Comité de thèse
Première année

Coordination temporelle en ligne pour la décision décentralisée dans l'incertain

Emmanuel RACHELSON

sous la direction de
Frédérick GARCIA
Florent TEICHTEIL

2 février 2007

Table des matières

Introduction

Imaginons une situation d'incendie en forêt, dans laquelle est plongé un robot pompier. Ce robot, sorte de rover terrestre autonome équipé d'une lance à eau et d'un réservoir, a pour mission d'éteindre l'incendie au milieu duquel il se trouve. Or, seul, il n'est pas très efficace car il dispose de peu d'information sur l'état global du feu dans la forêt. Heureusement, le concepteur du système lui a adjoint un binôme, un drone hélicoptère, capable éventuellement, lui aussi de s'attaquer au feu, mais surtout disposant d'un champ de vision beaucoup plus large. Pour éviter la perte des deux agents en cas de défaillance d'un seul, il a été décidé qu'il n'y aurait pas de supérieur hiérarchique et que chaque agent déciderait par lui-même des actions qu'il entreprend. On a également doté les agents d'une sorte de modèle d'incendie leur permettant d'anticiper l'évolution vraisemblable du feu sur les minutes à venir. On se pose à présent la question de savoir comment on va donner de l'autonomie décisionnelle aux agents, en particulier comment on va leur permettre de construire un plan d'action.

Cette situation présente un certain nombre de caractéristiques qui nous permettent de cerner le type de problème auquel on s'attaque :

- On est dans un cadre *biagent*
- On souhaite mettre en place un processus de décision *décentralisé*
- On souhaite permettre à nos agents d'adapter leurs décisions *en ligne*
- Chaque agent doit effectuer une *planification* pour lui-même qui présente les caractéristiques suivantes :
  - Les actions de l'agent ont des issues *incertaines*
  - L'environnement évolue avec le temps, il est *instationnaire*.

L'objectif des travaux de recherche que l'on présente ici et qui constituent le travail de la première année de thèse, est d'apporter une réponse à ce type de problèmes de décision décentralisée, en ligne, dans l'incertain et en environnement instationnaire. On pourra, si les résultats s'y prêtent, ajouter d'autres épithètes, qualifiant alors les possibilités de communication entre les agents, l'observabilité du milieu, etc.

Pour l'instant, on s'intéresse spécifiquement au problème mentionné ci-dessus, qu'on décompose hiérarchiquement en deux problèmes :

1. Un problème bi(multi)agent décentralisé où le protocole de communication, de négociation et de décision est à définir.
2. Un problème de planification monoagent en fonction du temps, en environnement dynamique et dans l'incertain.

On s'est intéressé en premier lieu au problème monoagent. Partant du cadre des Processus Décisionnels de Markov (MDP), on a amélioré un modèle de décision intégrant une variable temporelle explicite et continue proposé dans la littérature (le modèle TMDP) et proposé une

autre formulation (SMDP+). Puis on a exploré les différentes possibilités de résolution des deux formalismes pour parvenir à des algorithmes de résolution utilisables sur des cas concrets. Actuellement, ces algorithmes sont en partie implémentés et le travail continue pour parvenir à un planificateur permettant de résoudre des problèmes de décision dans l'incertain présentant une dépendance explicite à une variable temporelle.

Par ailleurs, le travail de fond effectué sur les deux formalismes précédents a permis de faire émerger la notion d'espaces d'actions continus (ou d'actions paramétriques) dans le cadre MDP instationnaire à temps observable. Cette notion englobe les deux formalismes précédents et propose une extension logique aux méthodes actuelles de traitement des MDP à variables d'état continues et discrètes.

Enfin, sur l'aspect biagent (ou multiagent), une première proposition de mécanisme en ligne de négociation et d'amélioration de politiques a été proposé et attend la disponibilité du planificateur mentionné plus haut pour être évaluée.

Ce rapport d'avancement s'articule de la façon suivante. Dans un premier temps, on s'intéresse au problème monoagent et à sa formalisation (chapitre 1). On présente le problème, ses aspects et sa portée, puis on introduit les différents cadres de modélisation classiques pour arriver à deux représentations adaptées à notre problème, les cadres TMDP et SMDP+. On présente alors brièvement les options dont on dispose pour la résolution et on détaille plus avant les spécificités du problème temporel ainsi que des situations concrètes auxquelles on s'intéresse. Puis, au chapitre 2, on présente en détail l'algorithme que l'on a développé pour la résolution par programmation dynamique des TMDP/SMDP+, suivi, au chapitre 3, par une méthode alternative de résolution par discrétisation optimale de la variable temporelle. Les travaux entrepris sur les deux formalismes précédents donnent naissance au chapitre 4 à une représentation générale des actions paramétriques continues dans le cadre MDP instationnaire, représentation qui, on le verra, dépasse le cadre des problèmes temporels et propose une extension aux problèmes généraux de MDP à variables continues et discrètes. Enfin, au chapitre 5, on présente la méthode de résolution que l'on propose pour notre problème de décision décentralisée.

