases} 1 & \text{si } o_j = pa_\delta(a_j) \\ 0 & \text{sinon} \end{cases}$$

Dans le cadre PDMF^3, le système n'est pas forcément collectivement totalement observable (on ne peut pas forcément reconstruire l'état du système à partir des observations de tous les agents).

Les politiques que l'on recherche peuvent être vues comme des cas particuliers de contrôleurs à états finis (voir définition 11) où pour chaque agent $j \in \{1, ..., m\}$ le contrôleur à états finis $< N_j, \eta_j^0, \eta_j, \psi_j >$ est donné par :

— $N_j = \Omega_j = \prod_{k/k \in pa_\delta^S(A_j)} \mathcal{S}_k \times \prod_{k/k \in pa_\delta^A(A_j)} \mathcal{A}_k$

— $\eta_j^0(n) = P^0(pa_\delta(a_j))$
— η_j est déterministe et $\eta_j(n, o, n') = 1$ si $n = o$ et 0 sinon
— $\psi_j(n, a) = \delta_j(a_j|pa_\delta(a_j))$.

Nous verrons que l'approche de résolution d'un PDMF^3 que nous proposons pourrait s'étendre au cas des Dec-POMDPs à espace d'état factorisé. Cependant, ce n'est pas le cadre dans lequel nous nous sommes placés car il n'est pas adapté pour la modélisation du problème d'agroécologie qui nous intéresse (voir chapitre 3).

2.1.4 Choix d'une structure pour la politique

Notre but est de résoudre des problèmes pour lesquels la structure de la politique est apparente dans l'expression du problème, dans lesquels elle fait partie des contraintes du problème. Par exemple, dans les problèmes agronomiques, elle peut être liée aux parcelles que possède un agriculteur (et qui ne sont pas forcément voisines). L'agriculteur ne dispose pas d'informations sur les parcelles qui ne lui appartiennent pas. Cependant, nous proposons ici une structure 'naturelle' pour les problèmes (sans arcs synchrones) dans lesquels la structure de la politique n'est pas donnée, notamment pour les problèmes aléatoires que nous générerons pour la validation expérimentale (voir section 2.4). Cette structure 'naturelle' consiste à décider de l'action A_k ($k \in \{1...m\}$) en se basant sur :

— les variables d'état qui interviennent dans les mêmes fonctions de récompense que A_k
— et les variables d'état qui influencent les mêmes variables d'état que A_k.

En effet, si on connaît les variables d'état qui influencent S_j avant de décider de A_k qui influence aussi S_j, on pourra mieux décider de A_k de manière à contrôler S_j.

Plus formellement, la structure que nous proposons lorsqu'une structure de politique n'émerge pas de la nature du problème est la suivante :

$$\begin{aligned}\forall k = 1...m,\ pa_\delta(A_k^t) \quad = \quad & \{S_i^t, i \in [\![1;n]\!]/\exists i' \in [\![1;n]\!], A_k^{t-1} \in pa_P(S_{i'}^t) \text{ et } S_i^{t-1} \in pa_P(S_{i'}^t)\} \\ & \bigcup \{S_j^t, j \in [\![1;n]\!]/\exists \alpha \in [\![1;r]\!], A_k^t \in pa_R(R_\alpha^t) \text{ et } S_j^t \in pa_R(R_\alpha^t)\}\end{aligned}$$

Le fait de ne prendre que des variables d'état permet de ne pas créer de cycles. C'est une généralisation de la structure choisie pour les PDMGs. L'inconvénient est que cette structure naturelle, sur les gros problèmes, peut conduire à un ensemble de parents pour A_k trop grand pour permettre la résolution du PDMF^3 (dans ce cas nous conseillons de ne garder que la condition sur les fonctions de récompense, qui nous semble plus importante en pratique).

Nous ne nous intéressons pas au problème de l'optimisation de la structure de la politique, c'est-à-dire à la recherche, sous des contraintes de parcimonie, de la meilleure politique stochastique factorisée de structure quelconque. Mais cela peut faire partie de perspectives à ce travail.

2.1.5 Contre-exemple montrant l'intérêt de considérer des PSFs

Cet exemple est un cas simple où, à structure de politique fixée, il existe une politique stochastique meilleure que toute politique déterministe. Il s'agit d'un PDMF^3 à $n = 2$

variables d'état binaires, $m = 1$ variable d'action binaire et $r = 1$ fonction de récompense. La figure 2.2 montre la structure de la transition, de la politique et de la récompense. La structure de la politique est donnée : $pa_\delta(A_1) = \{S_2\}$. Pour décider de la variable d'action A_1, on se base donc uniquement sur la variable d'état S_2. Les tables sont les suivantes :

$$P(S_1^{t+1} = s_1' | S_2^t = s_2, A_1^t = 1) = \begin{array}{c} \\ s_1' \end{array} \begin{array}{c} s_2 \\ \begin{pmatrix} 0.7 & 0.6 \\ 0.3 & 0.4 \end{pmatrix} \end{array}$$

$$P(S_1^{t+1} = s_1' | S_2^t = s_2, A_1^t = 2) = \begin{array}{c} \\ s_1' \end{array} \begin{array}{c} s_2 \\ \begin{pmatrix} 0.5 & 0.2 \\ 0.5 & 0.8 \end{pmatrix} \end{array}$$

$$P(S_2^{t+1} = s_2' | S_1^t = s_1, A_1^t = 1) = \begin{array}{c} \\ s_2' \end{array} \begin{array}{c} s_1 \\ \begin{pmatrix} 0.6 & 0.6 \\ 0.4 & 0.4 \end{pmatrix} \end{array}$$

$$P(S_2^{t+1} = s_2' | S_1^t = s_1, A_1^t = 2) = \begin{array}{c} \\ s_2' \end{array} \begin{array}{c} s_1 \\ \begin{pmatrix} 0.5 & 0.3 \\ 0.5 & 0.7 \end{pmatrix} \end{array}$$

$$R(s_1, a_1) = \begin{array}{c} \\ s_1 \end{array} \begin{array}{c} a_1 \\ \begin{pmatrix} 0.45 & 0.8 \\ 1 & 0.7 \end{pmatrix} \end{array}$$

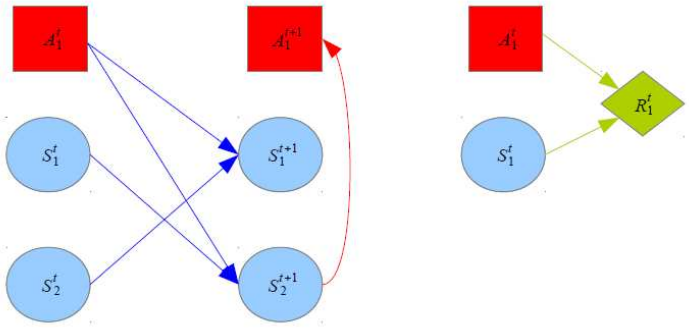


FIGURE 2.2 – Structure de la transition (en bleu), de la politique (en rouge) et de la récompense (en vert) d'un PDMF[3] pour lequel il existe une politique factorisée stochastique meilleure que toute politique factorisée déterministe.

Les quatre politiques déterministes possibles sont :

$$\delta_1(a_1|s_2) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_2 \\ \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \end{array}, \delta_2(a_1|s_2) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_2 \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}, \delta_3(a_1|s_2) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_2 \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array},$$

$$\delta_4(a_1|s_2) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_2 \\ \left(\begin{array}{cc} 0 & 0 \\ 1 & 1 \end{array} \right) \end{array}$$

Leurs valeurs exactes à horizon infini, pour $\gamma = 0.9$ et P^0 uniforme, sont respectivement :

$$V_{\delta_1}^{R,\infty}(P^0) = 6.4630, V_{\delta_2}^{R,\infty}(P^0) = 7.3256, V_{\delta_3}^{R,\infty}(P^0) = 7.3046, V_{\delta_4}^{R,\infty}(P^0) = 7.3326$$

Or la politique stochastique définie par

$$\delta_5(a_1|s_2) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_2 \\ \left(\begin{array}{cc} 0 & 0.5 \\ 1 & 0.5 \end{array} \right) \end{array}$$

a pour valeur $V_{\delta_5}^{R,\infty}(P^0) = 7.4975$. Donc dans cet exemple, pour la structure de politique factorisée considérée, il existe une politique stochastique meilleure que toute politique déterministe. Pour comprendre que cette politique stochastique est meilleure que toute politique déterministe, il suffit de constater que c'est S_1 qui intervient dans la récompense et que la probabilité que $S_1 = 1$ évolue au cours du temps, amenant à des meilleurs choix pour l'action, basés uniquement sur S_2, différents au cours du temps.
A noter que la politique globale optimale (déterministe), est en fait structurée selon S_1 :

$$\delta_{opt}(a_1|s_1) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_1 \\ \left(\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right) \end{array}$$

Sa valeur est de : $V_{\delta_{opt}}^{R,\infty} = 8.9855$. Choisir une structure de politique selon la règle 'naturelle' (voir section 2.1.4) aurait conduit à choisir $pa_\delta(A_1) = \{S_1, S_2\}$. La règle fonctionne donc dans cet exemple mais est trop conservatrice.

2.2 Évaluation des PSFs dans les PDMF^3

Nous nous intéressons ici à l'évaluation des politiques stochastiques factorisées (PSF) dans les PDMF^3. Une évaluation exacte n'étant pas possible quand n et m grandissent, nous nous intéresserons tout d'abord à une évaluation par simulations de Monte-Carlo (section 2.2.2). Puis, cette évaluation étant trop coûteuse en temps de calcul sur les problèmes de grande taille pour être intégrée à un algorithme d'optimisation, nous proposerons une méthode approchée plus rapide dans la section 2.2.3.

2.2.1 Définition de la valeur d'une PSF dans un PDMF^3

Pour un PDMF^3, défini par le tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, P, pa_\delta, R, P^0)$, nous appelons valeur d'une politique stochastique factorisée δ l'espérance de la somme des récompenses obtenues en suivant δ si la distribution initiale sur les états est P^0 :

$$V_\delta^{R,T}(P^0) = \mathbb{E}_{P_\delta^T}\left[\sum_{t=0}^{T} \gamma^t R(S^t, A^t) \Big| P^0, \delta\right] \tag{2.1}$$

Cette définition englobe à la fois le cas de l'horizon fini (il suffit de prendre $\gamma = 1$ et T égal à l'horizon du problème) et le cas de l'horizon infini avec facteur d'amortissement $0 < \gamma < 1$ (nous proposons alors en pratique de choisir un horizon T 'suffisamment grand' pour approcher l'horizon infini, voir section 2.2.4).

L'espérance est prise par rapport à la distribution P_δ^T sur les historiques à date T du processus. Sous la politique δ et la distribution initiale P^0, la probabilité de l'historique $(s,a)^{0:T} =< s^0, a^0, ..., s^T, a^T >$ est donnée par :

$$P_\delta^T((s,a)^{0:T}) = P^0(s^0) \times \prod_{t=0}^{T-1} \left(\prod_{i=1}^{n} P_i(s_i^{t+1} | pa_P(s_i^{t+1})) \prod_{j=1}^{m} \delta_j(a_j^t | pa_\delta(a_j^t)) \right) \times \prod_{j=1}^{m} \delta_j(a_j^T | pa_\delta(a_j^T))$$

Cette distribution est celle du réseau bayésien dynamique représentant les variables d'état et d'action, comme celui de la figure 2.1 (schéma de gauche).

2.2.2 Évaluation par la méthode de Monte-Carlo

La valeur étant une espérance, on peut naturellement la calculer par la méthode de Monte-Carlo. Il s'agit tout simplement de faire la moyenne empirique de la somme pondérée de récompenses obtenues pour un grand nombre nsim d'historiques $(s,a)^{0:T}$, simulés selon P_δ^T. On notera $\hat{V}_{\delta,\mathrm{MC}}^{R,T}(P^0)$ la valeur approchée obtenue par la méthode de Monte-Carlo.

D'après [OKV08] étendu au cas de l'horizon fini avec facteur d'amortissement (qui utilise un résultat théorique de [Hoe63]), pour que $\mathbb{P}(|\hat{V}_{\delta,\mathrm{MC}}^{R,T}(P^0) - V_\delta^{R,T}(P^0)| < \epsilon) \geq \kappa$ il faut que :

$$\mathrm{nsim} \geq \frac{(1-\gamma^{T+1})^2 (R_{max} - R_{min})^2}{2\epsilon^2 (1-\gamma)^2} \ln \frac{2}{1-\kappa} \tag{2.2}$$

où $R_{max} = \max_{s,a} R(s,a)$, $R_{min} = \min_{s,a} R(s,a)$.

Le nombre de simulations nécessaire pour obtenir une erreur absolue plus petite qu'ϵ avec probabilité κ est donc indépendant de la taille du problème. Ce résultat suppose un horizon fini T et ne prend donc pas en compte l'erreur de troncature dans le cas d'un horizon infini.

L'algorithme 5 donne les détails de la mise en œuvre de l'évaluation d'une politique stochastique factorisée par la méthode de Monte-Carlo. Lorsqu'il y a des arcs synchrones, la simulation des variables d'état et d'action doit se faire dans un certain ordre : les valeurs des variables parentes d'une certaine variable doivent être simulées avant celle-

ci.

Data: $\mathcal{S}, \mathcal{A}, \mathcal{T}, P, pa_\delta, R, P^0, \delta, \gamma$
Result: $\hat{V}^{R,T}_{\delta,\mathrm{MC}}(P^0)$
for $simu \leftarrow 1$ **to** $nsim$ **do**
  simuler les $s^0_i, i = 1...n$ selon P^0;
  simuler les $a^0_j, j = 1...m$ selon les $\delta_j(.|pa_\delta(a^0_j))$;
  for $t \leftarrow 0$ **to** T **do**
    Vcur $\leftarrow R(s^t, a^t)$;
    V(simu)$\leftarrow$V(simu)$+\gamma^t$Vcur;
    simuler les $s^t_i, i = 1...n$ selon les $P_i(.|pa_P(s^t_i))$;
    simuler les $a^t_j, j = 1...m$ selon les $\delta_j(.|pa_\delta(a^t_j))$;
  end
end
$\hat{V}^{R,T}_{\delta,\mathrm{MC}}(P^0) \leftarrow$mean(V);

Algorithm 5: Algorithme d'évaluation pour PDMF[3] basé sur la méthode de Monte-Carlo

Pour être accéléré, le code de l'évaluation Monte-Carlo peut être parallélisé (parallélisation de la simulation des différents historiques). Cependant, cela ne suffit pas à rendre le calcul suffisamment rapide sur des problèmes de grande taille pour rendre cette évaluation utilisable au sein d'un algorithme d'optimisation. C'est pourquoi nous proposons une autre méthode approchée, plus rapide, dans la section suivante.

2.2.3 Évaluation basée sur le calcul de marginales dans un modèle graphique

La valeur d'une politique stochastique factorisée δ pour la distribution initiale P^0 peut s'obtenir en calculant certaines marginales du modèle graphique associé à la distribution P^T_δ :

$$\begin{aligned} V^{R,T}_\delta(P^0) &= \mathbb{E}_{P^T_\delta}\left[\sum_{t=0}^{T}\gamma^t R(S^t,A^t)\Big|P^0,\delta\right] = \mathbb{E}_{P^T_\delta}\left[\sum_{t=0}^{T}\gamma^t\sum_{\alpha=1}^{r}R_\alpha(pa_R(R^t_\alpha))\Big|P^0,\delta\right] \\ &= \sum_{t=0}^{T}\gamma^t\sum_{\alpha=1}^{r}\mathbb{E}_{P^T_\delta}\left[R_\alpha(pa_R(R^t_\alpha))\Big|P^0,\delta\right] \\ &= \sum_{t=0}^{T}\gamma^t\sum_{\alpha=1}^{r}\sum_{pa_R(R_\alpha)} b^t_\alpha(pa_R(R_\alpha))R_\alpha(pa_R(R_\alpha)) \end{aligned}$$

où

$$b^t_\alpha(pa_R(R_\alpha)) = \sum_{(s,a)^{0:T}\setminus(pa_R(R^t_\alpha))} P^T_\delta((s,a)^{0:T})$$

est la distribution marginale des variables influençant la fonction de récompense α au temps t. Elle se calcule en marginalisant $P^T_\delta((s,a)^{0:T})$ par rapport à toutes les autres

variables d'état et d'action dans la trajectoire $(s,a)^{0:T}$. Remarquons que la valeur s'écrit comme une somme de fonctions de faible arité de portées $\{pa_R(R_\alpha^t)\}_{\alpha\in\{1\dots r\},t\in\{0,\dots,T\}}$. Elle fait intervenir toutes les variables d'état et d'action, des pas de temps 0 à T, et non uniquement les variables décrivant s^0 comme dans d'autres travaux ([SPF12] ou [KZT11] par exemple). Cette décomposition est exacte et découle directement des hypothèses du cadre.

Quand le problème est de taille suffisamment petite, l'algorithme *Junction Tree* (voir par exemple [YFW05]) peut être utilisé pour calculer de manière exacte ces marginales, et obtenir ainsi une évaluation exacte de δ. Mais lorsque le problème est de grande taille, ces marginales doivent être calculées de manière approchée. Différents algorithmes de type propagation de message, dérivés de l'algorithme *Junction Tree*, ont été proposés qui sont rapides et retournent de bonnes approximations en pratique (même si les résultats théoriques sur ces méthodes sont rares). Ces méthodes ne sont pas sans lien avec les méthodes variationnelles [YFW05], dont le principe est d'approcher une distribution de probabilité complexe par son meilleur représentant dans une famille plus simple.

Dans cette thèse, nous utiliserons l'algorithme *Loopy Belief Propagation* [KFL01], qui est détaillé en annexe section A.2. Mais il existe d'autres algorithmes de propagation de messages, comme l'algorithme *Generalized Belief Propagation* [YFW05] ou l'algorithme *Tree-reweighted Belief Propagation* [WJW03]. L'algorithme *Factored Frontier* [MW01] a été développé spécifiquement pour les réseaux bayésiens dynamiques, mais *Loopy Belief Propagation* reste plus précis (*Factored Frontier* est équivalent à une seule itération de LBP).

Dans l'algorithme de résolution d'un PDMG MF-API [SPF12], une approximation en champ moyen des marginales du modèle graphique est utilisée pour l'évaluation des politiques. Cette approximation est la plus naïve parmi les méthodes variationnelles.

On remarque que :

$$V_\delta^{R,T}(P^0) = \sum_{\alpha=1}^{r} \sum_{pa_R(R_\alpha)} R_\alpha(pa_R(R_\alpha)) \sum_{t=0}^{T} \gamma^t b_\alpha^t(pa_R(R_\alpha))$$

C'est à partir de cette dernière expression que l'on peut calculer la valeur d'une politique stochastique factorisée donnée de manière la plus efficace, selon l'algorithme 6. La figure 2.3 représente sous forme de *factor graph* (modèle graphique générique, voir annexe section A.1.2 et [KFL01]) la distribution $P_\delta^T((s,a)^{0:T})$ du PDMF^3 de la figure 2.1. Lorsque l'évaluation est utilisée dans un algorithme d'optimisation, où on évalue successivement plusieurs politiques, le *factor graph* de la figure 2.3 n'est pas à reconstruire à chaque fois,

il suffit de mettre à jour les tables associées à la politique.

Data: $\mathcal{S}, \mathcal{A}, \mathcal{T}, P, pa_\delta, R, P^0, \delta, \gamma$
Result: V
construction du *factor graph* d'horizon T (version générale ou version PDMG);
calcul des tables marginales b^t_α;
$V \leftarrow 0$;
for $\alpha \leftarrow 1$ **to** r **do**
    $V2 \leftarrow zeros$;
    for $t \leftarrow 0$ **to** T **do**
        $V2 \leftarrow V2 + \gamma^t b^t_\alpha$;
    end
    $V \leftarrow V + \sum V2. * R_\alpha$;
end

Algorithm 6: Algorithme d'évaluation approchée pour PDMF^3 basé sur le calcul de marginales

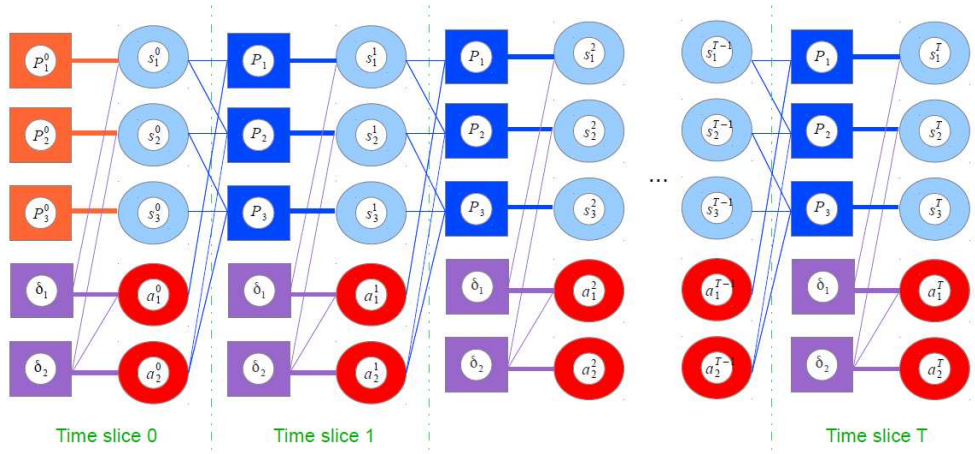


FIGURE 2.3 – Exemple de *factor graph* représentant la distribution $P^T_\delta((s,a)^{0:T})$ du PDMF^3 de la figure 2.1 ; dans un *factor graph* les variables sont représentées par des cercles et les facteurs par des carrés

2.2.4 Approximation de l'horizon infini par un horizon fini

Pour choisir un horizon fini pour approcher le cas d'un problème à horizon infini, deux critères sont possibles : basé sur l'erreur absolue (théorique) ou sur l'erreur relative (théorique).

Critère basé sur l'erreur absolue

Soit :

$$C(t) = \sum_{\alpha=1}^{r} \mathbb{E}_{P_\delta^T}\left[R_\alpha(pa_R(R_\alpha^t))\middle| P^0, \delta\right] = \sum_{\alpha=1}^{r} C_\alpha(t)$$

où

$$C_\alpha(t) = \mathbb{E}_{P_\delta^T}\left[R_\alpha(pa_R(R_\alpha^t))\middle| P^0, \delta\right]$$

En supposant que $C(t)$ est calculé de manière exacte, l'erreur absolue de troncature, commise en approchant l'horizon infini par un horizon fini T, notée EA, vérifie :

$$\begin{aligned}
0 \quad &\leq \quad EA = |V_\delta^{R,\infty}(P^0) - V_\delta^{R,T}(P^0)| = \left|\sum_{t=0}^{+\infty}\gamma^t C(t) - \sum_{t=0}^{T}\gamma^t C(t)\right| = \left|\sum_{t=T+1}^{+\infty}\gamma^t C(t)\right| \\
&= \quad \sum_{t=T+1}^{+\infty}\gamma^t \sum_{\alpha=1}^{r} C_\alpha(t) = \sum_{t=0}^{+\infty}\gamma^{t+T+1}\sum_{\alpha=1}^{r} C_\alpha(t+T+1) = \gamma^{T+1}\sum_{t=0}^{+\infty}\gamma^t\sum_{\alpha=1}^{r} C_\alpha(t+T+1) \\
&\leq \quad \gamma^{T+1}\sum_{t=0}^{+\infty}\gamma^t R_{\max} = \frac{\gamma^{T+1}}{1-\gamma}R_{\max} \text{ car } |\gamma| < 1
\end{aligned}$$

Si on prend comme critère d'arrêt $\phi(T) = \frac{\gamma^{T+1}}{1-\gamma}R_{\max} \leq \epsilon$, on a donc la garantie que l'erreur absolue (théorique) est bornée par ϵ. Avec ce critère d'arrêt, T peut s'exprimer en fonction de γ, ϵ et $R_{\max}$:

$$T = \frac{\log\left(\frac{(1-\gamma)\epsilon}{R_{\max}}\right)}{\log\gamma} - 1 \tag{2.3}$$

Lorsque le calcul de $C(t)$ est approché (pour les problèmes de grande taille), cette borne sur l'erreur n'est pas vérifiée en général. Mais dans le cas d'une évaluation par la méthode de Monte-Carlo (voir section 2.2.2), on peut obtenir simplement une borne sur l'erreur absolue. Supposons que T vérifie l'équation (2.3) et que le nombre de simulations vérifie l'équation (2.2). On a alors :

$$\begin{aligned}
& \mathbb{P}(|\hat{V}_{\delta,\mathrm{MC}}^{R,T}(P^0) - V_\delta^{R,T}(P^0)| < \epsilon) \geq \kappa \\
\Leftrightarrow \quad & \mathbb{P}(|\hat{V}_{\delta,\mathrm{MC}}^{R,T}(P^0) - V_\delta^{R,T}(P^0)| + |V_\delta^{R,\infty}(P^0) - V_\delta^{R,T}(P^0)| < 2\epsilon) \geq \kappa \\
\Leftrightarrow \quad & \mathbb{P}(|\hat{V}_{\delta,\mathrm{MC}}^{R,T}(P^0) - V_\delta^{R,\infty}(P^0)| < 2\epsilon) \geq \kappa
\end{aligned}$$

par l'inégalité triangulaire. Avec ces deux hypothèses, l'erreur absolue prenant en compte l'approximation par la méthode de Monte-Carlo et la troncature de l'horizon est donc bornée par 2ϵ avec probabilité κ.

Critère basé sur l'erreur relative

Pour ce qui est de l'erreur relative ER, en supposant que $C(t)$ est calculée de manière exacte, on a :

$$ER = \frac{|V_\delta^{R,\infty}(P^0) - V_\delta^{R,T}(P^0)|}{V_\delta^{R,\infty}(P^0)} \leq \frac{\phi(T)}{V_\delta^{R,\infty}(P^0)} \leq \frac{\phi(T)}{V_\delta^{R,T}(P^0)}$$

Un critère permettant de borner l'erreur relative (théorique) par ϵ consisterait à s'arrêter lorsque l'horizon T vérifie $\frac{\phi(T)}{V_\delta^{R,T}(P^0)} \leq \epsilon$. L'avantage de l'erreur relative est qu'elle est insensible à une transformation affine de la récompense. Par contre, avec ce critère l'horizon ne peut pas être déterminé à l'avance. Ce critère ne peut donc pas être utilisé dans l'algorithme 6 car les marginales à calculer sont celles du modèle graphique pour un horizon T connu. Il pourrait par contre être utilisé dans l'algorithme 5 car la longueur des trajectoires simulées peut ne pas être définie au départ.

2.2.5 Bilan

Pour évaluer une politique stochastique factorisée dans un PDMF^3, lorsque le problème est de petite taille et d'horizon fini, une évaluation exacte reposant sur un calcul de marginales dans un modèle graphique est possible par exemple en utilisant l'algorithme *Junction Tree* et la méthode décrite dans la section 2.2.3. Lorsque le problème est de grande taille, une évaluation approchée est possible en utilisant un algorithme d'inférence approchée dans les modèles graphiques, comme *Loopy Belief Propagation*. Pour une évaluation plus précise, mais moins rapide, dans les problèmes de grande taille, on pourra utiliser la méthode de Monte-Carlo décrite dans la section 2.2.2 (notamment pour comparer des politiques issues de différents algorithmes d'optimisation). Par contre, cette dernière méthode est trop coûteuse en temps pour des appels multiples au sein d'un algorithme d'optimisation de la politique.

Les deux méthodes peuvent permettre d'évaluer aussi des politiques factorisées stochastiques non stationnaires (la seule différence est que les tables associées à la politique, dans le *factor graph* représentant $P_\delta^T((s,a)^{0:T})$, sont différentes à chaque pas de temps au lieu d'être identiques).

2.3 Optimisation des PSF dans les PDMF^3

2.3.1 Formulation du problème d'optimisation et complexité

Si maintenant nous nous intéressons à calculer, pour une structure de politique donnée, une politique stochastique factorisée optimale, nous sommes face à un problème d'optimisation continue sous contraintes. Les variables d'optimisation de ce problème sont les valeurs permettant de remplir les tables de probabilité $\delta_j(a_j|pa_\delta(a_j))$. Il y a en tout

$$N = \sum_{j=1}^{m} |\mathcal{A}_j| \prod_{k/A_k \in pa_\delta^A(A_j)} |\mathcal{A}_k| \prod_{k'/S_{k'} \in pa_\delta^S(A_j)} |\mathcal{S}_{k'}|$$

valeurs. On notera $\bar{\delta} = (\bar{\delta}_1, ..., \bar{\delta}_N)$ le vecteur listant ces N valeurs dans un ordre arbitraire.

Le problème d'optimisation est le suivant :

Problème 1

$$\begin{array}{ll} \max\limits_{\bar{\delta}\in(\mathbb{R}^+)^N} & V_{\bar{\delta}}^{R,T}(P^0) \\ \text{s.c} & \sum_{a_j\in\mathcal{A}_j}\delta_j(a_j|pa_\delta(a_j)) = 1\ \forall j, \forall pa_\delta(a_j) \end{array}$$

Dans ce problème, il y a donc N variables d'optimisation/paramètres et

$$c = \sum_{j=1}^{m} \prod_{k/A_k\in pa_\delta^A(A_j)} |\mathcal{A}_k| \prod_{k'/S_{k'}\in pa_\delta^S(A_j)} |\mathcal{S}_{k'}|$$

contraintes. Pour simplifier, quand $\forall j = 1...m, |\mathcal{A}_j| = |\mathcal{A}_1|, |pa_\delta^A(a_j)| = |pa_\delta^A(a_1)| = z$ et $\forall i = 1...n, |\mathcal{S}_i| = |\mathcal{S}_1|, |pa_\delta^S(a_j)| = |pa_\delta^S(a_1)| = y$, il y a

$$N = m|\mathcal{A}_1|^{z+1}|\mathcal{S}_1|^y$$

paramètres, ce qui est beaucoup plus petit que le nombre de paramètres d'une politique globale déterministe ($m|\mathcal{S}| = m|\mathcal{S}_1|^n$) quand $z << m$ et $y << n$.

Théorème 9. *La formulation en problème de décision du problème 1 est un problème NP^{PP}-complet.*

Preuve : voir annexe B

2.3.2 Analyse du problème d'optimisation

Parmi les problèmes d'optimisation continue, les problèmes de minimisation convexes (dont l'objectif et le domaine sont convexes) présentent des propriétés intéressantes : dans ces problèmes tout optimum local est global [BV04]. Nous avons donc essayé de reformuler le problème 1 comme un problème de minimisation convexe, mais sans succès. Néanmoins, la démarche que nous avons menée nous semble suffisamment intéressante pour être décrite. Nous nous sommes appuyés sur les notions de posynôme et de programme géométrique (voir [BV04, BKVH07]). Un posynôme n'est autre qu'un polynôme à coefficients strictement positifs, et un programme géométrique est un problème de minimisation d'un posynôme qui présente certaines propriétés permettant sa transformation en un programme convexe équivalent :

Définition 21. *Un programme géométrique est un problème d'optimisation de la forme*

$$\begin{array}{ll} \underset{x\in\mathcal{D}}{min} & f_0(x) \\ s.c & f_i(x) \leq 1,\ i = 1,\dots,m \\ & g_j(x) = 1, j = 1,\dots p \end{array}$$

où $\mathcal{D} = (R^{+*})^n$, les $f_i, i = 0...m$ sont des posynômes, et les $g_j, j = 1...p$ des monômes (à coefficients strictement positifs).

Propriété 3. *Si f est un posynôme, la fonction $F(y) = \log f(e^y)$ est convexe.*

Propriété 4. *Tout programme géométrique est transformable en un programme convexe équivalent, où on fait le changement de variable* $\forall i, y_i = \log(x_i)$ *ainsi qu'une transformation en log de l'objectif et des contraintes :*

$$\begin{array}{ll} \underset{y}{min} & \log f_0(e^y) \\ s.c & \log f_i(e^y) \leq 0,\ i = 1, \ldots, m \\ & \log g_j(e^y) = 0, j = 1, \ldots p \end{array}$$

Remarquons que $V_{\bar{\delta}}^{R,T}(P^0)$ est un posynôme en $\bar{\delta} = (\bar{\delta}_1, ..., \bar{\delta}_N)$. D'après les propriétés ci-dessus, le problème 1 est donc transformable en un problème de maximisation dont la fonction objectif est convexe mais le domaine ne l'est pas.

On peut relâcher les contraintes d'égalité du problème de départ (problème 1). En effet, le critère étant croissant en les paramètres, le maximum est atteint sur la frontière du domaine. On obtient alors le problème 2, équivalent au problème 1 :

Problème 2

$$\begin{array}{ll} \underset{\bar{\delta} \in (\mathbb{R}^+)^N}{\max} & V_{\bar{\delta}}^{R,T}(P^0) \\ \text{s.c} & \sum_{a_j \in \mathcal{A}_j} \delta_j(a_j | pa_\delta(a_j)) \leq 1\ \forall j, \forall pa_\delta(a_j) \end{array}$$

Par changement de variable en log et transformation en log de l'objectif et des contraintes, ce problème est équivalent à un problème de maximisation d'une fonction convexe sur un domaine convexe. Mais ces problèmes ne sont pas faciles à résoudre (voir par exemple [EBK06]), contrairement aux problèmes de minimisation d'une fonction convexe.

Dans le problème 1, on peut se ramener à un problème de minimisation tout en gardant un posynôme grâce à un changement des fonctions de récompense. L'idée est de considérer les récompenses transformées suivantes :

$$\bar{R}_\alpha(pa_R(R_\alpha)) = \left(\max_{pa_R(R_\alpha)} R_\alpha(pa_R(R_\alpha)) \right) - R_\alpha(pa_R(R_\alpha))$$

On a alors

$$\mathbb{E}_{P_\delta^T}\left[\sum_{\alpha=1}^{r} \bar{R}_\alpha(pa_R(R_\alpha)) \right] = \sum_{\alpha=1}^{r} \max_{pa_R(R_\alpha)} R_\alpha(pa_R(R_\alpha)) - \mathbb{E}_{P_\delta^T}\left[\sum_{\alpha=1}^{r} R_\alpha(pa_R(R_\alpha)) \right]$$

d'où

$$\begin{aligned} V_\delta^{\bar{R},T}(P^0) &= \sum_{t=0}^{T} \gamma^t \mathbb{E}_{P_\delta^T}\left[\sum_{\alpha=1}^{r} \bar{R}_\alpha(pa_R(R_\alpha^t)) \right] \\ &= \left(\sum_{t=0}^{T} \gamma^t \sum_{\alpha=1}^{r} \max_{pa_R(R_\alpha)} R_\alpha(pa_R(R_\alpha)) \right) - V_\delta^{R,T}(P^0) \\ &= \frac{1}{1-\gamma} \left(\sum_{\alpha=1}^{r} \max_{pa_R(R_\alpha)} R_\alpha(pa_R(R_\alpha)) \right) - V_\delta^{R,T}(P^0) = B - V_\delta^{R,T}(P^0) \end{aligned}$$

où $B = \frac{1}{1-\gamma}\sum_{\alpha=1}^{r} \max_{pa_R(R_\alpha)} R_\alpha(pa_R(R_\alpha)) > 0$, donc $\sup_{\bar{\delta}} V_{\bar{\delta}}^{R,T}(P^0) = B - \inf_{\bar{\delta}} V_{\bar{\delta}}^{\bar{R},T}(P^0)$.

Le problème 1 est donc équivalent au problème 3 de minimisation :

Problème 3

$$\begin{array}{ll} \min\limits_{\bar{\delta}\in(\mathbb{R}^+)^N} & V_{\bar{\delta}}^{\bar{R},T}(P^0) \\ \text{s.c} & \sum_{a_j\in\mathcal{A}_j} \delta_j(a_j|pa_\delta(a_j)) = 1 \ \forall j, \forall pa_\delta(a_j) \end{array}$$

$V_{\bar{\delta}}^{\bar{R},T}$ est bien un posynôme en $\bar{\delta}$. Le problème 3 pourrait donc se transformer en un problème de minimisation dont la fonction objectif serait convexe, mais le domaine ne le serait pas.

On peut ensuite limiter la taille de l'espace de recherche en utilisant des contraintes d'inégalité au lieu de contraintes d'égalité, en remarquant que :

$$\forall j, \forall pa_\delta(a_j),\ \delta_j(|\mathcal{A}_j||pa_\delta(a_j)) = 1 - \sum_{a_j\in\{1,\ldots,|\mathcal{A}_j|-1\}} \delta_j(a_j|pa_\delta(a_j))$$

Soit $\tilde{\delta} = (\tilde{\delta}_1, ..., \tilde{\delta}_{N'})$ le vecteur de coordonnées l'ensemble de valeurs $\{\delta_j(a_j|pa_\delta(a_j)), \forall j, \forall a_j \neq |A_j|, \forall pa_\delta(a_j)\}$ listées dans un ordre arbitraire, où

$$N' = \sum_{j=1}^{m}(|\mathcal{A}_j| - 1) \prod_{k/A_k\in pa_\delta^A(A_j)} |\mathcal{A}_k| \prod_{k'/S_{k'}\in pa_\delta^S(A_j)} |\mathcal{S}_{k'}| = N - c.$$

Le problème 3 est équivalent au problème 4 suivant :

Problème 4

$$\begin{array}{ll} \min\limits_{\tilde{\delta}\in(\mathbb{R}^+)^{N'}} & V_{\tilde{\delta}}^{\bar{R},T}(P^0) \\ \text{s.c} & \sum_{a_j\in\{1,\ldots,|\mathcal{A}_j|-1\}} \delta_j(a_j|pa_\delta(a_j)) \leq 1 \ \forall j, \forall pa_\delta(a_j) \end{array}$$

$V_{\tilde{\delta}}^{\bar{R},T}(P^0)$ n'est plus un posynôme en $\tilde{\delta}$. On ne peut donc pas transformer le programme 4 en un programme convexe. Mais cette reformulation est particulièrement intéressante car elle permet de réduire le nombre de variables d'optimisation. Par exemple, lorsque les variables d'action sont toutes binaires ($\forall j = 1...m, |\mathcal{A}_j| = 2$), on a $N' = N/2$.

Une dernière formulation possible peut s'obtenir en reparamétrant le problème 3 (ou le problème 1) de manière à ne plus avoir de contraintes. La transformation, souvent utilisée dans les méthodes de résolution de PDM dites *de gradient* (voir par exemple [PKMK00]), est la suivante : à chaque élément $\delta_j(a_j|pa_\delta(a_j))$ on associe une quantité réelle $\theta_j(a_j|pa_\delta(a_j))$ telle que :

$$\delta_j(a_j|pa_\delta(a_j)) = \frac{e^{\theta_j(a_j|pa_\delta(a_j))}}{\sum_{a'_j\in\mathcal{A}_j} e^{\theta_j(a'_j|pa_\delta(a_j))}}$$

Soit $\bar{\theta} = (\bar{\theta}_1, ..., \bar{\theta}_N)$ le vecteur de coordonnées $\{\theta_j(a_j|pa_\delta(a_j)), \forall j, \forall a_j \in A_j, \forall pa_\delta(a_j)\}$. Le problème 3 est équivalent au problème suivant :

Problème 5

$$\min_{\bar{\theta} \in (\mathbb{R})^N} \quad V_{\bar{\theta}}^{\bar{R},T}(P^0)$$

Si les minima globaux coïncident, les minima locaux de $V_{\bar{\theta}}^{\bar{R},T}(P^0)$ et $V_{\bar{\delta}}^{\bar{R},T}(P^0)$ ne coïncident pas forcément, et ne sont pas forcément aussi nombreux. Enfin, $V_{\bar{\theta}}^{\bar{R},T}(P^0)$ n'est pas un posynôme en $\bar{\theta}$, et ne semble pas convexe en $\bar{\theta}$.

Contrainte d'égalité sur certaines politiques locales

Nous envisageons ici le cas où il y a des contraintes d'égalité sur certaines politiques locales. Par exemple, dans le cas d'une grille, on peut obliger à ce que les politiques en tous les sites de la grille soient identiques (sauf sur les bords). Ou encore on peut avoir deux types d'acteurs, et ne calculer que deux types de politiques locales, une pour chaque type d'acteur. Plus formellement, on se donne $(Q_1, ..., Q_L)$ une partition de $\{1, 2, ..., m\}$ (l'ensemble des indices des variables d'action) :

$$\bigcup_{i=1}^{L} Q_i = \{1, 2, ..., m\} \text{ et } \forall (l, l') \in \{1, ..., L\}^2,\ Q_l \cap Q_{l'} = \emptyset$$

La partition $(Q_1, ..., Q_L)$ doit être définie de manière à ce que :

$$\forall l \in \{1, ..., L\}, \forall (j, j') \in (Q_l)^2 \mathcal{A}_j = \mathcal{A}_{j'}$$

De plus, les variables parentes de A_j et de $A_{j'}$ pour la structure de politique, si elles ne sont pas identiques, doivent être en même nombre et de mêmes domaines : $\forall l \in \{1, ..., L\}, \forall (j, j') \in (Q_l)^2$, si $pa_\delta(A_j) = \{S_{i_1}, ..., S_{i_{n_j}}, ..., A_{k_1}, ..., A_{k_{N_j}}\}$ alors

$$\begin{aligned} \exists (i'_1, ...i'_{n_j}, ..., k'_1, ..., k'_{N_j}) \qquad & pa_\delta(A_{j'}) = \{S_{i'_1}, ..., S_{i'_{n_j}}, ..., A_{k'_1}, ..., A_{k'_{N_j}}\} \\ & \forall v = 1...n_j, \mathcal{S}_{i_v} = \mathcal{S}_{i'_v} \\ & \forall w = 1...N_j, \mathcal{A}_{i_w} = \mathcal{A}_{i'_w} \end{aligned}$$

Sous ces conditions, on souhaite optimiser $V_{\delta}^{\bar{R},T}(P^0)$ sous les contraintes suivantes :

$$\forall l \in \{1, ..., L\}, \forall (j, j') \in (Q_l)^2, \forall a_j \in \mathcal{A}_j, \forall pa_\delta(a_j),\ \delta_j(a_j|pa_\delta(a_j)) = \delta_{j'}(a_j|pa_\delta(a_j))$$

Soient $q_1, ..., q_L$ tels que :

$$\forall l \in \{1, ..., L\},\ q_l \in Q_l$$

Le nombre de paramètres N'' à optimiser sera inférieur à N (si $L < m$) :

$$N'' = \sum_{l=1}^{L} |\mathcal{A}_{q_l}| \prod_{k/A_k \in pa_\delta^A(A_{q_l})} |\mathcal{A}_k| \prod_{k'/S_{k'} \in pa_\delta^S(A_{q_l})} |\mathcal{S}_{k'}|$$

On notera $\hat{\delta} = (\hat{\delta}_1, ..., \hat{\delta}_{N''})$ le vecteur de coordonnées $\{\delta_{q_1}(a_{q_1}|pa_\delta(a_{q_1})), ..., \delta_{q_L}(a_{q_L}|pa_\delta(a_{q_L}))\}$, et $\hat{\theta} = (\hat{\theta}_1, ..., \hat{\theta}_{N''})$ le vecteur associé avec la transformation habituelle. Le problème d'optimisation 5 devient par exemple :

Problème 5 bis

$$\underset{\hat{\theta} \in \mathbb{R}^{N''}}{\text{minimize}} \quad V_{\hat{\theta}}^{\bar{R},T}(P^0)$$

Dans le cas du problème 4 où le nombre de variables d'optimisation sans contraintes d'égalité est de $N' = \sum_{j=1}^{m}(|\mathcal{A}_j| - 1)\prod_{k/A_k \in pa_\delta^A(A_j)}|\mathcal{A}_k|\prod_{k'/S_{k'} \in pa_\delta^S(A_j)}|\mathcal{S}_{k'}|$, il devient de :

$$N''' = \sum_{l=1}^{L}(|\mathcal{A}_{q_l}| - 1) \prod_{k/A_k \in pa_\delta^A(A_{q_l})} |\mathcal{A}_k| \prod_{k'/S_{k'} \in pa_\delta^S(A_{q_l})} |\mathcal{S}_{k'}|$$

si l'on prend en compte les contraintes d'égalité. Le problème 4 devient donc,
en notant $\hat{\hat{\delta}} = (\hat{\hat{\delta}}_1, ..., \hat{\hat{\delta}}_{N'''})$ le vecteur de coordonnées
$\{\delta_{q_1}(a_{q_1}|pa_\delta(a_{q_1})), ..., \delta_{q_L}(a_{q_L}|pa_\delta(a_{q_L}))/\forall l = 1...L, a_{q_l} \neq |\mathcal{A}_{q_l}|\}$:

Problème 4 bis

$$\begin{array}{ll} \min\limits_{\hat{\hat{\delta}} \in (\mathbb{R}^+)^{N'''}} & V_{\hat{\hat{\delta}}}^{\bar{R},T}(P^0) \\ \text{s.c} & \sum_{a_{q_l} \in \{1,...,|\mathcal{A}_{q_l}|-1\}} \delta_{q_l}(a_{q_l}|pa_\delta(a_{q_l})) \leq 1 \ \forall l = 1...L, \forall pa_\delta(a_{q_l}) \end{array}$$

Bilan

Nous n'avons pas pu trouver de reformulation du problème d'optimisation en un programme convexe, ce qui n'est pas complètement étonnant au vu du résultat de complexité sur ce problème (voir théorème 9). En effet, la convexité est un indice de 'facilité' pour un problème d'optimisation, alors que le fait qu'il soit ici NP^{PP}-difficile est une preuve de difficulté.

Utiliser une boîte à outils d'optimisation, lorsque l'évaluation est approchée et relativement longue, ne donne pas de bons résultats. De plus, la plupart des méthodes d'optimisation sans gradient ne sont pas applicables pour des problèmes à plus de 1000 variables d'optimisation [RS13] (nous dépasserons largement ce nombre dans la partie expérimentale). C'est pourquoi nous avons implémenté des algorithmes d'optimisation spécifiquement pour notre problème.

Dans la suite, nous nous intéresserons particulièrement aux formulations 4 et 5 du problème d'optimisation. En effet, le problème 4 présente l'avantage, lorsque les variables d'action sont binaires, de diviser par deux le nombre de variables d'optimisation. Nous pourrons dans ce cas appliquer un algorithme de descente par coordonnées (voir section 2.3.3). Dans le cas général, nous utiliserons le problème 5 qui présente l'avantage de ne pas avoir de contraintes et donc de rendre la mise en œuvre plus facile. Nous proposons pour résoudre cette formulation du problème d'utiliser un algorithme de descente de gradient (voir section 2.3.4).

Les deux méthodes sont des méthodes dites de descente. Celles-ci génèrent une suite de points $(x_q)_{q\geq 0} \in \mathcal{D}$ dans le domaine d'optimisation $\mathcal{D} \subset \mathbb{R}^n$ qui vérifient, si f est la fonction objectif à minimiser :

$$\forall q \geq 0, f(x_{q+1}) \leq f(x_q)$$

Les points de la suite $(x_q)_{q\geq 0}$ sont générés de la manière suivante :

$$\forall q \geq 0,\ x_{q+1} = x_q + s_q d_q$$

où $d_q \in \mathbb{R}^n$ est la direction de descente et $s_q \in \mathbb{R}$ le pas, choisi de manière à faire décroître la fonction objectif. Les différentes méthodes de descente diffèrent par la manière de choisir la direction de descente et le pas. La recherche du pas est appelée recherche linéaire car elle correspond à l'optimisation d'une variable réelle.

2.3.3 Un algorithme d'optimisation pour le cas de variables d'action binaires : la descente par coordonnées

Dans le cas de variables d'action binaires ($\forall j = 1...m, \mathcal{A}_j = \{1, 2\}$), le problème 4 s'écrit :

Problème 4*

$$\min_{\tilde{\delta}\in]0;1]^{N/2}} \quad V_{\tilde{\delta}}^{\bar{R},T}(P^0)$$

Le domaine d'optimisation étant une 'boîte', un algorithme de descente par coordonnées présente la garantie de converger vers un minimum local. Cet algorithme consiste à fixer tour à tour toutes les coordonnées sauf une et à optimiser la coordonnée restante. La direction de descente est donc celle d'un axe de coordonnées. Le pas optimal est le pas qui conduit à la meilleure amélioration de la fonction objectif dans la direction de descente. Nous avons mis en œuvre de deux manières la recherche de ce pas optimal : une recherche par pas fixe s (on avance de s dans la direction de descente jusqu'à ce que la fonction objectif ne soit plus améliorée) et une méthode de recherche dichotomique basée sur le nombre d'or (*golden section search*), implémentée dans la fonction `fminbnd` de Matlab. L'algorithme de descente par coordonnées s'arrête lorsqu'aucun changement n'a plus lieu sur aucune coordonnée de $\tilde{\delta}$.

Dans le cas d'un problème avec contraintes quelconques, cet algorithme n'a pas la garantie de converger vers un minimum local. Pour s'en convaincre, considérons l'exemple simple suivant :

$$\begin{aligned} \min_{(x,y)\in\mathbb{R}^2} \quad & x^2 + y^2 \\ \text{s.c} \quad & x + y \geq 1 \end{aligned}$$

Si le point courant est le point de coordonnées $(\frac{1}{4}, \frac{3}{4})$, on ne peut améliorer la fonction en ne modifiant qu'une seule coordonnée. Or $(\frac{1}{4}, \frac{3}{4})$ n'est pas un point de minimum local. On ne peut donc pas utiliser l'algorithme de descente par coordonnées pour un PDMF[3] en dehors du cas de variables d'action toutes binaires, où il n'y a que des contraintes de 'boîte'.

Dans le cas où l'évaluation de $V_{\tilde{\delta}}^{\bar{R},T}(P^0)$ est approchée, l'algorithme de descente par coordonnées conduit à un minimum local de l'approximation de $V_{\tilde{\delta}}^{\bar{R},T}(P^0)$, notée $\hat{V}_{\tilde{\delta}}^{\bar{R},T}(P^0)$. Il n'y a pas de garantie théorique que ce minimum local coïncide avec un minimum local de $V_{\tilde{\delta}}^{\bar{R},T}(P^0)$.

2.3.4 Un algorithme d'optimisation générique : la descente de gradient

Lorsque les variables d'action ne sont pas binaires, la réduction du nombre de variables d'optimisation dans le problème 4 par rapport au problème 1 est moins intéressante. Nous avons donc choisi, pour le cas général, de proposer un algorithme de descente de gradient sur le problème 5, qui présente l'avantage de ne pas avoir de contraintes et donc de simplifier l'implémentation : **Problème 5**

$$\min_{\bar{\theta}\in(\mathbb{R})^N} \quad V_{\bar{\theta}}^{\bar{R},T}(P^0)$$

De plus, l'avantage de la descente de gradient est que c'est un algorithme qui peut être parallélisé, contrairement à la descente par coordonnées. Sur les problèmes de grande taille que nous envisageons de traiter, cela permettra d'avoir un gain de temps non négligeable. Par contre, la descente de gradient n'a pas la garantie de fournir un minimum local, mais seulement un point critique, c'est-à-dire un point en lequel le gradient s'annule. Il s'agit d'une condition nécessaire mais pas suffisante d'optimalité locale.

La fonction objectif $V_{\bar{\theta}}^{\bar{R},T}(P^0)$ est différentiable par rapport à $\bar{\theta}$. Dans les algorithmes de descente de gradient, la direction de descente est opposée au gradient :

$$d_q = \frac{-\nabla V_{\bar{\theta}_q}^{\bar{R},T}(P^0)}{\|\nabla V_{\bar{\theta}_q}^{\bar{R},T}(P^0)\|}$$

En effet, on peut montrer que c'est la direction de plus forte descente (celle où la pente est la plus forte).

Calcul du gradient

On doit donc calculer le gradient de $V_{\bar{\theta}}^{\bar{R},T}(P^0)$, c'est-à-dire le vecteur de coordonnées $\frac{\partial V_{\bar{\theta}}^{\bar{R},T}(P^0)}{\partial \bar{\theta}_k}, k = 1...N$. On rappelle que :

$$\forall k = 1...N,\ \bar{\delta}_k = \frac{e^{\bar{\theta}_k}}{\sum_{l\in G(k)} e^{\bar{\theta}_l}}$$

$G(k)$ est défini comme suit : si $\bar{\delta}_k$ correspond à $\delta_j(a_j|pa_\delta(a_j))$, alors $g \in G(k)$ si et seulement si $\exists a'_j \neq a_j,\ \bar{\delta}_g$ correspond à $\delta_j(a'_j|pa_\delta(a_j))$. Si $g \in G(k), G(g) = G(k)$.

On peut montrer que chaque dérivée partielle $\frac{\partial V_{\bar{\theta}}^{\bar{R},T}(P^0)}{\partial \theta_k}$ peut s'obtenir en calculant les marginales d'un réseau bayésien dynamique, légèrement modifié par rapport à celui du calcul de $V_{\bar{\delta}}^{\bar{R},T}(P^0)$ (voir section 2.2.3).

Les dérivées partielles peuvent aussi s'obtenir par la méthode des différences finies :

$$\forall k = 1...N,\ \frac{\partial V_{\bar{\theta}}^{\bar{R}}(P^0)}{\partial \bar{\theta}_k} \approx \frac{V_{\bar{\theta}^+}^{\bar{R}}(P^0) - V_{\bar{\theta}}^{\bar{R}}(P^0)}{\epsilon}$$

où $\bar{\theta}_k^+ = \bar{\theta}_k + \epsilon$ et $\forall l = 1...N, l \neq k,\ \bar{\theta}_l^+ = \bar{\theta}_l$. Un pas de ϵ sur la k-ème coordonnée de $\bar{\theta}$ se traduit par une modification de plusieurs coordonnées dans le vecteur décrivant la politique, $\bar{\delta}$. Au vecteur $\bar{\theta}^+$ correspond le vecteur $\bar{\delta}^+$ tel que :

$$\begin{aligned}
\bar{\delta}_k^+ &= \frac{e^{\bar{\theta}_k+\epsilon}}{\sum_{\substack{g\in G(k)\\ g\neq k}} e^{\bar{\theta}_g} + e^{\bar{\theta}_k+\epsilon}} = \frac{e^{\bar{\theta}_k}e^{\epsilon}}{\sum_{\substack{g\in G(k)\\ g\neq k}} e^{\bar{\theta}_g} + e^{\bar{\theta}_k}e^{\epsilon}} \\
&= \frac{e^{\bar{\theta}_k}e^{\epsilon}}{\sum_{g\in G(k)} e^{\bar{\theta}_g} + (e^{\epsilon}-1)e^{\bar{\theta}_k}} = \frac{\bar{\delta}_k \sum_{g\in G(k)} e^{\bar{\theta}_g}e^{\epsilon}}{\sum_{g\in G(k)} e^{\bar{\theta}_g} + (e^{\epsilon}-1)\bar{\delta}_k \sum_{g\in G(k)} e^{\bar{\theta}_g}} \\
&= \frac{\bar{\delta}_k e^{\epsilon}}{1 + (e^{\epsilon}-1)\bar{\delta}_k} \\
\forall g \in G(k), g \neq k,\ \bar{\delta}_g^+ &= \frac{e^{\bar{\theta}_g}}{\sum_{\substack{l\in G(k)\\ l\neq k}} e^{\bar{\theta}_l} + e^{\bar{\theta}_k+\epsilon}} = \frac{e^{\bar{\theta}_g}}{\sum_{l\in G(k)} e^{\bar{\theta}_l} + (e^{\epsilon}-1)e^{\bar{\theta}_k}} \\
&= \frac{e^{\bar{\theta}_g}}{\sum_{l\in G(k)} e^{\bar{\theta}_l} + (e^{\epsilon}-1)\bar{\delta}_k \sum_{l\in G(k)} e^{\bar{\theta}_l}} = \frac{e^{\bar{\theta}_g}}{e^{\bar{\theta}_g}/\bar{\delta}_g + (e^{\epsilon}-1)\bar{\delta}_k e^{\bar{\theta}_g}/\bar{\delta}_g} \\
&= \frac{1}{(1 + (e^{\epsilon}-1)\bar{\delta}_k)/\bar{\delta}_g} = \frac{\bar{\delta}_g}{1 + (e^{\epsilon}-1)\bar{\delta}_k} \\
\forall h \notin G(k),\ \bar{\delta}_h^+ &= \bar{\delta}_h.
\end{aligned}$$

Il est important de remarquer que, dans le cas où l'évaluation de $V_{\bar{\theta}}^{\bar{R},T}(P^0)$ est approchée par $\hat{V}_{\bar{\theta}}^{\bar{R},T}(P^0)$, si l'on utilise la méthode des différences finies on approche le gradient de l'approximation $\hat{V}_{\bar{\theta}}^{\bar{R},T}(P^0)$, alors que si on utilise la méthode du calcul de marginales on approche le gradient de la valeur exacte $V_{\bar{\theta}}^{\bar{R},T}(P^0)$. On ne peut pas savoir à l'avance ce qui donnerait les meilleurs résultats empiriques, mais on peut penser qu'il est plus raisonnable, étant donné que l'algorithme recherche un minimum local de l'approximation $\hat{V}_{\bar{\theta}}^{\bar{R},T}(P^0)$, d'approcher le gradient de cette approximation, en utilisant la méthode des différences finies, qui est également plus simple à implémenter. C'est le choix que nous avons fait dans la partie expérimentale. Dans les deux cas, le calcul du gradient peut être parallélisé, puisque le calcul de chaque dérivée partielle est indépendant.

Si on considérait des politiques factorisées non stationnaires, dans le cas d'une approximation de $V_{\bar{\theta}}^{\bar{R},T}(P^0)$ basée sur l'algorithme LBP (voir section 2.2.3) on pourrait

utiliser l'algorithme *back-belief-propagation* (BBP) présenté dans [EG09] pour estimer le gradient. Cet algorithme permet en effet, dans un *factor graph*, de calculer le gradient d'une fonction des marginales par rapport à un facteur. Avec cet algorithme, on calculerait donc une approximation du gradient de la valeur approchée par LBP. Mais cet algorithme ne peut pas être utilisé dans le cas qui nous intéresse de politiques factorisées stationnaires. En effet, dans ce cas le facteur par rapport auquel on dérive apparaît plusieurs fois dans le *factor graph*.

Recherche linéaire

Pour ce qui est de la recherche linéaire, le pas peut être choisi fixe ou optimal comme dans le cas de la descente par coordonnées. Mais utiliser un pas fixe ou optimal ne conduit pas à un nombre d'itérations optimal. En effet, les itérés peuvent avoir un comportement en zigzag. Un pas trop petit atténue le comportement en zigzag mais augmente le nombre d'itérations nécessaires. Tandis qu'un pas trop grand peut conduire l'algorithme à diverger. Il existe des conditions, appelée conditions de Wolfe, pour vérifier que le pas n'est ni trop grand ni trop petit. La condition pour que le pas ne soit pas trop court est la suivante :

$$\nabla f(x_k + s_k d_k)^T d_k \geq \epsilon_2 (\nabla f(x_k)^T d_k)$$

La condition pour que le pas ne soit pas trop long est la suivante (elle est parfois appelée condition d'Armijo ou de Goldstein) :

$$f(x_k + s_k d_k) \leq f(x_k) + \epsilon_1 s_k (\nabla f(x_k)^T d_k)$$

On doit avoir $0 < \epsilon_1 < \epsilon_2 < 1$. En général on prend $\epsilon_1 = 10^{-4}$ et $\epsilon_2 = 0.99$. Le pseudo-code de la recherche linéaire de Wolfe est donné dans l'algorithme 7.

Pour l'application au problème 5, nous avons implémenté la condition de Wolfe de manière différente afin de ne pas avoir à recalculer un gradient : nous regardons *a posteriori* si le pas de l'itération précédente était trop court.

Critère d'arrêt

Nous avons implémenté l'algorithme de descente de gradient avec trois conditions d'arrêt possibles :

1. nombre maximum d'itérations atteint
2. norme du gradient proche de zéro (condition nécessaire d'optimalité locale) :

$$\| \nabla V_{\bar{\theta}} \| < \epsilon_g$$

   où $V_{\bar{\theta}}$ est une notation abrégée pour $V_{\bar{\theta}}^{\bar{R},T}(P^0)$
3. stagnation de la fonction objectif et du paramètre $\bar{\theta}$:

$$|V_{\bar{\theta}} - V_{\bar{\theta}}^{old}| < \epsilon_V (1 + |V_{\bar{\theta}}^{old}|) \text{ et } \| \bar{\theta} - \bar{\theta}^{old} \| < \epsilon_\theta (1 + \| \bar{\theta}^{old} \|)$$

Data: f,x,d,s_0,ϵ_1,ϵ_2
Result: s^*

k $\leftarrow 0$;
$s_- \leftarrow 0$;
$s_+ \leftarrow +\infty$;
while *non(armijo) ou non(wolfe)* **do**
  if *non(armijo)* **then**
    $s_+ \leftarrow s_k$;
    $s_{k+1} \leftarrow \frac{s_- + s_+}{2}$;
  else
    $s_- \leftarrow s_k$;
    if $s_+ < +\infty$ **then**
      $s_{k+1} \leftarrow \frac{s_- + s_+}{2}$;
    else
      $s_{k+1} \leftarrow 2s_k$;
    end
    $k \leftarrow k + 1$;
  end
  $s^* \leftarrow s_k$
end

Algorithm 7: Recherche linéaire de Wolfe

Les différents paramètres de l'algorithme de descente de gradient que nous proposons sont donc : la méthode de recherche linéaire (pas fixe, pas optimal ou recherche de Wolfe), le nombre maximum d'itérations maxit, le paramètre ϵ de précision de l'estimation par différences finies du gradient, le paramètre ϵ_g de précision sur la norme du gradient, le paramètre ϵ_V de précision sur la valeur, et le paramètre ϵ_θ de précision sur le paramètre θ associé à la politique.

2.3.5 Bilan

Nous avons proposé une méthode générique de résolution pour les PDMF[3], de type itération de la politique approchée, basée sur l'alternance entre :

— une étape d'évaluation : étant donnée une PSF courante stockée dans le vecteur $\bar{\delta}_q$, $V_{\bar{\delta}_q}^{\bar{R},T}$ est évaluée de manière exacte ou approchée, en utilisant par exemple *Junction Tree*, *Loopy Belief Propagation* ou la méthode de Monte-Carlo

— une étape d'amélioration : $\bar{\delta}_q$ est améliorée en $\bar{\delta}_{q+1}$ en utilisant une approche de type descente par coordonnées ou descente de gradient.

Le choix de considérer la récompense transformée $\bar{R}$ et un problème de minimisation est purement arbitraire, on peut aussi garder la récompense de départ R et considérer un problème de maximisation.

L'approximation peut venir de la méthode d'évaluation utilisée lorsqu'elle n'est pas exacte, mais aussi du fait que les algorithmes d'optimisation utilisés sont des algorithmes d'optimisation locale.

Comme nous l'avons détaillé plus haut, les problèmes d'optimisation considérés ne sont pas exactement les mêmes pour la descente par coordonnées et la descente de gradient. La descente par coordonnées n'est applicable que quand le problème est à variables d'action binaires.

Cette méthode est bien générique puisque d'autres méthodes d'évaluation ou d'optimisation peuvent être envisagées. En particulier, pour la méthode d'évaluation basée sur le calcul de marginales, de nouveaux algorithmes d'inférence continuent à être proposés dans les recherches récentes.

Dans la suite, nous nommerons les différentes instanciations de cet algorithme sous la forme Optim-Eval, où Optim est la méthode d'optimisation utilisée (CD pour la descente par coordonnées, GD pour la descente de gradient) et Eval la méthode d'évaluation utilisée (LBP pour *Loopy Belief Propagation*, JT pour *Junction Tree* et MC pour la méthode de Monte-Carlo). Ainsi, par exemple, CD-LBP est l'acronyme pour l'algorithme de descente par coordonnées utilisant l'algorithme *Loopy Belief Propagation* pour l'évaluation approchée.

2.4 Evaluation expérimentale des algorithmes proposés

Nous avons mis en œuvre un certain nombre d'expériences numériques afin de comprendre le comportement des algorithmes proposés. Nous nous sommes placés dans le cas de l'horizon infini ($\gamma = 0.9$), qui est le cas le plus défavorable pour notre algorithme

puisque plus l'horizon est grand plus le *factor graph* associé au problème est grand, ce qui conduit à un temps d'évaluation plus grand. De plus, dans le cas de l'horizon infini, nous faisons une approximation supplémentaire : celle liée à la troncature de l'horizon.

Nous avons utilisé une machine avec 2 processeurs de 8 coeurs (donc 16 coeurs en tout) et 128 Go de RAM. Nous avons utilisé le logiciel `Matlab R2014a`, ainsi que la toolbox de parallélisation associée. Pour l'évaluation par la méthode de Monte-Carlo (voir section 2.2.2), nous avons utilisé systématiquement un horizon de $T = 40$, un nombre de simulations de nsim $= 4000$ et une parallélisation. Pour les algorithmes d'inférence dans les modèles graphiques (*Loopy Belief propagation, Junction Tree...*), nous avons utilisé l'interface Matlab de la librairie libDAI [Moo10]. Pour l'algorithme MF-API (et l'évaluation en champ moyen), nous avons utilisé un code existant sous `Scilab 5.4.1`. Les temps de calcul sont à prendre comme des ordres de grandeur car ils dépendent de la charge du serveur de calcul utilisé. Les paramètres pour l'algorithme de descente de gradient utilisés dans toute cette section sont de : maxit $= 1000$, $\epsilon = \epsilon_g = \epsilon_V = \epsilon_\theta = 0.01$. Si cela n'est pas mentionné, l'algorithme GD-LBP s'est arrêté pour condition nécessaire d'optimalité locale vérifiée (norme du gradient proche de zéro).

2.4.1 Génération de PDMF^3 aléatoires

Nous décrivons ici le protocole utilisé pour générer des PDMF^3 aléatoires lors des expériences numériques. L'algorithme de génération de PDMF^3 aléatoires prend en entrée le nombre de variables d'état n, le nombre de variables d'action m, le nombre de fonctions de récompense r, le nombre maximum d'états possibles par variable d'état n_s et le nombre maximum d'actions possibles par variable d'action n_a. Enfin, une dernière entrée, appelée v, représente le nombre de voisins maximum pour les facteurs dans le *factor graph* associé au PDMF^3. Pour le cas sans arcs synchrones, les étapes sont les suivantes :

1. Tirage aléatoire de la taille de chaque variable d'état entre 2 et n_s et de la taille de chaque variable d'action entre 2 et n_a.
2. Pour chaque fonction de parentèle (pa_P^S,pa_P^A,pa_δ^S,pa_δ^A,pa_R^S,pa_R^A), par exemple pour pa_P^S :
   — tirage aléatoire du nombre de parents de type état $n_{pa}^S(i)$ pour chaque variable d'état i entre 0 et $min(m, (v-1)/2)$
   — tirage aléatoire sans remise des $n_{pa}^S(i)$ variables d'état pour chaque variable d'état i.

   Prendre $min(m, (v-1)/2)$ comme maximum pour le nombre de parents permet d'assurer que pour les facteurs de transition, le nombre total de voisins (variables d'état et variables d'action) ne sera pas supérieur à v (voir figure 2.3).
3. Génération aléatoire des tables $P^0, P_i, i = 1...n, R_\alpha, \alpha = 1...r$ avec des nombres entre 0 et 1. Celles qui correspondent à des probabilités sont normalisées. On peut éventuellement générer une PSF aléatoire δ de structure pa_δ (lorsqu'il s'agit d'études sur l'évaluation).

Pour le cas avec arcs synchrones, il faut tirer un ordre sur les variables d'état et un ordre sur les variables d'action. Pour le cas particulier des PDMGs, le protocole est similaire, mais les parentèles sont identiques pour transition, politique et récompense.

2.4.2 Méthodes d'évaluation des PSFs

Nous nous sommes d'abord intéressés à l'impact du choix de la méthode de calcul et de l'horizon de troncature sur la qualité et le temps de calcul de l'évaluation des PSFs.

Pour la mise en œuvre pratique de l'évaluation approchée basée sur le calcul de marginales (voir section 2.2.3), nous avons utilisé l'interface Matlab de la librairie libDAI [Moo10]. Cette librairie utilise une représentation des distributions de probabilité sous forme de *factor graph* [KFL01] (voir annexe, section A.1.2). Les variables dont dépendent les marginales $b^t_\alpha(pa_R(R_\alpha))$ sont liées à la structure de la récompense, qui n'a pas forcément de lien avec celle de la transition ou de la politique. Pour des raisons pratiques liées à l'utilisation de la librairie libDAI, il est donc nécessaire de rajouter dans le *factor graph* de la figure 2.3 des facteurs additionnels $f_\alpha(pa_R(R_\alpha))$ dépendant des mêmes variables que les marginales à calculer, et dont la table associée ne contient que des 1. La figure 2.4 représente les facteurs additionnels, à ajouter à ceux déjà présents sur le *factor graph* de la figure 2.3 pour l'exemple de PDMF^3 servant de fil rouge. Cette astuce permet de récupérer les marginales $b^t_\alpha(pa_R(R_\alpha))$ mais augmente la taille du *factor graph* fourni en entrée à libDAI, donc le temps de calcul pour l'évaluation d'une politique stochastique factorisée. Cependant, dans le cas des PDMG notamment, cette manipulation n'est pas nécessaire puisque les marginales dépendent des mêmes variables que les facteurs politique : $S^t_{N(i)} \cup A^t_i$. Nous avons donc développé un code d'évaluation spécifique aux PDMF^3 qui vérifient cette condition, on l'appellera dans la suite code d'évaluation pour PDMG+.

D'autres librairies d'inférence dans les modèles graphiques ont été développées récemment qui seraient plus intéressantes pour notre cas d'utilisation [1]. Par exemple, la librairie openGM2 [ATK12] permet de ne stocker qu'une fois les facteurs qui apparaissent de manière répétée, ce qui n'est pas le cas de libDAI, et qui est très utile dans le cas de résaux bayésiens dynamiques.

Pour la comparaison empirique des méthodes d'évaluation, nous avons généré 100 PDMF^3 aléatoires avec PSF associée selon la méthode décrite section 2.4.1, avec $n = m = r = 6$, $n_s = n_a = 2$, $v = 7$. Ces problèmes sont de taille suffisamment petite pour que la PSF puisse être évaluée de manière exacte (à horizon infini) par transformation en un PDM et utilisation de la MDP toolbox [2] [CCC+14]. La table 2.1 donne l'erreur relative moyenne et le temps d'évaluation moyen sur les 100 PDMF^3, pour différentes méthodes d'évaluation et différents horizons T de troncature. La table 2.2 donne les résultats obtenus pour l'évaluation de politiques factorisées déterministes (il s'agit des politiques stochastiques évaluées pour la table 2.1 mais déterminisées). MC représente l'évaluation

1. voir `http://www.cs.ubc.ca/~murphyk/Software/bnsoft.html` pour une revue des librairies d'inférence dans les modèles graphiques

2. `http://www7.inra.fr/mia/T/MDPtoolbox/`

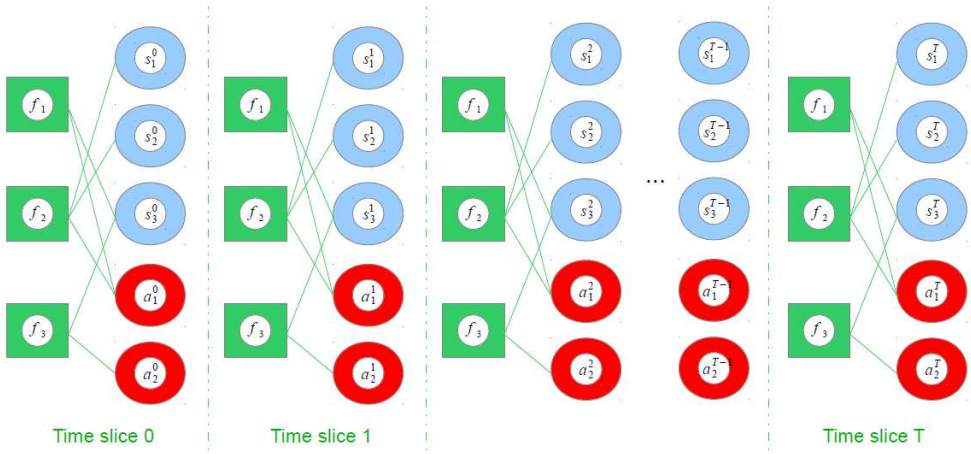


FIGURE 2.4 – Représentation des facteurs additionnels $f_\alpha^t(pa_R(R_\alpha)), \alpha = 1...3$ nécessaires dans le *factor graph* de la figure 2.3 pour l'évaluation d'une politique stochastique factorisée du PDMF^3 de la figure 2.1

par la méthode de Monte-Carlo (voir section 2.2.2), JT représente l'évaluation par calcul de marginales avec l'algorithme exact *Junction Tree* et LBP représente l'évaluation par calcul de marginales avec l'algorithme approché *Loopy Belief Propagation* (voir section 2.2.3). Le temps de calcul pour l'évaluation exacte comprend la transformation en PDM et l'évaluation à proprement parler, mais le temps de l'évaluation uniquement est également donné entre parenthèses.

méthode d'évaluation	ERM	temps moyen (sec)
MC $T = 40$	0.01	23.45
JT $T = 20$	0.11	0.085
LBP $T = 20$	0.11	0.049
MC $T = 100$	6.8×10^{-4}	66.05
JT $T = 100$	2.4×10^{-5}	1.1
LBP $T = 100$	0.009	0.23
exact $T = \infty$		21.76 (0.0022)

TABLE 2.1 – Résultats d'évaluation de politiques factorisées stochastiques sur 100 PDMF^3 aléatoires ($n = m = r = 6, n_s = n_a = 2$)

Pour un horizon de $T = 20$, un facteur d'amortissement de $\gamma = 0.9$, et une récompense maximum de $R_{max} = 6$, l'erreur absolue de troncature (voir section 2.2.4) est de $\frac{\gamma^{T+1}}{1-\gamma} R_{max} \approx \epsilon_1 = 6.6$. Elle est d'environ $\epsilon_1 = 0.8$ pour $T = 40$ et $\epsilon_1 = 0.0014$ pour $T = 100$.

Avec un nombre de simulations de nsim = 4000 et un horizon de $T = 40$, l'erreur absolue d'évaluation avec la méthode de Monte-Carlo (comprenant aussi l'erreur de

méthode d'évaluation	ERM	temps moyen (sec)
MC $T = 40$	0.01	28.28
JT $T = 20$	0.11	0.095
LBP $T = 20$	0.11	0.054
MC $T = 100$	6.7×10^{-4}	98.57
JT $T = 100$	2.4×10^{-5}	1.46
LBP $T = 100$	0.02	0.31
exact $T = \infty$		20.24 (0.0022)

TABLE 2.2 – Résultats d'évaluation de politiques factorisées déterministes sur 100 PDMF[3] aléatoires ($n = m = r = 6, n_s = n_a = 2, v = 7$)

méthode d'évaluation	ERM	temps moyen (sec)	temps moyen (sec)
code d'évaluation		générique	pour PDMG+
MC $T = 40$	0.01	51.04	-
JT $T = 20$	0.11	0.16	0.099
LBP $T = 20$	0.11	0.14	0.062
MF $T = 20$	0.13	0.027	-
MC $T = 100$	6.5×10^{-4}	124.34	-
JT $T = 100$	2.4×10^{-5}	1.86	1.28
LBP $T = 100$	0.01	0.90	0.27
MF $T = 100$	0.071	0.032	-
exact $T = \infty$		22.8 (0.0027)	-

TABLE 2.3 – Résultats d'évaluation de politiques factorisées déterministes sur 100 PDMG aléatoires ($n = 6, n_s = n_a = 2, v = 7$)

troncature) est plus petite que $\epsilon_1 + \epsilon_2 = 0.8 + 1.15 = 1.95$ avec probabilité $\kappa = 0.9$. Pour des valeurs de l'ordre de 200, 1.95 est une erreur absolue raisonnable, qui donne une erreur relative de l'ordre de 0.01. C'est bien ce que l'on retrouve expérimentalement dans la table 2.1.

Junction Tree étant un algorithme d'inférence exact, l'erreur observée avec cet algorithme est uniquement due à la troncature de l'horizon. De ce fait, pour $T = 20$, l'erreur commise avec l'algorithme approché *Loopy Belief Propagation* est négligeable par rapport à l'erreur de troncature. Mais le gain en temps de calcul n'est pas négligeable.

Pour un horizon $T = 100$, les trois méthodes donnent de bons résultats en termes d'erreur relative, mais sont trop lentes pour être incorporées dans un algorithme d'optimisation pour des problèmes de grande taille. C'est pourquoi dans la suite nous utiliserons dans nos algorithmes de type itération de la politique la méthode d'évaluation LBP avec un horizon de $T = 20$. Pour comparer les valeurs des politiques retournées par ces algorithmes de manière plus précise, nous utiliserons pour les problèmes de grande taille une évaluation basée sur la méthode de Monte-Carlo, avec nsim = 4000 et $T = 40$.

Les erreurs relatives moyennes pour des politiques déterministes (voir table 2.2) sont proches de celles pour des politiques stochastiques, sauf dans le cas de l'évaluation par LBP pour $T = 100$ (l'erreur est plus importante dans le cas de politiques déterministes). Cette observation a déjà été faite dans la littérature [SG14]. L'algorithme *Conditioned Belief Propagation* (CBP, [EG09]) serait plus performant dans le cas de facteurs déterministes. Mais nous avons fait l'essai pour des politiques déterministes, et les résultats n'étaient pas meilleurs qu'avec LBP.

Les résultats obtenus sur des PDMG aléatoires à $n = 6$ noeuds étaient similaires à ceux obtenus pour les PDMF[3]. Dans le cas de politiques déterministes, nous avons également comparé avec l'évaluation en champ moyen utilisée dans l'algorithme MF-API : voir table 2.3. L'erreur commise par cette méthode est plus élevée que celle commise avec l'algorithme *Loopy Belief Propagation*. Cela confirme les résultats théoriques, puisque *Loopy Belief Propagation* est une méthode variationnelle plus précise que l'approximation en champ moyen. Enfin, on peut constater que le temps d'évaluation est considérablement réduit lorsqu'on utilise le code spécifique aux PDMG+ (qui ne nécessite pas d'ajouter des facteurs additionnels dans le modèle graphique, voir plus haut).

L'erreur de troncature et l'erreur commise par la méthode de Monte-Carlo sont indépendantes de la taille du problème considéré. Par contre, il se pourrait que l'erreur commise par l'algorithme *Loopy Belief Propagation* soit plus importante sur les problèmes de grande taille. C'est pourquoi nous vérifierons la qualité de l'évaluation par LBP sur des problèmes de grande taille en comparaison avec l'évaluation par la méthode de Monte-Carlo (voir section 2.4.4), pour des politiques fournies par l'algorithme d'optimisation cette fois.

2.4.3 Expériences préliminaires sur des problèmes aléatoires de petite taille

Nous allons maintenant nous intéresser à la comparaison de différentes instanciations de l'algorithme d'itération de la politique approchée (différents couples Optim-Eval) sur

des PDMF[3] aléatoires de petite taille, en comparaison avec une résolution exacte par l'algorithme d'itération de la politique de la MDP toolbox [CCC+14]. La résolution exacte donne la politique globale optimale. Celle-ci n'a pas forcément la structure supposée, l'erreur relative moyenne est donc une borne supérieure de l'erreur relative par rapport à la meilleure politique factorisée avec la structure supposée.

Les politiques obtenues sont souvent quasiment déterministes, c'est pourquoi nous donnons en plus de l'erreur relative moyenne celle obtenue pour les politiques déterminisées (dans la colonne 'ERM det'), qui est souvent légèrement inférieure.

L'algorithme de descente de gradient, contrairement à la descente par coordonnées, peut s'appliquer dans le cas de variables d'action non binaires. La parallélisation du calcul approché du gradient (par la méthode des différences finies) devient alors intéressante en termes de temps de calcul. Dans la table 2.5, l'algorithme GD-LBP est donc parallélisé, tandis qu'il ne l'est pas dans la table 2.4. Les temps de calcul sont plus faibles que dans la table 2.4 car le nombre de variables et de fonctions de récompense est plus faible ($n = m = r = 5$). L'erreur relative moyenne est quant à elle du même ordre.

Dans le but de valider expérimentalement la structure 'naturelle' proposée dans la section 2.1.4, nous avons comparé les résultats obtenus pour cette structure et pour une structure aléatoire basée sur le même nombre de variables. A chaque fois, les résultats sont légèrement meilleurs avec la structure naturelle, ce qui permet de penser que ce n'est pas un mauvais choix par rapport à une structure aléatoire, lorsque la structure de politique n'est pas dans les données du problème. Cependant, cette validation expérimentale a des limites, car sur un petit problème il est difficile que les deux structures soient très différentes puisque la structure 'naturelle' a des chances d'englober une grande partie des variables.

Lorsque l'algorithme d'évaluation est *Junction Tree*, l'erreur d'évaluation vient uniquement de la troncature de l'horizon. Les autres sources d'erreur par rapport à la politique globale optimale sont dues :

— au fait que l'algorithme d'optimisation est local

— au fait que l'on suppose une structure de politique donnée.

L'erreur est beaucoup plus importante lorsqu'on utilise l'évaluation approchée basée sur l'algorithme *Loopy Belief Propagation* (on passe de 2% à 12% environ), ce qui prouve que l'utilisation de LBP est à l'origine de la majeure partie de l'approximation dans les algorithmes GD-LBP et CD-LBP. Passer d'un horizon $T = 20$ à $T = 40$ dans le cas de l'algorithme CD-LBP n'améliore pas l'erreur relative moyenne. Par contre, cela augmente le temps de calcul, qui est multiplié par $3,4$. C'est pourquoi dans la suite nous utiliserons un horizon de $T = 20$ pour approcher l'horizon infini dans le cas d'une évaluation par LBP. Les algorithmes GD-JT et CD-JT ne seront malheureusement pas applicables sur des problèmes de grande taille.

Concernant la méthode de choix du pas pour la descente de gradient, on constate qu'utiliser un pas fixe, à condition de bien le choisir, est la méthode la plus rapide. Mais comme on ne connaît pas forcément à l'avance la bonne valeur pour le pas, il vaut mieux, comme le confirment les résultats, utiliser la méthode de Wolfe qu'un pas optimal. C'est donc le choix que nous ferons dans la suite des expérimentations.

Les pas pour l'algorithme de descente par coordonnées ne sont pas du même ordre de grandeur, puisqu'ils se font directement sur la politique (paramètres entre 0 et 1) plutôt que sur un paramètre θ réel. Là aussi, bien choisir la valeur du pas permet de gagner beaucoup de temps de calcul. En effet, passer d'un pas de $0,01$ à un pas de $0,1$ divise par 12 le temps de calcul pour des résultats de qualité équivalente.

La résolution exacte est plus rapide mais elle ne permet pas de trouver la meilleure politique factorisée avec la structure contrainte, tandis que CD-JT ou GD-JT fournissent une politique factorisée localement optimale présentant cette structure donnée.

	ERM	ERM det	temps moyen
GD-JT struct. nat. T=20, pas fixe de 20	0.0200	0.0192	3.25 min
GD-JT struct. al. T=20, pas fixe de 20	0.048	0.047	3.66 min
GD-LBP struct. nat. T=20, pas fixe de 20	0.1228	0.1226	1.88 min
GD-LBP struct. nat. T=20, pas fixe de 10	0.1214	0.1206	2.47 min
GD-LBP struct. nat. T=20, pas de wolfe	0.1216	0.1207	2.48 min
GD-LBP struct. nat. T=20, pas optimal	0.1217	0.1206	2.63 min
CD-JT struct. nat. T=20, pas=0.1	0.0222	0.0222	5.14 min
CD-LBP struct. nat. T=20, pas=0.01	0.1274	0.1231	18.37 min
CD-LBP struct. nat. T=20, pas=0.1	0.1228	0.1230	1.53 min
CD-LBP struct. nat. T=40, pas=0.1	0.1231	0.1232	5.22 min
CD-LBP struct. al. T=40, pas=0.1	0.1327	0.1327	5.45 min
exact, struct. globale, $T = \infty$	-	-	1.95 min

TABLE 2.4 – Résultats d'optimisation sur 100 PDMF^3 aléatoires ($n = m = r = 6, n_s = n_a = 2, v = 7$) - point de départ : politique uniforme

	ERM	ERM det	temps moyen
GD-LBP struct. nat. T=20, pas fixe de 20	0.1288	0.1287	39.74s
exact, struct. globale, $T = \infty$	-	-	43.44s

TABLE 2.5 – Résultats d'optimisation sur 100 PDMF^3 aléatoires ($n = m = r = 5, n_s = 2, n_a = 3, v = 7$) - point de départ : politique uniforme

La table 2.6 donne les résultats obtenus pour 100 PDMG aléatoires à $n = 6$ nœuds. Dans ce cas, nous avons pu comparer CD-LBP et GD-LBP avec l'algorithme MF-API spécifique aux PDMG. Le point de départ, pour cet algorithme, doit être déterministe. Nous avons choisi comme point de départ, pour tous les algorithmes, la politique gloutonne, qui consiste à choisir l'action qui maximise la récompense locale :

$$\forall a^t \in \mathcal{A}, \forall s^t \in \mathcal{S},\ \delta^{glout}(a^t|s^t) = \prod_{i=1}^{m} \delta_i^{glout}(a_i^t|s_{N(i)}^t)$$

où

$$\delta_i^{glout}(a_i^t|s_{N(i)}^t) = \begin{cases} 1 & \text{si } a_i^t = \text{argmax}_{a_i' \in \mathcal{A}_i} R_i(s_{N(i)}, a_i') \\ 0 & \text{sinon} \end{cases}$$

Les trois algorithmes retournent très souvent le point de départ, la politique gloutonne, ce qui explique les faibles temps de calcul. MF-API donne de meilleurs résultats, bien que la méthode d'évaluation utilisée dans cet algorithme, l'évaluation en champ moyen, soit moins précise que celle basée sur l'algorithme *Loopy Belief Propagation* (voir section 2.4.2).

	ERM	ERM det	temps moyen (s)
GD-LBP struct. nat. T=20, pas fixe de 10	0.0078	0.0078	5.75
CD-LBP struct. nat. T=20, pas=0.1	0.0065	0.0066	4.45
MF-API struct. nat. T=20	0.0015	-	1.79
exact, struct. globale, $T = \infty$	-	-	115.16

TABLE 2.6 – Résultats d'optimisation sur 100 PDMG aléatoires à $n = 6$ noeuds - point de départ : politique gloutonne

Application au contre-exemple de la figure 2.2

Sur l'exemple de la figure 2.2 (voir section 2.1.5), CD-LBP renvoie une politique stochastique proche de la politique δ_5 :

$$\delta_5(a_1|s_2) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_2 \\ \begin{pmatrix} 0 & 0.5 \\ 1 & 0.5 \end{pmatrix} \end{array}$$

dont la valeur exacte est $V_{\delta_5}^{R,\infty}(P^0) = 7.4975$. GD-LBP renvoie une autre politique stochastique :

$$\delta_6(a_1|s_2) = \begin{array}{c} \\ a_1 \end{array} \begin{array}{c} s_2 \\ \begin{pmatrix} 0.23 & 0.3 \\ 0.77 & 0.7 \end{pmatrix} \end{array}$$

dont la valeur exacte est $V_{\delta_6}^{R,T}(P^0) = 7.491$. Cela montre que les deux algorithmes ne renvoient pas forcément les mêmes politiques, et qu'il est intéressant, lorsque les variables d'action sont binaires, de comparer les résultats des deux algorithmes.

2.4.4 Problèmes aléatoires de grande taille

Nous nous intéressons maintenant à des problèmes de grande taille. Nous allons d'abord considérer des problèmes aléatoires, plutôt que des problèmes spécifiques. Les problèmes étant de plus grande taille, nous ne pourrons considérer que 5 problèmes aléatoires et non plus 100, pour des raisons de temps de calcul. De plus, l'évaluation

a posteriori sera une évaluation basée sur la méthode de Monte-Carlo car une évaluation exacte n'est plus possible. L'algorithme de descente de gradient est maintenant systématiquement parallélisé, et utilise la méthode de Wolfe pour la recherche linéaire.

On constate dans la table 2.7 que les valeurs des politiques obtenues avec GD-LBP et CD-LBP sont proches (il faut tenir compte de l'aléa de l'évaluation par Monte-Carlo), mais que GD-LBP est beaucoup plus rapide que CD-LBP. A chaque fois, environ 4 itérations de GD-LBP sont nécessaires pour obtenir un gradient dont la norme est proche de zéro. Dans le cas de variables d'action non binaires, CD-LBP n'est pas applicable, c'est pourquoi dans la table 2.8 nous avons comparé la valeur par la méthode de Monte-Carlo obtenue en sortie de l'algorithme GD-LBP avec celle de la politique uniforme (qui sert de point de départ à GD-LBP) et avec celle d'une politique aléatoire, de même structure évidemment. GD-LBP renvoie des politiques de valeur supérieure à ces deux politiques d'environ 20%. Les politiques obtenues avec GD-LBP et CD-LBP sont systématiquement déterministes.

L'erreur relative entre l'évaluation approchée par l'algorithme LBP et l'évaluation plus précise par la méthode de Monte-Carlo est inférieure à 10% dans le cas de variables d'action binaires (comme pour les petits problèmes, voir section 2.4.2). Mais l'ordre est respecté pour les valeurs des politiques obtenues avec GD-LBP ou avec CD-LBP. Cette erreur est de l'ordre de 20% dans le cas de variables d'action non binaires.