Chapitre 1

Présentation et modélisation du problème de planification en fonction d'un temps explicite

1.1 Exemples de problèmes dépendant du temps

Imaginons devoir planifier notre trajet dans Toulouse pour aller de l'INRA à l'ONERA. Les différentes options qui se présentent sont de prendre la voiture, le bus ou le métro sur différentes parties du trajet. En fonction de l'heure, le trafic automobile peut être plus ou moins fluide sur certains axes, différent dans les petites rues, etc. Par ailleurs, en fonction de l'horaire de la journée, il y a plus ou moins de bus qui circulent et ceux-ci sont — dans une moindre mesure — affectés par le trafic routier. Enfin le métro représente une solution envisageable mais il est malheureusement loin de la maison et il faut prévoir un trajet vers la station la plus proche. Ce problème, variante de celui présenté dans [BL01], présente les caractéristiques principales des problèmes auxquels on s'intéresse :

Incertitude sur le résultat des actions : une rue barrée, un chauffeur de bus qui oublie un arrêt, un métro annulé ; on décrit le résultat d'une action entreprise dans un état donné comme l'ensemble des résultats possibles, pondérés chacun d'une probabilité d'occurrence.

Dépendance explicite au temps : Le modèle dépend explicitement de la variable "date courante" (horaires de passage des bus, dépendance entre le trafic et l'heure ...), cette dernière est *continue* et *observable* par l'agent qui peut donc spécifier sa stratégie en fonction de l'heure qu'il est.

Incertitude sur la date de fin des actions : l'heure d'arrivée d'un métro ou d'un bus, la durée d'un trajet en voiture, sont sujets à incertitude.

D'autres problèmes présentent ce type de caractéristiques. Un exemple que l'on retrouvera plus loin dans le document concerne la mission d'un robot pompier devant traverser une forêt en feu pour remplir son réservoir d'eau. Cet agent dispose d'un modèle d'évolution de l'incendie et construit son plan en fonction. Le problème de planification des activités d'un satellite d'observation de la Terre en fonction de l'heure de la journée peut également rentrer dans ce cadre : l'incertitude porte sur le résultat des prises de vue et l'environnement du satellite change continûment avec le temps (sa position orbitale par exemple). Enfin, parmi les problèmes faisant appel à cette modélisation d'un environnement dynamique et incertain, il y a les problèmes de coordination temporelle (de rendez-vous) comme par exemple le problème de deux entreprises

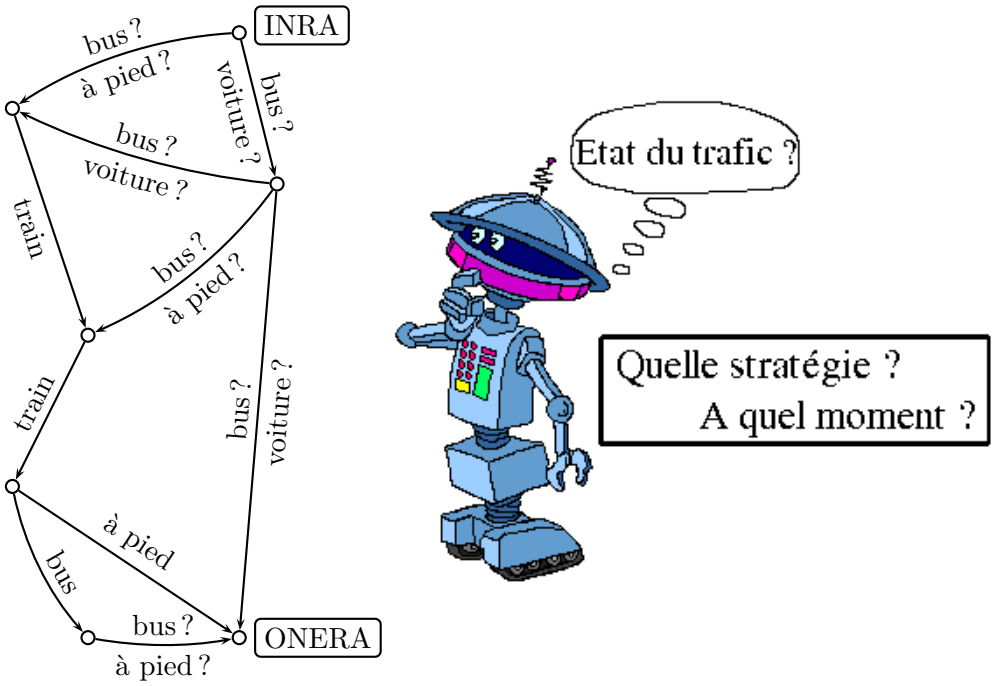

FIGURE 1.1 – Exemple

qui construisent ensemble un avion et qui doivent coordonner leurs actions sans qu'aucune puisse imposer un planning à l'autre : l'exercice revient alors à maximiser un profit économique en coordonnant au mieux sa propre stratégie de production avec l'"environnement", ce dernier incluant les actions prévues par l'autre entreprise.

1.2 Modélisation

1.2.1 Des MDP aux SMDP

Afin de trouver des plans dans l'incertain, plusieurs approches existent dans la littérature. Nous distinguons d'une part les approches qui recherchent un plan séquentiel explicite, comme les approches de planification dans l'espace des plans partiels présentées dans [KHW95], [DHW94] ou [ML98]. Par ailleurs, il existe des approches de recherche de politiques robustes et valuées permettant de définir des stratégies sur tout l'espace d'états du problème. C'est dans ce second cadre que l'on se situe.

Afin de modéliser notre problème, nous allons partir d'un cadre devenu classique en décision dans l'incertain, les Processus Décisionnels de Markov (MDP) ([Put94]). Dans cette section, nous présenterons brièvement les bases des MDP puis nous verrons comment les dépendances au temps s'expriment dans les différents modèles dérivés.

Le modèle MDP

Un problème représenté sous forme de MDP est constitué d'un espace d'états dénombrable S, d'un espace d'actions dénombrable A, d'un modèle de transition $P(s'|s,a)$ associant une

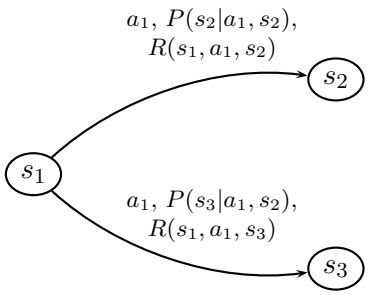

FIGURE 1.2 – Illustration des transitions possibles pour une action

probabilité à chaque transition (s, a, s'), et d'un modèle de récompense R associant une valeur réelle à chaque transition. On peut représenter un MDP comme un graphe d'états-transitions (figure 1.2).

Lorsqu'on cherche à construire une fonction qui indique l'action à entreprendre à chaque étape de l'exécution, il apparait que cette action dépend de l'état courant, de l'état initial et de la séquence des actions qui ont mené dans l'état courant. Ainsi, la politique (ou le plan conditionnel) que l'on recherche semble dépendre de l'historique des actions entreprises. Sous les hypothèses de stationnarité du problème, on peut montrer que cette action ne dépend, pour un problème à horizon infini, que de l'état courant. On définit ainsi des fonctions qui, à chaque état courant, associent une action. Ces politiques sont dites Markoviennes car elles ne dépendent que du dernier état visité. L'objectif est alors d'optimiser ces politiques.

Pour cela, on se dote d'un critère : on cherche à trouver la politique $\pi : S \to A$ qui maximise la fonction de valeur :

$$V_\gamma^\pi = E\left(\sum_{\delta=0}^{\infty} \gamma^{t_\delta} r_\delta^\pi | s_0\right) \tag{1.1}$$

Dans l'expression précédente, r_δ^π désigne la récompense obtenue à l'étape δ en suivant la politique π, et t_δ désigne la date à laquelle la $\delta^{\text{ième}}$ action est entreprise. Ce critère s'appelle "γ-pondéré" et son pendant pour $\gamma = 1$ est appelé "critère total". On peut considérer ce facteur γ comme une probabilité de non-panne, ou encore comme une pénalisation sur les récompenses obtenues loin dans le futur ; on a garantie de convergence de la somme si $\gamma < 1$ et R borné.

Dans un MDP classique, on considère toutes les durées d'action comme unitaires et on a $t_\delta = \delta$. Le critère précédent se traduit par une équivalence entre fonction de valeur maximale V^* et politique optimale π^* et on en tire l'équation de Bellman pour les MDP : $V^* = LV^*$:

$$V^*(s) = \max_{a \in A} \left(\sum_{s' \in S} P(s'|s, a) \left(R(s', a, s) + \gamma V^*(s') \right) \right) \tag{1.2}$$

On se ramène à une formulation plus simple en écrivant le modèle de récompense $r(s, a)$ comme la moyenne des récompenses accessibles à partir de (s, a) pondérée par les probabilités de transition : $r(s, a) = \sum_{s' \in S} P(s'|s, a) R(s, a, s')$.

On peut notamment résoudre les MDP par programmation dynamique ([Bel57]) en construisant la suite des $V_{n+1} = LV_n$, qui converge asymptotiquement vers V^*. On peut également effectuer une résolution par programmation linéaire qui converge en un nombre d'itérations borné. De nombreuses techniques ont été développées autour du cadre MDP afin de contrer le *curse of dimensionnality* de Bellman ([Bel57]) qui limite la taille des problèmes traitables dans un cadre où on énumère tous les états. Notre propos n'est pas d'effectuer une liste exhaustive de ces techniques, nous mentionnerons toutefois les techniques d'exploitation d'une décomposition de l'espace d'états présentées dans [HMK+98], [Par98], [DL95] et [Sab02]. Par ailleurs, la représentation sous forme de variables d'état, discrètes ou continues et la représentation factorisée du problème associé ont été abordés dans [BDG99], [HSHB00]. La résolution par programmation linéaire des problèmes factorisés est abordée dans [HK04] et [GHK04]. Enfin, des approches de hiérarchisation de l'espace d'actions permettant une exploitation dans le cadre de l'apprentissage par renforcement ont été développées notamment dans [Die98] et [Die00].

Le modèle MDP permet donc de considérer des actions de durée unitaire dans un environnement stationnaire. L'optimisation d'un MDP (par programmation dynamique par exemple) sur un horizon infini (donc avec un nombre de "coups" infini) détermine une fonction de valeur $V^*(s)$ qui représente l'espérance des gains atteignables à partir de tout état s.

À partir de cette base de modélisation, on va chercher à prendre en compte les durées incertaines des actions dans le modèle.

Le modèle SMDP

Le modèle des Processus Décisionnels Semi-Markoviens ou SMDP ([Put94]) intègre la notion de coût de durée d'action. Il enrichit le modèle MDP en transformant le modèle de transition en une fonction $Q(\tau, s'|s, a)$ qui décrit la densité de probabilité que la prochaine décision soit prise τ unités de temps dans l'avenir, dans l'état s', sachant qu'on entreprend actuellement l'action a dans l'état s. On décompose généralement la fonction Q en : $Q(\tau, s'|s, a) = P(s'|s, a) \cdot F(\tau|s, a)$, cela traduit une hypothèse forte sur le modèle : on suppose la durée de transition indépendante de l'état d'arrivée. On définit également les fonctions de taux de coût $c(s', a, s)$.