La table 2.9 compare les résultats obtenus avec les algorithmes GD-LBP, CD-LBP et MF-API sur 5 PDMG aléatoires à $n = 15$ noeuds. GD-LBP et CD-LBP renvoient la politique gloutonne qui sert de point de départ, mais pas MF-API. Cependant, les politiques retournées par MF-API sont de valeur très proche de la valeur de la politique gloutonne. On peut remarquer aussi que l'évaluation en champ moyen utilisée par MF-API sous-estime systématiquement la valeur, tandis que l'évaluation par LBP la surestime. L'erreur relative d'évaluation est systématiquement plus élevée pour l'évaluation en champ moyen.

	PDMF^3 n^o 1	PDMF^3 n^o 2	PDMF^3 n^o 3	PDMF^3 n^o 4	PDMF^3 n^o 5
nombre de paramètres (N/N')	3554/1777	1840/920	1558/779	1702/851	3616/1808
V^{LBP} de δ_{GD-LBP}, $T = 20$	88.75	91.27	94.31	95.87	94.48
V^{LBP} de δ_{CD-LBP}, $T = 20$	91.01	93.52	97.04	94.24	94.23
V^{MC} de δ_{GD-LBP}, $T = 40$	80.99	84.68	86.65	**88.82**	**87.11**
V^{MC} de δ_{CD-LBP}, $T = 40$	**83.53**	**86.70**	**89.46**	87.24	86.76
ER LBP pour δ_{GD-LBP}	9.6%	7.8%	8.8%	7.9%	8.5%
ER LBP pour δ_{CD-LBP}	9.0%	7.9%	8.5%	8.0%	8.6%
temps GD-LBP (sec)	**475.06**	**223.84**	**938.16**	**371.99**	**745.36**
temps CD-LBP (sec)	5220.4	3050.9	4847.0	2241.6	6913.7

TABLE 2.7 – Résultats sur 5 PDMF^3 aléatoires à variables d'action binaires - structure de politique aléatoire - point de départ : politique uniforme - $n = m = r = 15, n_s = n_a = 2, v = 10$

	PDMF^3 n^o 1	PDMF^3 n^o 2	PDMF^3 n^o 3	PDMF^3 n^o 4	PDMF^3 n^o 5
nombre de paramètres (N)	4627	4558	3040	3234	4529
V^{LBP} de δ_{GD-LBP}, $T = 20$	92.58	90.30	94.14	87.64	96.67
V^{MC} de δ_{GD-LBP} , $T = 40$	**77.04**	**74.05**	**77.68**	**72.10**	**81.48**
V^{MC} de politique uniforme, $T = 40$	65.26	67.52	65.77	62.59	66.57
V^{MC} de politique aléatoire, $T = 40$	65.57	67.52	65.97	62.98	66.69
ER LBP pour δ_{GD-LBP}	20.2%	21.9%	21.2%	21.6%	18.6%
temps GD-LBP (sec)	1299.6	1179.2	955.1	554.4	502.4

TABLE 2.8 – Résultats sur 5 PDMF^3 aléatoires à variables d'action non binaires - structure de politique aléatoire - point de départ : politique uniforme - $n = m = r = 15, n_s = 2, n_a = 3, v = 10$

	PDMG n° 1	PDMG n° 2	PDMG n° 3	PDMG n° 4	PDMG n° 5
nombre de paramètres (N/N')	952/476	1456/728	1232/616	952/476	2304/1152
V^{MF} de δ_{MF-API}, $T = 20$	88.09	92.42	90.53	88.09	90.91
V^{LBP} de δ_{GD-LBP}, $T = 20$	103.25	107.45	106.15	103.25	106.99
V^{LBP} de δ_{CD-LBP}, $T = 20$	103.25	107.45	106.15	103.25	106.99
V^{MC} de δ_{MF-API}, $T = 40$	**97.54**	**102.30**	**100.36**	**97.58**	**100.64**
V^{MC} de δ_{GD-LBP}, $T = 40$	97.53	101.84	100.20	97.49	100.63
V^{MC} de δ_{CD-LBP}, $T = 40$	97.53	101.84	100.20	97.49	100.63
ER MF pour δ_{MF-API}	9.7%	9.7%	9.8%	9.7%	9.7%
ER LBP pour δ_{GD-LBP}	5.9%	5.5%	5.9%	5.9%	6.3%
ER LBP pour δ_{CD-LBP}	6.0%	5.5%	6.0%	6.0%	6.4%
temps MF-API (sec)	**1.07**	**1.06**	**0.79**	**1.09**	**1.81**
temps GD-LBP (sec)	23.67	47.18	29.85	21.58	77.26
temps CD-LBP (sec)	149.64	504.39	286.65	154.69	799.74

TABLE 2.9 – Résultats sur 5 PDMG aléatoires - point de départ : politique gloutonne - $n = 15, n_s = n_a = 2, v = 10$

2.4.5 Problèmes d'épidémiologie à l'échelle du paysage

Utilisation des symétries des paysages parcellaires

Dans toute cette section, nous étudions des problèmes d'épidémiologie dans des paysages agricoles théoriques de type parcellaire régulier en forme de grille. Les décisions se prennent à l'échelle de la parcelle en fonction d'informations similaires. Nous pouvons donc utiliser les symétries de ces parcellaires pour réduire le temps de calcul. Ainsi, il suffit de calculer les politiques locales pour les parcelles en gras dans la figure 2.5, les autres peuvent s'obtenir par symétrie. Nous utilisons dans cette section la méthode décrite section 2.3.2 en basant la partition des indices des variables d'action sur les symétries de la grille.

1	**2**	**3**	4	5
6	**7**	**8**	9	10
11	12	**13**	14	15
16	17	18	19	20
21	22	23	24	25

FIGURE 2.5 – Paysage théorique de type parcellaire 5×5 : par symétrie, il suffit de calculer les politiques locales pour les parcelles en gras (1, 2, 3, 7, 8 et 13)

Problème épidémiologique de type PDMG

Nous nous intéressons maintenant à des problèmes de gestion de maladies contagieuses à l'échelle du paysage agricole. Supposons tout d'abord que les agriculteurs agissent à l'échelle de la parcelle en fonction de l'état (sain ou infecté) des parcelles voisines. Agir consiste à traiter et laisser le champ en jachère. Ce problème est décrit dans [SPF12] et peut se modéliser sous forme de PDMG (voir aussi [CLCI13]).

Nous considérons tout d'abord un paysage sous forme de grille 5x5 à $n = 25$ parcelles. Chaque parcelle i peut donc être dans deux états : saine ($s_i = 1$) ou infectée ($s_i = 2$). L'action qui consiste à ne rien faire est codée par $a_i = 1$ et l'action qui consiste à traiter la maladie et laisser la parcelle en jachère est codée par $a_i = 2$. On considère que la maladie se propage principalement vers les parcelles du voisinage d'ordre 1 (celles à gauche, à droite, au-dessus et en-dessous). On note $N(i)$ l'ensemble constitué de la parcelle i et des ses voisines (au plus quatre). La probabilité de transition est factorisée :

$$P(s'|s,a) = \prod_{i=1}^{n} P_i(s'_i|s_{N(i)}, a_i)$$

et les probabilités de transition locales sont données dans la table 2.10. La probabilité pour un champ sain de devenir infecté l'année suivante en l'absence de traitement est fonction de l'état d'infection des champs voisins :

$$F(s_{N(i)}, a_{N(i)}) = F(s_{N(i)}) = \epsilon + (1-\epsilon)(1-(1-p)^{n_i})$$

où ϵ est la probabilité de contamination à longue distance, p la probabilité de contamination à courte distance, et n_i est le nombre de parcelles voisines infectées : $n_i = \sum_{j \in N(i)} \mathbb{1}_{s_j=2}$. Un champ infecté qui est traité et laissé en jachère a une probabilité q de redevenir sain l'année suivante. La récompense est additive et il y a une fonction de récompense par parcelle : $R(s,a) = \sum_{i=1}^{n} R_i(s_i, a_i)$ où :

$$\forall i = 1...n,\ R_i(s_i, a_i) = \begin{pmatrix} \rho & 0 \\ \rho/2 & 0 \end{pmatrix}, \rho > 0$$

La récompense représente en effet le rendement. Celui-ci est nul lorsque le champ est traité et laissé en jachère, le rendement est maximal égal à ρ lorsque le champ est sain et non traité, et ce rendement est divisé par deux lorsque le champ est infecté et non traité.

	$a_i = 1$		$a_i = 2$	
	$s_i = 1$	$s_i = 2$	$s_i = 1$	$s_i = 2$
$s'_i = 1$	$1 - F(s_{N(i)}, a_{N(i)})$	0	1	q
$s'_i = 2$	$F(s_{N(i)}, a_{N(i)})$	1	0	1-q

TABLE 2.10 – Probabilités de transition locales pour le problème épidémiologique $P_i(s'_i | s_{N(i)}, a_i)$

La structure de politique recherchée est basée sur les états des parcelles voisines :

$$\delta(a^t | s^t) = \prod_{i=1}^{n} \delta_i(a_i^t | pa_\delta(a_i^t))$$

où $pa_\delta(a_i^t) = s_{N(i)}^t$.

Nous avons pris $\epsilon = 0.01$, $p = 0.2$, $q = 0.9$, $\rho = 100$ et utilisé comme initialisation pour les algorithmes CD-LBP, GD-LBP et MF-API la politique gloutonne (qui consiste à ne jamais traiter aucune parcelle).

La table 2.11 donne les résultats obtenus pour une grille 5×5, et la table 2.12 pour une grille 10×10. Les temps de calcul des algorithmes CD-LBP et GD-LBP sont donnés sans prise en compte des symétries de la grille (pour comparaison avec MF-API qui n'en tient pas compte) et avec prise en compte des symétries en dessous (le nombre de paramètres N''' est alors inférieur à N', et le temps de calcul est moins élevé).

Nous avons utilisé un horizon de $T = 20$ dans les trois algorithmes d'optimisation, et un horizon de $T = 40$ pour l'évaluation *a posteriori* par la méthode de Monte-Carlo. Nous avons vérifié qu'utiliser un horizon de $T = 40$ dans les trois algorithmes d'optimisation ne changeait pas la politique finale renvoyée. La politique obtenue avec CD-LBP est déterministe et identique à celle obtenue avec l'algorithme MF-API : il s'agit de la politique 'triviale', dépendant uniquement de l'état de la parcelle considérée, qui consiste à traiter si la parcelle est infectée, et à ne pas traiter sinon. Pour des valeurs de p (contamination courte distance) plus importantes, nous avons obtenu les mêmes résultats. Cette politique améliore la valeur de la politique gloutonne d'environ 60% pour les deux tailles

de grille. L'algorithme GD-LBP, quant à lui, renvoie la politique de départ, la politique gloutonne. Par contre, si on lui donne comme point de départ la politique uniforme, il renvoie la même politique que CD-LBP et MF-API. La politique gloutonne est donc un point critique [3] de la valeur estimée par LBP pour le problème 5, reparamétré, utilisé par l'algorithme GD, mais pas pour le problème 4, non reparamétré et utilisé par l'algorithme CD. On peut constater enfin que l'algorithme MF-API, qui recherche parmi l'ensemble (plus restreint) des politiques factorisées déterministes, est beaucoup plus rapide.

3. Rappelons qu'un point critique est un point d'annulation du gradient, ce qui est une condition nécessaire mais pas suffisante d'optimalité locale.

	temps	iter	nb éval	nb param	V MC $T = 40$	V LBP $T = 20$	V MF $T = 20$	politique obtenue
références								
politique gloutonne					13953	13836	13004	-
politique uniforme					11393	11258	-	-
départ : politique gloutonne								
CD-LBP, pas=0.1	15 min 4.64 min	4	4507	$N' = 512$ $N''' = 136$	21878	17361	19520	triviale
GD-LBP	17s 8s	1	1025	$N = 1024$ $N'' = 272$	13953	13836	13004	gloutonne
MF-API	0.82s	2		-	21878	17361	19520	triviale
départ : politique uniforme								
GD-LBP	1.16 min 36s	3	4112	$N = 1024$ $N'' = 272$	21878	17361	19520	triviale

TABLE 2.11 – Résultats sur le problème épidémiologique (PDMG) grille 5x5 - structure de politique classique - Les temps de calcul des algorithmes CD-LBP et GD-LBP sont donnés sans prise en compte des symétries de la grille (pour comparaison avec MF-API qui n'en tient pas compte) et avec prise en compte des symétries en dessous (le nombre de paramètres N''' est alors inférieur à N').

	temps	iter	nb éval	nb param	V MC $T = 40$	V LBP $T = 20$	V MF $T = 20$	politique obtenue
références								
politique gloutonne					55226	50114	50113	-
politique uniforme					44983	40038	-	-
départ : politique gloutonne								
CD-LBP, pas=0.1	6h34 58 min	4	22359 3390	$N' = 2592$ $N''' = 392$	86326	57373	76976	triviale
GD-LBP	8.87 min 1.4 min	1	5185	$N = 5184$ $N'' = 784$	55226	50114	50113	gloutonne
MF-API	3.58s	2		-	86326	57373	76976	triviale
départ : politique uniforme								
GD-LBP	57.84 min 7.27 min	6	31123	$N = 5184$ $N'' = 784$	86326	57373	76976	triviale

TABLE 2.12 – Résultats sur le problème épidémiologique (PDMG) grille 10x10 - structure de politique classique - Les temps de calcul des algorithmes CD-LBP et GD-LBP sont donnés sans prise en compte des symétries de la grille (pour comparaison avec MF-API qui n'en tient pas compte) et avec prise en compte des symétries en dessous (le nombre de paramètres N''' est alors inférieur à N').

Problème épidémiologique de type PDMF^3

Nous nous intéressons maintenant à une version du problème plus réaliste pour certaines maladies, où le traitement a lieu avant la propagation de la maladie et réduit la probabilité de contamination vers les parcelles voisines. Dans ce cas, le problème n'est plus un PDMG mais un PDMF^3, puisque les probabilités de transition locales dépendent des actions sur les parcelles voisines :

$$P(s'|s,a) = \prod_{i=1}^{n} P_i(s'_i|s_{N(i)}, a_{N(i)})$$

On considère que le traitement réduit la probabilité de propagation de la maladie à courte distance à $p_2 < p_1$. Les probabilités de transition locales sont données par la table 2.10 où :

$$F(s_{N(i)}, a_{N(i)}) = \epsilon + (1-\epsilon)(1-(1-p_1)^{n_1}) + (1-\epsilon)(1-p_1)^{n_1}(1-(1-p_2)^{n_2})$$

n_1 représente le nombre de voisins infectés et non traités, p_1 la probabilité de contamination venant d'un voisin non traité, n_2 le nombre de voisins infectés et traités et p_2 la probabilité de contamination venant d'un voisin traité.

Nous avons pris comme valeurs de paramètres des valeurs pour lesquelles, sur une grille à 4 noeuds, la politique optimale exacte n'est pas triviale :

$$\epsilon = 0.01, q = 0.9, r = 100, p_1 = 0.6, p_2 = 0.4$$

Remarquons que dans ce cas précis on peut utiliser le code d'évaluation LBP pour PDMG+ (qui ne nécessite pas de rajouter des facteurs artificiels, voir section 2.4.2), puisque récompenses locales et politiques locales dépendent des mêmes variables.

CD-LBP renvoie la politique triviale (qui consiste à traiter la parcelle quand elle est infectée et à ne pas la traiter quand elle est saine). Dans le cas de la grille 5×5 on améliore de 23.8% la politique gloutonne, et dans le cas de la grille 10×10 de 19.2%. GD-LBP renvoie la politique gloutonne pour les deux grilles.

	temps	iter	nb éval	nb param	V MC (det) , T=40	V LBP
politique uniforme					10367	10078 (T=40)
politique gloutonne					13191	13167 (T=40)
politique aléatoire					9411	9502 (T=40)
CD-LBP, pas=0.1	1h16 23.2 min	3	3892	$N' = 512$ $N''' = 136$	17264 (17300)	15106 (T=20)
GD-LBP	21.4 min 2.83 min	3	4125	$N = 1024$ $N'' = 272$	13195 (13191)	14701 (T=20)

TABLE 2.13 – Résultats sur le problème épidémiologique (PDMF^3) - grille 5x5 - point de départ : politique uniforme - structure de politique classique - Les temps de calcul des algorithmes CD-LBP et GD-LBP sont donnés sans prise en compte des symétries de la grille et avec prise en compte des symétries en dessous (le nombre de paramètres N''' est alors inférieur à N').

	temps	iter	nb éval	nb param	V MC (det) (T=40)	V LBP
politique uniforme					40495	39671 (T=40)
politique gloutonne					52575	52521 (T=40)
politique aléatoire					40329	39177 (T=40)
CD-LBP, pas=0.1	43h42 7.28h	3	22720 3597	$N' = 2592$ $N''' = 392$	65068	59484 (T=20)
GD-LBP	10h20 41 min	3	20765	$N = 5184$ $N'' = 784$	52574 (52576)	58657 (T=20)

TABLE 2.14 – Résultats sur problème épidémiologique (PDMF^3) - grille 10x10 - point de départ : politique uniforme - structure de politique classique - Les temps de calcul des algorithmes CD-LBP et GD-LBP sont donnés sans prise en compte des symétries de la grille et avec prise en compte des symétries en dessous (le nombre de paramètres N''' est alors inférieur à N').

Problème épidémiologique de type 'réseau de surveillance'

Nous nous intéressons maintenant au cas d'un réseau de surveillance : un suivi épidémiologique est effectué sur 4 parcelles témoins, régulièrement réparties, dans un paysage parcellaire de taille 5×5 (voir figure 2.6). Les caractéristiques du problème (transition, récompense...) sont les mêmes que celles du problème épidémiologique de type PDMG. La seule différence est dans la structure de la politique : la décision du traitement sur chaque parcelle doit se faire en fonction de l'état des quatre parcelles témoins centrales (saine/infectée). On a donc :

$$\forall j = 1...25, pa_\delta(A_j) = \{S_7, S_9, S_{17}, S_{19}\}$$

Dans la table 2.15, on donne les résultats obtenus avec les algorithmes CD-LBP et GD-LBP, pour les valeurs de paramètres suivantes : $\epsilon = 0.01, q = 0.9, r = 100, p = 0.2$. MF-API ne peut pas traiter ce problème car, à cause de la structure de la politique, le problème n'est pas un PDMG, mais un PDMF[3]. On compare avec la politique uniforme, la politique gloutonne (qui consiste à ne jamais traiter), et une politique aléatoire de même structure. On compare aussi avec deux politiques qui paraissent naturelles :

— politique 1 : traiter la parcelle si la ou les parcelles témoins les plus proches sont infectées [4]

— politique 2 : traiter la parcelle si 3 ou 4 des parcelles témoins sont infectées.

Que le point de départ soit uniforme ou aléatoire, GD-LBP s'arrête pour stagnation et non pour norme du gradient proche de 0. L'algorithme a donc des difficultés à converger vers un point critique. La valeur Monte-Carlo est supérieure pour la politique stochastique obtenue en sortie de l'algorithme que pour la politique déterminisée. Utiliser un horizon de $T = 40$ ou $T = 100$ dans l'algorithme GD-LBP n'améliore pas les résultats. La politique obtenue avec GD-LBP ou CD-LBP est de valeur plus élevée que la politique uniforme, la politique aléatoire et la politique gloutonne, mais moins élevée que les politiques 'de bon sens' 1 et 2. Celles-ci sont tout de même des points critiques de $V_\delta^{LBP}(P^0)$.

La table 2.16 donne les résultats obtenus pour une probabilité de dispersion plus importante : $p = 0.8$. Que l'on parte de la politique uniforme ou d'une politique aléatoire, l'algorithme s'arrête pour norme du gradient proche de 0 et renvoie la politique gloutonne, qui consiste à ne jamais traiter. Celle-ci est de valeur supérieure à la politique uniforme ou une politique aléatoire, mais inférieure aux politiques 1 et 2, qui sont encore une fois des points critiques de $V_\delta^{LBP}(P^0)$. Utiliser l'évaluation Monte-Carlo, avec l'algorithme CD-MC, conduit curieusement à des résultats de moins bonne qualité, alors que le nombre d'évaluations (donc le nombre d'itérations de l'algorithme) est plus important. Et le temps de calcul est bien entendu très élevé.

4. La parcelle témoin la plus proche de la parcelle 25 est la parcelle 19. Les parcelles témoins les plus proches de la parcelle 14 sont les parcelles 9 et 19.

1	2	3	4	5
6	**7**	8	**9**	10
11	12	13	14	15
16	**17**	18	**19**	20
21	22	23	24	25

FIGURE 2.6 – Réseau de surveillance - grille 5×5 - en gras : les parcelles du réseau de surveillance

	temps	iter	nb éval	nb param	V MC (det)	V LBP*	remarque
références							
politique aléatoire	-	-	-	-	11020	8471.7 (T=20)	
politique uniforme	-	-	-	-	11393	10080 (T=20)	
politique gloutonne	-	-	-	-	13953	12634 (T=20)	
politique 1	-	-	-	-	18911	15182 (T=20)	gradient nul
politique 2	-	-	-	-	18349	13959 (T=20)	gradient nul
départ : politique uniforme							
GD-LBP T=20	43.18s 36s	2	1622	$N = 800$ $N'' = 192$	14427 (13935)	15597 (T=20)	stagnation
GD-LBP T=40	3.8 min 1.18 min	2	1622	$N = 800$ $N'' = 192$	14480 (13940)	14511 (T=40)	stagnation
GD-LBP T=100	14.6 min 3.30 min	2	1622	$N = 800$ $N'' = 192$	14458 (13941)	14362 (T=100)	stagnation
CD-LBP, pas=0.1, T=20	47.25 min 11.87 min	21	12041 1554	$N' = 400$ $N''' = 96$	15055 (14408)	15609 (T=20)	
départ : politique aléatoire							
GD-LBP T=20	4.27 min 1.58 min	7	5626	$N = 800$ $N'' = 192$	14438 (14147)	15549 (T=20)	stagnation

TABLE 2.15 – Résultats sur problème épidémiologique 'réseau de surveillance' (PDMF^3) grille 5x5 - $p = 0.2$ - Les temps de calcul des algorithmes CD-LBP et GD-LBP sont donnés sans prise en compte des symétries de la grille et avec prise en compte des symétries en dessous (le nombre de paramètres N''' est alors inférieur à N').

	temps	iter	nb éval	nb param	V MC (det)	V LBP*	remarque
références							
politique aléatoire	-	-	-	-	9408.5	8471.7 (T=20)	
politique uniforme	-	-	-	-	9534.7	11286 (T=20)	
politique gloutonne	-	-	-	-	13087	11877 (T=20)	
politique 1	-	-	-	-	17386	10221 (T=20)	gradient nul
politique 2	-	-	-	-	16434	12800 (T=20)	gradient nul
départ : politique uniforme							
GD-LBP T=20	50.59s 36.26s	1	1626	$N = 800$ $N'' = 192$	13090 (13088)	14612 (T=20)	gradient nul
CD-LBP, pas=0.1, T=20	14.07min 4.35 min	3	2802 577	$N' = 400$ $N''' = 96$	13094	11877 (T=20)	
CD-MC, pas=0.1, T=40	5.8 jours 2.1 jours	5	4029 960	$N' = 400$ $N''' = 96$	13029 (12876)	12169 (T=40)	
départ : politique aléatoire							
GD-LBP T=20	1.43 min 1.00 min	3	3230	$N = 800$ $N'' = 192$	13090 (13089)	14612 (T=20)	gradient nul

TABLE 2.16 – Résultats sur problème épidémiologique 'réseau de surveillance' (PDMF[3]) grille 5x5 - $p = 0.8$ - Les temps de calcul des algorithmes CD-LBP et GD-LBP sont donnés sans prise en compte des symétries de la grille et avec prise en compte des symétries en dessous (le nombre de paramètres N''' est alors inférieur à N').

2.4.6 Problème de conservation d'une espèce en danger

Nous allons maintenant considérer un problème légèrement plus complexe, celui d'un problème de conservation, inspiré d'un problème réel[5]. Plusieurs mares se situent dans une zone susceptible d'inondation, et contiennent à la fois l'espèce endémique (E) que l'on cherche à conserver et une espèce invasive (I). On notera n le nombre de mares. Lorsqu'il y a une inondation, les mares suffisamment proches sont mises en relation et il y a des déplacements de poissons entre ces mares, dites voisines. Un graphe décrit les liens entre les mares. On note $N(i)$ l'ensemble constitué de la mare i et de ses voisines. L'état d'une mare i est codé par deux booléens : $S_i^t = (S_{iE}^t, S_{iI}^t)$, où $S_{iE}^t = 1$ si l'espèce endémique est présente dans la mare i et 0 sinon, $S_{iI}^t = 1$ si l'espèce invasive est présente dans la mare i et 0 sinon.

Pour modéliser le fait que l'espèce invasive est susceptible de réapparaître facilement, une mare source, virtuelle, est ajoutée aux n mares. Cette mare contient toujours l'espèce invasive (sa dynamique est déterministe et indépendante des états des autres mares). Elle est reliée à toutes les mares avec une certaine probabilité de colonisation. Il y a trois actions possibles sur une mare i donnée : ne rien faire ($a_i = 1$), empoisonner ($a_i = 2$) ou réintroduire l'espèce endémique ($a_i = 3$). On cherche à savoir comment agir de manière à maximiser l'espérance de la présence de l'espèce endémique au cours du temps.

Modèle de dynamique

La probabilité de transition est factorisée. La dynamique de l'espèce invasive est indépendante de celle de l'espèce endémique. Par contre, la dynamique de l'espèce endémique dépend de celle de l'espèce invasive :

$$P(s'|s,a) = \prod_{i=1}^{n} P_{iI}(s'_{iI}|s_{N(i)I}, a_i) P_{iE}(s'_{iE}|s_{N(i)E}, s_{iI}, a_i)$$

La table 2.17 définit les différents paramètres du modèle de dynamique et donne leur valeur (sauf pour les probabilités de déplacement car elles dépendront du graphe considéré). La table 2.18 donne les probabilités locales $P_{iI}(s'_{iI}|s_{N(i)I}, a_i)$ de transition pour l'espèce invasive, où :

— $F^I(s_{N(i)I})$ représente la probabilité que l'espèce invasive soit absente dans la mare i sachant qu'elle était absente au pas de temps précédent et que la mare n'a pas été empoisonnée :

$$F^I(s_{N(i)I}) = \epsilon \prod_{j \in N(i)/s_{jI}=1} (1 - \rho_{ji}^I) + (1 - \epsilon)$$

5. Nous remercions Sam Nicol et Iadine Chadès pour nous avoir permis de nous inspirer de leur travail de modélisation sous forme de PDMG d'un problème de conservation, concernant *Scaturiginichthys vermeilipinnis* (*red-finned blue-eye*), un poisson osseux endémique du Queensland en Australie et menacé de disparition.

paramètre	définition	valeur
ϵ	probabilité d'inondation	0.8
ρ^I_{ji}	probabilité de déplacement de l'espèce invasive de la mare j vers la mare i en cas d'inondation	
$\rho^{E\|I=0}_{ji}$	probabilité de déplacement de l'espèce endémique de la mare j vers la mare i en cas d'inondation si l'espèce invasive est absente de la mare i	
$\rho^{E\|I=1}_{ji}$	probabilité de déplacement de l'espèce endémique de la mare j vers la mare i en cas d'inondation si l'espèce invasive est présente dans la mare i	
m^I	probabilité de mortalité de l'espèce invasive	0.5
$m^{E\|I=0}$	probabilité de mortalité de l'espèce endémique si aucune action n'est faite et que l'espèce invasive est absente	0.5
$m^{E\|I=1}$	probabilité de mortalité de l'espèce endémique si aucune action n'est faite et que l'espèce invasive est présente	1
p^I	probabilité de succès du poison pour l'espèce invasive	0.8
p^E	probabilité de succès du poison pour l'espèce endémique	0.8
$r^{E\|I=0}$	probabilité de succès de la réintroduction de l'espèce endémique si l'espèce invasive est absente	1
$r^{E\|I=1}$	probabilité de succès de la réintroduction de l'espèce endémique si l'espèce invasive est présente	1

TABLE 2.17 – Paramètres du modèle de dynamique dans le problème de conservation

— $P^{I|I=0}(s_{N(i)I})$ représente la probabilité que l'espèce invasive soit absente dans la mare i sachant qu'elle était absente au pas de temps précédent et que la mare a été empoisonnée :

$$P^{I|I=0}(s_{N(i)I}) = F^I(s_{N(i)I}) + (1 - F^I(s_{N(i)I}))p^I$$

— $P^{I|I=1}$ représente la probabilité que l'espèce invasive disparaisse d'une mare suite à un empoisonnement :

$$P^{I|I=1} = (1 - p^I)m^I + p^I$$

La table 2.19 donne les probabilités locales $P_{iE}(s'_{iE}|s_{N(i)E}, s_{iI}, a_i)$ de transition pour l'espèce endémique, où $\forall x \in \{0, 1\}$:

— $F^{E|I=x}(s_{N(i)E})$ représente la probabilité que l'espèce endémique soit absente de la mare i sachant qu'elle était absente au pas de temps précédent, que l'espèce invasive était dans l'état x au pas de temps précédent, et que la mare n'a subi aucune action :

$$F^{E|I=x}(s_{N(i)E}) = \epsilon \prod_{j \in N(i)/s_{jE}=1} (1 - \rho_{ji}^{E|I=x}) + (1 - \epsilon)$$

— $P^{E|E=0,I=x}(s_{N(i)E})$ représente la probabilité que l'espèce endémique soit absente de la mare i sachant qu'elle était absente au pas de temps précédent, que l'espèce invasive était dans l'état x au pas de temps précédent, et que la mare a été empoisonnée :

$$P^{E|E=0,I=x}(s_{N(i)E}) = F^{E|I=x}(s_{N(i)E}) + (1 - F^{E|I=x}(s_{N(i)E}))p^E$$

— $P^{E|E=1,I=x}$ représente la probabilité que l'espèce endémique soit absente d'une mare sachant qu'elle était présente au pas de temps précédent, que l'espèce invasive était dans l'état x au pas de temps précédent et que la mare a été empoisonnée :

$$P^{E|E=1,I=x} = (1 - p^E)m^{E|I=x} + p^E$$

— $R^{E|I=x}(s_{N(i)E})$ représente la probabilité que l'espèce invasive soit présente dans la mare i sachant qu'elle était absente au pas de temps précédent, que l'espèce invasive était dans l'état x au pas de temps précédent et qu'il y a eu une action de réintroduction :

$$R^{E|I=x}(s_{N(i)E}) = r^{E|I=x} + (1 - r^{E|I=x})(1 - F^{E|I=x}(s_{N(i)E}))$$

Modèle de récompense

La récompense est additive et il y a une fonction de récompense par mare : $R(s, a) = \sum_{i=1}^{n} R_i(s_i, a_i)$. La récompense associée à la mare i est de 1 si E est présente, mais de

	$a_i = 1$		$a_i = 2$		$a_i = 3$	
	$s_{iI} = 0$	$s_{iI} = 1$	$s_{iI} = 0$	$s_{iI} = 1$	$s_{iI} = 0$	$s_{iI} = 1$
$s'_{iI} = 0$	$F^I(s_{N(i)I})$	m^I	$P^{I\|I=0}(s_{N(i)I})$	$P^{I\|I=1}$	$F^I(s_{N(i)I})$	m^I
$s'_{iI} = 1$	$1 - F^I(s_{N(i)I})$	$1 - m^I$	$1 - P^{I\|I=0}(s_{N(i)I})$	$1 - P^{I\|I=1}$	$1 - F^I(s_{N(i)I})$	$1 - m^I$

TABLE 2.18 – Probabilités de transition locales pour l'espèce invasive $P_{iI}(s'_{iI}|s_{N(i)I}, a_i)$

	$s_{iI} = 0$		$s_{iI} = 1$	
	$s_{iE} = 0$	$s_{iE} = 1$	$s_{iE} = 0$	$s_{iE} = 1$
$s'_{iE} = 0$	$F^{E\|I=0}(s_{N(i)E})$	$m^{E\|I=0}$	$F^{E\|I=1}(s_{N(i)E})$	$m^{E\|I=1}$
$s'_{iE} = 1$	$1 - F^{E\|I=0}(s_{N(i)E})$	$1 - m^{E\|I=0}$	$1 - F^{E\|I=1}(s_{N(i)E})$	$1 - m^{E\|I=1}$

(a) Cas d'une action nulle : $a_i = 1$

	$s_{iI} = 0$		$s_{iI} = 1$	
	$s_{iE} = 0$	$s_{iE} = 1$	$s_{iE} = 0$	$s_{iE} = 1$
$s'_{iE} = 0$	$P^{E\|E=0,I=0}(s_{N(i)E})$	$P^{E\|E=1,I=0}$	$P^{E\|E=0,I=1}(s_{N(i)E})$	$P^{E\|E=1,I=1}$
$s'_{iE} = 1$	$1 - P^{E\|E=0,I=0}(s_{N(i)E})$	$1 - P^{E\|E=1,I=0}$	$1 - P^{E\|E=0,I=1}(s_{N(i)E})$	$1 - P^{E\|E=1,I=1}$

(b) Cas d'une action d'empoisonnement : $a_i = 2$

	$s_{iI} = 0$		$s_{iI} = 1$	
	$s_{iE} = 0$	$s_{iE} = 1$	$s_{iE} = 0$	$s_{iE} = 1$
$s'_{iE} = 0$	$1 - R^{E\|I=0}(s_{N(i)E})$	$m^{E\|I=0}$	$1 - R^{E\|I=1}(s_{N(i)E})$	$m^{E\|I=1}$
$s'_{iE} = 1$	$R^{E\|I=0}(s_{N(i)E})$	$1 - m^{E\|I=0}$	$R^{E\|I=1}(s_{N(i)E})$	$1 - m^{E\|I=1}$

(c) Cas d'une action de réintroduction : $a_i = 3$

TABLE 2.19 – Probabilités de transition locales pour l'espèce endémique $P_{iE}(s'_{iE}|s_{N(i)E}, s_{iI}, a_i)$

-10 si E est présente et que l'on fait une action de réintroduction :

$$R_i(s_i, a_i) = \begin{cases} 0 & \text{si } s_{iE} = 0 \\ -10 & \text{si } s_{iE} = 1 \text{ et } a_i = 3 \\ 1 & \text{sinon} \end{cases}$$

Structure de la politique

Les mares sont numérotées dans le sens des abscisses (sens horizontal). C'est le sens de parcours des personnes chargées d'inspecter et d'agir sur les mares. Celles-ci vont utiliser une règle de décision basée sur les mares qu'elles ont vu précédemment dans leur parcours.

La probabilité initiale sur les états est uniforme sauf pour la mare source (qui contient toujours l'espèce invasive). Étant donné que les variables d'action ne sont pas binaires, on ne peut utiliser que l'algorithme d'optimisation GD.

Nous comparerons les politiques obtenues par optimisation avec les politiques expertes suivantes (basées au plus sur l'état de la mare considérée) :

1. politique uniforme (point de départ des algorithmes d'optimisation) : elle attribue la même probabilité à chaque action
2. politique 0 : ne rien faire (c'est une des politiques gloutonnes, qui maximise la récompense immédiate)
3. politique 1 : si E est absente, réintroduire ; si E est présente, ne rien faire
4. politique 2 :
   — état 1 (E et I absentes) : réintroduire
   — état 2 (E présente et I absente) : ne rien faire
   — état 3 (E absente et I présente) : empoisonner
   — état 4 (E et I présentes) : empoisonner
5. politique 3 : idem que politique 2 mais dans l'état 4 ne rien faire
6. politique 4 : idem que politique 2 mais dans l'état 4 réintroduire
7. politique 5 : politique uniforme mais sur deux actions au lieu de trois (ne rien faire ou réintroduire dans les états 1 et 2 où I est absente, ne rien faire ou empoisonner dans les états 3 et 4 où I est présente).

Graphe linéaire à 5 mares

Nous avons tout d'abord considéré le cas simple d'un graphe linéaire à 5 mares (voir figure 2.7). L'inondation conduit à un échange de poissons dans les deux sens, mais le sens de parcours des inspecteurs se fait de gauche à droite. Les valeurs de paramètres sont données dans la table 2.17. La probabilité de colonisation est de 0.6065 sauf pour l'espèce endémique en présence de l'espèce invasive où elle est divisée par deux : $\forall (i,j) \in \{1, ...n\}^2, \rho^I_{ji} = 0.6065, \rho^{E|I=0}_{ji} = 0.6065$ et $\rho^{E|I=1}_{ji} = 0.3033$.

Afin de répondre à la contrainte du sens de parcours, nous avons envisagé deux structures de politiques différentes :

— structure A : On décide en une mare en fonction de son état et de l'action prise sur les mares précédentes : $\forall i = 1...n, pa_\delta(A_i) = \{S_i, A_{i-1}, A_{i-2}, ..., A_1\}$; cela fait 1452 paramètres.

— structure S : On décide en une mare en fonction de son état et de l'état des mares précédentes : $\forall i = 1...n, pa_\delta(A_i) = \{S_i, S_{i-1}, S_{i-2}, ..., S_1\}$; cela fait 4092 paramètres.

Nous avons également comparé les résultats obtenus avec ceux obtenus pour une structure de politique de type PDMG (basée sur la structure de la transition et de la récompense, c'est-à-dire sur le graphe), afin de voir si changer l'organisation des 'inspecteurs' pouvait leur permettre d'améliorer la conservation de l'espèce endémique. Nous appellerons cette structure G (avec cette structure, il y a 672 paramètres) : $\forall i = 1...n, pa_\delta(A_i) = S_{N(i)}$.

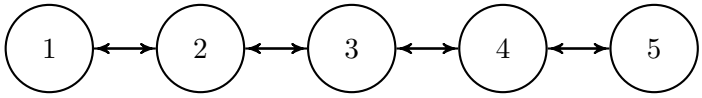


FIGURE 2.7 – Graphe en ligne de 5 mares

La table 2.20 donne les résultats obtenus avec les algorithmes GD-LBP (pour toutes les structures) et MF-API (pour la structure de politique G). On peut constater que MF-API est plus performant en temps et en valeur : il renvoie la politique 2 qui est celle de plus grande valeur (estimée par la méthode de Monte-Carlo) parmi les politiques expertes que nous avons envisagées, et ce en un temps très court. La qualité des politiques renvoyées par GD-LBP dépend de la structure de politique et du point de départ, mais la meilleure est de valeur 11.65, ce qui reste inférieur à la valeur des politiques expertes 1, 2 et 3. Le gradient des politiques 0, 1, 2 et 3 pour la valeur estimée par LBP est proche du vecteur nul, ce qui montre que ce sont des points critiques (mais pas forcément des minima locaux) pour la valeur reparamétrée et estimée par LBP, $V_\theta^{LBP}(P^0)$.

	temps	iter	nb éval	nb param	V MC (det)	V LBP	V MF	remarque
références								
politique uniforme (pu)					-34.54	-31.00		
politique 0 (p0)					4.13	4.17		gradient nul
politique 1 (p1)					18.98	**14.56**		gradient nul
politique 2 (p2)					**24.9**	14.10	22.00	gradient nul
politique 3 (p3)					21.76	13.17		gradient nul
politique 4 (p4)					-123.60	-82.14		
politique 5 (p5)					-19.09	-13.62		
GD-LBP struct A dép. pu	10.02 min	3	5824	$N = 1452$	11.65 (11.63)	9.21		gradient nul
GD-LBP struct S dép. pu	26.45 min	2	12304	$N = 4092$	8.54 (8.46)	7.86		gradient nul
GD-LBP struct A dép. p5	5.24 min	1	2931	$N = 1452$	4.15 (4.18)	4.16		gradient nul
GD-LBP struct S dép. p5	26.11 min	2	12304	$N = 4092$	5.01 (4.95)	5.21		gradient nul
GD-LBP struct G dép. pu	3.15 min	2	2044	$N = 672$	4.15 (4.10)	4.19		proche p0, gradient nul
GD-LBP struct G dép. p5	2.26 min	1	1371	$N = 672$	4.30 (4.14)	4.16		gradient nul
MF-API struct G dép. p0	0.50s	2			**24.9**	14.10	22.00	p2

TABLE 2.20 – Résultats sur le problème de conservation - graphe linéaire à 5 mares

Graphe quelconque à 9 mares

Nous nous sommes ensuite intéressés à une configuration plus réaliste, avec 9 mares réparties de manière aléatoire dans le paysage (voir figure 2.8). Le sens de parcours des inspecteurs est toujours le sens des abscisses, dans l'ordre croissant des numéros de mare. La probabilité de colonisation décroit avec la distance :

$$\forall (i,j) \in \{1...n\}^2,\ \rho_{ji} = \begin{cases} e^{-\alpha D_{ji}} & \text{si } e^{-\alpha D_{ji}} \geq 0.2 \\ 0 & \text{sinon} \end{cases}$$

où $\alpha = 5$ et D_{ji} représente la distance euclidienne entre les mares i et j. Deux mares sont donc considérées comme voisines si elles sont à une distance inférieure à 0.32. La probabilité de colonisation de la mare source vers chacune des mares est de 0.2. Les autres paramètres restent identiques à ceux du problème linéaire à 5 mares (voir table 2.17).

Afin de répondre à cette contrainte du sens de parcours tout en gardant un nombre de paramètres raisonnable, nous avons envisagé deux structures de politiques différentes :

- structure A : On décide en une mare en fonction de son état et des actions prises sur les trois mares précédentes : $\forall i = 1...n, pa_\delta(A_i) = \{S_i, A_{i-1}, A_{i-2}, A_{i-3}\}$; cela fait 2100 paramètres.
- structure S : On décide en une mare en fonction de son état et de l'état des deux mares précédentes : $\forall i = 1...n, pa_\delta(A_i) = \{S_i, S_{i-1}, S_{i-2}\}$; cela fait 1404 paramètres.

Si on envisageait une structure de politique de type PDMG comme dans le précédent problème cela ferait 26496 paramètres, ce qui est trop important. MF-API ne serait pas capable non plus de traiter ce problème, qui est de trop grande taille.

La table 2.21 donne les résultats obtenus avec l'algorithme GD-LBP. A nouveau, la valeur obtenue dans le meilleur des cas est de 37.81, ce qui reste inférieur à la valeur des politiques expertes 1, 2 et 3, qui sont des points critiques de la valeur reparamétrée et obtenue avec l'algorithme LBP, $V_\theta^{LBP}(P^0)$.

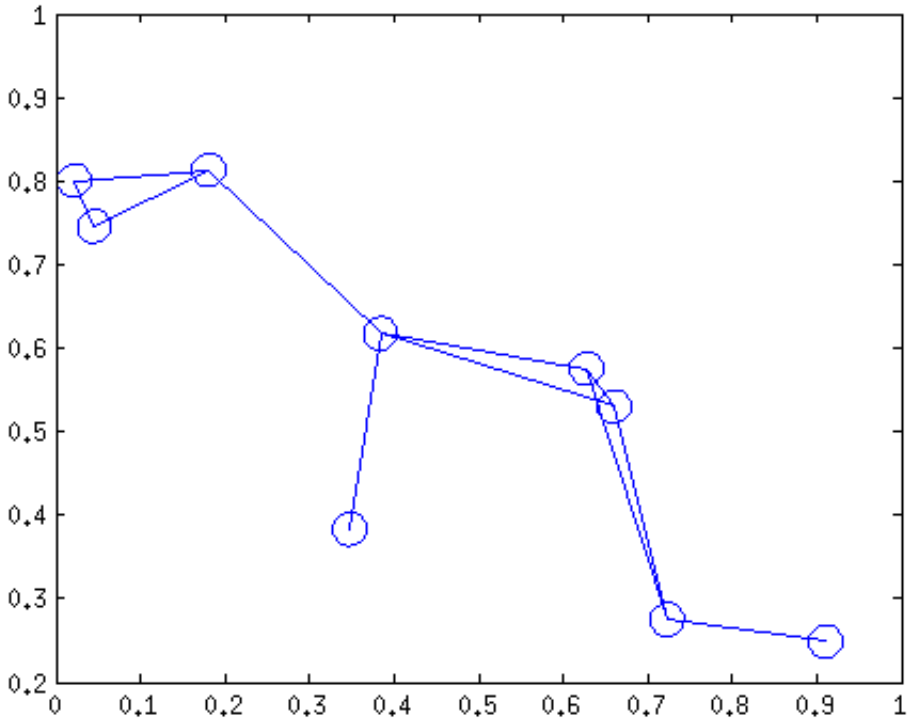


FIGURE 2.8 – Représentation du graphe des 9 mares ; le sens de parcours se fait de gauche à droite, en suivant l'axe des abscisses

	temps	iter	nb éval	nb param	V MC (det)	V LBP	remarque
références							
politique uniforme (pu)					-72.64	-65.83	
politique 0 (p0)					7.16	7.27	gradient nul
politique 1 (p1)					44.70	**32.49**	gradient nul
politique 2 (p2)					**50.24**	32.04	gradient nul
politique 3 (p3)					47.72	31.03	gradient nul
politique 4 (p4)					-90.71	-65.16	
politique 5 (p5)					-108.96	-91.16	
GD-LBP struct A dép. pu	2.68h	4	10527	$N = 2100$	12.09 (12.14)	12.91	gradient nul
GD-LBP struct S dép. pu	1.47h	3	5646	$N = 1404$	8.93 (9.05)	10.38	gradient nul
GD-LBP struct A dép. p5	4.21h	7	16819	$N = 2100$	37.81 (37.77)	25.43	gradient nul
GD-LBP struct S dép. p5	2.16h	5	8440	$N = 1404$	26.18 (26.27)	17.34	gradient nul

TABLE 2.21 – Résultats sur le problème de conservation - graphe quelconque à 9 mares

2.4.7 Bilan des résultats

Nous avons testé les algorithmes de résolution de PDMF^3, en comparaison avec une résolution exacte, un algorithme de résolution pour PDMGs ou bien des politiques 'expertes', sur des problèmes de complexité croissante. L'ensemble des résultats expérimentaux montre que la méthode à utiliser dépend du problème considéré :

- pour un PDMG, il vaut mieux utiliser MF-API, qui est plus rapide et donne des résultats similaires ou supérieurs à CD-LBP ou GD-LBP dans les expériences effectuées
- pour un PDMF^3 général à variables d'action binaires : le mieux est d'essayer les deux algorithmes CD-LBP et GD-LBP, avec des points de départ différents, car l'un ou l'autre peut conduire à de meilleurs résultats
- pour un PDMF^3 général à variables d'action non binaires : utiliser GD-LBP avec des points de départ différents. C'est le seul algorithme existant pour traiter de tels problèmes.