Le modèle SMDP considère en fait deux processus distincts hiérarchisés [Put94] : un processus réel de "bas niveau" qui comporte tous les états temporaires que traverse le système et le processus de "haut niveau" qu'on étudie (le SMDP lui-même). Le processus de bas niveau permet une description fine des différents taux de coût (par unité de temps) appliqués à l'agent pendant une transition du processus de haut niveau. Les deux processus concordent aux dates de décision. Le SMDP peut être vu comme une version hiérarchisée de ces deux processus. Les deux processus sont liés par la fonction $p(j|t, s, a)$ qui donne la probabilité que le processus réel soit dans l'état j, t unités de temps après avoir pris la décision a au point s. L'étude du modèle de récompense du processus réel permet de définir une fonction de récompense du SMDP $k(s, a)$ sous la forme :

$$k(s,a) = r(s,a) + \int_0^\infty \sum_{j \in S} \left[\int_0^u \gamma^t c(j,s,a) p(j|t,s,a) dt \right] F(du|s,a) \tag{1.3}$$

Et l'évaluation de la politique devient :

$$V^\pi(s) = k_\pi(s) + \sum_{s' \in S} \int_0^\infty \gamma^\tau \cdot V^\pi(s') \cdot Q_\pi(d\tau, s'|s) \tag{1.4}$$

On a alors, avec $m_\pi(j|s) = \int_0^\infty \gamma^t \cdot Q_\pi(d\tau, s'|s)$:

$$V^*(s) = \max_{a \in A}\{r(s,a) + \sum_{s' \in S} m(s'|s,a)V^*(s')\} \tag{1.5}$$

On est donc ramené à la résolution d'un MDP classique. On remarque qu'en introduisant une durée de transition entre états, on a pris en compte des coûts de transition associés à cette durée. Cependant, le modèle que l'on considère est toujours stationnaire et ne permet toujours pas de représenter notre problème initial. En effet, c'est le modèle lui-même qui doit dépendre explicitement du temps. Pour cela la variable temps doit être rendue observable pour qu'on puisse planifier en fonction d'elle. Prendre en compte des durées d'action ne suffit pas, il faut pouvoir observer la date courante, on cherche donc des modèles qui étendraient le cadre MDP aux problèmes instationnaires.

D'autres approches de prise en compte d'une variable temporelle

D'autres approches — ne prenant pas en compte les trois aspects, d'incertitude sur le résultat des actions, d'incertitude sur la durée des actions et de dépendance du modèle aux temps — proposent des solutions dans des cadres différents. On peut notamment citer les algorithmes de recherche de plus court chemin ou les Stochastic Time Dependent Network (STDN, [WFL95]) qui s'inscrivent dans le cadre de transitions déterministes. Les deux seuls modèles proposés à ce jour qui intègrent les trois aspects évoqués ci-dessus sont — à notre connaissance — le modèle SMDP+ et le modèle TMDP, présentés dans les deux sections suivantes et dont l'équivalence est démontrée par la suite.

1.2.2 Le modèle SMDP+

Le modèle SMDP+ constitue la première contribution de la thèse. Il s'inspire du modèle SMDP et cherche à y intégrer les deux aspects suivants :

— La dépendance explicite à la date courante dans le modèle de transition et de récompense
— La dépendance possible entre état d'arrivée et durée de transition.

Précisons ce dernier point : dans le modèle SMDP, lorsque l'on écrit $Q(\tau, s'|s, a) = P(s'|s, a) \cdot F(\tau|s, a)$, on suppose implicitement deux choses :

— Le modèle est stationnaire (pas de dépendance en t dans Q)
— la durée τ de la transition et l'état d'arrivée sont indépendants.

Partant de ce constat, on définit un SMDP+ comme un quadruplet $< \Sigma, A, Q, R >$:

— Σ un espace d'états $\sigma = (s, t)$ augmenté qui se décompose en :
  – Un espace d'états discrets $s \in S$
  – Un axe du temps continu $t \in \mathbb{R}$
— A un espace d'actions discrètes
— $Q(\sigma'|\sigma, a)$ une densité de probabilité de transition qui se décompose en $Q(\sigma'|\sigma, a) = P(s'|s, t, a) \cdot F(t'|s, t, a, s')$
— $R(\sigma', a, \sigma)$ une fonction de récompense.

Une politique sur un SMDP+ se définit comme une fonction de $\mathbb{R} \times S$ dans A et on évalue une telle politique selon l'équation :

$$V^{\pi}(\sigma) = \sum_{s' \in S} \int_0^{\infty} \left(r(s', t+\tau, \pi(\sigma), \sigma) + \gamma^{\tau} V^{\pi}(\sigma') \right) \cdot F(\tau | \sigma, \pi(\sigma), s') P(s' | \sigma, \pi(\sigma)) d\tau = L_{\pi}^{t}(V^{\pi})(\sigma) \tag{1.6}$$

On peut remarquer qu'une faiblesse du modèle SMDP+ réside dans l'absence d'une action "attendre" clairement définie. En effet, on peut définir des actions "attendre τ secondes" ou "attendre jusqu'à T" (définir une action revient à savoir écrire les fonctions P, F et R correspondantes) mais on ne peut pas définir d'action "ne rien faire" car alors on ne saurait pas écrire la fonction F correspondante (on peut toutefois définir une dynamique d'état P et une fonction de coût R). On verra en section 1.4, et plus spécifiquement en section 1.4.4, comment on peut contourner ce problème et réintroduire une action "ne rien faire" ou une action "attendre une date où il faut agir".

Pour l'instant, on se contente de définir une politique plus généralement : on définit un espace d'actions augmenté $A+ = A \cup \{attendre\}$ et on définit une politique π comme une fonction de $S \times \mathbb{R}$ dans $A+$. Appliquer la politique revient à exécuter l'action $\pi(s,t)$ en s à l'instant t. On abordera le problème du critère d'optimisation de la politique et la cohérence avec le modèle à la section 1.4.

Le modèle TMDP est une spécialisation au modèle SMDP+ dans le cadre du critère total et de certaines formes de fonctions de transition, récompense et densités de probabilités. Il a été proposé par [BL01], nous l'abordons à la section suivante.

1.2.3 Le modèle TMDP

Le modèle TMDP décompose chaque transition issue d'une action a entreprise en s en une série de *réalisations* possibles μ, chaque réalisation désignant un état d'arrivée et une description de la durée de transition. Formellement, un TMDP se définit comme :

- S : un espace d'états discret.
- A : un espace d'actions discret.
- M : espace discret de *réalisations* $\mu = (s'_{\mu}, T_{\mu}, P_{\mu})$:
  - s'_{μ} étant un état résultant.
  - T_{μ} étant un booléen indiquant si la distribution P_{μ} porte sur des dates ou des durées.
  - $P_{\mu}(\theta)$ étant une densité de probabilité décrivant la probabilité que la réalisation se finisse à $t = \theta$ si $T_{\mu} = \text{ABS}$ ou après un temps $\tau = \theta$ si $T_{\mu} = \text{REL}$.
- $L(\mu | s, t, a)$ décrit la probabilité de réaliser μ.
- $R(\mu, t, t')$ décrit la récompense associée à la réalisation μ, commençant en t et finissant en t'.
- $K(s,t)$ décrivant le coût instantané associé à l'action "attendre" dans l'état s.

La dynamique d'un TMDP est illustrée figure 1.3. Dans l'état s_1, entreprendre l'action a_1 permet d'atteindre la réalisation μ_1 avec une probabilité 0.8 et μ_2 avec une probabilité 0.2. μ_1 décrit le passage vers s_2 et la date de fin de transition est donnée par P_{μ_1} tandis que μ_2 décrit l'échec du départ de s_1 avec une durée donnée par P_{μ_2}.

Afin d'assurer le sens physique de la modélisation et la cohérence du modèle, il est important d'imposer que la date t à laquelle on déclenche une réalisation μ soit inférieure à toutes les dates

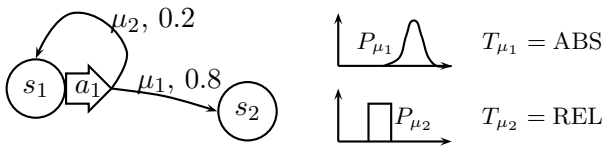

FIGURE 1.3 – TMDP - éléments de base

de fin possibles pour cette réalisation. De façon plus formelle, si on pose :
$Dep_{\mu,s,a} = \{t \in \mathbb{R} / L(\mu|s,t,a) \neq 0\}$,
$M_{ABS} = \{\mu \in M / T_\mu = ABS\}$ et
$Arr_\mu = \{t' \in \mathbb{R} / P_\mu(t') \neq 0\}$, alors :
$\forall (\mu,s,a) \in M_{ABS} \times S \times A, \quad \forall t \in Dep_{\mu,s,a}, \quad \forall t' \in Arr_\mu, \quad t < t'.$

Tout comme le modèle SMDP+, le modèle TMDP n'intègre pas d'action "attendre" clairement définie. La dynamique de l'état peut être rajoutée au modèle, ainsi que le coût instantané de l'action "attendre", mais on ne peut pas y définir de densité de probabilité P_μ.

Dans le cadre TMDP, on cherche des politiques sous la forme suivante : une politique TMDP est une fonction qui, à chaque paire $(s,t) \in S \times \mathbb{R}$, associe une paire $(t',a) \in \mathbb{R} \times A$ indiquant qu'en s, à la date t, il faut attendre la date t' pour entreprendre a.

Une question qui se pose immédiatement concerne les différences et l'éventuelle équivalence des modèles SMDP+ et TMDP, ainsi que l'équivalence des politiques SMDP+ et des politiques TMDP. Le premier point sera discuté à la section suivante ; le second nécessite de définir plus précisément le critère d'optimisation pour pouvoir comparer des politiques optimales et sera prouvé à la section 1.4.2.

1.2.4 Equivalence des modèles SMDP+ et TMDP

Tels qu'ils ont été présentés, il est quasi-immédiat de remarquer l'équivalence entre les modèles SMDP+ et TMDP. Cela apparait d'autant plus nettement en écrivant que tout TMDP est un SMDP+ que l'on écrit :

$$P(s'_\mu|s,a,t) = L(\mu|s,a,t) \tag{1.7}$$
$$F(t'|s,a,t,s'_\mu) = P_\mu(t') \tag{1.8}$$
$$R(s,t,a,s'_\mu,t') = R(\mu,t,t') \tag{1.9}$$

On illustre ainsi l'équivalence des deux formulations et le fait que la formulation TMDP repose sur la factorisation des transitions par les actions, plus spécifiquement : une transition est consituée d'abord par le choix d'une action, suivie par l'occurrence d'une réalisation prise parmi les réalisations atteignables, qui elle-même mène enfin vers un état résultant de la transition (comme illustré à la figure 1.3).

La notation TMDP sera conservée par la suite pour sa simplicité et pour l'intérêt que représente la variable T_μ. En effet, l'usage de T_μ (possible en TMDP comme en SMDP+) permet de spécifier des durées de transitions dépendant de la nature de l'agent (l'action "un pas en avant" prend une durée de 3 secondes) mais aussi des événements exogènes (l'action "prendre le métro" se termine à la date 9h10, indépendante de la date de début d'action, à la condition

qu'on ait pu effectivement monter dans le métro).