Dans les deux derniers cas, il peut être intéressant de déterminiser la politique obtenue en sortie car elle peut être de meilleure valeur.

Une utilisation raisonnable de CD-LBP et GD-LBP est de les appliquer dans une boucle itérative politique experte (point de départ) - optimisation - interprétation par un expert. On peut aussi envisager de combiner les différents algorithmes :

1. utiliser GD-LBP pour se rapprocher rapidement d'un optimum local
2. utiliser CD-LBP pour améliorer la valeur courante (si les variables d'action sont binaires)
3. utiliser CD-MC ou GD-MC pour optimiser une estimation asymptotiquement non biaisée de la valeur.

2.5 Positionnement par rapport à l'état de l'art

Nous avons donc proposé d'une part un nouveau cadre de PDM à espace d'état et d'action factorisés, caractérisé par la recherche de politiques stochastiques factorisées de structure donnée, ainsi qu'une famille d'algorithmes de résolution associés, de type itération de la politique approchée. Dans le cas général de variables d'action non binaires, l'algorithme d'optimisation vers lequel nous nous sommes orientés est un algorithme de descente de gradient. D'autres auteurs ont utilisé cette approche dans le domaine de l'apprentissage par renforcement (sans hypothèse sur la fonction de transition). Ainsi, [PKMK00] a proposé un algorithme de descente de gradient pour un sous-cadre de Dec-POMDP à horizon infini. Les politiques cherchées sont factorisées et pour chaque agent la politique est représentée par un contrôleur à états finis stochastique de taille donnée. Par contre, l'espace d'état n'est pas factorisé. Le gradient est calculé par simulation. Notre cadre peut être vu comme un cas particulier du cadre Dec-POMDP (voir section 2.1.3). Un résultat théorique de l'article serait donc vrai dans notre cas si on ne faisait pas d'approximation de la valeur et du gradient : 'tout équilibre de Nash strict est un optimum local pour la descente de gradient' (mais la réciproque n'est pas forcément

vraie). Un équilibre de Nash strict est une politique dont aucun agent ne peut dévier individuellement sans faire décroître la valeur.

[BA09] a plus récemment proposé l'algorithme *facored policy-gradient planner* (FPG). L'espace d'état et d'action peuvent être grands (jusqu'à $|\mathcal{S}| = 2^{250}$ et $|\mathcal{A}| = 2^{500}$), et les variables d'état et d'action peuvent être non binaires (les variables d'état peuvent même être continues). Une notion de ressource réelle peut notamment être prise en compte (temps, argent...). Le critère considéré est le critère moyen, et une extension au critère γ-pondéré n'est pas évidente. En théorie, l'algorithme peut chercher des politiques stochastiques factorisées aussi générales que les nôtres. Cependant, en pratique dans l'article, l'hypothèse est faite que tous les 'agents' partagent la même observation (ce qui revient à dire dans notre cadre que les parents de toutes les variables d'action sont les mêmes). De plus, une hypothèse supplémentaire de paramétrisation de la politique est faite, au choix parmi :

— approximateur linéaire : il y a alors un paramètre par couple (agent,observation) soit $\dim(\Omega) \times n$ paramètres réels (tandis que dans notre cas il y en a $|\Omega| \times n$)

— arbre d'experts : pour chaque action, la politique est un arbre dont certains nœuds sont des règles de décision déterministes expertes, et d'autres sont des règles de décision stochastiques paramétrées ; il y a donc $\dim(\Omega) \times d \times n$ paramètres, où d, le nombre de nœuds de décision stochastiques, est en général petit.

L'espace de recherche est donc plus restreint que le nôtre, ce qui en théorie peut amener à obtenir des politiques de moins bonne qualité (mais permet de traiter des problèmes de plus grande taille). Nous pourrions avec notre approche traiter ces choix de paramétrisation, mais à condition de 'remplir les tables' de la politique (cette hypothèse nous permet d'utiliser la méthode d'évaluation basée sur l'utilisation d'algorithmes d'inférence). Enfin, dans FPG le gradient est estimé par simulations (l'algorithme est donc un algorithme de gradient stochastique).

Parmi les autres approches récentes, [KZT11] applique l'algorithme EM de [TS06] aux Dec-POMDPs à espace d'état factorisé, mais avec hypothèse de fonction de valeur additive ($V_\pi(s^0)$ doit s'écrire comme une somme de fonctions de faible arité sur les variables décrivant s^0), ce qui est valable seulement pour certains cadres de Dec-POMDPs. L'algorithme EM obtenu est parallélisable et s'applique que l'horizon soit fini ou infini. La politique est représentée grâce à des contrôleurs à états finis. Contrairement à notre approche, c'est le problème d'optimisation dans son ensemble qui est vu comme un problème d'inférence, et non seulement l'étape d'évaluation. [PP11], sans hypothèse sur la fonction de valeur, propose un algorithme EM approché pour le cas de l'horizon infini, mais les problèmes résolus n'ont pas plus de 10 agents. Autant que nous le sachions, aucune approche de planification par inférence n'a été proposée qui permette de résoudre des PDMF-AF généraux avec autant de variables que ce que nous considérons dans nos expérimentations numériques.

Enfin, [OWS13] a proposé une approche de résolution pour les Dec-POMDPs à espace d'état factorisé et à horizon fini, basée sur les jeux bayésiens graphiques collaboratifs. Le nombre d'agents peut être très grand mais l'horizon est limité ($T = 3$ ou $T = 6$). L'algorithme *factored frontier* (voir [MW01]) est utilisé pour résoudre un problème

d'inférence, mais celui-ci conduit à une approximation plus importante que LBP.

2.6 Perspectives

Perspectives sur le cadre

Nous avons choisi de considérer des politiques stochastiques factorisées stationnaires (voir section 2.1.2). Or la meilleure PSF n'est pas forcément stationnaire, même à horizon infini. Notre approche pourrait s'étendre au cas des politiques stochastiques factorisées non stationnaires (mais de structure stationnaire). Le nombre de paramètres d'optimisation serait multiplié par l'horizon du problème.

Le cadre que nous avons proposé, le cadre PDMF^3, prend déjà en compte en partie le cas d'une observabilité partielle, via la structure de la politique. Mais nous pourrions étendre facilement nos algorithmes au cas général des Dec-POMDPs à espace d'état factorisé, en ajoutant des variables d'observation pour lesquelles la probabilité d'observation est factorisée. Le principe de l'approche serait inchangé, à condition que le graphe représentant les dépendances entre variables issues des structures de la transition, de la politique *et* de l'observation soit acyclique.

Nous pourrions également envisager de prendre en compte les indépendances contextuelles dans la transition et la récompense ; par exemple, si on utilisait des diagrammes de décision algébriques ou des tables creuses, il serait possible d'utiliser l'algorithme de [GD13] (*structured message passing*) pour l'évaluation, ce qui permettrait des économies en temps et en mémoire.

Perspectives sur les algorithmes

Pour améliorer les algorithmes de résolution que nous avons proposés, nous pourrions envisager une approche multifidélité [ALG$^+$99], c'est-à-dire utiliser à la fois une évaluation basse fidélité, de mauvaise qualité mais rapide (celle basée sur le calcul approché de marginales), et une évaluation haute fidélité, de meilleure qualité mais plus lente (basée sur la méthode Monte-Carlo), à laquelle l'algorithme d'optimisation ferait appel ponctuellement.

Nous pourrions aussi envisager d'évaluer et optimiser la PSF en même temps (comme en apprentissage par renforcement), en s'arrêtant avant convergence dans l'algorithme LBP ou en faisant peu de simulations dans le cas de l'évaluation par la méthode de Monte-Carlo.

Perspectives d'élargissement du problème

Enfin, nous pourrions considérer des problèmes pour lesquels on cherche à optimiser également la structure de la politique. Il faudrait alors étudier la complexité de ce nouveau problème et proposer un algorithme d'optimisation mixte, puisque l'on aurait à la fois des variables d'optimisation discrètes (représentant la structure de la politique, avec des contraintes de parcimonie), et des variables continues (représentant la politique stochastique factorisée). Cela aurait un intérêt par exemple pour les problèmes dans lesquels il n'y a pas de contrainte particulière sur la structure de la politique, et pour

lesquels la structure ‘naturelle’ que nous avons proposée conduit à un trop grand nombre de variables.

Chapitre 3

Application à un problème d'agroécologie à l'échelle du paysage

L'objectif de ce chapitre est d'illustrer l'intérêt du cadre PDMF^3 et des algorithmes de résolution associés (voir chapitre 2) pour la résolution à l'échelle du paysage de problèmes complexes de compromis entre services écosystémiques [Ass05] ou de conflits entre acteurs, intéressés par différents services. Nous nous appuierons pour cela sur un cas d'étude particulier : celui d'un conflit potentiel entre agriculteurs, apiculteurs et citoyens dans un paysage de grandes cultures autour des services fournis par les cultures, les plantes adventices et les pollinisateurs (voir section 3.4).

Après avoir introduit les enjeux agroécologiques actuels (sections 3.1 à 3.3) et avoir présenté la modélisation du cas d'étude considéré sous forme de PDMF^3 (sections 3.4 à 3.6), nous présenterons les résultats obtenus sur ce problème (voir section 3.7) : nous comparons les stratégies *land sparing* et *land sharing* de la littérature et proposons une méthode pour obtenir des stratégies intermédiaires permettant d'atteindre de meilleurs compromis. Nous donnerons enfin des perspectives à ce travail, pour l'instant illustratif, pour aller vers une utilisation du modèle comme support à la réflexion sur la fourniture de services écosystémiques dans les paysages de grandes cultures (section 3.8).

3.1 Vers une agriculture plus durable

Après la seconde guerre mondiale, l'agriculture en France et dans les pays développés en général, s'est construite dans un objectif de production. Cela a conduit à une intensification de l'agriculture, une extension des monocultures, une utilisation d'intrants (pesticides, fertilisants) dont l'effet est néfaste pour l'environnement, et une chute de la biodiversité [TKK$^+$05]. Face à ce constat, l'agroécologie (qui a été évoquée pour la première fois dans les années 1930 [Ben28]), s'est développée dans les années 1970, à la fois comme mouvement socio-politique de terrain et comme science [WBD$^+$09].

L'agroécologie repose sur la valorisation de processus écologiques (notamment les régulations biologiques) afin de diminuer l'utilisation des intrants chimiques. Elle donne par conséquent une place importante à la biodiversité et fait intervenir le concept de service écosystémique. Ce concept, c'est-à-dire la formalisation des bénéfices que l'humain peut tirer de la biodiversité, a été popularisé par le *Millenium Ecosystem Assessment* [Ass05] dans les années 2000. Celui-ci a classé les services écosystémiques en quatre groupes (voir figure 3.1) :

1. Les services d'approvisionnement : eau, production agricole, bois, énergie...
2. Les services de régulation : protection contre les catastrophes naturelles, contre la pollution de l'eau, pollinisation...
3. Les services culturels : esthétiques, récréationnels, éducatifs...
4. Les services de support : photosynthèse, production primaire... (ce sont les services nécessaires pour la délivrance des trois autres types de services).

Si cette notion est critiquée pour véhiculer une vision utilitaire de la biodiversité [Mar14], elle présente l'avantage pour certains de valoriser la biodiversité, de mettre en valeur son intérêt pour l'homme.

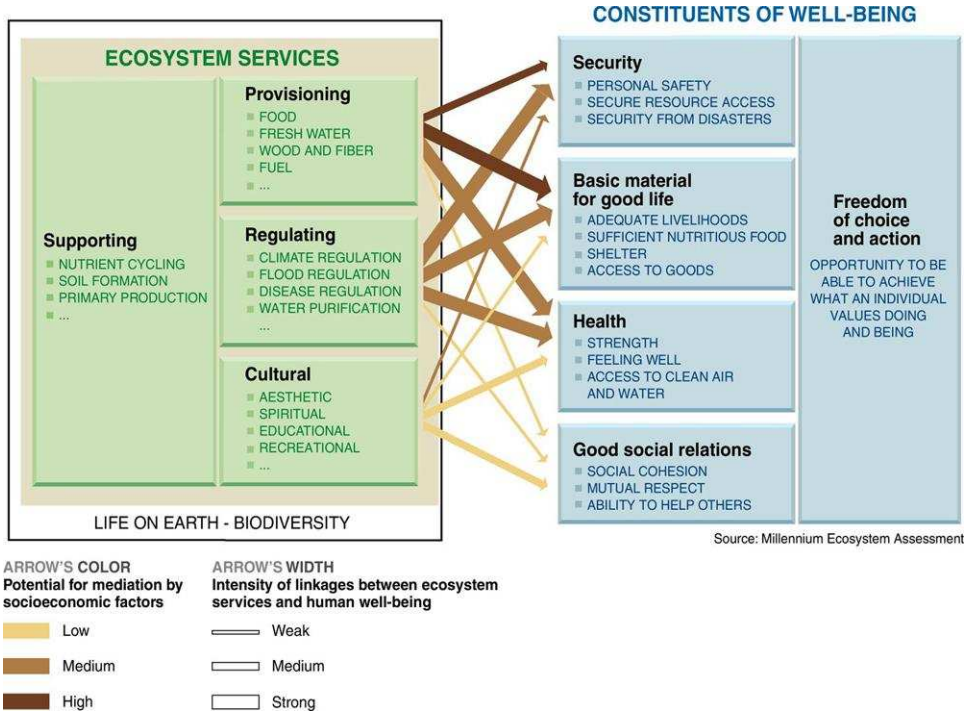


FIGURE 3.1 – Figure tirée de [Ass05] illustrant les liens entre biodversité, services écosystémiques et bien-être humain

Les relations entre services écosystémiques sont complexes et parfois mal connues [RBB+06, BPG09]. Les différents services écosystémiques peuvent être en synergie. Par

exemple, une augmentation de la provision du service de pollinisation entraîne une augmentation de la provision du service de production des cultures à fleurs. Mais il peuvent aussi être en conflit. Par exemple, dans le cas d'une utilisation de fertilisants, la production agricole augmente mais la qualité de l'eau diminue. Ces situations de conflit ou de synergie entre services peuvent être dues à une interaction directe entre services (cas du premier exemple) ou à un facteur extérieur influençant les deux services (l'utilisation de fertilisants dans le deuxième exemple) [BPG09]. Concevoir des systèmes agricoles permettant de fournir plusieurs services écosystémiques, et pas seulement des services de production, reste donc un challenge [GLB+15]. De plus, les différents acteurs (producteurs, consommateurs, citoyens...) sont intéressés par différents services écosystémiques et ne leur accordent pas forcément la même valeur, ce qui peut les mettre également en opposition ou en synergie pour atteindre leurs objectifs.

3.2 Comment atteindre un compromis entre production et biodiversité à l'échelle de la mosaïque paysagère ?

Dans la littérature en écologie théorique, deux principales stratégies font débat pour l'agencement spatial des paysages agricoles dans le but d'un compromis production-biodiversité [GCSB05]. La première stratégie, appelée *land sparing*, consiste à obtenir des rendements importants sur les parcelles cultivées de manière à 'économiser de la terre' pour des habitats semi-naturels dans lesquels une biodiversité importante va pouvoir se maintenir. La seconde stratégie, appelée *land sharing* (ou *wild-life friendly farming*), consiste à utiliser des pratiques plus respectueuses de la biodiversité dans les parcelles agricoles quitte à avoir de plus faibles rendements, mais mieux répartis dans le paysage (voir [GCSB05]). Autrement dit, la stratégie *land sparing* est une stratégie de spécialisation tandis que la stratégie *land sharing* est une stratégie d'intégration. Comme le montre [FBD+08] (voir aussi figure 3.2), tout un *continuum* de stratégies est possible entre ces deux stratégies extrêmes. Plusieurs auteurs se sont attachés à choisir entre ces deux stratégies grâce à une analyse expérimentale ou un modèle dans une situation donnée (voir par exemple [MDGMJ07, EM12, HKT+10]), mais ce cadre demande encore à être amélioré [FAB+14, vWAB+14]. Il ne prend pas en compte par exemple la dynamique temporelle des assolements. Or, des systèmes avec le même usage des terres mais des histoires différentes ne supportent pas la même biodiversité. De plus, les rendements agricoles peuvent varier au cours du temps et dépendre de la rotation des cultures.

La plupart des travaux sur les services écosystémiques consistent à cartographier les services dans le temps pour différents scénarios de gestion (voir par exemple [NMR+09, BHM+13]). Plusieurs auteurs se sont cependant déjà intéressés à la question de l'optimisation de la provision de services écosystémiques à l'échelle du paysage (voir par exemple [DLP09, PCB14]), mais souvent sans prendre en compte l'aspect temporel (voir par exemple [CSC+06, PNL+05, BAD08, GRC14, PLPN14]).

Dans ce chapitre, nous poserons l'hypothèse que les conflits entre services écosystémiques ou entre acteurs (intéressés par différents services écosystémiques) ne peuvent se résoudre qu'à l'échelle du paysage agricole. En effet, certaines espèces vivant dans

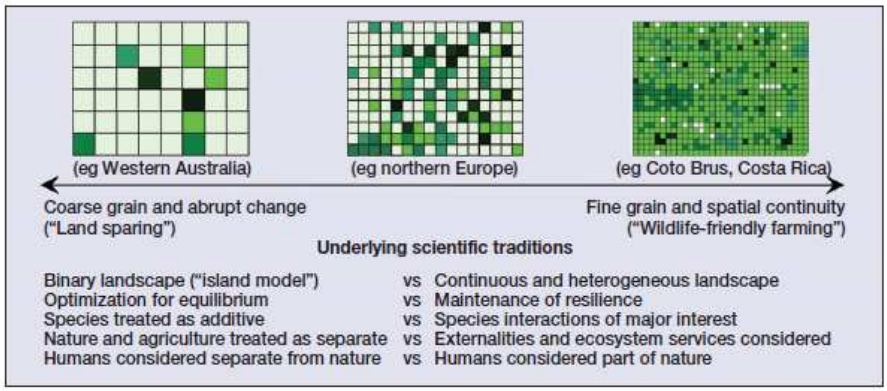


FIGURE 3.2 – Figure tirée de [FBD+08] illustrant le *continuum* entre les stratégies *land sparing* et *land sharing*

les agroécosystèmes (oiseaux, pollinisateurs...) se déplacent, et certains processus écologiques s'expriment à l'échelle du paysage. De ce fait, les services écosystémiques autres que le service de production (service culturel de conservation de la biodiversité, service de pollinisation...) s'évaluent à l'échelle du paysage plutôt que de la parcelle.

En procédant par optimisation plutôt que par évaluation-comparaison de stratégies expertes, nous espérons aboutir à des stratégies nouvelles, impensées par les experts car trop éloignées des pratiques habituelles. L'utilisation des processus décisionnels de Markov nous permettra de prendre en compte la dynamique temporelle des assolements [TPA+13]. L'analyse des stratégies obtenues nous permettra de répondre à des questions comme : 'est-ce qu'avec ces stratégies, les parcelles sont spécialisées dans l'espace (c'est-à-dire plutôt dédiées à la production ou plutôt dédiées à la protection de la biodiversité, comme dans l'approche *land sparing*) ?' ou 'est-ce qu'avec ces stratégies les parcelles sont spécialisées dans le temps ?'.

3.3 Les adventices au cœur d'un conflit potentiel entre production et conservation de la biodiversité

Les adventices sont les plantes sauvages des milieux agricoles (communément appelées mauvaises herbes [God84]). Celles-ci présentent des fonctions antagonistes. En effet, elles sont à la fois en compétition pour les ressources (lumière, nutriments...) avec la culture [Oer06], et bénéfiques à la conservation de la biodiversité puisqu'à la base de réseaux trophiques dans les agroécosystèmes [MBB+03].

C'est parce qu'elles sont en compétition pour les ressources avec la culture, qu'en agriculture intensive on cherche jusqu'à présent à les éradiquer, via notamment l'utilisation d'herbicides. Cependant, les herbicides sont néfastes pour l'environnement et la santé, c'est pourquoi il y a une certaine pression sociale et politique pour limiter leur utilisation (plan Ecophyto). Les changements dans l'agriculture (utilisation de cultures

d'hiver, d'herbicides, prédominance des terres cultivées etc.) ont entraîné une diminution dans l'abondance des adventices mais aussi de leur stock semencier, et une modification de la composition des communautés d'espèces adventices [MBB+03, FCX08, FKG12].

Le déclin des adventices explique en partie celui des populations d'oiseaux et d'insectes [MBB+03], et en particulier des pollinisateurs [GVBB+13]. Cette prise de conscience récente du rôle de support de la biodiversité joué par les adventices a conduit plusieurs auteurs à proposer de prendre en compte le rôle complexe des adventices dans les agrosystèmes et de les gérer de manière plus durable [GBHT03, PBLG+11]. Un exemple du rôle complexe des plantes adventices : elles sont une ressource pour les pollinisateurs entre les pics de floraison du colza et du tournesol [ROT+15]. Or, la pollinisation entomophile permet une augmentation de la production des cultures de colza ou de tournesol [SKS13, CGODJ11], permettant ainsi une augmentation du rendement. Par ce rôle dans le processus de pollinisation, les plantes adventices seraient bénéfiques à la production agricole. Un autre exemple : elles sont une ressource trophique pour des oiseaux et des petits mammifères, qui peuvent permettre la régulation des maladies de la plante cultivée [Alt99a]. Mais certaines adventices peuvent être aussi des vecteurs ou des réservoirs de pathogènes. Les espèces adventices sont donc très variées, et certains auteurs ont tenté de les classifier en fonction de leur réponse à l'environnement et/ou de leur impact sur l'environnement [Sto06, SMC10].

Les adventices sont ainsi au cœur d'un conflit entre production agricole et conservation de la biodiversité dans les agroécosystèmes. Dans la suite, nous nous intéresserons à une partie de ce conflit, en prenant en compte leur impact sur le rendement et leur qualité de ressource trophique pour les pollinisateurs.

3.4 Présentation du cas d'étude : le problème Cultures-Adventices-Pollinisateurs (CAP)

Nous illustrerons la résolution par optimisation de conflits entre services écosystémiques autour des adventices sur un problème d'interactions cultures-adventices-pollinisateurs dans un paysage agricole. Ce problème est décrit en détails dans [BG15] (voir aussi figure 3.3). On considérera trois cultures : blé, colza et prairie (luzerne en mesure agri-environnementale). Dans les luzernes en mesure agri-environnementale, la fauche a lieu après floraison, et la luzerne est une ressource pour les abeilles domestiques [ROT+15].

Les abeilles domestiques se nourrissent principalement des fleurs de colza (au printemps) et de tournesol (en été). En dehors des périodes floraison de ces deux cultures, leurs ressources florales sont les plantes adventices que l'on trouve dans les cultures de céréales (par exemple le blé) et dans les prairies ou les luzernes [ROT+15]. Dans cette étude, nous n'avons pas considéré le tournesol et posons l'hypothèse que la luzerne constitue la ressource alimentaire principale en été. Les abeilles domestiques ne contribuent pas à la reproduction des adventices (elles ne pollinisent pas les adventices, même si elles s'en nourrissent).

Les pollinisateurs sauvages, quant à eux, se nourrissent principalement des adventices présentes dans les cultures ou les habitats semi-naturels [RBD+13] et contribuent à

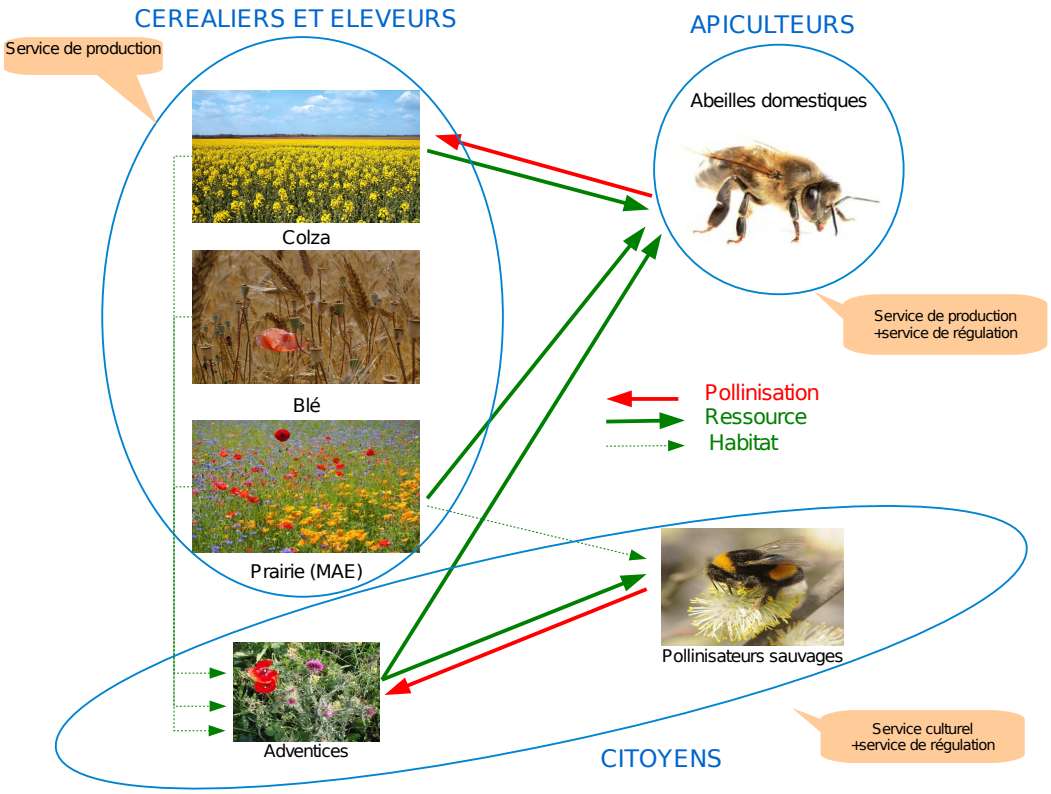


FIGURE 3.3 – Représentation schématique des conflits et synergies entre services/acteurs dans le problème Cultures-Adventices-Pollinisateurs (CAP)

la reproduction de ces adventices. Leur habitat se situe en général dans les bordures, prairies, bois et autres habitats semi-naturels. Dans le modèle que nous considérons, leur habitat est les prairies, puisque c'est le seul habitat semi-naturel que nous prenons en compte.

Les adventices ont pour habitat aussi bien les cultures de colza et de blé que les prairies, dans lesquelles elles sont souvent plus abondantes et diverses. Elles contribuent au maintien des pollinisateurs sauvages et domestiques dans le paysage agricole.

Les deux céréales peuvent se reproduire même en l'absence de pollinisateurs. Cependant, la pollinisation du colza par les abeilles domestiques augmenterait le rendement du colza de 30 à 40% par rapport à une fécondation par le vent [SKS13]. Certaines études tendent à montrer un rôle des pollinisateurs sauvages dans la pollinisation des cultures à fleurs [GSDW$^+$13], mais il n'y a pas de consensus. Nous supposerons donc que les pollinisateurs sauvages ne pollinisent pas le colza. Il peut y avoir un effet bénéfique indirect des pollinisateurs sauvages et des adventices sur le rendement du colza (les pollinisateurs sauvages favorisent les adventices, qui permettent de maintenir les abeilles domestiques, qui à leur tour augmentent le rendement du colza). Mais les adventices peuvent aussi réduire le rendement du blé et du colza, il y a donc des questions de compromis.

L'intérêt des céréaliers et des éleveurs est de maintenir une production suffisante de manière stable dans le temps. Ils auraient donc intérêt à maintenir les abeilles domestiques qui pollinisent les cultures en maintenant les adventices sans toutefois que ces dernières n'induisent des pertes de production trop importantes. L'intérêt des apiculteurs est de maintenir une production de miel au cours du temps, donc l'abondance d'abeilles domestiques. Enfin, dans le territoire agricole, les citoyens accordent un poids plus important aux services culturels, tels que la conservation de la biodiversité et la présence d'habitats semi-naturels, sources de loisirs et de tourisme.

On peut se demander comment allouer les cultures (blé/colza/prairie) dans l'espace et dans le temps de manière à concilier les objectifs des différents acteurs. Autrement dit, comment concilier services de production (rendement en céréales, en fourrage et en miel), services de régulation (régulation des populations d'espèces adventices et de pollinisateurs sauvages) et services culturels (conservation d'espèces adventices et pollinisateurs sauvages). Pour répondre à cette question, nous nous appuierons sur le cadre PDMF^3 présenté au chapitre 2.

3.5 Rappel du cadre PDMF^3

La figure 3.4 rappelle de manière schématique le cadre PDMF^3. $S^t \in \mathcal{S}$ représente l'état du système au temps t et $A^t \in \mathcal{A}$ représente l'action permettant d'agir sur le système au temps t. Les espaces d'état et d'action sont factorisés : $\mathcal{S} = \prod_{i=1}^{n} \mathcal{S}_i$, $\mathcal{A} = \prod_{j=1}^{m} \mathcal{A}_j$, c'est-à-dire que l'état du système et l'action sont décrits par plusieurs variables. $P(S^{t+1}|S^t, A^t)$ représente la probabilité que le système passe de l'état S^t à l'état S^{t+1} sous l'action A^t. Elle doit s'écrire comme un produit de probabilités de transition pour chaque variable d'état S_i^{t+1}, ne dépendant que d'un sous-ensemble des variables

contenues dans (S^t, A^t), noté $pa_P(S_i^{t+1})$:

$$P(S^{t+1}|S^t, A^t) = \prod_{i=1}^{n} P_i(S_i^{t+1}|pa_P(S_i^{t+1}))$$

Nous utiliserons dans ce chapitre une définition de la récompense un peu plus générale que celle donnée dans le chapitre 2 : $R(S^t, A^t, A^{t-1})$ représente la récompense réelle obtenue au pas de temps t si l'état du système est S^t, l'action prise est A^t et l'action prise au pas de temps précédent est A^{t-1}. Le modèle de récompense doit être additif, c'est-à-dire que la récompense doit s'écrire comme une somme de fonctions de récompenses R_α faisant intervenir un petit sous-ensemble des variables contenues dans (S^t, A^t, A^{t-1}), noté $pa_R(R_\alpha)$:

$$R(S^t, A^t, A^{t-1}) = \sum_{\alpha=1}^{r} R_\alpha(pa_R(R_\alpha))$$

P^0 est la distribution initiale sur les états du système, T est l'horizon de temps considéré.

Une politique (factorisée stationnaire stochastique) δ est une fonction de $\mathcal{A} \times \mathcal{S}$ dans $[0; 1]$ qui se factorise sous forme de produit :

$$\delta(a|s) = \prod_{j=1}^{m} \delta_j(a_j|pa_\delta(a_j))$$

$\delta(a|s)$ représente la probabilité de choisir l'action $a \in \mathcal{A}$ si le système est dans l'état $s \in \mathcal{S}$; $\delta_j(a_j|pa_\delta(a_j))$ représente la probabilité de choisir l'action a_j pour la variable d'action A_j si un certain nombre de variables d'état et d'action sont dans l'état joint noté $pa_\delta(a_j)$. Ces variables sont définies par ce que l'on appelle la structure de la politique, et représentent les informations à partir desquelles on peut décider de la variable d'action A_j.

Nous considérons un horizon de temps fini T. L'algorithme de résolution GD-LBP (voir section 2.3.4) permet de calculer de manière approchée une politique factorisée stochastique qui maximise l'espérance de la somme des récompenses obtenues au cours du temps jusqu'à l'horizon de temps $T - 1$ (appelée valeur de la politique δ) :

$$V_\delta(P^0) = \mathbb{E}\left[R(S^0, A^0) + \sum_{t=1}^{T-1} R(S^t, A^t, A^{t-1}) \middle| P^0, \delta\right]$$

3.6 Modélisation du problème CAP sous forme de PDMF[3]

Les paramètres de dynamique seront supposés connus. Nous nous intéresserons à l'effet de l'objectif que l'on se donne (associé à un modèle de récompense, mesurant par exemple la marge économique ou la biodiversité), sur la politique optimale d'allocation des cultures obtenue avec l'algorithme de résolution GD-LBP, et aux trajectoires de paysages auxquels elle conduit.

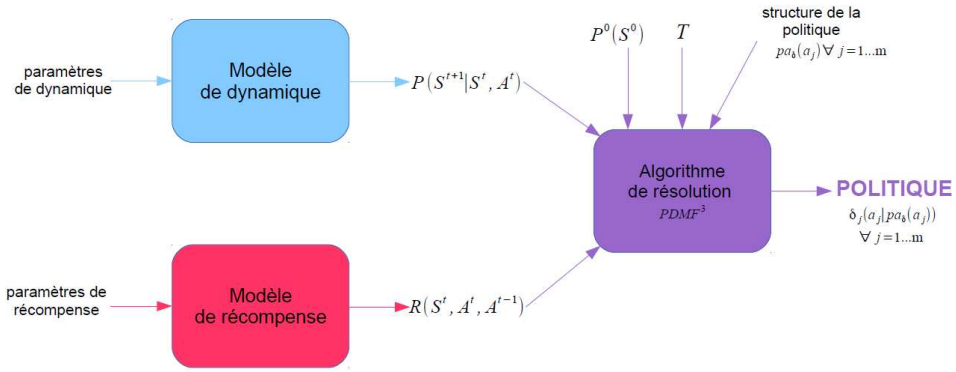


FIGURE 3.4 – Représentation schématique du cadre théorique utilisé : le cadre PDMF^3

Dans un premier temps, un travail de modélisation dans le cadre PDMF^3 a permis de formaliser la dynamique des adventices et des pollinisateurs et de caractériser des fonctions de récompense pour des objectifs de production, de maintien de la biodiversité ou de compromis entre les services.

3.6.1 Hypothèses

Nous considérons un paysage théorique de type parcellaire dans une zone d'agriculture conventionnelle, avec des prairies en mesure agri-environnementales. Le parcellaire est plus précisément un carré de k parcelles par k parcelles, avec un total de $P = k^2$ parcelles. Nous nous intéressons uniquement aux prairies comme habitats semi-naturels, nous ne prenons pas en compte les autres habitats semi-naturels (bordures, bois...). Nous supposons que l'abeille domestique est généraliste, tandis que les pollinisateurs sauvages sont spécialistes dans leur utilisation de la ressource florale [RBD$^+$13]. Nous supposons aussi qu'il y a une source de pollinisateurs sauvages proche du paysage considéré, et que si les conditions redeviennent favorables, ceux-ci peuvent réapparaître. Nous ne prenons pas en compte la compétition entre adventices, ni entre pollinisateurs sauvages et domestiques.

Les paramètres du modèle ont été validés par des experts, et ont pu être appuyés dans certains cas par une étude bibliographique. Ils ont été choisis de manière à respecter les hiérarchies de rentabilité, de qualité d'habitat etc. des différentes cultures considérées (colza, blé, prairie).

3.6.2 Description des variables d'état

L'état du système (le paysage) l'année t est décrit par la présence/absence de $I = 5$ groupes d'espèces adventices sur chaque parcelle : $S^t = \{S^t_{pi}, p = 1...P, i = 1...I\}$; i correspond à l'indice du groupe et p à l'indice de la parcelle, $S^t_{pi} \in \{0, 1\}$. Il y a donc $n = PI$ variables d'état, et $\forall i = 1...n,\ \mathcal{S}_i = \{0, 1\}$. C'est donc sur les adventices que

porte le modèle de dynamique. Il y a 4 groupes de fleurs (dicotylédones) et un groupe d'herbes (monocotylédones).

On considére les I groupes adventices comme identiques en termes de réponse à l'écosystème (mêmes paramètres de dynamique). La survie de ces groupes adventices est considérée comme identique dans les parcelles de blé et de colza, et plus importante dans les parcelles de luzerne (prairie). En effet, dans les prairies, il n'y a ni travail du sol ni traitement herbicide.

A chaque groupe adventice sont associés une saison de floraison et un niveau de nuisibilité :

— groupe 1 : fleurs qui fleurissent au printemps-été avec un impact fort sur le rendement
— groupe 2 : fleurs qui fleurissent au printemps-été avec un impact faible sur le rendement
— groupe 3 : fleurs qui fleurissent toute l'année avec un impact faible sur le rendement
— groupe 4 : fleurs à émergence printanière stricte avec un impact fort sur le rendement
— groupe 5 : herbes avec un impact fort sur le rendement.

Ces 5 groupes représentent les grands groupes fonctionnels des espèces adventices présentes dans les agroécosystèmes en France.

Nous considérons, en plus de l'abeille domestique, 4 groupes de pollinisateurs sauvages. Nous avons donc $B = 5$ groupes de pollinisateurs. On se base sur l'hypothèse que l'abeille domestique est généraliste tandis que les pollinisateurs sauvages sont spécialistes dans leur utilisation de la ressource florale [RBD$^+$13] pour proposer un réseau trophique cohérent décrivant les relations entre adventices et pollinisateurs (voir figure 3.5).

On note $RT(b)$ l'ensemble des groupes de fleurs adventices butinées par le pollinisateur sauvage b : $RT(1) = \{1\}, RT(2) = \{2\}, RT(3) = \{3\}, RT(4) = \{4\}$. On note $RT^{-1}(i)$ l'ensemble des pollinisateurs sauvages qui butinent les espèces du groupe adventice i : $RT^{-1}(1) = \{1\}, RT^{-1}(2) = \{2\}, RT^{-1}(3) = \{3\}, RT^{-1}(4) = \{4\}, RT^{-1}(5) = \emptyset$.

L'abondance des pollinisateurs sauvages et domestiques est modélisée de manière déterministe à partir de la présence-absence des groupes adventices et des cultures (voir section 3.6.4).

3.6.3 Description des variables d'action

L'action globale sur le paysage l'année t est notée $A^t = \{A^t_p, p = 1...P\}$. Il y a donc $m = P$ variables d'action. Pour $j = 1, ..., m$, $\mathcal{A}_j = \{$blé, colza, prairie$\}$.

On recherche une politique stationnaire (qui ne varie pas au cours du temps), qui va conduire à la meilleure évolution de la composition et de la configuration du paysage pour un objectif donné. Cette politique, éventuellement stochastique, décrit avec quelle probabilité choisir chaque culture sur chaque parcelle en fonction d'un certain nombre d'informations (qui dépendent de la structure de politique qu'on s'est donnée, voir section 3.6.9). Dans la suite, on pourra utiliser le terme de 'stratégie' comme synonyme du terme 'politique'.

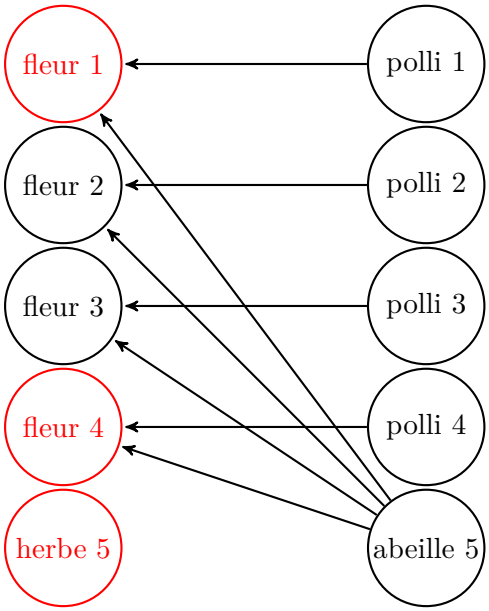


FIGURE 3.5 – Réseau trophique représentant les liens entre adventices et pollinisateurs ; en rouge les groupes adventices les plus nuisibles au rendement

A la différence de [PNL+05], qui optimise la composition et la configuration du paysage de manière à faire un compromis entre un critère économique et un critère de biodiversité, et de [BAD08], qui optimise la composition et la configuration du paysage pour le service de pollinisation, dans notre modèle la composition et la configuration du paysage ne sont pas fixes mais peuvent varier au cours du temps.

3.6.4 Quantification des pollinisateurs en fonction des cultures et des adventices

Dans notre modèle, on déduit de manière déterministe le score d'abondance des B groupes de pollinisateurs l'année t des cultures et des occurrences d'adventices l'année t (voir figure 3.6). Ce score d'abondance se calcule par parcelle et représente en quelque sorte une probabilité de visite de la parcelle par le groupe de pollinisateurs.

Le score PO^t_{pb} d'abondance du pollinisateur b sur la parcelle p l'année t dépend de l'état et de la culture des parcelles du voisinage $V^{\alpha_b}(p)$ de la parcelle p. Ce voisinage dépend de la portée α_b du pollinisateur, qui représente sa capacité à se déplacer. De plus, le score d'abondance du pollinisateur sauvage b ne dépend que de la présence du groupe adventice $RT(b)$ dont il se nourrit, tandis que le score d'abondance de l'abeille domestique dépend de la présence de tous les groupes adventices de type fleur :

$$\forall p = 1...P, \forall b = 1...B-1,\ PO^t_{pb} = g_1(S^t_{V^{\alpha_b}(p)RT(b)}, A^t_{V^{\alpha_b}(p)}) \tag{3.1}$$

$$\forall p = 1...P,\ PO^t_{p5} = g_2(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}) \tag{3.2}$$

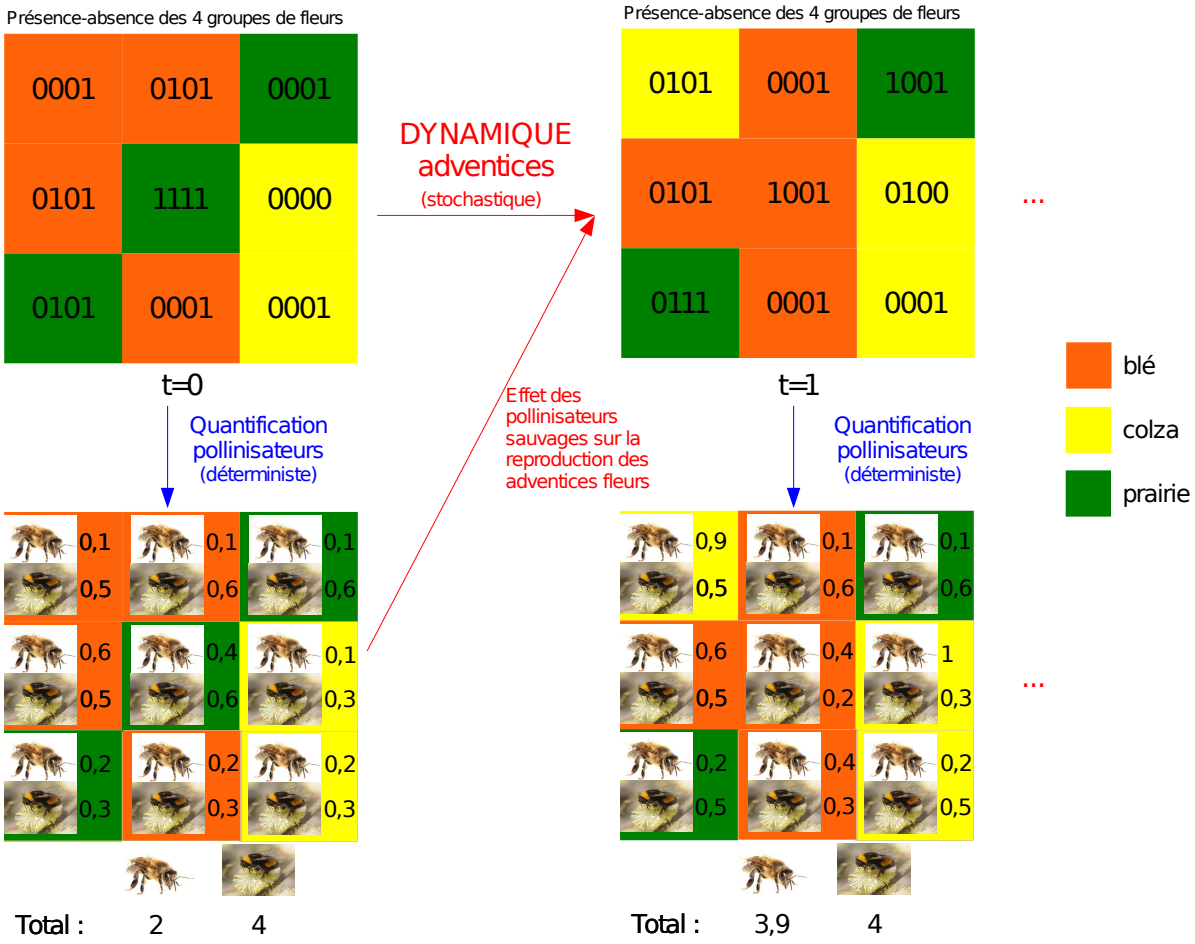


FIGURE 3.6 – Représentation de la dynamique des adventices à partir des occurrences des adventices, des cultures et des scores d'abondance des pollinisateurs sauvages l'année précédente, et de la quantification des pollinisateurs à partir des cultures et des adventices fleurs la même année ; pour simplifier, dans la partie inférieure nous n'avons représenté que le groupe des abeilles domestiques et un groupe de pollinisateurs sauvages.