1.3 La particularité de la variable temporelle

Les exemples et les modèles présentés précédemment mettent en évidence le fait que des problèmes que l'on sait traiter dans un cadre stationnaire présentent de nouvelles difficultés dans un cadre instationnaire. Le traitement d'un temps continu, variable observable par l'agent, sort un peu des cadres classiques dans lesquels on pourrait chercher à le ranger. Est-ce une ressource ? Si oui, alors elle n'est pas bornée, ou alors c'est qu'on travaille à horizon fini. Mais alors qu'est-ce que l'horizon ? S'agit-il d'une date de fin de mission ou d'un nombre d'actions que l'agent peut entreprendre ? Le temps est-il une variable d'état ? Alors on doit pouvoir mettre en évidence l'effet des actions dessus. Toutes ces questions proviennent en partie d'une imprécision d'appellation qui a été mise en évidence par les derniers développements de la thèse et dont l'explication constitue l'ouverture aux modèles plus généraux présentés à la section 4. On va tout d'abord s'attacher à cerner cette variable temporelle en fonction de laquelle on veut planifier. Puis on s'interrogera sur les valeurs qu'elle peut prendre et sur la notion d'horizon. Enfin, cette discussion permettra de définir sous quelle forme on cherche le plan-solution de nos problèmes.

1.3.1 Trois sens différents pour une même variable ?

Des techniques de résolution de MDP à variables continues existent dans la littérature. Cependant, peu d'entre elles, à notre connaissance, s'attaquent à la variable temporelle continue quand elle n'est pas à horizon borné. De fait, le temps est une variable quelque peu déroutante car c'est la seule qui peut potentiellement affecter notre critère (via l'exposant du γ dans un critère γ-pondéré), mais qui reste une variable d'état puisqu'elle constitue une variable interne, observable par l'agent, nécessaire à la prise de décision, et enfin qui constitue une variable non contrôlable qui ne fait que croître lors de l'exécution.

En fait, il faut bien séparer les différents "temps" que l'on considère :

— On considère le temps de la chaîne de Markov, celui qu'on a pris soin de noter δ à la section 1.2.1. Ce temps est discret, il représente la succession des instants de décision. En ces différents instants de décision, les variables d'état, continues ou discrètes prennent des valeurs données lors de l'exécution.

— La variable d'état temporelle t. Cette variable d'état est une variable continue mais non bornée (car on souhaite planifier à horizon infini), on ne la considère donc pas comme une ressource mais bien comme une variable représentative de l'état de l'agent.

— Enfin, il y a le temps de l'action "attendre" qui n'est ni une variable d'état, ni le temps de la chaîne et qui correspond, en fait, au paramètre de l'action continue "attendre".

La variable temporelle couple ainsi des aspects non contrôlables (le temps de la chaîne) et des aspects contrôlables (l'état du système). On peut retrouver cette caractéristique dans une moindre mesure sur d'autres variables d'action continues comme la position dans le cas d'actions de déplacement, mais sans affecter la dynamique de la chaîne. Cette discussion sera poursuivie à la section 4 où l'on étendra le travail effectué sur la planification en fonction d'un temps continu explicite à la définition d'espaces d'actions continus.

1.3.2 Notion d'horizon, de pseudo-horizon

Le second aspect qui rend la variable temporelle particulière par rapport à une autre variable continue réside dans le fait que son domaine de définition est potentiellement non borné.

En effet, lorsqu'on entreprend de construire un modèle de décision dans l'incertain en fonction d'un temps explicite et continu, la première question qui se pose est celle de l'horizon de planification et de l'horizon temporel. Il est important de faire la différence entre la succession des instants de décision qui, en fait, correspond au nombre d'actions entreprises (au temps de la chaîne de Markov lors de l'exécution), et la variable "date courante", continue, observable et croissant indépendemment de l'action de l'agent. A horizon de planification fini, c'est-à-dire en cherchant une séquence de N actions consécutives, il peut être acceptable de considérer un temps discrétisé suffisamment finement pour correctement représenter notre problème. Cependant, on souhaite conserver au système la possibilité d'entreprendre un nombre d'actions non borné et on cherche donc des solutions à horizon infini, d'autant que l'incertitude introduite à l'origine dans le modèle MDP induit la possibilité de cycles et donc qu'on se sait pas nécessairement le nombre de pas nécessaires pour parvenir au but.

On s'intéresse donc à la modélisation de notre problème à horizon de planification infini. La connaissance de l'instationnarité du problème s'étend jusqu'à une date dans l'avenir que l'on note T et que l'on appelle pseudo-horizon temporel. Au delà, le problème est considéré stationnaire. On note immédiatement que dans le cadre d'une planification en ligne ou d'un apprentissage du modèle, ce pseudo-horizon est glissant et c'est principalement dans cette optique (dans l'optique, par exemple, d'un besoin de réparation de la politique courante, ou d'extension) que l'on considère des problèmes à horizon de planification infini et à pseudo-horizon connu.

Partant de ce constat, on peut considérer alors que planifier en fonction du temps revient à planifier dans un cadre entièrement stationnaire avec une ressource "temps restant" qui vaut T dans l'état initial et spécifier ainsi les modèles de transition et de récompense en fonction de cette ressource. On peut alors mettre en oeuvre des méthodes approchées de résolution de MDP à espace d'état continu comme ceux présentés dans [YS04], [MBB$^+$05] ou [GHK04].

Cependant, on souhaite conserver à la variable temporelle sa place à part et mettre en évidence ses spécificités dans l'algorithme de planification. La principale difficulté qui se pose alors vis-à-vis d'un problème à variables continues réside dans la situation particulière de l'action "attendre". Il est parfois préférable, dans une stratégie donnée, de ne rien faire à un moment pour gagner plus, plus tard. Il y a donc un intérêt à disposer d'une action "attendre", cependant, sa définition pose problème. En effet, toute action définie dans un MDP est décrite – via le modèle de transition – par ses effets sur les variables du problème. Par exemple, l'action "prendre la route A" est décrite par les distributions de probabilité sur son effet sur l'état du système en fonction de l'état de départ (le changement de position sur la carte). Pour l'action "attendre", on ne peut pas écrire de fonction de transition car il manque un paramètre : la durée de l'attente. On verra en section 4 comment cette réflexion peut s'étendre à d'autres variables continues. Les deux méthodes de résolution proposées pour le modèle SMDP+ / TMDP proposent une solution à cet écueil de modélisation de nos problèmes instationnaires à temps continu.

1.4 Méthodes de résolution

A présent que l'on a présenté le type de problèmes auxquels on s'intéresse et introduit les différentes approches de modélisation, on va s'attacher à proposer des méthodes permettant de générer des plans, optimaux vis-à-vis d'un critère, qui permettent d'atteindre le but que la planification s'est fixé.

On propose dans les sections suivantes trois approches différentes. Dans un premier temps, on cherche à effectuer une résolution par programmation dynamique sur le problème TMDP (section 1.4.1). De cette résolution, on tire une preuve d'équivalence (section 1.4.2) des politiques SMDP+ (introduites en section 1.2.2) et TMDP (section 1.2.3, où la question avait été introduite). Puis on propose un cadre de résolution par programmation linéaire (section 1.4.3) et on présente enfin une méthode de discrétisation du temps dans le cadre SMDP+ qui permet un traitement par programmation dynamique (section 1.4.4).

1.4.1 Programmation dynamique

On cherche à résoudre un problème posé sous forme de TMDP par programmation dynamique. On cherche donc à optimiser une politique via une équation de Bellman traduisant un critère d'optimalité. L'idée est de trouver de façon itérative la fonction de valeur optimale. On étend l'équation de Bellman au modèle TMDP selon les équations suivantes :

$$V(s,t) = \sup_{t' \geq t} \left(\int_t^{t'} K(s,\theta) d\theta + \overline{V}(s,t') \right) \tag{1.10}$$

$$\overline{V}(s,t) = \max_{a \in A} Q(s,t,a) \tag{1.11}$$

$$Q(s,t,a) = \sum_{\mu \in M} L(\mu|s,t,a) \cdot U(\mu,t) \tag{1.12}$$

$$U(\mu,t) = \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{si } T_\mu = \text{ABS} \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{si } T_\mu = \text{REL} \end{cases} \tag{1.13}$$

La première équation indique que la valeur en s à t correspond au maximum de gain que l'on peut espérer obtenir en attendant jusqu'à t' puis en agissant. En effet, d'après les trois équations suivantes, $\overline{V}(s,t)$ représente le maximum de la valeur que l'on peut espérer obtenir en agissant immédiatement à t. Ainsi la résolution par programmation dynamique d'un TMDP alterne une phase d'optimisation d'un processus où on agit immédiatement et un calcul de la durée optimale de l'attente avant d'agir. On obtient alors une politique $\pi(s,t) = (t',a)$ qui indique qu'en s, à t, l'action à entreprendre est "attendre jusqu'à t' puis entreprendre a" (on note qu'on peut avoir $t' = t$). Cette approche traduit en fait une optimisation selon le critère total où la récompense obtenue à chaque coup s'écrit :

$$r_\delta^\pi = \int_{t_\delta}^{t'_\pi} K(s_\delta,\theta) d\theta + R(s_\delta, t_\delta, t'_p i) \tag{1.14}$$

Afin d'étendre le modèle à des cas plus réalistes, on peut définir une fonction $W : S \times \mathbb{R}^2 \to S$ décrivant la dynamique d'état lors des phases d'attente. L'état $s' = W(s,t,t')$ représente l'état dans lequel on se trouve lorsqu'on a attendu, en s, à t et jusqu'à t'. L'équation 1.10 se réécrit alors :

$$V(s,t) = \sup_{t' \geq t} \left(\int_t^{t'} K(W(s,t,\theta),\theta) d\theta + \overline{V}(W(s,t,t'),t') \right) \tag{1.15}$$

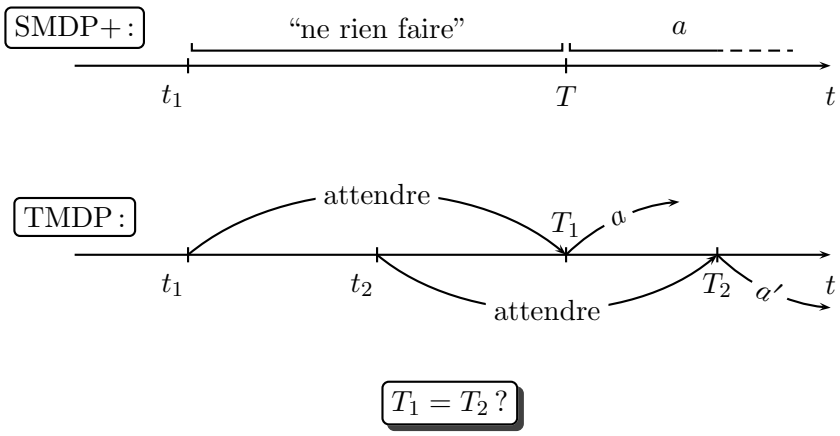

FIGURE 1.4 – Équivalence des politiques SMDP+ et TMDP

L'étude complète de ce cadre de résolution est présentée à la section 2, notamment les améliorations que nous avons apportées à la méthode proposée par [BL01], les différentes preuves d'existence de solutions, ainsi que la discussion sur les cas de résolution exacte ou approchée.