Pour calculer ces scores d'abondance (qui peuvent également être interprétés comme des probabilités de présence), nous nous inspirons du modèle présenté dans [LKR+09]. L'idée de ce modèle est qu'il y a d'autant plus de pollinisateurs qui visitent une parcelle p qu'il y d'habitats et de ressources florales autour (avec une décroissance de l'importance des parcelles exponentielle en la distance). Le score d'abondance est calculé comme un produit d'un score d'habitabilité, représentant le niveau d'habitabilité du voisinage de la parcelle p pour le pollinisateur considéré, et d'un score de ressource florale, représentant le niveau de ressource florale disponible dans le voisinage de la parcelle p pour le pollinisateur considéré. Dans la suite, nous détaillons le modèle que nous proposons pour ces deux types de scores, d'abord pour les pollinisateurs sauvages, puis pour l'abeille domestique.

Pollinisateurs sauvages

Considérons d'abord les pollinisateurs sauvages ($b = 1...4$).

Portée : La portée des pollinisateurs sauvages est plus faible que celle de l'abeille domestique. Une discussion avec des experts nous a conduit à choisir un paramètre de portée de $\alpha_b = 0.5$ pour tous les pollinisateurs sauvages. Nous considérons les parcelles à distance D de p et telles que $e^{-D/\alpha_b} < 0.07$ comme négligeables. Ce qui revient à négliger les parcelles qui sont à distance de p supérieure ou égale à $\sqrt{2}$. Le voisinage $V^{\alpha_b}(p)$ de la parcelle p pour le pollinisateur sauvage b est donc constitué des 4 parcelles les plus proches et de la parcelle p (voir figure 3.7). Dans la suite, le voisinage d'une parcelle p pour un groupe de pollinisateur sauvage $b \in \{1, ..., 4\}$ sera noté $V(p)$ puisque tous les groupes de pollinisateurs sauvages ont la même portée.

Score d'habitabilité potentielle $H_{qb}(A_q^t)$ de la parcelle $q \in V(p)$ pour le pollinisateur b : l'habitat des pollinisateurs sauvages se trouvant dans les prairies (voir section 3.4), nous définissons ce score comme égal à 1 si A_q^t =prairie, et 0 sinon.

Score de ressource florale potentielle $F_{qb}(S_{qRT(b)}^t)$ de la parcelle $q \in V(p)$ pour le pollinisateur b : ce score est défini comme égal à 1 si le groupe adventice dont se nourrit le pollinisateur sauvage, $RT(b)$, est présent sur q, et 0 sinon (voir réseau trophique de la figure 3.5).

Score d'habitabilité $\mathrm{SH}_{pb}(A_{V(p)}^t)$ de la parcelle p pour le pollinisateur b : le score d'habitabilité de la parcelle p pour le pollinisateur b est une somme pondérée des scores d'habitabilité potentielle des parcelles dans le voisinage de p :

$$\mathrm{SH}_{pb}(A_{V(p)}^t) = \frac{\sum_{q \in V(p)} \mathrm{H}_{qb}(A_q^t) e^{-D_{qp}/\alpha_b}}{\sum_{q' \in V(p)} e^{-D_{q'p}/\alpha_b}}$$

où D_{qp} représente la distance entre les parcelles p et q. Les parcelles voisines étant toutes à la même distance, elles ont toutes le même poids. La parcelle p a par contre plus de

poids que ses voisines. On a :

$$\mathrm{SH}_{pb}(A^t_{V(p)}) = \sum_{q \in V(p)} c_{qpb} \mathrm{H}_{qb}(A^t_q)$$

où $c_{qpb} = \frac{e^{-D_{qp}/\alpha_b}}{\sum_{q' \in V(p)} e^{-D_{q'p}/\alpha_b}}$ $\forall b = 1...B-1$. On a (pour une parcelle centrale qui a 4 voisins), $c_{qpb} = 0.6488$ si $p = q$, et $c_{qpb} = 0.0878$ si $q \in V(p) \backslash p$.

Score de ressource florale $\mathrm{SF}_{pb}(S^t_{V(p)RT(b)})$ de la parcelle p pour le pollinisateur b : de même que le score d'habitabilité, il s'obtient à partir des scores de ressource florale potentielle des parcelles voisines de p :

$$\mathrm{SF}_{pb}(S^t_{V(p)RT(b)}) = \frac{\sum_{q \in V(p)} \mathrm{F}_{qb}(S^t_{qRT(b)}) e^{-D_{qp}/\alpha_b}}{\sum_{q' \in V(p)} e^{-D_{q'p}/\alpha_b}} = \sum_{q \in V(p)} c_{qpb} \mathrm{F}_{qb}(S^t_{qRT(b)})$$

Score PO^t_{pb} du pollinisateur b sur la parcelle p : le score final d'abondance du pollinisateur b dans la parcelle p est le produit du score d'habitabilité et du score de ressource florale de la parcelle p :

$$PO^t_{pb} = g_1(S^t_{V(p)RT(b)}, A^t_{V(p)}) = \mathrm{SH}_{pb}(A^t_{V(p)}) \times \mathrm{SF}_{pb}(S^t_{V(p)RT(b)})$$

Ce score est entre 0 et 1.

Bilan : On a donc, si on somme les scores des pollinisateurs sauvages dans le paysage :

$$\begin{aligned} \sum_{p=1}^{P} \sum_{b=1}^{B-1} PO^t_{pb} &= \sum_{p=1}^{P} \sum_{b=1}^{B-1} \left(\sum_{q_1 \in V(p)} c_{q_1pb} F_{q_1b}(S^t_{q_1RT(b)}) \right) \left(\sum_{q_2 \in V(p)} c_{q_2pb} H_{q_2b}(A^t_{q2}) \right) \\ &= \sum_{p=1}^{P} \sum_{b=1}^{B-1} \sum_{q_1 \in V(p)} \sum_{q_2 \in V(p)} h(S^t_{q_1RT(b)}, A^t_{q_2}) \end{aligned}$$

où $h(S^t_{q_1RT(b)}, A^t_{q_2}) = c_{q_1pb} F_{q_1b}(S^t_{q_1RT(b)}) c_{q_2pb} H_{q_2b}(A^t_{q2})$.

Abeille domestique

Considérons maintenant l'abeille domestique ($b = B = 5$).

Portée : L'abeille domestique peut aller plus loin que les pollinisateurs sauvages pour se nourrir. Une discussion avec des experts nous a conduits à choisir un paramètre de portée de $\alpha_5 = 0.8$. Comme nous négligeons les parcelles à distance D de p et telles que $e^{-D/\alpha_5} < 0.07$, les parcelles à distance de p supérieure ou égale à $\sqrt{5}$ sont négligeables pour l'abeille, et le voisinage $V^{\alpha_5}(p)$ de la parcelle p pour l'abeille est constitué des 12 parcelles les plus proches (au plus) et de la parcelle p (voir figure 3.7). On a alors (pour une parcelle centrale qui a 12 voisins), $c_{qp5} = 0.3167$ si $p = q$, $c_{qp5} = 0.0907$ si q est à

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

FIGURE 3.7 – Représentation sur un parcellaire 5×5 du voisinage V (en bleu) et $V^{0.8}$ (bleu+rouge) de la parcelle 13 ; $V(13) = \{8, 12, 13, 14, 18\}, V^{0.8}(13) = \{3, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 23\}$.

distance 1 de p, $c_{qp5} = 0.0541$ si q est à distance $\sqrt{2}$ de p, et $c_{qp5} = 0.0260$ si q est à distance 2 de p.

Modélisation : Comme l'habitat des abeilles domestiques est géré par l'homme (placement de ruches), nous considérons que le **score d'habitabilité** de toutes les parcelles pour l'abeille est égal à 1 quelle que soit la culture (blé, colza, prairie).

Comme dans [LKR+09], nous prenons en compte les saisons. Nous considérons $K = 3$ saisons : l'indice $k = 1$ est associé au printemps, l'indice $k = 2$ à l'été et l'indice $k = 3$ à l'automne. Le score d'abondance de l'abeille sur la parcelle p l'année t est donné par :

$$\begin{aligned} PO^t_{p5} = g_2(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}) &= \mathrm{SF}_{p5}(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}) \\ &= \sum_{k=1}^{K} w_k \frac{\sum_{q \in V^{\alpha_5}(p)} \mathrm{F}_{q5k}(S^t_{q,-5}, A^t_q) e^{-D_{qp}/\alpha_5}}{\sum_{q' \in V^{\alpha_5}(p)} e^{-D_{q'p}/\alpha_5}} \\ &= \sum_{k=1}^{K} w_k \sum_{q \in V^{\alpha_5}(p)} c_{qp5} \mathrm{F}_{q5k}(S^t_{q,-5}, A^t_q) \end{aligned}$$

où $c_{qp5} = \frac{e^{-D_{qp}/\alpha_5}}{\sum_{q' \in V^{\alpha_5}(p)} e^{-D_{q'p}/\alpha_5}}$ et w_k représente le poids d'importance de la saison k pour l'abeille. Comme dans [LKR+09], nous prenons un poids de $w_1 = 0.4$ pour le printemps, un poids de $w_2 = 0.4$ pour l'été et un poids de $w_3 = 0.2$ pour l'automne. Le score d'abondance de l'abeille est donc d'autant plus important qu'elle a des ressources florales disponibles dans le voisinage tout au long de l'année (et notamment au printemps et en été).

Scores de ressource florale potentielle $F_{q5k}(S^t_{q,-5}, A^t_q)$ de la parcelle $q \in V^{\alpha_5}(p)$ pour l'abeille en fonction de la saison k : Ces scores sont indépendants de la présence-absence du groupe adventice 5 (groupe des herbes), ils ne dépendent que de la culture et des groupes adventices fleurs dans la parcelle.

La saison de floraison des différents groupes adventices est connue, on peut donc quand on connait la présence-absence des différents groupes dans la parcelle q l'année t, en déduire si il y a présence d'adventices en fleur ou non à chaque saison. Les scores de ressource florale potentielle d'une parcelle q pour l'abeille sont donnés dans la table

3.1. Quand la culture est en fleur à la saison considérée le score est maximal car la ressource florale pour l'abeille est importante, la présence ou non d'adventices est alors négligeable. Le colza est en fleurs au printemps, et la luzerne en été (en mesure agri-environnementale elle n'est pas fauchée avant floraison). Si la culture n'est pas en fleur à la saison considérée mais qu'il y a des adventices en fleur, le score est divisé par deux. Enfin, le score est nul si il n'y a ni culture ni adventices en fleur à la saison considérée.

Les scores résultants sur l'année $\sum_{k=1}^{K} w_k F_{q5k}(S^t_{q,-5}, A^t_q)$ sont donnés dans la table 3.2, dont les lignes correspondent aux différents états possibles pour les 4 groupes de fleurs. On peut constater que les scores en blé sont systématiquement inférieurs et que les scores en colza et en prairie sont systématiquement égaux sauf dans un cas (présence du groupe 4 seulement, à émergence printanière stricte).

PRINTEMPS	absence d'adventices en fleur	présence d'adventices en fleur
colza	1	1
prairie	0	0.5
blé	0	0.5
ETE	absence d'adventices en fleur	présence d'adventices en fleur
colza	0	0.5
prairie	1	1
blé	0	0.5
AUTOMNE	absence d'adventices en fleur	présence d'adventices en fleur
colza	0	0.5
prairie	0	0.5
blé	0	0.5

TABLE 3.1 – Scores de ressource florale potentielle pour l'abeille d'une parcelle donnée en fonction de la saison considérée, de la culture et de la présence d'adventices en fleurs sur la parcelle

état	colza	blé	prairie
0000	0.4	0	0.4
1000	0.6	0.4	0.6
0100	0.6	0.4	0.6
1100	0.6	0.4	0.6
0010	0.7	0.5	0.7
1010	0.7	0.5	0.7
0110	0.7	0.5	0.7
1110	0.7	0.5	0.7
0001	0.4	0.2	0.6
1001	0.6	0.4	0.6
0101	0.6	0.4	0.6
1101	0.6	0.4	0.6
0011	0.7	0.5	0.7
1011	0.7	0.5	0.7
0111	0.7	0.5	0.7
1111	0.7	0.5	0.7

TABLE 3.2 – Scores de ressource florale potentielle pour l'abeille d'une parcelle donnée sur l'année en fonction de la culture et des groupes adventices présents sur la parcelle - exemple : l'état 0001 correspond à l'état d'une parcelle où tous les groupes de fleurs adventices sont absents sauf le groupe 4

3.6.5 Modèle de dynamique spatio-temporel des adventices

Nous décrivons maintenant le modèle de dynamique des adventices dans le cadre PDMF[3], inspiré du modèle proposé dans [DKRP10].

Modèle de dynamique

La dynamique des adventices est markovienne et pour une parcelle p et un groupe adventice i donnés on a :

$$P(S_{pi}^{t+1}|S^t, A^t) = f_1(S_{pi}^{t+1}, S_{V(p)i}^t, PO_{pRT^{-1}(i)}^t, A_p^t)$$

D'où en combinant avec l'équation (3.1) :

$$P(S_{pi}^{t+1}|S^t, A^t) = f_2(S_{pi}^{t+1}, S_{V(p)i}^t, A_{V(p)}^t)$$

Nous supposons que sur une année les événements se passent dans l'ordre suivant : dispersion éventuelle des groupes adventices, puis mise en place de la culture, puis extinction éventuelle des groupes adventices. On rappelle que $S_{pi}^t = 1$ si il y a des plantes levées du groupe i dans la parcelle p l'année t, et 0 sinon. Nous considérons les groupes adventices comme indépendants entre eux (nous considérons qu'il n'y a pas de compétition) :

$$P(S^{t+1}|S^t, A^t) = \prod_{p=1}^{P} \prod_{i=1}^{I} P_{pi}\left(S_{pi}^{t+1} \middle| S_{V(p)i}^t, A_{V(p)}^t\right)$$

Comme dans [DKRP10], nous utilisons une distribution de Bernouilli :

$$S_{pi}^{t+1}|S_{V(p)i}^t, A_{V(p)}^t \sim Bernouilli\left((\phi_i + (1-\phi_i)\gamma_i)S_{pi}^t + \gamma_i(1-S_{pi}^t)\right)$$

où ϕ_i correspond au paramètre de persistance locale (probabilité que le groupe adventice i, si il était présent l'année t, ait survécu jusqu'à l'année $t+1$), et γ_i au paramètre de colonisation locale (probabilité que le groupe adventice i s'installe avec succès dans la parcelle p l'année $t+1$). Dans le cas d'espèces adventices, la dispersion peut venir d'une résurgence liée au stock de graines de la parcelle ou à une dispersion depuis une parcelle voisine. A la différence de [DKRP10], le paramètre de colonisation locale pour le groupe i dépend du nombre de parcelles voisines contenant le groupe i :

$$\gamma_i = \gamma_i(S_{V(p)i}^t, A_p^t) = \epsilon + (1-\epsilon)\left[1 - (1-\kappa(A_p^t))(1-\nu)^{\sum_{j\in V(p)} S_{ji}^t}\right]$$

où ϵ représente la probabilité d'une dispersion de longue distance (par exemple dispersion par les machines), $\kappa(A_p^t)$ représente la probabilité que le groupe adventice réapparaisse à partir du stock de graines de la parcelle (cette probabilité peut dépendre de la culture) et ν représente la probabilité de dispersion d'une parcelle à une autre.

Pour ce qui est du paramètre de persistance locale, il dépend de la quantité de pollinisateurs sauvages visitant le groupe adventice i :

$$\forall i = 1...4,\ \phi_i = \phi_i(S_{V(p)i}^t, A_{V(p)}^t) = \frac{\sum_{b\in RT^{-1}(i)} PO_{pb}(S_{V(p)RT(b)}^t, A_{V(p)}^t) + \eta(A_p^t)}{|RT^{-1}(i)| + \eta(A_p^t)}$$

où $|RT^{-1}(i)|$ représente le nombre de groupes de pollinisateurs sauvages visitant le groupe adventice i (égal à 1 pour le réseau trophique de la figure 3.5). Le paramètre $\eta(A_p^t)$, qui peut dépendre de la culture, représente la probabilité de reproduction par le vent. Pour le groupe des herbes (groupe 5), les pollinisateurs sauvages ne jouent pas sur la reproduction :

$$\phi_5 = \phi_5(S^t_{V(p)5}, A^t_{V(p)}) = \eta(A_p^t)$$

Ainsi, le paramètre de persistance locale d'un groupe de fleurs dont le groupe de pollinisateurs sauvages qui doivent la polliniser est absent est plus faible que le paramètre de persistance locale d'une herbe.

Paramètres de dynamique

Il y a donc quatre paramètres dans ce modèle de dynamique, dont deux peuvent dépendre de la culture A_p^t :

1. $\eta(A_p^t)$: probabilité de reproduction par le vent
2. $\kappa(A_p^t)$: probabilité que le groupe réapparaisse à partir du stock sachant qu'il était absent l'année d'avant
3. ν : probabilité que le groupe arrive par dispersion d'une parcelle voisine
4. ϵ : probabilité que le groupe arrive par dispersion longue distance (par exemple due aux machines).

Nous supposons qu'aucun de ces paramètres ne dépend du groupe adventice (groupes identiques en terme de réponse) [1]. Pour proposer ces paramètres, nous nous sommes appuyés sur les hypothèses suivantes :

1. La survie des adventices est identique dans une parcelle en blé et dans une parcelle en colza, et elle est supérieure dans une parcelle en prairie (ceci est vrai aussi bien pour la probabilité de reproduction par le vent $\eta(A_p^t)$ que pour la probabilité de réapparition à partir du stock $\kappa(A_p^t)$). En effet, dans les prairies il n'y a ni travail du sol ni traitements herbicides.
2. Pour une même culture A_p^t, $\eta(A_p^t) = \kappa(A_p^t)$.
3. La dispersion longue distance est inférieure à la dispersion courte distance : $\epsilon < \nu$.
4. La dispersion temporelle (via la banque de graines) est plus importante que la dispersion spatiale : $\forall A_p^t,\ \kappa(A_p^t) > \nu$.

Les valeurs des paramètres de dynamique ont été choisies en collaboration avec des experts, de manière à respecter les hypothèses ci-dessus : $\epsilon = 0.01$, $\nu = 0.1$ et $\eta(A_p^t) = \kappa(A_p^t) = 0.6$ si A_p^t=prairie, $\eta(A_p^t) = \kappa(A_p^t) = 0.1$ si $A_p^t \neq$prairie.

1. Par contre, les macro-paramètres ϕ_i et γ_i dépendent indirectement du groupe adventice i (de sa présence sur les parcelles ou de la quantité de pollinisateurs qui peuvent le visiter).

Comportement du modèle

Dans l'annexe C.1, des figures décrivent le comportement moyen du modèle sur un grand nombre de simulations pour des politiques de type monoculture (l'ensemble du paysage contient une seule culture, qui ne varie pas au cours du temps) :

1. Dans une monoculture de blé, les différents groupes adventices décroissent jusqu'à atteindre une présence stable autour de 20%. Les pollinisateurs sauvages sont absents à cause d'une absence d'habitats (prairies). L'abeille domestique peut se maintenir avec un score moyen de l'ordre de 24% grâce à la ressource florale adventice.
2. Dans une monoculture de colza, la seule différence avec la monoculture de blé est qu'il y a plus d'abeilles domestiques (le colza est une ressource importante au printemps), avec une présence dans le paysage d'environ 52%. Cela n'affecte pas les adventices puisqu'elles ne sont pas pollinisées par l'abeille domestique.
3. Dans une monoculture de prairie, les adventices fleurs et les pollinisateurs sauvages croissent jusqu'à atteindre le score maximal au bout de 6 ans environ. Le groupe des herbes est plus présent que dans les monocultures de blé ou de colza, et les abeilles domestiques aussi.

3.6.6 Objectifs sur les services écosystémiques

Nous nous intéressons aux services écosystémiques suivants :

1. Services de production :
   — production de miel
   — production de colza
   — production de blé
   — production de fourrage en prairie
2. Service culturel :
   — conservation des pollinisateurs (sauvages et domestiques) et des adventices.

Les services de régulation (régulation des populations d'espèces adventices et de pollinisateurs) sont pris en compte dans l'ensemble du modèle mais ne sont pas intégrés directement dans les objectifs, qui portent uniquement sur les services de production et culturel.

Dans un premier temps (voir section 3.7.1), nous comparerons les stratégies optimales obtenues pour les objectifs simples suivants, basés sur un seul service écosystémique :

1. **Objectif 1** : maximiser la marge en colza-blé (point de vue des céréaliers)
2. **Objectif 1 bis** : maximiser la marge en colza-blé-prairie (point de vue des éleveurs)
3. **Objectif 2** : maximiser la biodiversité en adventices et pollinisateurs sauvages et domestiques (point de vue des citoyens)

4. **Objectif 2 bis** : maximiser la biodiversité en adventices et pollinisateurs sauvages (point de vue des citoyens)
5. **Objectif 2 ter** : maximiser la biodiversité en adventices uniquement (point de vue des citoyens)
6. **Objectif 3** : maximiser la marge en miel, c'est-à-dire l'abondance d'abeilles domestiques (point de vue des apiculteurs).

Dans un second temps (voir section 3.7.2), nous nous intéresserons aux résultats obtenus pour différents types de compromis entre services écosystémiques :

1. **Objectif C1** : Nous nous intéresserons ensuite aux politiques optimales associées aux paysages *land sparing* et *land sharing* étudiés dans la littérature pour un compromis rendement-biodiversité (voir section 3.2) ; nous considérerons qu'un paysage *land sparing* consiste à optimiser la marge économique sur une moitié du paysage, et la biodiversité sur l'autre moitié, tandis qu'un paysage *land sharing* consiste à optimiser la marge économique sur une parcelle, et la biodiversité sur ses parcelles adjacentes.
2. **Objectif C2** : Enfin, nous nous intéresserons à des compromis de type 'durabilité' : pour ces compromis, des domaines admissibles de valeur, à l'échelle de l'exploitation [2], seront définis pour les différents services écosystémiques, et nous maximiserons le nombre moyen d'années et d'exploitations pour lesquelles les services écosystémiques sont dans ces domaines admissibles.

3.6.7 Modélisation de la marge économique

Nous proposons maintenant un modèle économique simple pour la marge économique obtenue à l'échelle de la parcelle (cas du colza, du blé ou de la prairie) ou à l'échelle du paysage (cas du miel).

Céréales : Soit $m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p)$ la marge obtenue sur la parcelle p l'année t (score entre 0 et 1). Pour le colza, cette marge est fonction des adventices et de l'abondance des abeilles domestiques, tandis que pour le blé elle est fonction uniquement des adventices (voir section 3.4). Dans les deux cas, la marge économique est également fonction de la culture précédente sur la parcelle p. En effet, si la culture précédente était une prairie, nous considérons qu'il y a une économie en utilisation d'azote, notée $e(A^{t-1}_p)$. Soit $a(S^t_p)$ la perte due aux adventices, r^{colza}_{min} le rendement minimum du colza en l'absence d'adventices, r^{colza}_{max} le rendement maximal du colza en l'absence d'adventices, r^{ble} le rendement du blé en l'absence d'adventices, c^{colza}_{fixe} le coût fixe associé au colza, et c^{ble}_{fixe} le coût fixe associé au blé.

Dans le cas du blé, on a :

$$\text{si } A^t_p = \text{blé},\ m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p) = m_p(S^t_p, A^{t-1}_p) = gain^{ble}_p(S^t_p) - cout^{ble}_p(A^{t-1}_p)$$

2. Une exploitation est un ensemble de parcelles appartenant à un même agriculteur et qui ne sont pas forcément adjacentes.

où

$$gain_p^{ble}(S_p^t) = r^{ble}\left[1 - a(S_p^t)\right]$$

$$cout_p^{ble}(A_p^{t-1}) = c_{fixe}^{ble} - e(A_p^{t-1})$$

Dans le cas du colza, le score d'abondance d'abeilles domestiques sur la parcelle p fait croître linéairement la production (le gain) :

si $A_p^t = \text{colza},\ m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A_p^{t-1}) = gain_p^{colza}(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}) - cout_p^{colza}(A_p^{t-1})$

où

$$gain_p^{colza}(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}) = \left[r_{min}^{colza} + (r_{max}^{colza} - r_{min}^{colza}) g_2(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)})\right]\left[1 - a(S_p^t)\right]$$

$$cout_p^{colza}(A_p^{t-1}) = c_{fixe}^{colza} - e(A_p^{t-1})$$

Nous avons choisi les valeurs de paramètres en collaboration avec des experts : $e(A_p^{t-1}) = 0.015$ si A_p^{t-1}=prairie, $a(S_p^t) = 0.2$ si il y a présence d'adventices très nuisibles dans la parcelle, $a(S_p^t) = 0.1$ si il y a seulement présence d'adventices peu nuisibles dans la parcelle et $a(S_p^t) = 0$ si il n'y a pas d'adventices dans la parcelle (voir [WW90]). De plus, nous avons pris : $r_{min}^{colza} = 0.7, r_{max}^{colza} = 1$, $r^{ble} = 0.9$, et $c_{fixe}^{colza} = c_{fixe}^{ble} = 0.1$. Avec ces valeurs de paramètres, comme le montre la figure 3.8, la marge en colza devient supérieure à la marge en blé à partir d'un score d'abondance d'abeilles environ égal à 0.67.

Prairies : Pour les prairies (fourrage), la marge est supposée constante, elle ne dépend ni des pollinisateurs ni des adventices. Une prairie rapporte moins la première année à cause du coût d'implantation :

$$m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A_p^{t-1}) = m_p(A_p^t, A_p^{t-1}) = \begin{cases} 0.5 & \text{si } A_p^t = prairie \text{ et } A_p^{t-1} \neq prairie \\ 0.8 & \text{si } A_p^t = prairie \text{ et } A_p^{t-1} = prairie \end{cases}$$

Apiculteurs : Enfin, pour le miel, nous supposons que le gain et le coût sont proportionnels à l'abondance d'abeilles domestiques à l'échelle du paysage. Il en va donc de même pour la marge économique :

$$m_{miel}(S^t, A^t) = \sum_{p=1}^{P} PO_{p5}^t = \sum_{p=1}^{P} g_2(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}) = \sum_{p=1}^{P}\sum_{k=1}^{K} w_k \sum_{q \in V^{\alpha_5}(p)} c_{qp5} \mathrm{F}_{q5k}(S^t_{q,-5}, A_q^t)$$

Dans notre modèle, maximiser la marge économique en miel revient donc à maximiser l'abondance des abeilles domestiques à l'échelle du paysage. La marge économique pour le miel telle que nous l'avons modélisée n'est pas comparable à la marge économique pour une culture donnée (prairie, colza ou blé).

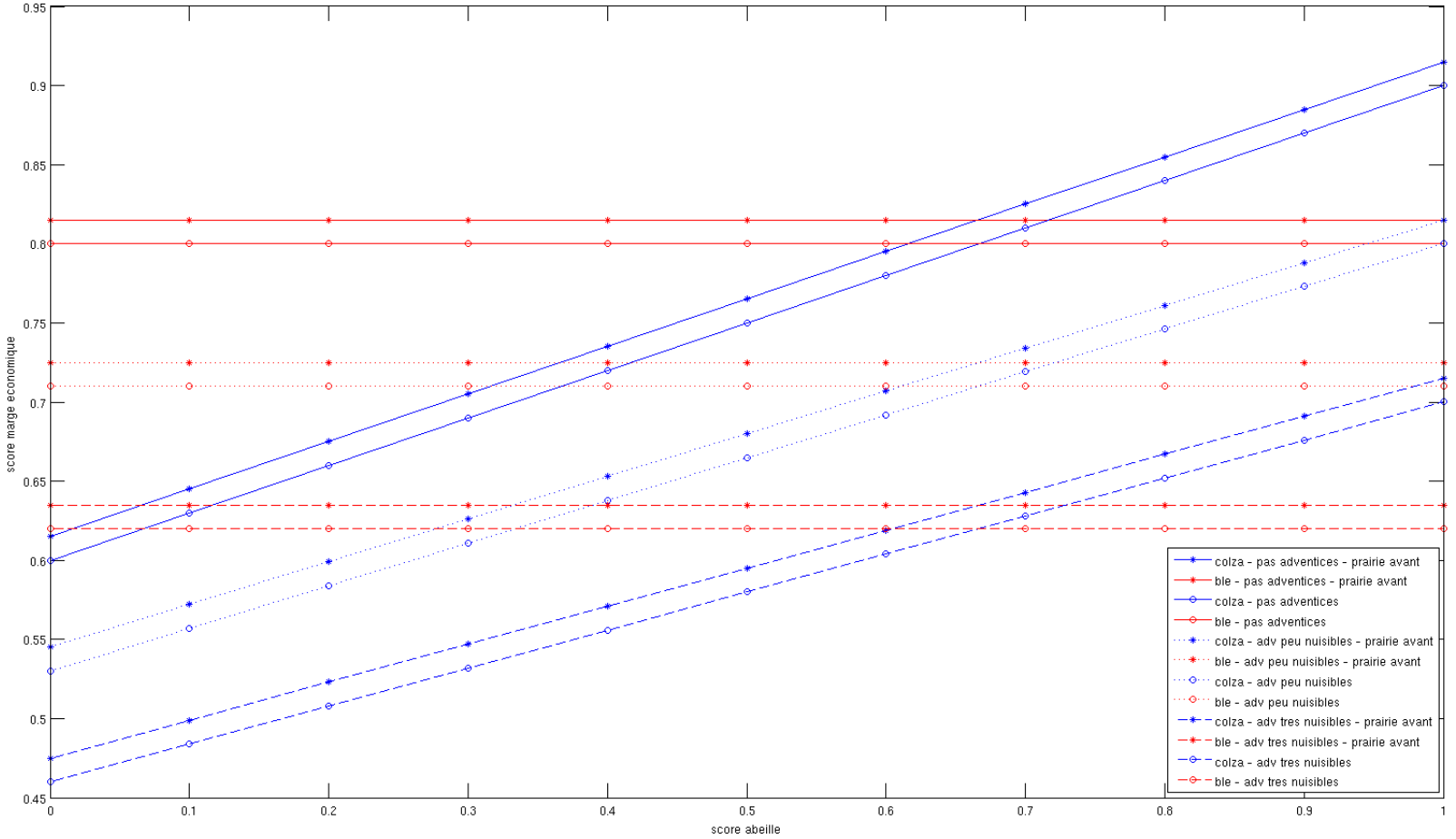


FIGURE 3.8 – Courbes représentant la marge économique des cultures de colza et de blé en fonction du score d'abondance d'abeilles domestiques sur la parcelle

3.6.8 Modèles de récompense associés aux différents objectifs

Dans l'annexe C.2, nous décrivons la modélisation des fonctions de récompense associées à chaque objectif considéré sous forme additive, comme le requiert le cadre PDMF[3]. En particulier, la marge économique en céréales et fourrage à l'échelle du paysage est mesurée par la somme des scores de marge économique à l'échelle de la parcelle :

$$\sum_{p=1}^{P} m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p)$$

Et la biodiversité à l'échelle du paysage est mesurée par la somme des scores de présence-absence des groupes adventices et des scores d'abondance des pollinisateurs dans l'ensemble du paysage :

$$\sum_{p=1}^{P}\sum_{i=1}^{I} S^t_{pi} + \sum_{p=1}^{P}\sum_{b=1}^{B} PO^t_{pb}$$

Ainsi, par exemple, la fonction de récompense associée à l'objectif 1 est :

$$R^1(S^t, A^t, A^{t-1}) = \sum_{p=1}^{P} m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p)\, \mathbb{1}_{A^t_p \neq prairie}$$

La fonction de récompense associée à l'objectif 2 est :

$$R^2(S^t, A^t, A^{t-1}) = R^2(S^t, A^t) = \sum_{p=1}^{P}\sum_{i=1}^{I} S^t_{pi} + \sum_{p=1}^{P}\sum_{b=1}^{B} PO^t_{pb}$$

La fonction de récompense associée à l'objectif C1, si E_1 est l'ensemble de parcelles sur lesquelles on souhaite maximiser la marge économique en blé-colza, et E_2 est l'ensemble de parcelles sur lesquelles on souhaite maximiser la biodiversité, est donnée par :

$$\begin{aligned} R^{C1}(S^t, A^t, A^{t-1}) &= \frac{1}{|E_1|}\sum_{p\in E_1} m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p)\, \mathbb{1}_{A^t_p \neq prairie} \\ &\quad + \frac{1}{|E_2|(I+B)}\sum_{p\in E_2}\left(\sum_{i=1}^{I} S^t_{pi} + \sum_{b=1}^{B} PO^t_{pb}\right) \end{aligned}$$

où I est le nombre de groupes adventices et B le nombre de groupes de pollinisateurs. Pour l'objectif C2, nous chercherons à maximiser le nombre moyen d'années et d'exploitations pour lesquelles la marge économique en blé est supérieure à un certain seuil β (si la parcelle n'est pas en blé elle ne compte pas) et le nombre de groupes adventices est supérieur à un certain seuil ζ. Soit L_r le nombre d'exploitations et $Q_r(l)$ l'ensemble des parcelles de l'exploitation $l \in \{1, ..., L_r\}$: $(Q_r(1), ..., Q_r(L_r))$ forme une partition de $\{1, ..., P\}$. La fonction de récompense associée à l'objectif C2 est donnée par :

$$R^{C2}(S^t, A^t, A^{t-1}) = \sum_{l=1}^{L_r} \mathbb{1}\left\{\sum_{p\in Q_r(l)} m_p(S^t_p, A^{t-1}_p)\, \mathbb{1}_{A^t_p = ble} \geq \beta\right\} \mathbb{1}\left\{\sum_{p\in Q_r(l)}\sum_{i=1}^{I} S^t_{pi} \geq \zeta\right\}$$

L'annexe C.2 donne la décomposition des différentes fonctions de récompense sous forme additive.

3.6.9 Structure de la politique

Nous envisagerons dans les expérimentations deux structures de politique possibles :

1. **structure 0** : décider de la culture à mettre sur chaque parcelle en fonction d'aucune information particulière :

$$\delta(A^t) = \prod_{p=1}^{P} \delta_p(A_p^t)$$

   On a alors $pa_\delta(A_p^t) = \emptyset \ \forall p = 1...P$. La politique étant stochastique, elle indique le pourcentage de chacune des cultures à utiliser sur la parcelle.

2. **structure S** : décider de la culture à mettre sur chaque parcelle en fonction de l'état de la parcelle (vecteur de présence/absence des 5 groupes adventices) :

$$\delta(A^t|S^t) = \prod_{p=1}^{P} \delta_p(A_p^t|S_p^t)$$

   On a alors $pa_\delta(A_p^t) = \{S_p^t\} \ \forall p = 1...P$.

Il pourrait être intéressant également de considérer des politiques basées sur les cultures mises en place la ou les années précédentes sur la parcelle, afin d'obtenir des règles de rotation. Cela fait partie des perspectives à court terme que nous envisageons pour ce travail.

3.7 Résultats

Comme le montre la figure 3.4, il reste à définir la distribution initiale sur les états du système P^0 et l'espace de temps $\mathcal{T}$. En l'absence d'informations suffisantes, nous choisissons une distribution initiale uniforme et indépendante pour chaque groupe adventice. Pour l'horizon de temps, nous choisissons de considérer un horizon fini $T = 10$ (la récompense est sommée des années 0 à 9 et le facteur d'amortissement est de $\gamma = 1$).

Étant donné que les variables d'action ne sont pas binaires, nous avons utilisé dans toute cette partie expérimentale l'algorithme GD-LBP pour optimiser les politiques (et l'évaluation par la méthode de Monte-Carlo pour une évaluation *a posteriori* plus précise). Nous avons utilisé une machine avec 10 coeurs et 256 Go de RAM, et le logiciel `Matlab R2015a`. Pour l'évaluation par la méthode de Monte-Carlo (voir section 2.2.2), nous avons utilisé systématiquement un horizon de $T = 10$, un nombre de simulations de nsim = 4000 et une parallélisation des simulations. Pour l'algorithme *Loopy Belief Propagation*, nous avons utilisé l'interface Matlab de la librairie libDAI [Moo10], comme dans le chapitre 2. Les paramètres pour l'algorithme de descente de gradient utilisés dans toute cette section sont identiques à ceux du chapitre 2 : maxit = 1000, $\epsilon = \epsilon_g = \epsilon_V = \epsilon_\theta = 0.01$.

Sauf indication contraire, le point de départ de l'algorithme GD-LBP est une politique uniforme et nous prenons en compte les symétries du parcellaire pour réduire le nombre

de politiques locales à optimiser (voir section 2.4.5). De plus, si cela n'est pas mentionné, l'algorithme GD-LBP s'est arrêté parce que la norme du gradient était proche de zéro (condition nécessaire d'optimalité locale).

3.7.1 Résultats pour des objectifs simples

Nous nous sommes tout d'abord intéressés à des objectifs simples, ne faisant intervenir qu'un seul service écosystémique. Les résultats sont donnés dans la table 3.3 pour un parcellaire de taille 3×3. Nous avons obtenu les mêmes résultats en utilisant des points de départ différents. Nous donnons les valeurs estimées par la méthode de Monte-Carlo des stratégies retournées par GD-LBP, et nous donnons également le rapport en pourcentage de ces valeurs par rapport à la valeur maximale possible (par exemple, pour l'objectif 1, la valeur maximale est de $PT = 90$ dans le cas du parcellaire 3×3, voir annexe C.2).

Pour l'objectif 1, qui consiste à maximiser la marge en blé-colza, nous obtenons une politique consistant à mettre du blé sur tout le paysage et chaque année (monoculture). Cela peut s'interpréter par le fait que pour le blé la marge économique est indépendante de la quantité d'abeilles présentes, tandis que pour le colza, si cette culture peut permettre d'atteindre une marge économique supérieure à celle du blé, elle nécessite qu'une certaine quantité d'abeilles soient présentes (voir figure 3.8). Pour l'objectif 1 bis, qui consiste à maximiser la marge en blé-colza-prairie, l'algorithme GD-LBP renvoie une politique de monoculture en prairie. Cela s'explique par le fait que, excepté la première année, une prairie rapporte une marge de 0.8, tandis qu'atteindre une telle marge en blé demande qu'il y ait peu d'adventices (voir figure 3.8).

Pour l'objectif 2, qui consiste à maximiser la biodiversité (adventices, pollinisateurs sauvages et domestiques), on obtient une monoculture de prairie, ce qui s'explique notamment par le fait que les pollinisateurs sauvages ont besoin de cet habitat. Pour l'objectif 2 bis, qui consiste à maximiser uniquement adventices et pollinisateurs sauvages, on obtient également une monoculture de prairie. Et pour l'objectif 2 ter, qui consiste à maximiser uniquement la biodiversité adventice, on obtient aussi une monoculture de prairie. En effet, les prairies sont plus favorables à la reproduction des adventices.

Enfin, pour l'objectif 3, qui consiste à maximiser la marge économique en production de miel (donc l'abondance d'abeilles domestiques), on obtient également une monoculture de prairie, avec une valeur Monte-Carlo de 62.17. Cela est plus étonnant car le colza est une ressource importante pour l'abeille au printemps. Mais nous avons pu vérifier qu'une monoculture de colza avait, sur un parcellaire 3×3, une valeur Monte-Carlo de 47.50, qu'un damier colza-prairie avait une valeur de 57.09, et qu'une politique uniforme colza-prairie avait une valeur de 57.36. La stratégie renvoyée par GD-LBP n'est donc pas contredite par la recherche de politiques expertes. Une monoculture de prairie doit avoir l'avantage de contenir plus d'adventices, qui sont nécessaires aux abeilles en l'absence de cultures en fleur (période automne-hiver). Ces résultats se confirment sur un parcellaire de taille 10×10 : voir table 3.4.

Pour les objectifs 1, 2 et 3, nous avons envisagé une structure de politique plus complexe, basée sur l'état de la parcelle (structure S, voir section 3.6.9). Les résultats

obtenus sont identiques, les politiques locales obtenues ne dépendent pas de l'état de la parcelle.

Maintenant que nous avons vérifié que l'algorithme GD-LBP donnait des résultats cohérents pour des objectifs simples, nous pouvons nous intéresser à des objectifs plus complexes de compromis entre services écosystémiques.

objectif	structure (nb param)	politique	valeur MC	valeur LBP	itérations (temps)
1	0 (9)	blé	62.11 (69.01%)	73.67	1 (4.53 min)
1 bis	0 (9)	prairie	69.3 (77%)	81	1 (4.12 min)
2	0 (9)	prairie	798.35 (88.71%)	798.41	1 (2.31 min)
2 bis	0 (9)	prairie	736.65 (90.94%)	736.23	1 (2.08 min)
2 ter	0 (9)	prairie	404.80 (89.96%)	404.61	1 (1.27 min)
3	0 (9)	prairie	62.17 (69.08%)	62.18	1 (1.44 min)
1	S (288)	blé	62.11 (69.01%)	73.67	2 (32.14 min)
2	S (288)	prairie	798.35 (88.71%)	798.41	1 (10.36 min)
3	S (288)	prairie	62.17 (69.08%)	62.18	1 (8.77 min)

TABLE 3.3 – Résultats obtenus sur un parcellaire de taille 3×3 avec GD-LBP pour des objectifs simples ; la valeur MC représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique ; la valeur LBP est une approximation plus grande mais plus rapide de la valeur de la politique, utilisée au sein de l'algorithme d'optimisation.

objectif	structure (nb param)	politique	valeur MC	valeur LBP	itérations (temps)
1	0 (45)	blé	681.72 (68.17%)	809.87	1 (5.03h)
2	0 (45)	prairie	8895.9 (88.96%)	8897.6	1 (8.65h)
2 bis	0 (45)	prairie	8206.1 (91.18%)	8206.5	1 (7.82h)
2 ter	0 (45)	prairie	4514.7 (90.29%)	4515.3	1 (2.90h)
3	0 (45)	prairie	690.97 (69.10%)	691.01	1 (2.36h)

TABLE 3.4 – Résultats obtenus sur un parcellaire de taille 10×10 avec GD-LBP pour des objectifs simples ; la valeur MC représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique ; la valeur LBP est une approximation plus grande mais plus rapide de la valeur de la politique, utilisée au sein de l'algorithme d'optimisation.

3.7.2 Résultats pour des objectifs de compromis entre services

Dans cette section, nous nous intéresserons à deux services : celui de la production en blé et colza (objectif 1), et celui de la biodiversité en adventices et pollinisateurs (objectif 2). Nous essaierons de trouver des politiques d'allocation des cultures permettant de faire un compromis entre ces deux services à l'échelle du paysage.

1. Étude de quelques politiques 'expertes'

Dans un premier temps, nous allons étudier quelques politiques simples, qui viennent à l'esprit logiquement suite aux résultats obtenus pour des objectifs uniques :

- les monocultures de blé, de colza et de prairie (elles ne permettront pas de faire un compromis satisfaisant entre les deux services de production et de biodiversité mais serviront de référence) : notées 'blé', 'colza' ou 'prairie'
- la politique uniforme qui, chaque année et sur chaque parcelle, tire au hasard de manière uniforme entre les trois cultures possibles (blé, colza, prairie) : notée 'uniforme BCP'
- la politique uniforme qui, chaque année et sur chaque parcelle, tire au hasard de manière uniforme entre blé et prairie : notée 'uniforme BP'
- la politique déterministe qui consiste à alterner blé et prairie en forme de damier (sur une parcelle donnée, la culture ne change donc pas au cours du temps) : notée 'alternance BP'.

La figure 3.9 représente la valeur Monte-Carlo des différentes politiques considérées, pour l'objectif 1 et l'objectif 2, pour un parcellaire 10×10. Elles sont toutes sur le front de Pareto empirique (c'est-à-dire qu'elles sont non dominées par une autre stratégie parmi celles envisagées), sauf la politique uniforme blé-prairie.

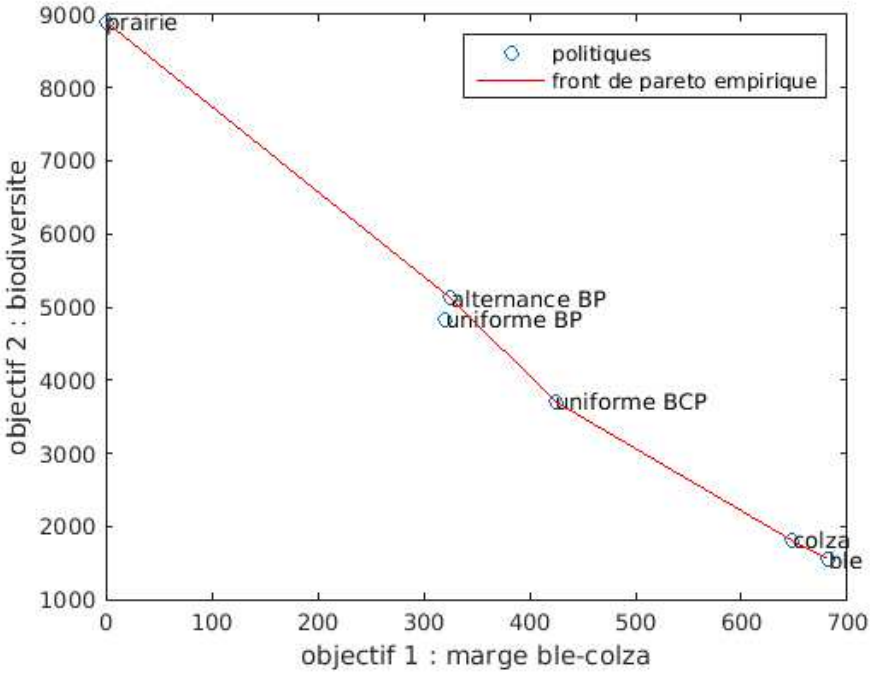


FIGURE 3.9 – Valeur Monte-Carlo des politiques expertes pour les objectifs 1 et 2 et représentation du front de Pareto empirique - parcellaire 10×10 - $T = 10$ - la valeur Monte-Carlo représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique.

2. Objectifs de type *land sparing/land sharing* (objectif C1)

Nous allons tout d'abord étudier les deux stratégies proposées dans la littérature : les stratégies *land sparing* et *land sharing* (voir section 3.2). Nous considérons que la stratégie *land sparing* consiste à optimiser le rendement sur une moitié du paysage, et la biodiversité sur l'autre moitié, et que la stratégie *land sharing* consiste à optimiser le rendement sur une parcelle, et la biodiversité sur ses parcelles adjacentes. Nous utilisons donc le terme de stratégie mais il s'agit plutôt de deux types d'objectif. Nous allons, en utilisant l'algorithme GD-LBP, étudier à quels types de stratégies, *ie* à quels types de politiques d'allocation des cultures, correspondent ces deux objectifs.

Avec la définition qu'on s'est donnée, il s'agit dans les deux cas de maximiser la marge en blé-colza sur les parcelles d'un ensemble E_1, et la biodiversité (adventices et pollinisateurs) sur les parcelles d'un ensemble E_2 tels que $E_1 \cup E_2 = \{1, ..., P\}$ et $E_1 \cap E_2 = \emptyset$. Les ensembles E_1 et E_2 différent selon l'objectif considéré :

- paysage *land sparing* : le paysage est divisé en deux dans le sens de la largeur avec une partie dont l'objectif est la marge économique et une partie dont l'objectif est la biodiversité (voir figure 3.10 gauche)
- paysage *land sharing* : on associe l'objectif de marge économique à une parcelle sur deux, et l'objectif de biodiversité à une parcelle sur deux (voir figure 3.10 droite).

La table 3.5 donne les résultats obtenus avec l'algorithme GD-LBP pour un parcellaire 10×10, sans aucune contrainte d'égalité entre politiques locales. L'objectif *land sparing* et l'objectif *land sharing* conduisent à la même politique localement optimale : mettre du blé sur les parcelles dont l'objectif est la marge économique, et de la prairie sur les parcelles dont l'objectif est la biodiversité. Un paysage de type *land sharing*, tel qu'on l'a défini, est donc un damier blé-prairie (politique experte que l'on a appelée 'alternance BP').

Ce qu'il est plus intéressant de constater est que l'objectif *land sparing* conduit à des résultats légèrement meilleurs sur le plan global, que ce soit pour la marge économique ou la biodiversité. Il y a plus d'adventices dans le blé du paysage *land sharing*, donc plus de compétition avec le blé, ce qui explique les moins bons résultats en termes de marge économique. Si on s'intéresse uniquement aux adventices, la stratégie *land sharing* est cependant gagnante.

Pour une structure de politique basée sur l'état de la parcelle, et pour des parcellaires plus petits, on obtient le même type de politiques et les mêmes conclusions. La stratégie *land sparing* domine donc (légèrement) la stratégie *land sharing* (qui correspond à 'alternance BP') : voir figure 3.11.

Maintenant, si on s'intéresse à la variabilité temporelle de chaque stratégie, mesurée par la moyenne sur les trajectoires simulées de la variance des récompenses obtenues à chaque pas de temps, on peut constater que cette variabilité est toujours inférieure dans le cas de la stratégie *land sharing* par rapport à la stratégie *land sparing* (sauf dans le cas des pollinisateurs dans les parcelles de blé). Si la stratégie *land sparing* domine légèrement la stratégie *land sharing* pour les objectifs de marge économique et de biodiversité, la stratégie *land sharing* offre donc quant à elle plus de stabilité dans le

temps de la marge économique et de la biodiversité.

Nous n'avons pas pris en compte le fait que dans les paysages de type *land sharing*, les pratiques agricoles sont supposées être plus respectueuses de la biodiversité (on parle de *wild-life friendly farming* [GCSB05]). Il serait intéressant de prendre en compte cela dans une future étude et de vérifier si les conclusions restent les mêmes.

Par ailleurs, les résultats pourraient être modifiés si on prenait en compte l'abondance des adventices et non seulement leur occurrence (en général, plus les adventices sont abondantes, plus elles sont nuisibles au rendement). Ou si, en gardant un modèle de dynamique basé sur les occurrences, on prenait des paramètres de dynamique différents pour chaque groupe.

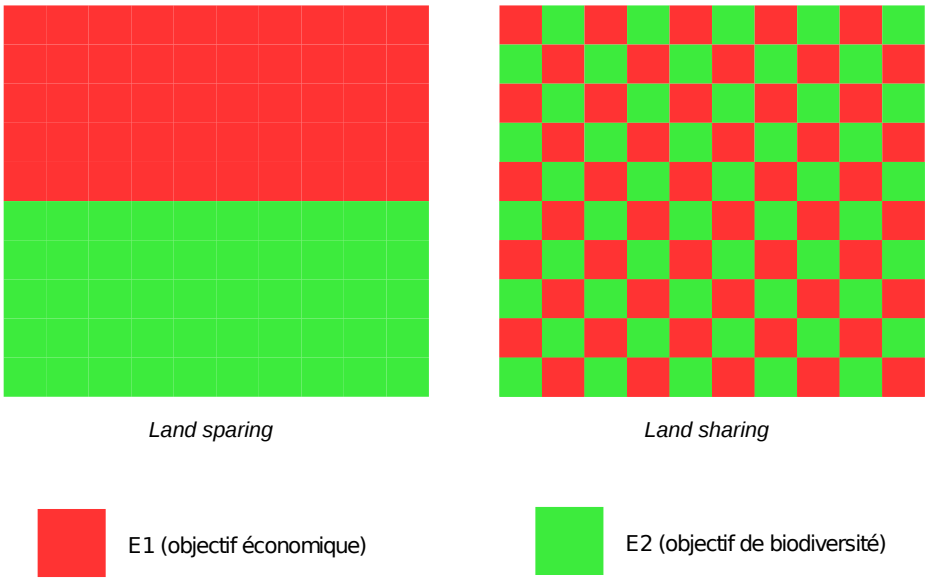


FIGURE 3.10 – Représentation de l'objectif C1 sur un parcellaire 10×10 (à gauche : *land sparing*, à droite : *land sharing*)

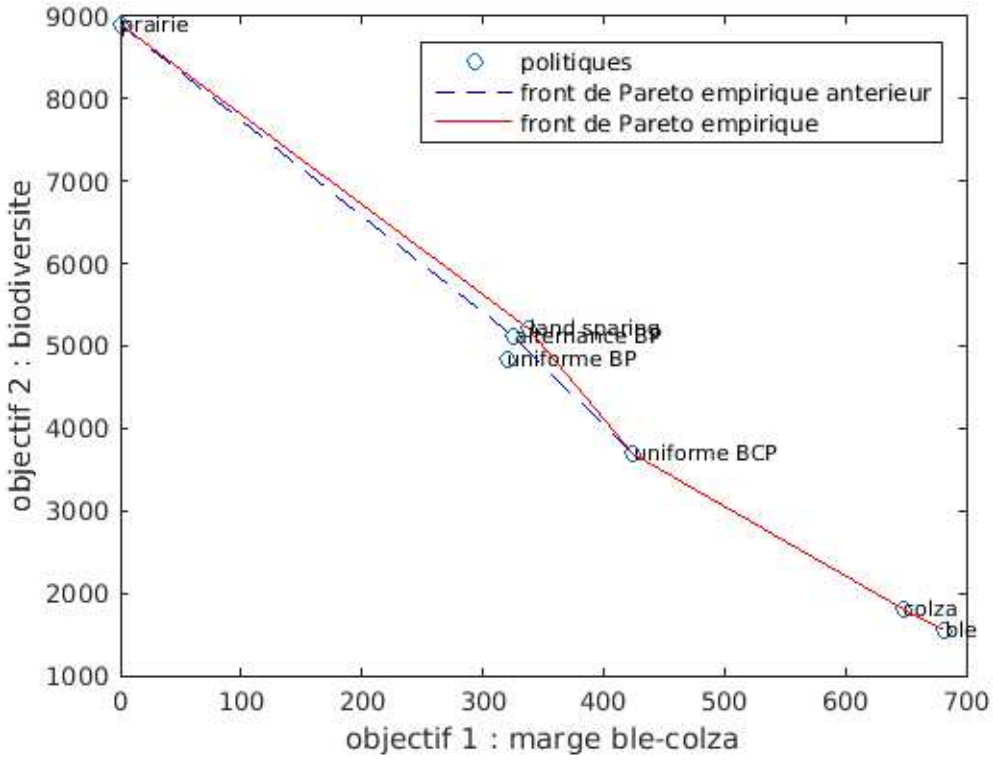


FIGURE 3.11 – Valeur Monte-Carlo des politiques pour les objectifs 1 et 2 - parcellaire 10×10 - $T = 10$ - ajout de la stratégie 'land sparing' - la valeur Monte-Carlo représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique.

objectif	*land sparing*	*land sharing*
nombre d'itérations	1	1
temps de calcul	14.05h	14.68h
objectif C1	**15.50** (**77.50**%)	13.08 (65.40%)
adventices sur blé	677.40 (27.10%) - **517.22**	**1167.2** (**46.69**%) - 117.27
adventices sur prairie	**2246** (**89.84**%) - **1392.4**	2076 (83.04%) - 920.28
total adventices	2923.4 (58.47%)	**3243.2** (**64.86**%)
pollinisateurs sur blé	165.84 (6.63%) - 8.50	**644.71** (**25.79**%) - **16.76**
pollinisateurs sur prairie	**2120.6** (**84.82**%) - **969.04**	1269.7 (50.79%) - 153.29
total pollinisateurs	**2286.4** (**45.73**%)	1914.4 (38.29%)
total biodiversité (objectif 2)	**5209.8** (**52.10**%)	5157.6 (51.58%)
marge économique (objectif 1)	**338.56** (**33.86**%) - **1.05**	319.48 (31.95%) - 0.14

TABLE 3.5 – Résultats obtenus avec l'algorithme GD-LBP pour l'objectif C1 - parcellaire 10×10 - structure de politique 0 (300 paramètres)
Le premier nombre représente la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique (estimation de la valeur par Monte-Carlo), le pourcentage entre parenthèses exprime le premier nombre en pourcentage de la valeur maximale possible pour l'objectif considéré ; enfin, le nombre après le tiret représente la moyenne sur les 4000 trajectoires de la variance des récompenses obtenues à chaque pas de temps.

3. Compromis de type 'durabilité' (domaines admissibles, objectif C2)

Nous allons maintenant proposer une autre manière de définir des objectifs permettant d'atteindre un compromis entre services à l'échelle du paysage. Nous proposons de définir des objectifs à l'échelle de l'exploitation (une exploitation est un ensemble de parcelles appartenant à un même agriculteur mais pas forcément adjacentes).

En guise d'exemple, nous essaierons de maximiser le nombre moyen d'années et d'exploitations pour lesquelles la marge en blé est supérieure à un certain seuil β (si la parcelle n'est pas en blé elle ne compte pas) et le nombre de groupes adventices est supérieur à un certain seuil ζ.

Nous avons envisagé trois configurations d'exploitations, dont les parcelles sont plus ou moins agrégées. Pour des raisons de mémoire et de temps de calcul, nous ne prenons pas en compte l'économie en utilisation d'azote. On considère la structure de politique 0.

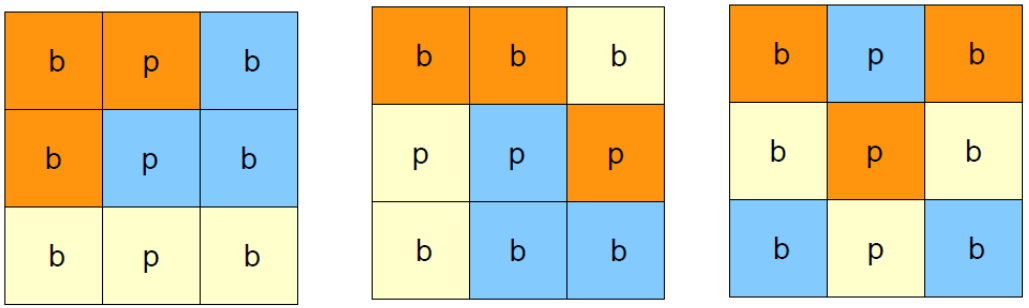
Parcellaire 3×3 : On considère tout d'abord un parcellaire de taille 3×3, avec $L_r = 3$ exploitations de 3 parcelles. On prend $\beta = 0.9$ (sur un maximum possible de 3) et $\zeta = 6$ (sur un maximum possible de $3I = 15$). Le point de départ est uniforme et on prend $\epsilon_g = 10^{-6}$ (paramètre de précision pour la norme du gradient). Quelle que soit la configuration, on obtient avec GD-LBP deux blés et une prairie par exploitation : voir figure 3.12 et table 3.6. On obtient une valeur Monte-Carlo de 25.86 sur $L_r T = 30$, donc l'objectif est atteint à 86.2%.

Le fait que les prairies soient à chaque fois disposées au milieu est certainement du

au hasard, car nous avons comparé avec un parcellaire avec deux blés et une prairie par exploitation, mais où les prairies ne sont pas toutes sur la deuxième ligne ou la deuxième colonne, et cela ne semble pas faire de différence en termes de valeur Monte-Carlo (même avec 10000 trajectoires simulées).

configuration	politique	Vmc obj C2	Vmc obj 1	Vmc obj 2	itérations (temps)
très agrégée	voir figure 3.12	25.86 (86.2%)	40.15 (44.61%)	339.31 (37.70%)	52 (37h)
moyennement agrégée	voir figure 3.12	25.86 (86.2%)	40.15 (44.61%)	339.31 (37.70%)	52 (36h)
peu aggrégée	voir figure 3.12	25.86 (86.2%)	40.15 (44.61%)	339.31 (37.70%)	58 (39h)

TABLE 3.6 – Résultats avec GD-LBP pour l'objectif C2 pour les différentes configurations d'exploitations testées - parcellaire 3×3 - $I = B = 5$ - structure de politique 0 (27 paramètres) - la valeur Monte-Carlo (Vmc) représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique.



(a) Très aggrégé (b) Moyennement aggrégé (c) Peu aggrégé

FIGURE 3.12 – Politiques obtenues avec GD-LBP pour l'objectif C2 pour les différentes configurations d'exploitations testées - parcellaire 3×3 - $I = B = 5$ - structure de politique 0 (27 paramètres) - chaque couleur correspond à une exploitation - b=blé, p=prairie

Parcellaire 6×6 : Nous allons vérifier si la stratégie optimale est la même pour un parcellaire 6×6, avec $L_r = 12$ exploitations de 3 parcelles. Pour des raisons d'espace mémoire, nous sommes obligés de travailler avec $I = B = 3$ (réseau trophique de la figure 3.13). On prend $\beta = 0.9$ (sur un maximum possible de 3) et $\zeta = 4$ (sur un maximum possible de $3I = 9$). On garde donc les mêmes rapports que pour le parcellaire 3×3. On prend $\epsilon_g = 10^{-2}$ (paramètre de précision pour la norme du gradient). Avec un point de départ uniforme, on obtient du blé partout pour les trois configurations. Avec un point de départ aléatoire, on obtient deux blés et une prairie par exploitation quelle que soit la configuration des exploitations, et cela donne de meilleurs résultats : voir résultats figure 3.14 et table 3.7. Les différences observées en termes de valeur Monte-Carlo entre les différentes configurations d'exploitations ne sont pas significatives. L'objectif est atteint à 25% seulement, contre 86.2% dans le cas du parcellaire 3×3.

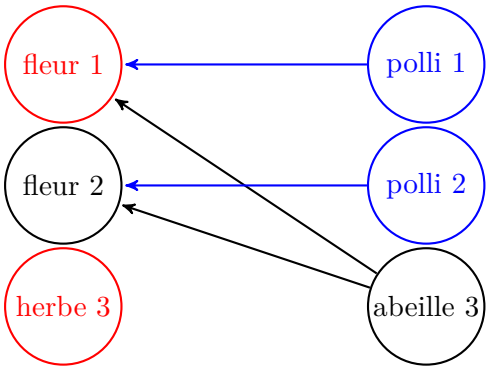
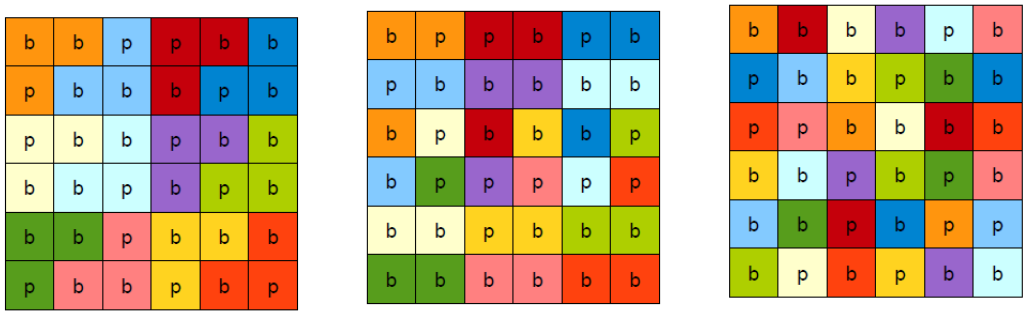


FIGURE 3.13 – Réseau trophique dans le cas où $I = B = 3$; en rouge les groupes adventices les plus nuisibles au rendement ; le groupe 1 fleurit au printemps-été et le groupe 2 toute l'année

configuration	politique	Vmc obj C2	Vmc obj 1	Vmc obj 2	itérations (temps)
point de départ uniforme					
très agrégée	blé	22.816 (6.34%)	261.94 (72.76%)	329.24 (15.24%)	1 (33 min)
moyennement agrégée	blé	22.816 (6.34%)	261.94 (72.76%)	329.24 (15.24%)	1 (33 min)
peu agrégée	blé	22.816 (6.34%)	261.94 (72.76%)	329.24 (15.24%)	1 (32 min)
point de départ aléatoire					
très agrégée	voir figure 3.14	92.67 (25.75%)	169.33 (47.04%)	825.54 (38.22%)	6 (55 min)
moyennement agrégée	voir figure 3.14	91.34 (25.37%)	170.36 (47.32%)	825.26 (38.21%)	9 (1.24 h)
peu agrégée	voir figure 3.14	93.7 (26.03%)	169.65 (47.12%)	827.70 (38.32%)	8 (1.09 h)

TABLE 3.7 – Résultats avec GD-LBP pour l'objectif C2 pour les différentes configurations d'exploitations testées - parcellaire 6×6 - $I = B = 3$ - structure de politique 0 (108 paramètres) - la valeur Monte-Carlo (Vmc) représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique.



(a) Très aggrégé (b) Moyennement aggrégé (c) Peu aggrégé

FIGURE 3.14 – Politiques obtenues avec GD-LBP initialisé avec une politique aléatoire - parcellaire 6×6 - $I = B = 3$ - structure de politique 0 (108 paramètres) - chaque couleur correspond à une exploitation - b=blé, p=prairie

Conclusion : La politique obtenue semble stable vis-à-vis du nombre de groupes d'adventices et de pollinisateurs et vis-à-vis de la taille du parcellaire. Cette expérience montre qu'il est possible de faire des compromis entre services à l'échelle du paysage en définissant des objectifs à l'échelle de l'exploitation. En effet, elle nous a permis, par optimisation, de repousser le front de Pareto empirique : voir figure 3.15. La politique avec deux blés pour une prairie (notée '2b1p') domine la politique uniforme blé-colza-prairie (il en va de même pour toutes les tailles de grilles que nous avons testées). Cette stratégie avec deux blés et une prairie peut être vue comme une stratégie de type *land sharing*, mais moins extrême que celle que nous avons testée avec l'objectif C1. Cela montre qu'entre les deux stratégies extrêmes *land sparing* et *land sharing*, il peut y avoir des stratégies intermédiaires permettant d'atteindre des compromis à l'échelle du paysage plus satisfaisants.

Remarque : Nous avons représenté les valeurs Monte-Carlo des différentes stratégies pour un parcellaire 10×10 et un horizon $T = 20$ et nous obtenons une représentation très similaire à celle obtenue pour un horizon $T = 10$: voir figure 3.16. Avoir utilisé un horizon $T = 10$ dans les expériences précédentes n'était donc pas un mauvais choix pour rechercher les meilleures stratégies à long terme.

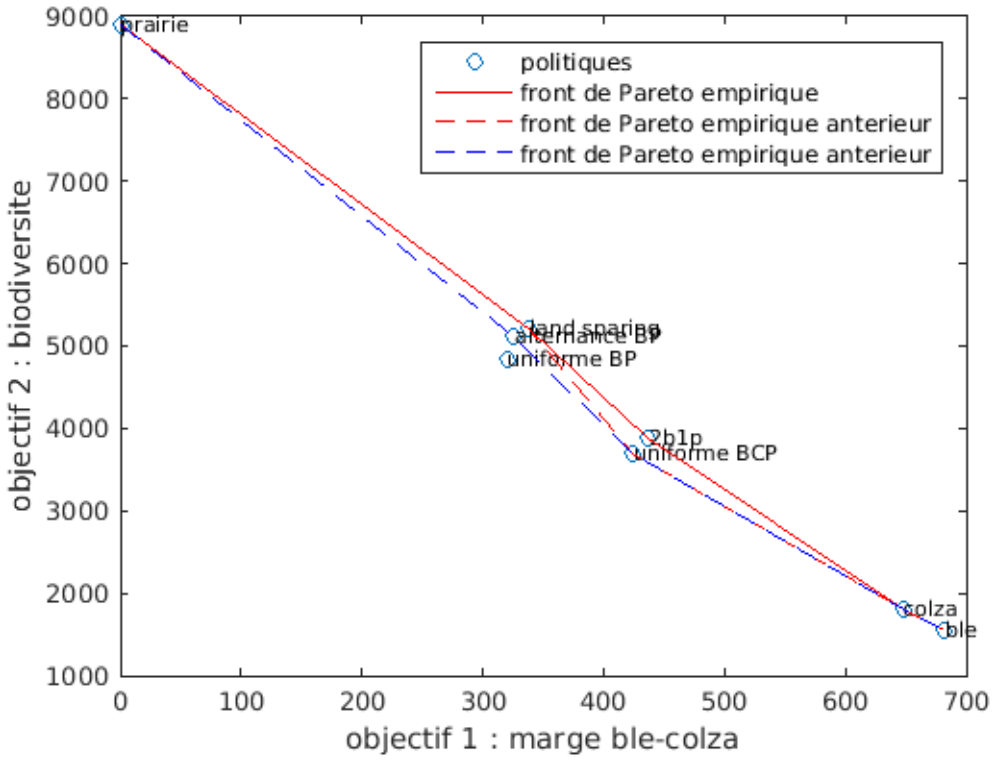


FIGURE 3.15 – Valeur Monte-Carlo des politiques pour les objectifs 1 et 2 - parcellaire 10×10 - $T = 10$ - ajout de la stratégie '2b1p' (deux blés pour une prairie) - la valeur Monte-Carlo représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique.

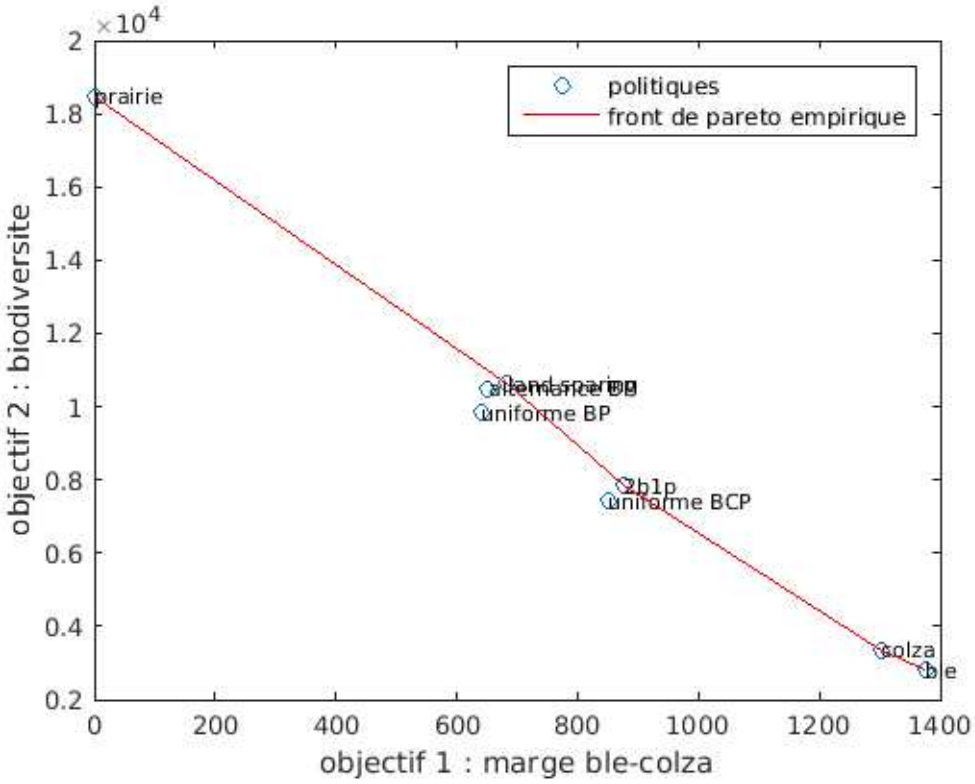


FIGURE 3.16 – Valeur Monte-Carlo des politiques pour les objectifs 1 et 2 - parcellaire 10×10 - $T = 20$ - la valeur Monte-Carlo représente l'évaluation de la politique par la méthode de Monte-Carlo, c'est-à-dire la moyenne empirique, sur 4000 trajectoires simulées, de la valeur de la politique.

3.8 Conclusion et perspectives

3.8.1 Conclusion

Dans ce chapitre, nous avons démontré la capacité de l'algorithme GD-LBP à traiter des problèmes de très grande taille (avec un espace d'état de taille $32^{100} = 2^{500}$ et un espace d'action de taille 3^{100} dans le cas d'un parcellaire 10×10). Nous avons défini un modèle de transition sur l'occurrence de différents groupes d'espèces adventices dans le paysage, et un modèle d'abondance de différents groupes de pollinisateurs, sauvages et domestiques. Nous avons également défini un ensemble de fonctions de récompense possibles, associées à différentes formes de compromis entre services écosystémiques. A partir de ces modèles, nous avons pu par optimisation tirer des hypothèses sur les stratégies d'allocation des cultures permettant d'atteindre des compromis entre services écosystémiques. Ces hypothèses sont les suivantes :

— un paysage de type *land sparing* est plus favorable à la marge économique des céréaliers et à la conservation de la biodiversité qu'un paysage de type *land sharing* (ce résultat va à l'encontre des résultats de [MDGMJ07], peut-être parce que nous n'avons pas pris en compte les différences de pratiques agricoles dans les paysages *land sparing* et *land sharing*, et que nous avons pris en compte les pollinisateurs en plus des adventices)

— il est possible de faire des compromis entre services écosystémiques à l'échelle du paysage en définissant des objectifs à l'échelle de l'exploitation, et il existe des stratégies satisfaisantes dans le *continuum land sparing-land sharing*.

Ces hypothèses seront à vérifier sur un modèle plus réaliste. Le travail de modélisation effectué nous a aussi permis de mettre en valeur les manques de connaissances en écologie qu'il faudrait combler pour préciser au mieux les valeurs de paramètres du modèle Cultures-Adventices-Pollinisateurs. Il s'agit en particulier des paramètres de dynamique, puisque les paramètres de marge économique peuvent s'obtenir à partir d'études de bases de données. Il faudrait notamment préciser la part de la reproduction, de la dispersion spatiale et de la dispersion temporelle dans la dynamique des adventices.

3.8.2 Perspectives

Il serait intéressant dans un futur proche d'étudier la sensibilité des résultats que nous avons obtenus aux valeurs des paramètres de dynamique adventice ou aux valeurs des paramètres de récompense (paramètres de modélisation économique notamment). En particulier, le fait que les conclusions soient les mêmes quelle que soit la taille de grille et quelle que soit la configuration des exploitations, laisse penser que l'aspect spatial du modèle ne joue pas un grand rôle. Il serait intéressant de tester si les conclusions sont les mêmes avec des paramètres de dispersion courte distance et longue distance des adventices plus importants. Ces paramètres sont en effet particulièrement mal connus des experts, et il se pourrait que la dispersion longue distance soit plus importante que ce qu'il était admis jusqu'à présent [PAC+13]. Nous pourrions aussi analyser l'effet d'un changement de réseau trophique (certains groupes adventices pourraient être visités par

plusieurs groupes de pollinisateurs sauvages par exemple).

A moyen terme, nous aimerions améliorer le modèle pour le rendre plus réaliste. Nous pourrions envisager un modèle de dynamique basé sur l'abondance plutôt que la présence-absence des groupes adventices. Cela permettrait notamment de mieux préciser leur effet sur le rendement et sur le maintien de la biodiversité. Nous pourrions aussi prendre en compte la compétition entre groupes adventices, et leur résistance plus ou moins importante aux perturbations, comme dans [DLP09]. Nous pourrions prendre en compte la dynamique des pollinisateurs, et intégrer des habitats boisés qui sont importants pour les pollinisateurs sauvages. Du côté agronomique, nous pourrions prendre en compte le fait que les prairies restent en général trois années consécutives en place, et que la flore adventice est réduite dans une parcelle de céréale qui suit une prairie [MMW$^+$10]. Il serait intéressant également d'envisager des parcelles ou des exploitations biologiques plutôt que conventionnelles. En effet, la flore adventice est souvent plus diverse et abondante dans ces parcelles, et les capacités de production ne sont pas les mêmes [GSKB13]. Nous aimerions aussi prendre en compte d'autres fonctions écologiques des adventices (ressource trophique pour les oiseaux et les insectes). Et considérer des structures de politique plus proches des manières de décider des agriculteurs, en choisissant la culture sur une parcelle en fonction de la ou des cultures mises en place l'année précédente, pour pouvoir fournir des règles de rotation, éventuellement stochastiques. Le cadre PDMF^3 permet de considérer ce type de structures de politiques. Enfin, il serait intéressant d'appliquer notre modèle à un paysage réel, et de comparer les stratégies obtenues avec les assolements observés.

De manière plus générale, étant donné que nous avons affaire à des problèmes multi-objectifs (faisant intervenir plusieurs services écosystémiques), il pourrait être intéressant de mettre en œuvre des méthodes spécifiques à ce type de problèmes (par exemple des *constrained MDP* [Alt99b]). Cependant, résoudre des processus décisionnels de Markov multi-objectifs est une question de recherche en soi [Wan14], et combiner cette problématique avec celle de la taille des espaces d'état et d'action aurait été trop complexe dans un premier temps.

Conclusion générale

Résultats obtenus et discussion

Nous avons proposé un nouveau cadre de PDM à espace d'état et d'action factorisés, caractérisé par la recherche de politiques stochastiques factorisées de structure donnée : le cadre PDMF^3. Nous avons montré que la résolution de ce type de problèmes était NP$^{\text{PP}}$-difficile. Nous avons proposé une famille d'algorithmes de résolution approchée, de type itération de la politique. Notre approche présente l'originalité de mêler des méthodes d'inférence dans les modèles graphiques pour l'évaluation (comme dans les méthodes de type *planning as inference* [TS06]), et des méthodes d'optimisation continue (comme dans les méthodes de type *policy gradient* [PKMK00]). Sa généricité permet d'intégrer facilement toute nouvelle méthode d'inférence dans les modèles graphiques, ou toute nouvelle méthode d'optimisation continue, afin d'améliorer la qualité ou la rapidité de la résolution. Les algorithmes proposés sont à disposition en ligne sous forme de *solver* Matlab [3]. Dans le chapitre 2, nous avons démontré la qualité de ces algorithmes sur de petits problèmes en comparaison avec la politique globale optimale, et sur de grands problèmes en comparaison avec un algorithme de l'état de l'art ou des politiques expertes.

Dans le chapitre 3, nous avons démontré la capacité de ces algorithmes à traiter des problèmes de très grande taille (avec un espace d'état de taille $32^{100} = 2^{500}$ et un espace d'action de taille 3^{100}). Nous avons aussi montré dans ce chapitre l'intérêt du cadre PDMF^3 pour la modélisation de problèmes de gestion des services écosystémiques à l'échelle du paysage. Nous nous sommes intéressés à un problème particulier, autour des services fournis par les cultures agricoles, les adventices et les pollinisateurs dans un paysage de grandes cultures. Nous avons défini un modèle de transition sur la présence/absence de différents groupes d'espèces adventices dans le paysage, sous l'effet d'actions (allocations spatio-temporelles des cultures), et un modèle d'abondance de différents groupes de pollinisateurs, sauvages et domestiques. Nous avons également défini un ensemble de fonctions de récompense possibles, associées à différentes formes de compromis entre services écosystémiques. A partir de ces modèles, nous avons pu par optimisation tirer des hypothèses sur les stratégies d'allocation des cultures permettant d'atteindre des compromis entre services écosystémiques. Ces hypothèses seront à vérifier sur un modèle plus réaliste, et éventuellement par expérimentation. Le travail de

3. `https://mulcyber.toulouse.inra.fr/projects/f3mdpsolver/`

modélisation effectué a aussi permis de mettre en valeur les manques de connaissances en écologie qu'il faudrait combler pour préciser au mieux les valeurs de paramètres du modèle Cultures-Adventices-Pollinisateurs.

Finalement, l'évaluation-comparaison de stratégies, sur les problèmes que nous avons considérés (dans le chapitre 2 ou le chapitre 3), donne souvent d'aussi bons voire de meilleurs résultats que l'approche par optimisation que nous avons proposée, et en moins de temps. De plus, comme nous utilisons un algorithme d'optimisation local dont l'évaluation est approchée (cas de l'évaluation basée sur *Loopy Belief Propagation*), nous n'avons pas avec notre approche de garantie d'optimalité de la stratégie retournée. Cependant, il se peut que l'absence de différence entre l'approche optimisation et l'approche évaluation-comparaison soit liée à une trop grande facilité de résolution des problèmes que nous avons envisagés. De plus, l'approche optimisation permet de vérifier qu'une stratégie experte est bien un point critique, en l'utilisant comme point de départ de l'algorithme d'optimisation, ou éventuellement de l'améliorer.

Perspectives

Méthodologiques

Comme nous l'avons vu dans la section 2.6, plusieurs perspectives à la partie méthodologique de la thèse sont envisageables. Premièrement, nous pourrions étendre notre approche aux politiques stochastiques factorisées non stationnaires. En effet, la meilleure politique stochastique factorisée n'est pas forcément stationnaire. Cela permettrait peut-être, dans l'application en agroécologie, d'obtenir des paysages qui varient au cours du temps, avec de meilleures performances. Nous pourrions également étendre notre approche au cas général des Dec-POMDPs à espace d'état factorisé, et comparer nos algorithmes aux approches récentes proposées dans ce cadre (par exemple [OWS13] ou [PP11]). Optimiser la structure de la politique stochastique factorisée, au lieu de faire l'hypothèse qu'elle fait partie des données du problème, serait une perspective intéressante mais plus complexe à mettre en œuvre, puisqu'elle demande de proposer une nouvelle approche, pour un problème d'optimisation mixte, avec des variables discrètes (pour la structure) et continues (pour les paramètres).

Appliquées

Sur la partie appliquée de la thèse, qui s'inscrit dans le projet ANR AgrobioSE [4], les suites à envisager doivent permettre de passer d'un modèle illustratif à un modèle pour l'aide à la décision qui pourra servir de support pour tester les hypothèses des écologues.

Dans un premier temps, nous étudierons la sensibilité des stratégies retournées par notre algorithme d'optimisation aux paramètres de dynamique et de récompense (paramètres économiques notamment). Une analyse de sensibilité utilisant les outils mathé-

4. Biodiversité et services écosystémiques en agroécosystèmes céréaliers intensifs : utilisation des concepts de l'agro-écologie pour atteindre les objectifs ECOPHYTO 2018

matiques appropriés [FIM+13] n'est pas possible, à cause du temps de calcul nécessaire à l'optimisation de la stratégie pour un jeu de paramètres donnés notamment. Mais nous pourrons envisager différents scénarios pour les valeurs de paramètres.

Nous envisagerons différentes améliorations du modèle pour le rendre plus réaliste (prise en compte de la dynamique des pollinisateurs, et des bordures et des haies par exemple), ainsi qu'une application à un paysage réel. Il est aussi prévu que le modèle proposé dans le chapitre 3 serve de support de discussion avec des acteurs de la zone atelier *Plaine et Val de Sèvre*[5] (agriculteurs, apiculteurs, citoyens), pour éventuellement envisager une co-construction de modèles de récompense et de stratégies d'allocation des cultures (conception participative [MMCD13]).

5. `http://www.zaplainevaldesevre.fr/`

Valorisation du travail de thèse

Les travaux présentés dans ce manuscrit ont fait l'objet des publications et communications suivantes :

Article de revue :

- [TPA+13] P. Tixier, N. Peyrard, J.N. Aubertot, S. Gaba, J. Radoszycki, G. Caron-Lormier, F. Vinatier, G. Mollot, and R. Sabbadin. Chapter Seven - Modelling interaction networks for enhanced ecosystem services in agroecosystems. *Advances in Ecological Research*, 49 : 437-480, 2013.

Articles de conférence avec comité de relecture :

- [RPS15a] J. Radoszycki, N. Peyrard and R. Sabbadin. Résolution de PDMF^3 : processus décisionnels de Markov à transitions, récompenses et politiques stochastiques factorisées. In *JFPDA, Rennes*, 2015. [présentation orale]
- [RPS14] J. Radoszycki, N. Peyrard and R. Sabbadin. Finding good stochastic factored policies for factored Markov decision processes. In *ECAI, Prague*, pages 1083-1084, 2014. [poster]
- [RPS15b] J. Radoszycki, N. Peyrard and R. Sabbadin. Solving F^{3}MDPs : collaborative multiagent Markov decision processes with factored transitions, rewards and stochastic policies. In *PRIMA, Bertinoro*, 2015. [présentation orale]

Communications :

- [Poster] Décision séquentielle sous incertitude - Application en agroécologie. *Doctoriales Midi-Pyrénées, Albi*, 2015.
- [Présentation orale] Optimisation à l'échelle du paysage pour un compromis entre services écosystémiques. *séminaire des doctorants de l'unité MIAT, Toulouse*, 2015.
- [Présentation orale] Optimisation à l'échelle du paysage pour un compromis entre services écosystémiques. *séminaire des doctorants de l'UMR Agroécologie, Dijon*, 2015.
- [Poster] Finding good stochastic factored policies for factored Markov decision processes. In *Journées MAS (Modélisation Aléatoire Statistique), Toulouse*, 2014.
- [Présentation orale] Résolution approchée de processus décisionnels de Markov factorisés, *séminaire de l'unité MIAT*, 2014.

- [Présentation orale] Résolution approchée de processus décisionnels de Markov factorisés, *séminaire DocToMe des doctorants en informatique de Toulouse*, 2014.
- [Présentation orale] Evaluation of stochastic policies for factored Markov decision processes by normalizing constants computation - *3rd Workshop on Algorithmic issues for Inference in Graphical Models (AIGM), Paris*, 2013.

Bibliographie

[ABZ07] C. Amato, D. S. Bernstein, and S. Zilberstein. Optimizing memory-bounded controllers for decentralized POMDPs. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2007.

[ADC07] R. Aras, A. Dutech, and F. Charpillet. Mixed integer linear programming for exact finite-horizon planning in decentralized POMDPs. In *International Conference on Automated Planning and scheduling*, 2007.

[ALG+99] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Optimization With Variable-Fidelity Models Applied to Wing Design. Technical report, 1999.

[Alt99a] M. A. Altieri. The ecological role of biodiversity in agroecosystems. *Agriculture, Ecosystems and Environment*, (74) :19–31, 1999.

[Alt99b] E. Altman. *Constrained Markov decision processes*. Chapman and Hall, 1999.

[Ass05] Millenium Ecosystem Assessment. Millenium ecosystem assessment, ecosystems and human well-being : a framework for assessment. World Resources Institute, 2005.

[ATK12] B. Andres, Beier T., and J. H. Kappes. OpenGM : A C++ library for discrete graphical models. *ArXiv e-prints*, 2012.

[Att03] H. Attias. Planning by probabilistic inference. In *Proceedings of the 9th international workshop on Artificial intelligence and Statistics*, AISTATS'03, 2003.

[BA09] O. Buffet and D. Aberdeen. The factored policy-gradient planner. *Artificial Intelligence*, 173 :722–747, 2009.

[BAD08] B. J. Brosi, P. R. Armsworth, and G. C. Daily. Optimal design of agricultural landscapes for pollination services. *Conservation letters*, 1 :27–36, 2008.

[BCSM08] A. Beynier, F. Charpillet, D. Szer, and A-I. Mouaddib. Dec-MDP/POMDP. In *Processus décisionnels de Markov en intelligence artificielle (volume 2)*, chapter 2, pages 51–79. Lavoisier, 2008.

[BDG95] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, IJCAI'95, pages 1104–1111, 1995.

[BDG00] C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121 :49–107, 2000.

[Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[Ben28] B. M. Bensin. *Agroecological characteristics description and classification of the local corn varieties chorotypes*. Publisher unknown so far, 1928.

[BG03] B. Bonet and H. Geffner. Labeled RTDP : improving the convergence of real-time dynamic programming. In *Proceedings of the 13th International Conference on Automated Planning and Scheduling*, page 12–21, 2003.

[BG15] V. Bretagnolle and S. Gaba. Weeds for bees ? A review. *Agronomy for a Sustainable Development*, pages 1–19, 2015.

[BGIZ02] D.S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4) :819–840, 2002.

[BHM+13] I. J. Bateman, A. R. Harwood, G. M. Mace, R. T. Watson, D. J. Abson, B. Andrews, A. Binner, A. Crowe, B. H. Day, S. Dugdale, C. Fezzi, J. Foden, D. Hadley, R. Haines-Young, M. Hulme, A. Kontoleon, A. A. Lovett, P. Munday, U. Pascual, J. Paterson, G. Perino, A. Sen, G. Siriwardena, D. van Soest, and M. Termansen. Bringing Ecosystem Services into Economic Decision-Making : Land Use in the United Kingdom. *Science*, 341(6141) :45–50, 2013.

[BHZ05] D. S. Bernstein, E. A. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized POMDPs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1287–1292, 2005.

[Bis07] C. M. Bishop. Pattern Recognition and Machine Learning. chapter Graphical models. Springer, 2007.

[BKVH07] S. Boyd, S-J. Kim, L. Vandenberghe, and A. Hassibi. A tutorial on geometric programming. *Optimization and Engineering*, 8(1) :67–127, 2007.

[Bou99] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *In International Joint Conference on Artificial Intelligence*, pages 478–485, 1999.

[BPG09] E. M. Bennett, G. D. Peterson, and L. J. Gordon. Understanding relationships among multiple ecosystem services. *Ecology Letters*, 12 :1394–1404, 2009.

[BPH+05] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, and A. Mihailidis. A decision-theoretic approach to task assistance for persons with dementia. In *IJCAI 2005*, pages 1293–1299, 2005.

[BT12] M. Botvinick and M. Toussaint. Planning as inference. *Trends in Cognitive Sciences*, 16 :485–488, 2012.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BZL04] R. Becker, S. Zilberstein, and V. R. Lesser. Decentralized Markov decision processes with event-driven interactions. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 302–309, 2004.

[BZLG04] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Solving transition independent decentralized Markov decision processes. *Journal of Artificial Intelligence Research*, 22 :423–455, 2004.

[CCC+14] I. Chades, G. Chapron, M. J. Cros, F. Garcia, and R. Sabbadin. MDPtoolbox : a multi-platform toolbox to solve stochastic dynamic programming problems. *Ecography*, 37 :916–920, 2014.

[CCTKL13] C. P. Carvalho Chanel, F. Teichteil-Königsbuch, and C. Lesire. Multi-target detection and recognition by UAVs using online POMDPs. In *AAAI Conference on Artificial Intelligence*, 2013.

[CGODJ11] E. D. Chambo, R. C. Garcia, N. T. E. de Oliveira, and J. B. Duarte-Junior. Honey bee visitation to sunflower : effects on pollination and plant genotype. *Scientia Agricola*, 68 :647 – 651, 2011.

[CLCI13] Q. Cheng, Q. Liu, F. Chen, and A. Ihler. Variational planning for graph-based MDPs. In *Advances in Neural Information Processing Systems 26*, pages 2976–2984, 2013.

[CM12] A. Canu and A-I. Mouaddib. Dynamic local interaction model. *Revue d'intelligence artificielle*, 26(5) :495–521, 2012.

[CSC+06] K. M. A. Chan, M. R. Shaw, D. R. Cameron, E. C. Underwood, and G. C. Daily. Conservation planning for ecosystem services. *Plos Biology*, 4(11) :2138–2152, 2006.

[DABC13] J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Optimally solving Dec-POMDPs as continuous-state MDPs. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 90–96. AAAI Press, 2013.

[DABC14] J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Exploiting separability in multiagent planning with continuous-state MDPs. In *Proceedings of the 13th international conference on Autonomous Agents and Multiagent Systems*, 2014.

[DAD12] J. S. Dibangoye, C. Amato, and A. Doniec. Scaling up decentralized MDPs through heuristic search. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, pages 217–226, 2012.

[DB98] R. Drechsler and B. Becker. *Binary Decision Diagrams - Theory and Implementation*. Springer, 1998.

[DD06] D.A. Dolgov and E.H. Durfee. Symmetric approximate linear programming for factored MDPs with application to constrained problems. *Annals of Mathematics and Artificial Intelligence*, 47 :273–293, 2006.

[Deg07] T. Degris. *Apprentissage par renforcement dans les processus markoviens factorisés*. PhD thesis, Université Paris VI, 2007.

[dFVR03] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6) :850–865, 2003.

[dFVR04] D. P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3) :462–478, 2004.

[DK89] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5 :142–150, 1989.

[DKRP10] R. M. Dorazio, M. Kery, J. A. Royle, and M. Plattner. Models for inference in dynamic metacommunity systems. *Ecology*, 91(8) :2466–2475, 2010.

[DLP09] M. Drechsler, R. Lourival, and H. P. Possingham. Conservation planning for successional landscapes. *Ecological modelling*, 220 :438–450, 2009.

[DLR77] A.P. Dempster, M.N. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 39 :1–22, 1977.

[DS08] T. Degris and O. Sigaud. Représentations factorisées. In *Processus décisionnels de Markov en intelligence artificielle (volume 2)*, chapter 2, pages 51–79. Lavoisier, 2008.

[EBK06] R. Enkhbat, B. Barsbold, and M. Kamada. A numerical approach for solving some convex maximization problems. *Journal of Global Optimization*, 35 :85–101, 2006.

[EG09] F. Eaton and Z. Ghahramani. Choosing a variable to clamp : approximate inference using conditioned belief propagation. In *Proceedings of the 12th international conference on Artificial intelligence and statistics*, 2009.

[EM12] J. F. Egan and D. A. Mortensen. A comparison of land-sharing and land-sparing strategies for plant richness conservation in agricultural landscapes. *Ecological Applications*, 22(2) :459–471, 2012.

[FAB+14] J. Fischer, D. J. Abson, V. Butsic, M. J. Chappell, J. Ekroos, J. Hanpspach, T. Kuemmerle, H. G. Smith, and H. von Wehrden. Land sparing versus land sharing : moving forward. *Conservation Letters*, 7(3) :149–157, 2014.

[FB09] T. Furmston and D. Barber. Solving deterministic policy (PO)MDPs using Expectation-Maximisation and antifreeze. In *Proceedings of the European Conference on Machine Learning (LEMIR workshop)*, 2009.

[FB10] T. Furmston and B. Barber. Variational methods for reinforcement learning. *Journal of Machine Learning Research - Proceedings Track*, 9 :241–248, 2010.

[FB12] T. Furmston and D. Barber. A unifying perspective of parametric policy search methods for Markov decision processes. *Proceedings of the annual conference on Neural Information Processing Systems*, 2012.

[FBD+08] J. Fischer, B. Brosi, G. C. Daily, P. R. Ehrlich, R. Goldman, J. Goldstein, D. B. Lindenmayer, A. D. Manning, H. A. Mooney, L. Pejchar, J. Ranganathan, and H. Tallis. Should agricultural policies encourage land sparing or wildlife-friendly farming ? *Frontiers in Ecology and The Environment*, 6 :380–385, 2008.

[FCX08] G. Fried, B. Chauvel, and Reboud X. Evolution de la flore adventice des champs cultivés au cours des dernières décennies : vers la sélection de groupes d'espèces répondant aux systèmes de culture. *Innovations agronomiques*, 3 :15–26, 2008.

[FGS09] N. Forsell, F. Garcia, and R. Sabbadin. Reinforcement learning for spatial processes. In *Proceedings of the international Congress on Modelling and Simulation*, pages 755–761, 2009.

[FH02] Z. Feng and E. A. Hansen. Symbolic Heuristic Search for Factored Markov Decision Processes. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 455–460, 2002.

[FIM+13] R. Faivre, B. Iooss, S. Mahévas, D. Makowski, and H. Monod. *Analyse de sensibilité et exploration de modèles - Application aux sciences de la nature et de l'environnement.* Quae, 2013.

[FKG12] G. Fried, E. Kazakou, and S. Gaba. Trajectories of weed communities explained by traits associated with species' response to management practices. *Agriculture, Ecosystems and Environment*, 158 :147–155, 2012.

[FM98] B. Frey and D. MacKay. A revolution : Belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems*, pages 479–485, 1998.

[FS06] N. Forsell and R. Sabbadin. Approximate linear-programming algorithms for graph-based markov decision processes. In *Proceedings of the 17th European Conference on Artificial Intelligence*, pages 590–594, 2006.

[GBB04] E. Greensmith, P. Barlett, and J. Baxter. Variance reduction techniques for gradient based estimates in reinforcement learning. *Journal of Machine Learning Research*, 5 :1471–1530, 2004.

[GBHT03] B. Gerowitt, E. Bertke, S-K. Hespelt, and C. Tute. Towards multifunctional agriculture - weeds as ecological goods ? *Weed Research*, 43 :227–235, 2003.

[GCSB05] R. E. Green, S. J. Cornell, J. P. Scharlemann, and A. Balmford. Farming and the fate of wild nature. *Science*, 307(5709) :550–555, 2005.

[GD13] V. Gogate and P. Domingos. Structured message passing. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 2013.

[GKP01] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*, pages 1523–1530, 2001.

[GKPV03] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research (JAIR)*, 19 :399–468, 2003.

[GLB+15] S. Gaba, F. Lescourret, S. Boudsocq, J. Enjalbert, P. Hinsinger, E-P. Journet, M-L. Navas, J. Wery, G. Louarn, E. Malézieux, E. Pelzer, M. Prudent, and H. Ozier-Lafontaine. Multiple cropping systems as drivers for providing multiple ecosystem services : from concepts to design. *Agronomy for Sustainable Development*, 35(2) :607–623, 2015.

[GLP02] C. Guestrin, M. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In *Proceedings of the 19th international conference on Machine learning*, ICML '02, 2002.

[God84] I. Godinho. Les définitions d' 'adventice' et de 'mauvaise herbe'. *Weed Research*, 24 :121–125, 1984.

[Gom09] C. P. Gomes. Computational Sustainability : Computational methods for a sustainable environment, economy, and society. *The Bridge*, 39(4) :5–13, 2009.

[GRC14] M. K. A. Gavina, J. F. Rabajante, and C. R. Cervancia. Mathematical Programming models for determining the optimal location of beehives. *Bulletin of Mathematical Biology*, 76 :997–1016, 2014.

[GSDW+13] L. A. Garibaldi, I. Steffan-Dewenter, R. Winfree, M. A Aizen, R. Bommarco, S. A. Cunningham, C. Kremen, L. G. Carvalheiro, L. D. Harder, O. Afik, et al. Wild pollinators enhance fruit set of crops regardless of honey bee abundance. *Science*, 339(6127) :1608–1611, 2013.

[GSKB13] D. Gabriel, S. M. Sait, W. E. Kunin, and T. G. Benton. Food production vs. biodiversity : comparing organic and conventional agriculture. *Journal of Applied Ecology*, 50 :355–364, 2013.

[GVBB+13] J.P. Gonzalez-Varo, J. C. Biesmeijer, R. Bonmarco, S. G. Potts, O. Schweiger, H. G. Smith, I. Steffan-Dewenter, H. Szentgyorgyi, M. Woyciechowski, and M. Vila. Combined effects of global change pressures on animal-mediated pollination. *Trends in Ecology and Evolution*, 28(9) :524–530, 2013.

[GZ04] C. V. Goldman and S. Zilberstein. Decentralized control of cooperative systems : Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22 :143–174, 2004.

[GZ08] C. V. Goldman and S. Zilberstein. Communication-based decomposition mechanisms for decentralized MDPs. *Journal of Artificial Intelligence Research*, 32 :169–202, 2008.

[HBZ04] E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *American Association for Artificial Intelligence*, pages 709–715, 2004.

[HKT+10] J. A. Hodgson, W. E. Kunin, C. D. Thomas, T. G. Benton, and D. Gabriel. Comparing organic farming and land sparing : optimizing yield and butterfly populations at a landscape scale. *Ecology Letters*, 13(11) :1358–1367, 2010.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301) :13–30, 1963.

[HSAHB99] J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. SPUDD : Stochastic planning using decision diagrams. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 279–288, 1999.

[HZ01] E. A. Hansen and S. Zilberstein. LAO* : A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129(1-2) :35–62, 2001.

[JB06] A. Jonsson and A. Barto. Causal graph based decomposition of factored MDPs. *Journal of Machine Learning Research*, 7 :2259–2301, 2006.

[JN07] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Publishing Company, Incorporated, 2nd edition, 2007.

[Kak02] S. Kakade. A natural policy gradient. In *Neural Information Processing Systems Foundation*, NIPS'02, pages 1531–1538, 2002.

[KD02] K-E. Kim and T. Dean. Solving factored MDPs with large action space using algebraic decision diagrams. In *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence*, pages 80–89, 2002.

[KDM00] K-E. Kim, T. L. Dean, and N. Meuleau. Approximate Solutions to Factored Markov Decision Processes via Greedy Search in the Space of Finite State Controllers. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, pages 323–330, 2000.

[KDMW12] A. Kolobov, P. Dai, Mausam, and D. S. Weld. Reverse Iterative Deepening for Finite-Horizon MDPs with Large Branching Factors. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 2012.

[KE12] T. Keller and P. Eyerich. PROST : Probabilistic Planning Based on UCT. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 2012.

[Ken86] J.O.S. Kennedy. *Dynamic programming applications to agriculture and natural resources*. Elsevier Science Pub. Co., 1986.

[KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models : Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[KFL01] F. R. Kschischang, B. J. Frey, and H-A. Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2) :498–519, 2001.

[KH08] B. Kveton and M. Hauskrecht. Partitioned linear programming approximations for MDPs. In *Proceedings of the international conference on Uncertainty in Artificial Intelligence*, pages 341–348, 2008.

[KHG06] B. Kveton, M. Hauskrecht, and C. Guestrin. Solving factored MDPs with hybrid state and action variables. *Journal of Artificial Intelligence Research*, 27 :153–201, 2006.

[KLC98] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101 :99–134, 1998.

[KMW12] A. Kolobov, Mausam, and D. S. Weld. Discovering hidden structure in factored MDPs. *Artificial Intelligence*, 189 :19–47, 2012.

[KP11] J. Kober and J. Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1-2) :171–203, 2011.

[KP14a] I. Kiselev and P. Poupart. A novel single-DBN generative model for optimizing pomdp controllers by probabilistic inference. In *Proceedings of the 28th AAAI conference on artificial intelligence*, 2014.

[KP14b] I. Kiselev and P. Poupart. Policy optimization by marginal-MAP probabilistic inference in generative models. In *Proceedings of the 13th international conference on autonomous agents and multiagent systems (AAMAS)*, 2014.

[KS06] L. Kocsis and C. Szepesvari. Bandit Based Monte-Carlo Planning. In *Proceedings of the 17th European Conference on Machine Learning*, page 282–293, 2006.

[KSM10] O. Kozlova, O. Sigaud, and C. Meyer. TeXDYNA : Hierarchical Reinforcement Learning in Factored MDPs. In *From Animals to Animats 11, 11th International Conference on Simulation of Adaptive Behavior, SAB*, pages 489–500, 2010.

[KZT11] A. Kumar, S. Zilberstein, and M. Toussaint. Scalable multiagent planning using probabilistic inference. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, 2011.

[LDK95] M.L. Littman, T.L. Dean, and L.P. Kaelbling. On the complexity of solving Markov decision problems. In *In Proc. of the eleventh International Conference on Uncertainty in Artificial Intelligence*, pages 394–402, 1995.

[LGM98] M.L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9 :1–36, 1998.

[Li09] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2009.

[LI13] Q. Liu and A. Ihler. Variational Algorithms for Marginal MAP. *Journal of Machine Learning Research*, 14 :3165–3200, 2013.

[LKR+09] E. Lonsdorf, C. Kremen, T. Ricketts, R. Winfree, N. Williams, and S. Greenleaf. Modelling pollination services across agricultural landscapes. *Annals of Botany*, 103 :1589–1600, 2009.

[Mar14] V. Maris. *Nature à vendre : les limites des services écosystémiques*. Editions Quae, 2014.

[MBB+03] E. J. P. Marshall, V. K. Brown, N. D. Boatman, P. J. W. Lutman, G. R. Squire, and L. K. Ward. The role of weeds in supporting biological diversity within crop fields. *Weed Research.*, 43 :77–89, 2003.

[MDGMJ07] D. Makowski, T. Dore, J. Gasquez, and N. Munier-Jolain. Modelling land use startegies to optimise crop production and protection of ecologically important weed species. *Weed Research*, 47 :202–211, 2007.

[MHC99] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *American Association for Artificial Intelligence*, 1999.

[MK07] J. M. Mooij and H. J. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Transactions on Information Theory*, 53(12) :4422–4437, 2007.

[MK12] Mausam and A. Kolobov. *Planning with Markov decision processes - An AI perspective*. Morgan and Claypool publishers, 2012.

[ML11] H. Mostafa and V. R. Lesser. Compact mathematical programs for DEC-MDPs with structured agent interactions. In *Proceedings of the international conference on Uncertainty in Artificial Intelligence*, pages 523–530, 2011.

[MMC98] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng. Turbo decoding as an instance of pearl's 'belief propagation' algorithm. *IEEE J. on Selected Areas in Communications*, 16(2) :140–152, 1998.

[MMCD13] G. Martin, R. Martin-Clouaire, and M. Duru. Farming system design to feed the changing world. A review. *Agronomy for Sustainable Development*, 33 :131–149, 2013.

[MMW+10] H. Meiss, S. Médiène, R. Waldhart, J. Caneill, and N. Munier-Jolain. Contrasting weed species composition in perennial alfalfas and six annual crops : implications for integrated weed management. *Agronomy for a sustainable development*, 30 :657–666, 2010.

[Moo10] J. M. Mooij. libDAI : A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11 :2169–2173, 2010.

[MPKK99] N. Meuleau, L. Peshkin, K-E. Kim, and L. P. Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the 15th conference on Uncertainty in Artificial intelligence*, pages 427–436, 1999.

[MT01] P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Transactions on automatic control*, 46(2) :191–209, 2001.

[Mur02] K. P. Murphy. *Dynamic Bayesian networks : representation, inference and learning*. PhD thesis, University of California, Berckley, 2002.

[MV11] F.S. Melo and M.M. Veloso. Decentralized MDPs with sparse interactions. *Artificial Intelligence*, 175(11) :1757–1789, 2011.

[MW01] K. P. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Proceedings of the international conference on Uncertainty in Artificial Intelligence*, 2001.

[MWJ99] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference : an empirical study. In *Proceedings of the international conference on Uncertainty in Artificial Intelligence*, 1999.

[NMR+09] E. Nelson, G. Mendoza, J. Regetz, S. Polasky, H. Tallis, D Richard Cameron, K. M. Chan, G. C. Daily, J. Goldstein, P. M. Kareiva, E. Lonsdorf, R. Naidoo, T. H. Ricketts, and M R. Shaw. Modeling multiple ecosystem services, biodiversity conservation, commodity production, and tradeoffs at landscape scales. *Frontiers in Ecology and the Environment*, 7(1) :4–11, 2009.

[NTY+03] R. Nair, M. Tambe, M. Yokoo, D. V. Pynadath, and S. Marsella. Taming decentralized POMDPs : towards efficient policy computation for multiagent settings. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 705–711, 2003.

[NVTY05] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs : A synthesis of distributed constraint optimization and POMDPs. In *Association for the Advancement of Artificial Intelligence*, pages 133–139, 2005.

[NW06] J. Nocedal and S. Wright. *Numerical Optimisation*. Springer, 2006.

[Oer06] E-C. Oerke. Crop losses to pests. *The Journal of Agricultural Science*, 144 :31–43, 2006.

[OKV08] F. A. Oliehoek, J. F. P. Kooij, and N. Vlassis. The cross-entropy method for policy search in decentralized POMDPs. *Informatica*, 32 :341–357, 2008.

[Oli10] F. A. Oliehoek. *Value-based planning for teams of agents in stochastic partially observable environments*. PhD thesis, Amsterdam University, 2010.

[OWS13] F. A. Oliehoek, S. Whiteson, and M. T. J. Spaan. Approximate solutions for factored Dec-POMDPs with many agents. In *Proceedings of the 12th international conference on autonomous agents and multiagent systems (AAMAS)*, 2013.

[PAC+13] S. Petit, A. Alignier, N. Colbach, A. Joannon, D. Le Coeur, and C. Thenail. Weed dispersal by farming at various spatial scales. A review. *Agronomy for sustainable development*, 33, 2013.

[PBLG+11] S. Petit, A. Boursault, M. Le Guilloux, N. Munier-Jolain, and X. Reboud. Weeds in agricultural landscapes. A review. *Agronomy for Sustainable Development*, 31(2), 2011.

[PBPS02] P. Poupart, C. Boutilier, R. Patrascu, and D. Schuurmans. Piecewise linear value function approximation for factored MDPs. In *Eighteenth national conference on Artificial intelligence*, pages 292–299, 2002.

[PCB14] M. Porto, O. Correia, and P. Beja. Optimization of landscape services under uncoordinated management by multiple landowners. *Plos One*, 9(1), 2014.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[PKMK00] L. Peshkin, K-E. Kim, N. Meuleau, and L. P. Kaelbling. Learning to cooperate via policy search. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, 2000.

[PLPN14] S. Polasky, D. J. Lewis, A. J. Plantinga, and E. Nelson. Implementing the optimal provision of ecosystem services. *Proceedings of the National Academy of Sciences*, 111(17) :6248–6253, 2014.

[PNL+05] S. Polasky, E. Nelson, E. Lonsdorf, P. Fackler, and A. Starfield. Conserving species in a working landscape : land use with biological and economic objectives. *Ecological applications*, 15(4) :1387–1401, 2005.

[Pow11] W. B. Powell. *Approximate dynamic programming*. Wiley Series in Probability and Statistics, 2011.

[PP11] J. Pajarinen and J. Peltonen. Efficient planning for factored infinite-horizon DEC-POMDPs. In *Proceedings of the 22th international joint conference on Artificial intelligence*, 2011.

[PS78] M. L. Puterman and M. C. Shin. Modified policy iteration algorithms for discounted Markov decision problems. *Management Science*, 24(11) :1127–1137, 1978.

[PS06] N. Peyrard and R. Sabbadin. Mean field approximation of the policy iteration algorithm for graph-based markov decision processes. In *Proceedings of the European Conference on Artificial Intelligence*, pages 595–599, 2006.

[PT87] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. 12(3) :441–450, 1987.

[Put94] M.L Puterman. *Markov decision processes*. John Wiley and Sons, 1994.

[PZ09] M. Petrik and S. Zilberstein. A bilinear programming approach for multiagent planning. *Journal of Artificial Intelligence Research*, 35 :235–274, 2009.

[RBB+06] J. P. Rodriguez, T. D. Beard, E. M. Bennett, G. S. Cumming, S. Cork, J. Agard, A. P. Dobson, and G. D. Peterson. Trade-offs across space, time, and ecosystem services. *Ecology and Society*, 11(1) :28, 2006.

[RBD+13] O. Rollin, V. Bretagnolle, A. Decourtye, J. Aptel, N. Michel, B. E. Vaissière, and M. Henry. Differences of floral ressource use between honey bees and wild bees in an intensive farming system. *Agriculture, Ecosystems and Environment*, 179 :78–86, 2013.

[RJF+12] A. Raghavan, S. Joshi, A. Fern, P. Tadepalli, and R. Khardon. Planning in factored action spaces with symbolic dynamic programming. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

[ROT+15] F. Requier, J-F. Odoux, T. Tamic, N. Moreau, M. Henry, A. Decourtye, and V. Bretagnolle. Honey bee diet in agricultural landscapes. *Ecological Applications*, 2015.

[RPS14] J. Radoszycki, N. Peyrard, and R. Sabbadin. Finding good stochastic factored policies for factored Markov decision processes. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic*, pages 1083–1084, 2014.

[RPS15a] J. Radoszycki, N. Peyrard, and R. Sabbadin. Résolution de PDMF^3 : processus décisionnels de Markov à transitions, récompenses et politiques stochastiques factorisées. In *JFPDA 2015 - 10èmes Journées Francophones Planification Décision Apprentissage, 1-3 Juillet 2015, Rennes*, 2015.

[RPS15b] J. Radoszycki, N. Peyrard, and R. Sabbadin. Solving F^{3}MDPs : Collaborative Multiagent Markov Decision Processes with Factored Transitions, Rewards and Stochastic Policies. In *PRIMA 2015 - Principles and Practice of Multi-Agent Systems, 26-30 October 2015, Bertinoro, Italy*, 2015.

[RS13] L. M. Rios and N. V. Sahinidis. Derivative-free optimization : a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56 :1247–1293, 2013.

[RTV12] K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics : Science and Systems*, 2012.

[SAHB00] R. St-Aubin, J. Hoey, and C. Boutilier. APRICODD : Approximate policy construction using decision diagrams. In *Advances in Neural Information Processing Systems*, pages 1089–1095, 2000.

[SB98] R. S. Sutton and A. G. Barto. *Reinforcement Learning : An Introduction*. A Bradford Book, 1998.

[SB08] O. Sigaud and O. Buffet. *Processus décisionnels de Markov en intelligence artificielle*, volume 1 - principes généraux et applications of *IC2 - informatique et systèmes d'information*. Lavoisier - Hermes Science Publications, 2008.

[SBSR04] A. Schaefer, M. Bailey, S. Schechter, and M. Roberts. Medical decisions using Markov decision processes. In *Handbook of operations research / Management science applications in health care*. Kluwer Academic Publishers, 2004.

[SCZ05] D. Szer, F. Charpillet, and S. Zilberstein. MAA* : a heuristic search algorithm for solving decentralized POMDPs. In *Proceedings of Uncertainty in Artificial Intelligence*, 2005.

[SG14] D. Smith and V. Gogate. Loopy belief propagation in the presence of determinism. In *Proceedings of the 17th international conference on Artificial intelligence and Statistics*, 2014.

[SH04] B. Sallans and G. E. Hinton. Reinforcement learning with factored states and actions. *Journal of Machine Learning Research*, 5 :1063–1088, 2004.

[Sha86] R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 33(6) :871–882, 1986.

[SKS13] D. A. Stanley, M. E. Knight, and J. C. Stout. Ecological variation in response to mass-flowering oilseed rape and surrounding landscape composition by members of a cryptic bumblebee complex. *PLoS ONE*, 8(6), 2013.

[SM05] S. Sanner and D. McAllester. Affine algebraic decision diagrams (AADDs) and their application to structured probabilistic inference. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1384–1390, 2005.

[SMC10] J. Storkey, S.R. Moss, and J.W. Cussans. Using assembly theory to explain changes in a weed flora in response to agricultural intensification. *Weed Science*, 58 :39–46, 2010.

[SPF12] R. Sabbadin, N. Peyrard, and N. Forsell. A framework and a mean-field algorithm for the local control of spatial processes. *International Journal of Approximate Reasoning*, 53(1) :66–86, 2012.

[SS85] P. Schweitzer and A. Seidman. Generalized polynomial approximations in markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110 :568–582, 1985.

[SSR07] R. Sabbadin, D. Spring, and C-E. Rabier. Dynamic reserve site selection under contagion risk of deforestation. *Ecological Modelling*, 201 :75–81, 2007.

[Sto06] J. Storkey. A functional group approach to the management of UK arable weeds to support biological diversity. *Weed Research*, 46 :513–522, 2006.

[SUD10] S. Sanner, W. Uther, and K. V. Delgado. Approximate dynamic programming with affine ADDs. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, pages 1349–1356, 2010.

[SZ07] S. Seuken and S. Zilberstein. Improved memory-bounded dynamic programming for decentralized pomdps. In *Proceedings of Uncertainty in Artificial intelligence*, 2007.

[TBCR13] E. Tancoigne, M. Barbier, J-P. Cointet, and G. Richard. Les services écosystémiques dans la littérature scientifique : démarche d'exploration et résultats d'analyse. rapport d'étude pour la phase d'exploration du métaprogramme ecoserv. Technical report, INRA, 2013.

[THS06] M. Toussaint, S. Harmeling, and A. Storkey. Probabilistic inference for solving (PO)MDP's. Technical report, Dec 2006.

[TKK+05] T. Tscharntke, A. M. Klein, A. Kruess, I. Steffan-Dewenter, and C. Thies. Landscape perspectives on agricultural intensification and biodiversity - ecosystem service management. *Ecology Letters*, 8 :857–874, 2005.

[TPA+13] P. Tixier, N. Peyrard, J.N. Aubertot, S. Gaba, J. Radoszycki, G. Caron-Lormier, F. Vinatier, G. Mollot, and R. Sabbadin. Chapter Seven – Modelling Interaction Networks for Enhanced Ecosystem Services in Agroecosystems. *Advances in Ecological Research*, 49 :437–480, 2013.

[TS06] M. Toussaint and A. J. Storkey. Probabilistic inference for solving discrete and continuous state Markov Decision Processes. In *Proceedings of the International Conference on Machine learning*, pages 945–952, 2006.

[vWAB+14] H. von Wehrden, D. J. Abson, M. Beckmann, A. F. Cord, S. Klotz, and R. Seppelt. Realigning the land-sharing/land-sparing debate to match

conservation needs : considering diversity scales and land-use history. *Landscape Ecology*, 29 :941–948, 2014.

[Wan14] W. Wang. *Multi-objective sequential decision making.* PhD thesis, Universite Paris-Sud, 2014.

[WBD+09] A. Wezel, S. Bellon, T. Dore, C. Francis, D. Vallod, and C. David. Agroecology as a science, a movement and a practice. A review. *Agronomy for sustainable development*, 29(4), 2009.

[WD10] S. Witwicki and D. Durfee. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *Proceedings of the 20th international conference on automated planning and scheduling*, 2010.

[Wil92] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8 :229–256, 1992.

[WJW03] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation via pseudo-moment matching. In *Workshop on Artificial Intelligence and Statistics*, 2003.

[WW90] B. J. Wilson and K. J. Wright. Predicting the growth and competitive effects of annual weeds in wheat. *Weed Research*, 30 :201–211, 1990.

[WY07] J. D. Williams and S. Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21 :393–422, 2007.

[YFW05] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7) :2282–2312, 2005.

Annexe A

Modèles graphiques

Dans cette partie, nous expliquons ce qu'est un modèle graphique (section A.1), et nous définissons en particulier deux modèles graphiques qui interviennent dans la littérature sur les PDMs et dans l'approche que nous développons au chapitre 2 : le réseau bayésien dynamique (*dynamic bayesian network*, DBN [Mur02]) et le *factor graph* [KFL01]. Nous décrivons ensuite dans la section A.2 l'algorithme *loopy belief propagation* (LBP) permettant le calcul approché de marginales dans un modèle graphique.

A.1 Plusieurs cadres

Un modèle graphique [KF09, Bis07] est un modèle probabiliste dans lequel un graphe permet de représenter les indépendances conditionnelles entre les variables aléatoires. Plus formellement, soit $S = (S_1, ..., S_n)$ un vecteur aléatoire à valeurs dans $\Omega = \Omega_1 \times ... \times \Omega_n$. On dit que la loi jointe P de S est un modèle graphique s'il existe un ensemble de fonctions positives $\{\psi_c, c \in \mathcal{C}\}$ indicées sur des sous-ensembles de $V = \{1, ..., n\}$ telles que :

$$\forall s \in \Omega, P(s) = \mathbb{P}(S = s) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(s_c)$$

où s_c est une notation pour $\{s_i, i \in c\}$, $\mathcal{C}$ désigne un ensemble de parties de V et Z est appelée fonction de partition. Les fonctions ψ_c sont appelées fonctions potentiel ou facteurs. Le graphe associé au modèle graphique ainsi défini est le graphe $G = (V, E)$ où les noeuds, correspondant à l'ensemble V, sont les indices des variables aléatoires. Lorsque deux variables aléatoires apparaissent dans la même fonction potentielle, leurs noeuds sont reliés par une arête. Les arêtes peuvent être orientées ou non, elles forment l'ensemble E. Lorsque le graphe est un graphe dirigé acyclique, on parle de réseau bayésien [JN07], et lorsque le graphe est non dirigé, on parle de champ de Markov [Li09]. Dans un réseau bayésien, les fonctions potentiel sont des probabilités conditionnelles, et $Z = 1$. Cette propriété est perdue dans le cas des champs de Markov.

A.1.1 Réseau bayésien dynamique

Dans le cadre des PDMs factorisés, des réseaux bayésiens dynamiques [Mur02] sont souvent utilisés pour représenter la fonction de transition (voir exemple figure 1.5 gauche). Nous définissons ici plus formellement un réseau bayésien dynamique en tant que modèle graphique. Une distribution de probabilité sur des variables aléatoires multidimensionnelles (à valeurs dans $\Omega = \Omega_1 \times ... \times \Omega_n$) indexées par un indice de temps discret t $(S^0, ..., S^t, ...)$ est un réseau bayésien dynamique si la propriété de Markov est vérifiée (conditionnellement à $(S^0, ..., S^{t-1})$, S^t ne dépend que de S^{t-1}) et :

$$\forall t \in \mathbb{N},\ \mathbb{P}(S^t|S^{t-1}) = \prod_{i=1}^{n} P_i(S_i^t|pa(S_i^t))$$

où $pa(S_i^t) \subset \{S_j^t, j = 1...n, j \neq i, S_k^{t-1}, k = 1...n\}$. La seule hypothèse des réseaux bayésiens dynamiques est que le graphe orienté associé à ce modèle graphique soit acyclique [Mur02]. Un réseau bayésien dynamique est un modèle graphique particulier, dont les fonctions potentiel sont les distributions conditionnelles $P_i, i = 1...n$.

A.1.2 *Factor graph*

Tout modèle graphique peut être représenté par un *factor graph* [KFL01]. Les *factor graphs* peuvent même représenter, plus généralement que des distributions jointes de probabilité, des fonctions réelles de plusieurs variables qui se factorisent en un produit de fonctions locales faisant intervenir un sous-ensemble des ces variables.

Un *factor graph* est un graphe bipartite, c'est-à-dire avec deux types de noeuds et des arcs seulement entre noeuds de type différent. Le premier type de noeuds, qui sera représenté par des cercles, correspond aux variables, le second type de noeuds, qui sera représenté par des carrés, correspond aux facteurs (ou fonctions potentiel). Il y a un arc entre un noeud variable et un noeud facteur si le facteur fait intervenir cette variable. La figure A.1 donne un exemple de *factor graph*.

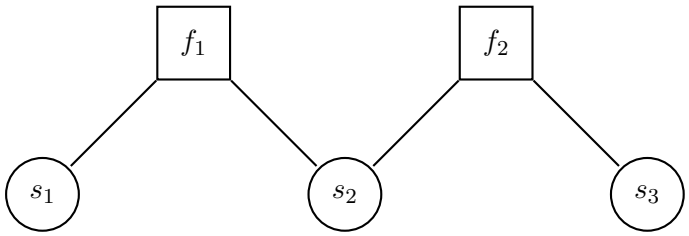


FIGURE A.1 – Exemple de *factor graph* représentant la fonction $g(s_1, s_2, s_3) = f_1(s_1, s_2) \times f_2(s_2, s_3)$

La figure A.2 représente sous forme de *factor graph* le réseau bayésien dynamique de la figure 1.5, correspondant à la distribution de probabilité conditionnelle :

$$P(S^{t+1}, S^t, A^t) = P_1(S_1^{t+1}|S_1^t, S_2^t, A^t) \times P_2(S_2^{t+1}|S_1^t, S_3^t, S_1^{t+1}, A^t) \times P_3(S_3^{t+1}|S_3^t, A^t)$$

Remarquons que plusieurs *factor graphs* peuvent représenter un même modèle graphique P ou une même fonction g, puisque certains facteurs peuvent être regroupés.

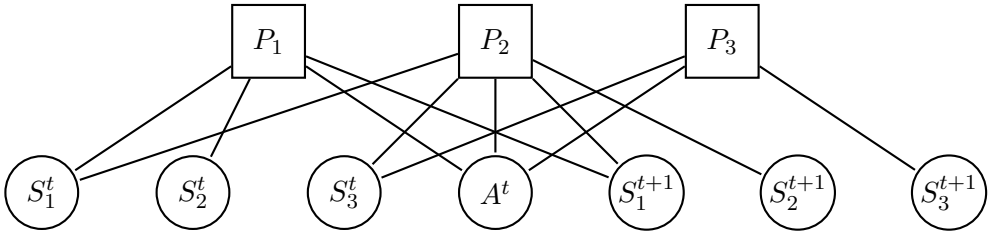


FIGURE A.2 – Représentation sous forme de *factor graph* du réseau bayésien dynamique de la figure 1.5 représentant la fonction de transition d'un PDMF

A.2 L'algorithme *loopy belief propagation* (LBP)

Dans ce manuscrit, nous nous intéressons à une tâche d'inférence particulière dans les réseaux bayésiens dynamiques, celle du calcul de lois marginales associées à un sous-ensemble de variables indicées par $e \in \mathcal{P}(V)$:

$$P_e(s_e) = \sum_{s \setminus s_e} P(s) = \sum_{s \setminus s_e} \prod_{c \in \mathcal{C}} \psi_c(s_c)$$

où $\sum_{s \setminus s_e}$ désigne la somme sur toutes les valeurs possibles pour les variables qui ne sont pas dans l'ensemble e. Le problème du calcul de lois marginales dans des modèles graphiques est en général complexe puisqu'il peut nécessiter de sommer sur un nombre exponentiel de termes.

Nous décrivons ici l'algorithme *loopy belief propagation* (LBP), connu aussi sous le nom d'algorithme *sum-product* [FM98, KFL01]. Cet algorithme est exact si le *factor graph* associé au modèle graphique est acyclique (on retrouve alors l'algorithme *belief propagation* décrit dans [Pea88]), mais l'algorithme est quand même défini et donne de bons résultats approchés en pratique sur des *factor graphs* quelconques.

L'algorithme LBP est un algorithme de passage de messages sur le *factor graph* associé au modèle graphique. Il y a deux types de messages :

— les messages d'un nœud variable i vers un nœud facteur c, fonctions de l'état s_i du nœud variable i et notés $n_{i \to c}(s_i)$

— les messages d'un nœud facteur c vers un nœud variable i, fonctions de l'état s_i du nœud variable i et notés $m_{c \to i}(s_i)$.

Les messages sont initialisés à 1 et mis à jour selon le schéma suivant :

$$n_{i \to c}^{t+1}(s_i) := \prod_{k \in N(i) \setminus c} m_{k \to i}^t(s_i)$$

$$m_{c \to i}^{t+1}(s_i) := \sum_{s_c \setminus s_i} \psi_c(s_c) \prod_{j \in N(c) \setminus i} n_{j \to c}^t(s_j)$$

où $N(i)$ désigne l'ensemble des indices des facteurs voisins du nœud variable i dans le *factor graph* et $N(c)$ désigne l'ensemble des indices des variables voisines du facteur c dans le *factor graph*. Plusieurs schémas pour le passage de messages sont possibles (séquentiel, parallèle *etc.*). La convergence du schéma de passage de messages n'est pas garantie, mais il existe des conditions suffisantes (mais pas nécessaires) de convergence vers un point fixe unique [MK07]. Dans le cas d'un *factor graph* acyclique, la convergence est garantie, et, pour un schéma de passage de messages bien choisi, il y a convergence au bout de deux itérations.

Une fois les messages calculés, une approximation de la marginale associée à chaque nœud variable i s'obtient par l'équation :

$$b_i(s_i) \propto \prod_{k \in N(i)} m_{k \to i}(s_i)$$

Une approximation de la marginale associée aux variables voisines du facteur c s'obtient par l'équation :

$$b_c(s_c) \propto \psi_c(s_c) \prod_{i \in N(c)} n_{i \to c}(s_i)$$

Il n'y a pas de résultats théoriques sur la qualité de l'approximation avec l'algorithme LBP. Le seul résultat théorique est que les points fixes de l'algorithme LBP correspondent aux points stationnaires de l'approximation de Bethe [YFW05], faisant le lien avec les méthodes variationnelles. Cependant, un certain nombre d'études ont montré une bonne qualité d'approximation en pratique [MWJ99, MMC98], et cet algorithme est donc maintenant largement utilisé pour approcher des marginales dans les modèles graphiques.

A.3 Bilan

Les réseaux bayésiens dynamiques, qui sont souvent utilisés dans les PDMs factorisés pour modéliser la transition, sont des modèles graphiques qui peuvent être représentés sous forme de *factor graph*. Le calcul de marginales dans ces modèles est difficile en général, mais l'algorithme LBP [KFL01] offre un bon compromis entre temps de calcul et qualité de l'approximation.

Annexe B

Démonstration de la complexité du problème d'optimisation dans les PDMF^3

Le but de cette annexe est de démontrer le théorème 9 de la section 2.3.1. Le problème de décision associé au problème 1 (voir section 2.3.1) est le suivant :

Définition 22 (problème 1D)**.** *Le problème de décision associé au problème 1 pour un PDMF^3 $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, P, pa_\delta, R, P^0)$ consiste à décider si il existe une PSF δ^* de structure pa_δ et de valeur $V_{\delta^*}^{R,T}(P^0)$ strictement supérieure à $\frac{1}{2}$.*

Pour montrer que ce problème est NP^{PP}-complet, nous allons montrer qu'il est NP^{PP}-difficile (voir section B.1) puis qu'il appartient à la classe NP^{PP} (voir section B.2).

B.1 Le problème de décision associé au problème d'optimisation dans les PDMF^3 est NP^{PP}-difficile

Pour montrer que le problème de décision associé au problème 1 est NP^{PP}-difficile, nous démontrons une réduction en temps polynomial du problème $EMAJSAT$, qui est connu pour être NP^{PP}- complet [LGM98].

B.1.1 Le problème $EMAJSAT$

Définition 23 (problème $EMAJSAT$)**.** *Soit ϕ une formule booléenne sur les variables $X_1, \ldots, X_n$ à m clauses ($\phi = (\mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_m)$), et soit $k \in \{1, \ldots, n\}$ un entier fixé. Le problème de décision $EMAJSAT$ consiste à décider si il existe une instanciation $(x_1, \ldots, x_k)$ des variables $(X_1, \ldots, X_k)$ telle que, pour la majorité des instanciations possibles $(x_{k+1}, \ldots, x_n)$ de $(X_{k+1}, \ldots, X_n)$ (il y en a 2^{n-k}) ϕ est satisfaite.*

On notera $pa_C(\mathcal{C}_j) \subset \{X_1, ..., X_n\}$ l'ensemble des variables (ou littéraux) intervenant dans la clause $\mathcal{C}_j$, pour tout j de 1 à m. La proposition suivante est connue :

Propriété 5 ([LGM98]). *Le problème $EMAJSAT$ est NP^{PP}-complet.*

Dans la suite, nous démontrerons une réduction du problème $EMAJSAT$ avec des 3-clauses (clauses qui ont au plus 3 littéraux) en un problème d'optimisation dans un PDMF^3, puisque toute instance d'un problème $EMAJSAT$ peut être réécrite en un problème $EMAJSAT$ équivalent contenant uniquement des 3-clauses (et qui peut contenir un nombre polynomialement plus grand de clauses et de variables que le premier problème $EMAJSAT$).

B.1.2 Réduction du problème $EMAJSAT$ en un problème d'optimisation dans un PDMF^3

Nous allons montrer une réduction polynomiale de toute instance (ϕ, k) d'un problème $EMAJSAT$ en une instance $M^* = (\mathcal{S}, \mathcal{A}, \mathcal{T}, P, pa_\delta, R, P^0)$ du problème 1D. Ainsi, la réponse au problème $EMAJSAT$ sera positive si et seulement si il existe une PSF δ^* (en fait déterministe) pour le PDMF^3 M^* de valeur strictement supérieure à $\frac{1}{2}$.

Soit le PDMF^3 $M^* = (\mathcal{S}, \mathcal{A}, \mathcal{T}, P, pa_\delta, R, P^0)$ représenté figure B.1 dont :

- Les variables d'action sont les variables $(X_1, \ldots, X_k)$ et les variables d'état sont les variables $(Y_1, \ldots, Y_k, X_{k+1}, \ldots, X_n, C_1, \ldots, C_m)$. Toutes les variables prennent leurs valeurs dans $\{0, 1\}$, sauf les variables $Y_l, l = 1...k$ qui prennent leurs valeurs dans $\{-1, 0, 1\}$. Nous noterons $A^t = (X_1^t, ..., X_k^t)$ l'action au temps t (le nombre de variables d'action est de k) et $S^t = (Y_1^t, \ldots, Y_k^t, X_{k+1}^t, \ldots, X_n^t, C_1^t, \ldots, C_m^t)$ l'état du système au temps t (le nombre de variables d'état est de $k+n-k+m = n+m$). On a donc $\mathcal{A} = \prod_{l=1}^{k} \mathcal{A}_l$ où $\forall l = 1...k,\ \mathcal{A}_l = \{0, 1\}$ et $\mathcal{S} = \prod_{l=1}^{n+m} \mathcal{S}_l$ où $\forall l = 1...k, \mathcal{S}_l = \{-1, 0, 1\}$ et $\forall l = k+1...n+m, \mathcal{S}_l = \{0, 1\}$.
- L'espace des temps est $\mathcal{T} = \{0, ..., m\}$, et le facteur d'amortissement est $0 \leq \gamma \leq 1$.
- La distribution de probabilité initiale P^0 est définie par :
  - $P^0(X_i^0 = 0) = P^0(X_i^0 = 1) = \frac{1}{2}, \forall i = k+1, \ldots, n$.
  - $P^0(C_j^0 = 0) = 1, \forall j = 1, \ldots, m$.
  - $P^0(Y_l^0 = -1) = 1, \forall l = 1, \ldots, k$.
- Les probabilités de transition sont définies par :
  - $\forall t = 0, \ldots, m-1, \forall l = 1, \ldots, k, \forall (x_l, x'_l) \in \{0, 1\}^2, x_l \neq x'_l, P_l(Y_l^{t+1} = x_l | X_l^t = x_l) = 1$ et $P_l(Y_l^{t+1} = x_l | X_l^t = x'_l) = 0$. On a donc $pa_P(Y_l^{t+1}) = \{X_l^t\}\ \forall t = 0, ..., m-1, \forall l = 1, ..., k$. Les variables Y_l sont des variables auxiliaires permettant, avec des fonctions de récompense auxiliaires décrites plus loin, d'assurer que des politiques non déterministes ne peuvent pas être optimales.
  - $\forall t = 0, \ldots, m-1, \forall i = k+1, \ldots, n, \forall (x_i, x'_i) \in \{0, 1\}^2, x_i \neq x'_i, P_i(X_i^{t+1} = x_i | X_i^t = x_i) = 1$ et $P_i(X_i^{t+1} = x_i | X_i^t = x'_i) = 0$. On a donc $pa_P(X_i^{t+1}) = \{X_i^t\}\ \forall t = 0, ..., m-1, \forall i = k+1, ..., n$. Les valeurs des variables X_i^0 sont tirées dans une distribution uniforme au départ, et restent inchangées au cours du temps.

— $P_{n+1}(C_1^{t+1} = 1 | C_1^t = 1 \vee (C_1^t = 0 \wedge X^t \models \mathcal{C}_1)) = 1$ et $\forall j = 2...m, P_{n+j}(C_j^{t+1} | C_j^t = 1 \vee (C_j^t = 0 \wedge C_{j-1}^t = 1 \wedge X^t \models \mathcal{C}_j)) = 1$. On a donc $pa_P(C_1^{t+1}) = \{C_1^t, pa_C^t(\mathcal{C}_1)\}$ et $\forall j = 2...m,\ pa_P(C_j^{t+1}) = \{C_j^t, C_{j-1}^t, pa_C^t(\mathcal{C}_j)\}$. Les probabilités locales de transition des variables C_j vérifient, pour toute instanciation $X^0 = (x_1, \ldots, x_n)$: au temps $t = 1$ C_1^1 prend la valeur 1 si et seulement si $(x_1, \ldots, x_n) \models \mathcal{C}_1$, puis, à tout temps $t \geq j > 1$, C_j^t prend la valeur 1 si et seulement si $(x_1, \ldots, x_n) \models \mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_j$.

- La structure de la politique est vide : $pa_\delta(X_l^t) = \emptyset, \forall l = 1 \ldots k$.
- Les fonctions de récompense sont définies comme suit :
  — Des fonctions de récompense $\{R_l, l = 1 \ldots k\}$ sont définies sur les paires de variables (X_l, Y_l) : $R_l(x_l, y_l) = 0$ si $x_l = y_l$ et $R_l(x_l, y_l) = -K$ si $x_l \neq y_l$, où $K > \frac{1}{\gamma^{2m}}$.
  — La fonction de récompense R_{k+1} est définie sur la variable C_m : $R_{k+1}(c_m) = \frac{1}{\gamma^m}$ si $c_m = 1$ et $R_{k+1}(c_m) = 0$ si $c_m = 0$.

  Il y a donc $r = k+1$ fonctions de récompense. Avec cette définition des fonctions de récompense et la définition des fonctions de transition locales, on a la garantie que toute trajectoire $x^0, \ldots, x^T$ conduit à une somme positive ou nulle de récompenses si et seulement si $x^0 = x^1 = \ldots = x^T$, et à une somme de récompenses négative sinon.

Remarquons que cette réduction est polynomiale en temps. Toutes les tables de transition et les fonctions de récompense ont une taille bornée par une constante. En particulier, comme nous considérons des 3-clauses, $pa_P(C_j^{t+1})$ comprend cinq variables au plus.

Nous allons montrer que :

(i) la politique optimale δ^* de M^* est déterministe,

(ii) δ^* est de valeur strictement supérieure à $\frac{1}{2}$ si et seulement si la réponse au problème $EMAJSAT$ est positive

(iii) δ^* détermine l'instanciation $(x_1, ..., x_k)$ des variables $X_1, \ldots, X_k$ qui conduit à une réponse positive au problème $EMAJSAT$.

Considérons le PDMF^3 M^* associé au problème $EMAJSAT$ et décrit ci-dessus. Soit δ une PSF arbitraire pour ce PDMF^3 :

$$V_\delta^{R,T}(P^0) = \sum_{t=0}^{m} \gamma^t \sum_{(s,a)^{0:t}} P_\delta^t \left((s,a)^{0:t}\right) R(s^t, a^t).$$

Remarquons tout d'abord que, étant données les dépendances déterministes des variables $\{Y_l^t, l = 1...k\}$ et $\{C_j^t, j = 1...m\}$ par rapport aux variables $\{X_i^t, i = 1...n\}$, la valeur $V_\delta^{R,T}(P^0)$ de toute PSF δ telle que $pa_\delta(X_i^t) = \emptyset, \forall i = 1..k$ peut s'exprimer

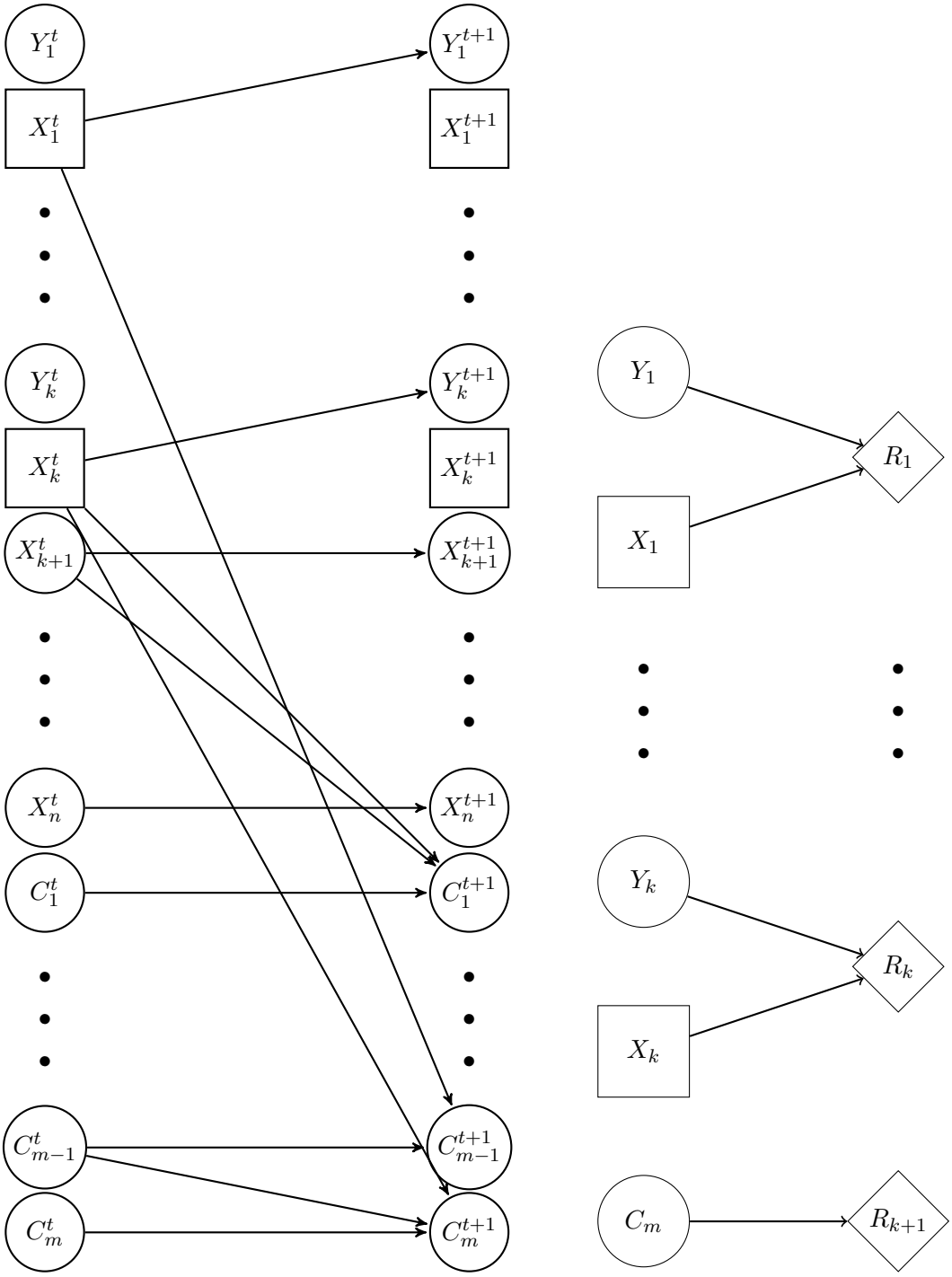


FIGURE B.1 – Représentation de la structure de la transition (à gauche) et de la structure de la récompense (à droite) pour un exemple de $PDMF^3$ correspondant à la réduction d'un problème EMAJSAT.

seulement en fonction des variables $\{X_i^0\}_{i=1..n}$ et $\{X_i^t\}_{i=1..k,t=1..m}$:

$$\forall \delta, V_\delta^{R,T}(P^0) = \sum_{(x_1^0,\ldots,x_n^0),\ldots,(x_1^m,\ldots,x_n^m)} \left(\prod_{t=0}^{m} \delta(x_1^t,\ldots,x_k^t)\right) \times P^0(x_{k+1}^0,\ldots,x_n^0) \times f\left(\{x_i^t, i=1...k, t=1...m\}\right).$$

où $\delta(x_1^t,\ldots,x_k^t) = \prod_{i=1}^k \delta_i(x_i^t)$ est la probabilité de l'instanciation $(x_1^t,\ldots,x_k^t)$ sous la PSF δ, et $f(\{x_i^t\}) < 0$ si $\exists l \in \{1,\ldots,k\}, t \in \{0,\ldots,m-1\}$ tels que $x_l^t \neq x_l^{t+1}$. Cela est du au fait que toute trajectoire dont deux vecteurs consécutifs $(x_1^t,\ldots,x_k^t)$ et $(x_1^{t+1},\ldots,x_k^{t+1})$ ne sont pas égaux reçoit une somme des récompenses inférieure ou égale à $\frac{1}{\gamma^m} - K\gamma^m$. Or, puisque $K > \frac{1}{\gamma^{2m}}$, $\frac{1}{\gamma^m} - K\gamma^m < 0$.

Donc $V_\delta^{R,T}(P^0)$ est majorée par une somme sur toutes les trajectoires de variables $(X_1^t = x_1,\ldots,X_n^t = x_n)$ constantes dans le temps [1] :

$$\forall \delta, V_\delta^{R,T}(P^0) \leq \sum_{x_1,\ldots,x_n} (\delta(x_1,\ldots,x_k))^m \times P^0(x_{k+1},\ldots,x_n) \times g(x_1,\ldots,x_n),$$

où $g(x_1,\ldots,x_n) = 1$ si $(x_1,\ldots,x_n) \models \phi$ et 0 sinon.

Soit $(x_1^*,\ldots,x_k^*)$ défini par :

$$(x_1^*,\ldots,x_k^*) = \operatorname*{argmax}_{x_1,\ldots,x_k} \sum_{x_{k+1},\ldots,x_n} P^0(x_{k+1},\ldots,x_n) \times g(x_1,\ldots,x_n).$$

Puisque pour toute PSF δ on a $\delta(x_1,\ldots,x_k) \leq 1 \forall x_1,\ldots,x_k$,

$$\begin{aligned} \forall \delta,\ V_\delta^{R,T}(P^0) &\leq \sum_{x_{k+1},\ldots,x_n} P^0(x_{k+1},\ldots,x_n) \times g(x_1,\ldots,x_n) \\ &\leq \sum_{x_{k+1},\ldots,x_n} P^0(x_{k+1},\ldots,x_n) \times g(x_1^*,\ldots,x_k^*,x_{k+1},\ldots,x_n). \end{aligned}$$

Mais le terme de droite est exactement $V_{\delta^*}^{R,T}(P^0)$, où $\delta^*(x_1,\ldots,x_k) = 1$ si et seulement si $(x_1,\ldots,x_k) = (x_1^*,\ldots,x_k^*)$. Donc la politique factorisée déterministe δ^* est optimale pour le PDMF^3 M^*.

Pour démontrer (ii), considérons maintenant une politique factorisée déterministe δ, correspondant à un vecteur $(x_1,\ldots,x_k)$, puisque $pa_\delta(X_l^t) = \emptyset, \forall l = 1...k$. P^0 détermine un vecteur $(x_{k+1},\ldots,x_n)$ aléatoire (tous les vecteurs ont la même probabilité $\frac{1}{2^{n-k}}$). Une fois que toutes les valeurs de variables sont fixées au temps $t = 0$, il est facile de vérifier que les transitions sont déterministes, et que $C_m^m = 1$ si et seulement si $(x_1,\ldots,x_n) \models \mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_m$. Si cela est vérifié, la trajectoire correspondante conduit à une somme pondérée des récompenses $\gamma^m R_{k+1}(1) = 1$.

1. Rappelons que, étant donnée la forme des probabilités locales de transition, les variables $(X_{k+1}^t,\ldots X_n^t)$ restent également constantes au cours du temps.

Par conséquent, la valeur de toute politique factorisée déterministe $\delta = (x_1, \ldots, x_k)$ est :

$$\begin{aligned} V_\delta^{R,T}(P^0) &= \sum_{x_{k+1},\ldots,x_n s.t. \{x_1,\ldots,x_n\} \models \phi} \frac{1}{2^{n-k}}, \\ &= \frac{1}{2^{n-k}} \times \#\Big\{\{x_{k+1}, \ldots, x_n\} / \{x_1, \ldots, x_n\} \models \phi\Big\}. \end{aligned}$$

Et cette valeur est supérieure à $\frac{1}{2}$ si et seulement si une majorité des 2^{n-k} instanciations $\{x_{k+1}, \ldots, x_n\}$ vérifie ϕ. Donc l'instanciation (ϕ, k) du problème $EMAJSAT$ se réduit en le problème 1D associé au PDMF^3 M^* décrit ci-dessus.

De plus, la politique factorisée déterministe obtenue définit un certificat positif $(x_1, ..., x_k)$ du problème $EMAJSAT$, donc (iii) est vérifiée.

Nous avons donc trouvé une réduction en temps polynomial de tout problème $EMAJSAT$ en un problème 1D, ce qui signifie que le problème 1D, donc le problème 1, est NP^{PP}-difficile.

B.2 Le problème de décision associé au problème d'optimisation dans les PDMF^3 appartient à la classe NP^{PP}

Cette partie est plus facile à démontrer. Vérifier si une PSF δ dans un PDMF^3 est de valeur supérieure à μ revient à évaluer les probabilités marginales d'un (petit) ensemble de variables dans un réseau bayésien. Ces valeurs peuvent être testées avec des oracles PP. Vérifier qu'un PDMF^3 admet une politique de valeur supérieure à μ^*, μ^* étant fixé, revient à deviner cette politique puis tester sa valeur. Ce problème appartient à NP si on suppose que l'on a un oracle PP pour le calcul de la valeur de la politique.

En conclusion, le problème de décision associé au problème d'optimisation dans les PDMF^3 est NP^{PP}-complet.

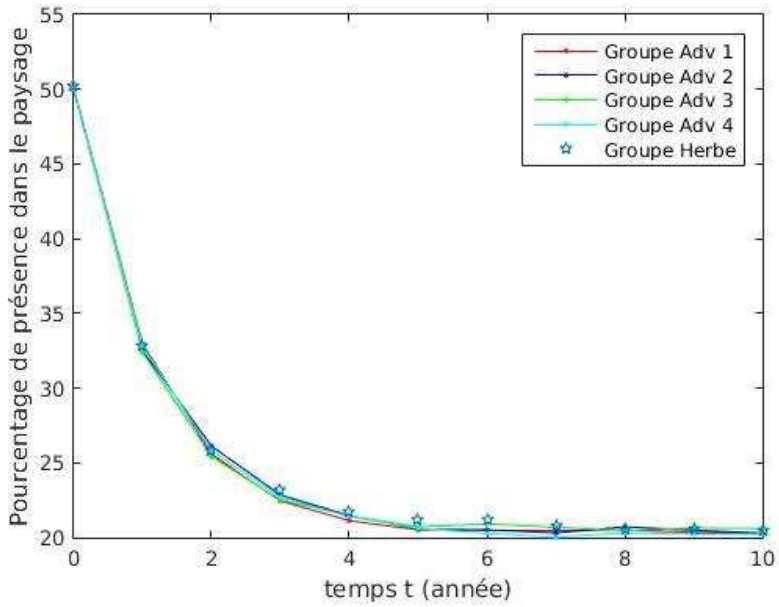
Annexe C

Application en agroécologie

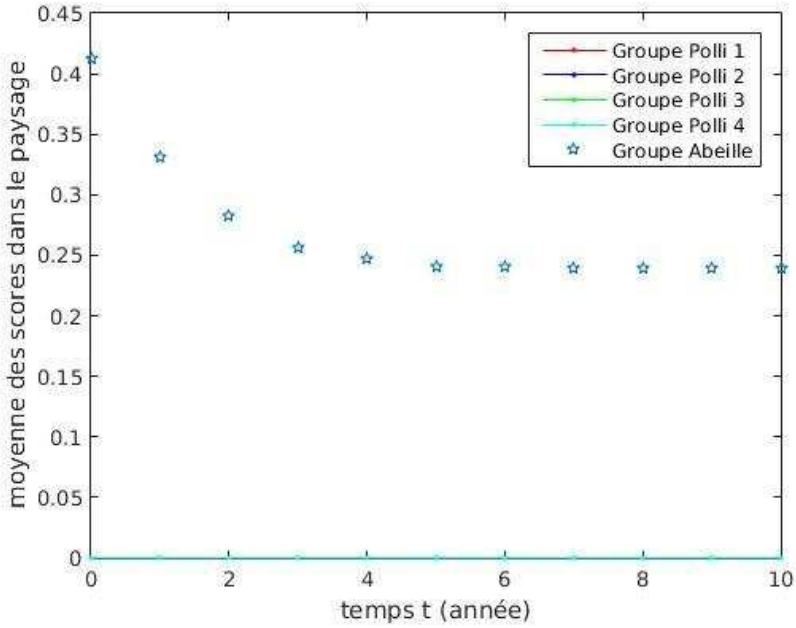
C.1 Comportement du modèle pour des monocultures

Dans cette section, nous illustrons le comportement du modèle pour des monocultures, c'est-à-dire pour des politiques simples consistant à mettre la même culture sur toutes les parcelles et tous les ans [1]. Les résultats sont commentés dans la section 3.6.5.

1. Je tiens à remercier Romain Alexandre qui a généré ces figures lors de son stage de Master 2.

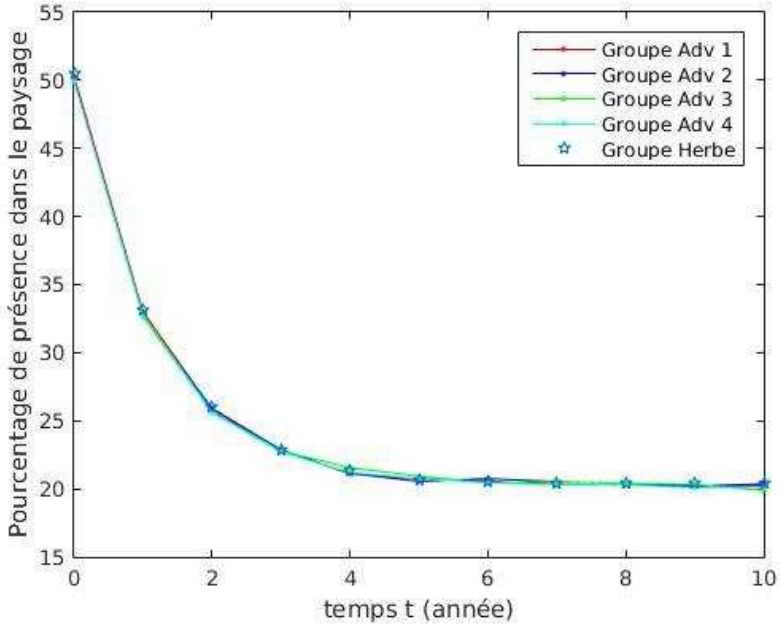


(a) Adventices

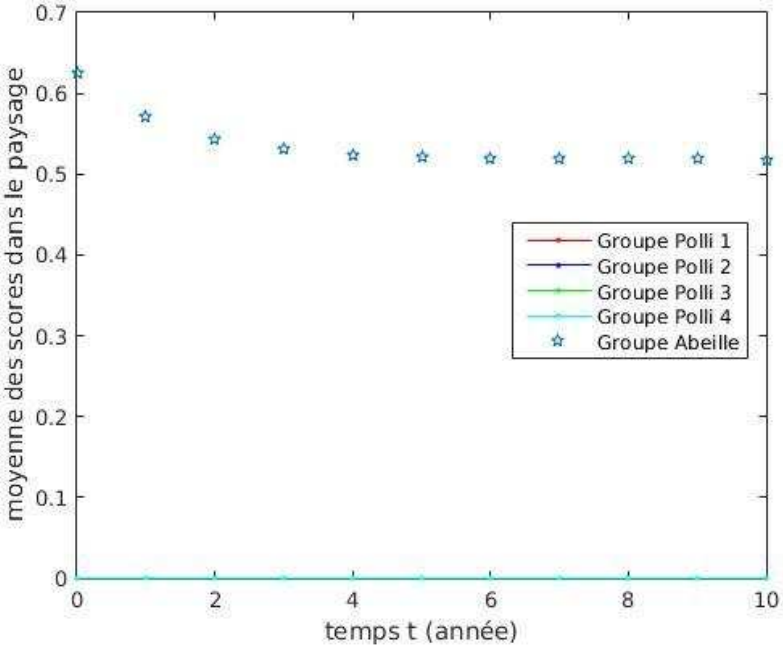


(b) Pollinisateurs

FIGURE C.1 – Monoculture de blé - Évolution moyenne des adventices et pollinisateurs (500 simulations) - paysage parcellaire 10×10

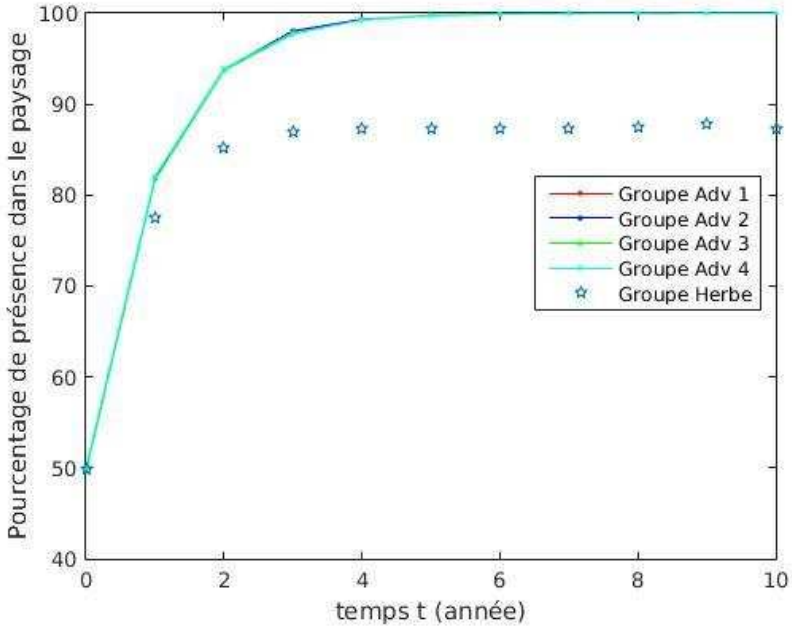


(a) Adventices

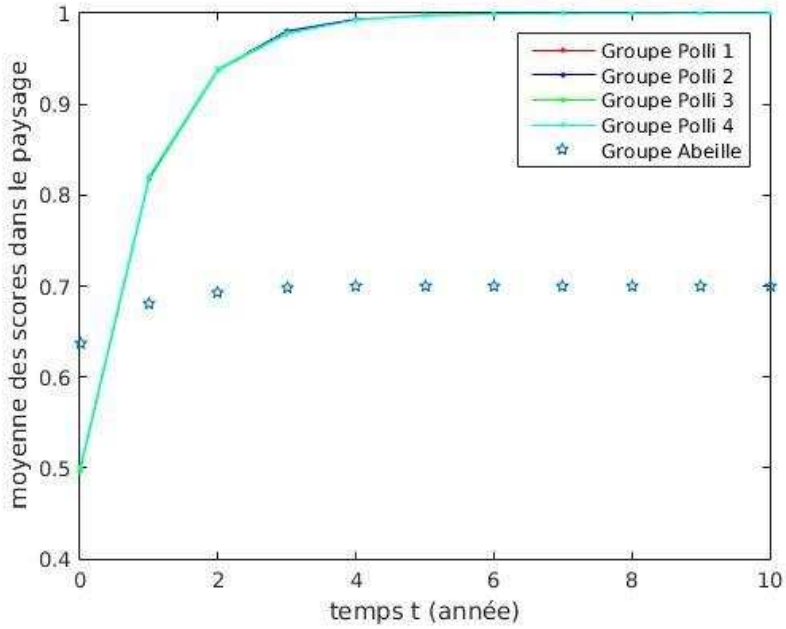


(b) Pollinisateurs

FIGURE C.2 – Monoculture de colza - Évolution moyenne des adventices et pollinisateurs (500 simulations) - paysage parcellaire 10×10



(a) Adventices



(b) Pollinisateurs

FIGURE C.3 – Monoculture de prairie - Évolution moyenne des adventices et pollinisateurs (500 simulations) - paysage parcellaire 10×10

C.2 Modélisation des fonctions de récompense associées aux différents objectifs

Ici, nous écrivons les fonctions de récompense associées aux différents objectifs considérés sous forme additive comme le requiert le cadre PDMF[3]. Il y a plusieurs manières de modéliser de manière additive les fonctions de récompense :

— minimiser le nombre de fonctions/facteurs de récompense

— minimiser la taille des facteurs de récompense.

Nous avons choisi de mettre la priorité sur le deuxième point, car cela nous semble conduire à des temps de calcul plus rapides.

Objectif 1 : On veut maximiser la marge en colza-blé :

$$\begin{aligned}
R^1(S^t, A^t, A^{t-1}) &= \sum_{p=1}^{P} m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p)\, \mathbb{1}_{A^t_p \neq prairie} \\
&= \sum_{p=1}^{P} \Bigg[\Bigg[r^{colza}_{min} + (r^{colza}_{max} - r^{colza}_{min}) \sum_{k=1}^{K} w_k \sum_{q \in V^{\alpha_5}(p)} c_{qp5} F_{q5k}(S^t_{q,-5}, A^t_q) \Bigg] \\
&\quad (1 - a(S^t_p)) - cout^{colza}_p(A^{t-1}_p) \Bigg] \mathbb{1}_{A^t_p = colza} \\
&\quad + [r^{ble}(1 - a(S^t_p)) - cout_{ble}(A^{t-1}_p)]\, \mathbb{1}_{A^t_p = ble} \\
&= \sum_{p=1}^{P} \Big[(r^{colza}_{min}(1 - a(S^t_p)) - cout^{colza}_p(A^{t-1}_p))\, \mathbb{1}_{A^t_p = colza} \Big] \\
&\quad + \Big[(r^{ble}(1 - a(S^t_p)) - cout^{ble}_p(A^{t-1}_p))\, \mathbb{1}_{A^t_p = ble} \Big] \\
&\quad + \sum_{p=1}^{P} \sum_{q \in V^{\alpha_5}(p)} \sum_{k=1}^{K} w_k c_{qp5} F_{q5k}(S^t_{q,-5}, A^t_q)(1 - a(S^t_p))(r^{colza}_{max} - r^{colza}_{min})\, \mathbb{1}_{A^t_p = colza} \\
&= \sum_{p=1}^{P} f_1(S^t_p, A^t_p, A^{t-1}_p) + \sum_{p=1}^{P} \sum_{q \in V^{\alpha_5}(p), q \neq p} f_2(S^t_p, S^t_{q,-5}, A^t_p, A^t_q)
\end{aligned}$$

où

$$\begin{aligned}
f_1(S^t_p, A^t_p, A^{t-1}_p) &= \Big[(r^{colza}_{min}(1 - a(S^t_p)) - cout^{colza}_p(A^{t-1}_p))\, \mathbb{1}_{A^t_p = colza} \Big] \\
&\quad + \Big[(r^{ble}(1 - a(S^t_p)) - cout^{ble}_p(A^{t-1}_p))\, \mathbb{1}_{A^t_p = ble} \Big] \\
&\quad + (1 - a(S^t_p))(r^{colza}_{max} - r^{colza}_{min})\, \mathbb{1}_{A^t_p = colza} c_{pp5} \sum_{k=1}^{K} w_k F_{p5k}(S^t_{p,-5}, A^t_p)
\end{aligned}$$

$$f_2(S^t_p, S^t_{q,-5}, A^t_p, A^t_q) = (1 - a(S^t_p))(r^{colza}_{max} - r^{colza}_{min})\, \mathbb{1}_{A^t_p = colza} c_{qp5} \sum_{k=1}^{K} w_k F_{q5k}(S^t_{q,-5}, A^t_q)$$

Pour l'objectif 1, il y a donc

$$r^1 = P + \sum_{p=1}^{P}(|V^{\alpha_5}(p) - 1|) = P + \sum_{p=1}^{P}|V^{\alpha_5}(p)| - P = \sum_{p=1}^{P}|V^{\alpha_5}(p)|$$

fonctions de récompense par pas de temps.
R^1 est entre 0 et P, donc V^1 est entre 0 et PT, où T est l'horizon.

Objectif 1 bis : On veut maximiser la marge agricole (colza-blé-prairie) :

$$\begin{aligned}
R^{1bis}(S^t, A^t, A^{t-1}) &= \sum_{p=1}^{P} m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p) \\
&= \sum_{p=1}^{P}\Bigg[\Bigg[r^{colza}_{min} + (r^{colza}_{max} - r^{colza}_{min})\sum_{k=1}^{K} w_k \sum_{q\in V^{\alpha_5}(p)} c_{qp5}F_{q5k}(S^t_{q,-5}, A^t_q)\Bigg] \\
&\quad (1 - a(S^t_p)) - cout^{colza}_p(A^{t-1}_p)\Bigg]\mathbb{1}_{A^t_p=colza} \\
&\quad +[r^{ble}(1 - a(S^t_p)) - cout^{ble}_p(A^{t-1}_p)]\,\mathbb{1}_{A^t_p=ble} \\
&\quad +m^{prairie}_1\mathbb{1}_{A^t_p=prairie,A^{t-1}_p\neq prairie} + m^{prairie}_2\mathbb{1}_{A^t_p=prairie,A^{t-1}_p=prairie} \\
&= \sum_{p=1}^{P}\Big[(r^{colza}_{min}(1 - a(S^t_p)) - cout^{colza}_p(A^{t-1}_p))\,\mathbb{1}_{A^t_p=colza}\Big] \\
&\quad +\Big[(r^{ble}(1 - a(S^t_p)) - cout^{ble}_p(A^{t-1}_p))\,\mathbb{1}_{A^t_p=ble}\Big] \\
&\quad +\sum_{p=1}^{P}\sum_{q\in V^{\alpha_5}(p)}\sum_{k=1}^{K} w_k c_{qp5}F_{q5k}(S^t_{q,-5}, A^t_q)(1 - a(S^t_p))(r^{colza}_{max} - r^{colza}_{min})\,\mathbb{1}_{A^t_p=colza} \\
&\quad +m^{prairie}_1\mathbb{1}_{A^t_p=prairie,A^{t-1}_p\neq prairie} + m^{prairie}_2\mathbb{1}_{A^t_p=prairie,A^{t-1}_p=prairie} \\
&= \sum_{p=1}^{P} f_{1bis}(S^t_p, A^t_p, A^{t-1}_p) + \sum_{p=1}^{P}\sum_{q\in V^{\alpha_5}(p),q\neq p} f_2(S^t_p, S^t_{q,-5}, A^t_p, A^t_q)
\end{aligned}$$

où

$$\begin{aligned}
f_{1bis}(S^t_p, A^t_p, A^{t-1}_p) &= \Big[(r^{colza}_{min}(1 - a(S^t_p)) - cout^{colza}_p(A^{t-1}_p))\,\mathbb{1}_{A^t_p=colza}\Big] \\
&\quad +\Big[(r^{ble}(1 - a(S^t_p)) - cout^{ble}_p(A^{t-1}_p))\,\mathbb{1}_{A^t_p=ble}\Big] \\
&\quad +(1 - a(S^t_p))(r^{colza}_{max} - r^{colza}_{min})\,\mathbb{1}_{A^t_p=colza}c_{pp5}\sum_{k=1}^{K} w_k F_{p5k}(S^t_{p,-5}, A^t_p) \\
&\quad +m^{prairie}_1\mathbb{1}_{A^t_p=prairie,A^{t-1}_p\neq prairie} + m^{prairie}_2\mathbb{1}_{A^t_p=prairie,A^{t-1}_p=prairie}
\end{aligned}$$

Pour l'objectif 1 bis, il y a donc

$$r^{1bis} = P + \sum_{p=1}^{P}(|V^{\alpha_5}(p) - 1|) = P + \sum_{p=1}^{P}|V^{\alpha_5}(p)| - P = \sum_{p=1}^{P}|V^{\alpha_5}(p)|$$

fonctions de récompense par pas de temps.
R^{1bis} est entre 0 et P, donc V^{1bis} est entre 0 et PT, où T est l'horizon.

Objectif 2 : On veut maximiser la biodiversité (adventices et pollinisateurs sauvages et domestiques). On a donc :

$$\begin{aligned}
R^2(S^t, A^t, A^{t-1}) &= R^2(S^t, A^t) = \sum_{p=1}^{P}\sum_{i=1}^{I} S^t_{pi} + \sum_{p=1}^{P}\sum_{k=1}^{K} w_k \sum_{q\in V^{\alpha_5}(p)} c_{qp5}\mathrm{F}_{q5k}(S^t_{q,-5}, A^t_q) \\
&\quad + \sum_{p=1}^{P}\sum_{b=1}^{B-1}\sum_{q_1\in V(p)}\sum_{q_2\in V(p)} h(S^t_{q_1 RT(b)}, A^t_{q_2}) \\
&= \sum_{p=1}^{P}\sum_{q\in V^{\alpha_5}(p), q\neq p}\sum_{k=1}^{K} w_k c_{qp5} F_{q5k}(S^t_{q,-5}, A^t_q) \\
&\quad + \sum_{p=1}^{P}\left(\sum_{i=1}^{I} S^t_{pi} + \sum_{k=1}^{K} w_k c_{pp5} F_{p5k}(S^t_{p,-5}, A^t_p)\right) \\
&\quad + \sum_{p=1}^{P}\sum_{b=1}^{B-1}\sum_{q_1\in V(p)}\sum_{q_2\in V(p)} h(S^t_{q_1 RT(b)}, A^t_{q_2}) \\
&= \sum_{p=1}^{P}\sum_{q\in V^{\alpha_5}(p), q\neq p} f(S^t_{q,-5}, A^t_q) \\
&\quad + \sum_{p=1}^{P} g(S^t_p, A^t_p) + \sum_{p=1}^{P}\sum_{b=1}^{B-1}\sum_{q_1\in V(p)}\sum_{q_2\in V(p)} h(S^t_{q_1 RT(b)}, A^t_{q_2})
\end{aligned}$$

où

$$f(S^t_{q,-5}, A^t_q) = c_{qp5}\sum_{k=1}^{K} w_k F_{q5k}(S^t_{q,-5}, A^t_q)$$

$$g(S^t_p, A^t_p) = \sum_{i=1}^{I} S^t_{pi} + c_{pp5}\sum_{k=1}^{K} w_k F_{p5k}(S^t_{p,-5}, A^t_p)$$

$$h(S^t_{q_1 RT(b)}, A^t_{q_2}) = c_{q_1 pb} F_{q_1 b}(S^t_{q_1 RT(b)}) c_{q_2 pb} H_{q_2 b}(A^t_{q2})$$

Pour l'objectif 2, il y a donc

$$\begin{aligned} r^2 &= \sum_{p=1}^{P}(|V^{\alpha_5}(p)|-1)+P+(B-1)\sum_{p=1}^{P}|V(p)|^2 \\ &= \sum_{p=1}^{P}|V^{\alpha_5}(p)|-P+P+(B-1)\sum_{p=1}^{P}|V(p)|^2 \\ &= \sum_{p=1}^{P}|V^{\alpha_5}(p)|+(B-1)\sum_{p=1}^{P}|V(p)|^2 \end{aligned}$$

fonctions de récompense par pas de temps.
R^2 est entre 0 et $P(I+B)$, donc V^2 est entre 0 et $PT(I+B)$, où T est l'horizon.

Objectif 2 bis : On veut maximiser la biodiversité (adventices et pollinisateurs sauvages) :

$$\begin{aligned} R^{2bis}(S^t,A^t,A^{t-1}) &= R^{2bis}(S^t,A^t)=\sum_{p=1}^{P}\sum_{i=1}^{I}S_{pi}^t+\sum_{p=1}^{P}\sum_{b=1}^{B-1}\sum_{q_1\in V(p)}\sum_{q_2\in V(p)}h(S^t_{q_1RT(b)},A^t_{q_2}) \\ &= \sum_{p=1}^{P}f_5(S_p^t)+\sum_{p=1}^{P}\sum_{b=1}^{B-1}\sum_{q_1\in V(p)}\sum_{q_2\in V(p)}h(S^t_{q_1RT(b)},A^t_{q_2}) \end{aligned}$$

où

$$f_5(S_p^t)=\sum_{i=1}^{I}S_{pi}^t$$

Pour l'objectif 2bis, il y a donc

$$r^{2bis} = P+(B-1)\sum_{p=1}^{P}|V(p)|^2$$

fonctions de récompense par pas de temps.
R^{2bis} est entre 0 et $P(I+B-1)$, donc V^{2bis} est entre 0 et $PT(I+B-1)$, où T est l'horizon.

Objectif 2 ter : On veut maximiser la biodiversité (adventices) :

$$R^{2ter}(S^t,A^t,A^{t-1})=R^{2ter}(S^t,A^t)=\sum_{p=1}^{P}\sum_{i=1}^{I}S_{pi}^t=\sum_{p=1}^{P}f_5(S_p^t)$$

Pour l'objectif 2ter, il y a donc

$$r^{2ter} = P$$

fonctions de récompense par pas de temps.
R^{2ter} est entre 0 et PI, donc V^{2ter} est entre 0 et PTI, où T est l'horizon.

Objectif 3 : On veut maximiser la marge en miel (maximiser l'abondance des abeilles dans le paysage). On a donc :

$$\begin{aligned} R^3(S^t, A^t, A^{t-1}) &= R^3(S^t, A^t) = marge_{miel}(S^t, A^t) = \sum_{p=1}^{P} PO_{p5}(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}) \\ &= \sum_{p=1}^{P}\sum_{k=1}^{K} w_k \sum_{q\in V^{\alpha_5}(p)} c_{qp5}\mathrm{F}_{q5k}(S^t_{q,-5}, A^t_q) \\ &= \sum_{p=1}^{P} \sum_{q\in V^{\alpha_5}(p)} c_{qp5} \sum_{k=1}^{K} w_k \mathrm{F}_{q5k}(S^t_{q,-5}, A^t_q) \\ &= \sum_{p=1}^{P} \sum_{q\in V^{\alpha_5}(p)} f(S^t_{q,-5}, A^t_q) \end{aligned}$$

Pour l'objectif 3, il y a donc

$$r^3 = \sum_{p=1}^{P} |V^{\alpha_5}(p)|$$

fonctions de récompense par pas de temps.
R^3 est entre 0 et P, donc V^3 est entre 0 et PT, où T est l'horizon.

Objectif C1 : Il s'agit ici de maximiser le rendement en colza-blé sur les parcelles de l'ensemble E_1, et la biodiversité (adventices-pollinisateurs) sur les parcelles de l'ensemble E_2. Autrement dit, remplir l'objectif 1 sur les parcelles de l'ensemble E_1 et l'objectif 2 sur les parcelles de l'ensemble E_2.

$$\begin{aligned} R^{C1}(S^t, A^t, A^{t-1}) &= \frac{1}{|E_1|}\sum_{p\in E_1} m_p(S^t_{V^{\alpha_5}(p)}, A^t_{V^{\alpha_5}(p)}, A^{t-1}_p)\, \mathbb{1}_{A^t_p \neq prairie} \\ &\quad + \frac{1}{|E_2|(I+B)} \sum_{p\in E_2} \left(\sum_{i=1}^{I} S^t_{pi} + \sum_{b=1}^{B} PO^t_{pb} \right) \\ &= \frac{1}{|E_1|}\sum_{p\in E_1} f_1(S^t_p, A^t_p, A^{t-1}_p) + \frac{1}{|E_1|}\sum_{p\in E_1}\sum_{q\in V^{\alpha_5}(p), q\neq p} f_2(S^t_p, S^t_{q,-5}, A^t_p, A^t_q) \\ &\quad + \frac{1}{|E_2|(I+B)} \sum_{p\in E_2}\sum_{q\in V^{\alpha_5}(p), q\neq p} f(S^t_{q,-5}, A^t_q) + \frac{1}{|E_2|(I+B)}\sum_{p\in E_2} g(S^t_p, A^t_p) \\ &\quad + \frac{1}{|E_2|(I+B)} \sum_{p\in E_2}\sum_{b=1}^{B-1}\sum_{q_1\in V(p)}\sum_{q_2\in V(p)} h(S^t_{q_1 RT(b)}, A^t_{q_2}) \end{aligned}$$

Ici, il est important de normaliser pour que les deux objectifs soient entre 0 et 1. Pour l'objectif C1, il y a

$$r^{C1} = \sum_{p\in E_1} |V^{\alpha_5}(p)| + \sum_{p\in E_2} |V^{\alpha_5}(p)| + (B-1)\sum_{p\in E_2} |V(p)|^2$$

fonctions de récompense par pas de temps. On a $r^1 \leq r^{C1} \leq r^2$. R^{C1} est entre 0 et 2, donc V^{C1} est entre 0 et $2T$, où T estl'horizon.

Objectif C2 : Il s'agit ici de maximiser le nombre moyen d'années et d'exploitations [2] pour lesquelles la marge en blé est supérieure à un certain seuil β (si la parcelle n'est pas en blé elle ne compte pas) et le nombre de groupes adventices est supérieur à un certain seuil ζ. Soit L_r le nombre d'exploitations et $Q_r(l)$ l'ensemble des parcelles de l'exploitation $l \in \{1, ..., L_r\}$.

$$\begin{aligned} R^{C2}(S^t, A^t, A^{t-1}) &= \sum_{l=1}^{L_r} \mathbb{1}\left\{ \sum_{p \in Q_r(l)} f_{5bis}(S_p^t, A_p^t, A_p^{t-1}) \geq \beta \right\} \mathbb{1}\left\{ \sum_{p \in Q_r(l)} f_5(S_p^t) \geq \zeta \right\} \\ &= \sum_{l=1}^{L_r} f_{C2}(S^t_{Q_r(l)}, A^t_{Q_r(l)}, A^{t-1}_{Q_r(l)}) \end{aligned}$$

où

$$f_{5bis}(S_p^t, A_p^t, A_p^{t-1}) = (r^{ble}(1 - a(S_p^t)) - cout_p^{ble}(A_p^{t-1}))\, \mathbb{1}_{A_p^t = ble}$$

$$f_{C2}(S^t_{Q_r(l)}, A^t_{Q_r(l)}, A^{t-1}_{Q_r(l)}) = \mathbb{1}\left\{ \sum_{p \in Q_r(l)} f_{5bis}(S_p^t, A_p^t, A_p^{t-1}) \geq \beta \right\} \mathbb{1}\left\{ \sum_{p \in Q_r(l)} f_5(S_p^t) \geq \zeta \right\}$$

Pour l'objectif C2, il y a donc

$$r^{C2} = L_r$$

fonctions de récompense par pas de temps. R^{C2} est entre 0 et L_r, donc V^{C2} est entre 0 et $L_r T$, où T est l'horizon.

2. Une exploitation est un ensemble de parcelles appartenant à un même agriculteur et qui ne sont pas forcément adjacentes.