1.4.2 Equivalence des politiques SMDP+ et TMDP

A présent qu'on dispose d'une caractérisation des politiques TMDP, on peut s'attacher à répondre à la question posée en section 1.2.3 : a-t-on équivalence entre les politiques définies dans le cadre SMDP+ et TMDP ? Une politique SMDP+ est donnée sous la forme “à chaque instant il existe une action optimale à entreprendre, cette action peut éventuellement être l'action “ne rien faire” ” ; en permanence l'agent se réfère à sa politique pour savoir ce qu'il doit faire (éventuellement rien) et dès qu'il a fini son action en cours, il recommence. Une politique TMDP, au contraire, définit en (s,t) des actions “attendre jusqu'à t'” pendant lesquelles l'agent n'agit pas et ne remet pas en cause son action de “ne rien faire”, puis entreprend une action. Le problème de l'équivalence de ces deux descriptions de la politique revient à montrer que quel que soit l'instant t'' entre t et t', l'action de la politique SMDP+ est bien “ne rien faire”, ou encore que quelle que soit la date t'' que l'on considère entre t et t', la politique TMDP trouvée stipule qu'en (s,t'') il faut “attendre jusqu'à t' puis agir” (le t' et l'action étant les mêmes que pour la politique considérée en t). On va s'attacher à prouver ce second point.

Pour clarifier les choses, prenons un exemple illustré figure 1.4 : supposons que l'on se trouve dans l'état s à la date t_1. La politique SMDP+ nous dit que l'action à entreprendre est l'action “ne rien faire”. Un observateur qui anticipe la politique remarque également que l'on commence à effectuer l'action a à la date T (entre temps l'action à effectuer est toujours “ne rien faire”). Par ailleurs, la politique TMDP nous dit que l'action à entreprendre est “attendre la date T_1 puis effectuer a”. En écrivant l'équivalence des modèles (équations 1.7 à 1.9) puis l'équation de Bellman pour le critère total (équation 1.6), on trouve que $T = T_1$. La principale question pour montrer l'équivalence entre les politiques SMDP+ et TMDP est de savoir si, en prenant une date t_2 entre t_1 et T_1, la politique TMDP (T_2, a') en t_2 est bien cohérente avec la politique SMDP+, c'est-à-dire si $T_2 = T_1$ et $a' = a$.

L'équivalence des actions à entreprendre en t_2 est immédiate grace à l'équation 1.11 : que

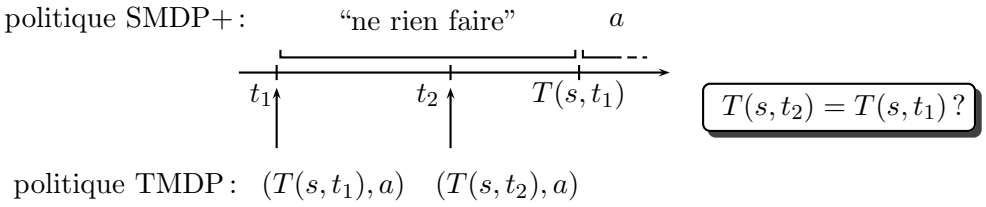

FIGURE 1.5 – Le problème de l'équivalence formalisé

l'on considère la politique en t_1 ou en t_2, si la date T_2 est bien la même que T_1, alors l'action à entreprendre sera l'action spécifiée par la fonction $\overline{V}(s,T_1)$. Il faut donc prouver que la date que l'on attend pour agir est bien la même, que l'on se situe à t_1 ou à t_2.

Pour cela on introduit la fonction $T(s,t)$:

$$T(s,t) : \left\{ \begin{array}{rcl} S \times \mathbb{R} & \rightarrow & \mathbb{R} \\ (s,t) & \mapsto & \underset{t' \geq t}{argsup} \left\{ \int_t^{t'} K(s,\theta)d\theta + \overline{V}(s,t') \right\} \end{array} \right. \tag{1.16}$$

On cherche alors à résoudre le problème :

Problème. *Soient un état s et une date t_1 telle que $T(s,t_1) > t_1$.*
Soit une date $t_2 \in \mathbb{R}$ telle que $t_2 \in [t, T(s,t_1)]$.
A-t-on $T(s,t_2) = T(s,t_1)$?

Ce problème est illustré à la figure 1.5, il est équivalent au problème de la figure 1.4.

Preuve. On a $\int_{t_1}^{t'} K(s,\theta)d\theta + \overline{V}(s,t') = \int_{t_1}^{t_2} K(s,\theta)d\theta + \int_{t_2}^{t'} K(s,\theta)d\theta + \overline{V}(s,t')$.

Or $T(s,t_1) > t_2$ donc $\underset{t' \geq t_1}{argsup} \left\{ \int_{t_1}^{t'} K(s,\theta)d\theta + \overline{V}(s,t') \right\} = \underset{t' \geq t_2}{argsup} \left\{ \int_{t_1}^{t'} K(s,\theta)d\theta + \overline{V}(s,t') \right\}$.

Donc :

$$\begin{array}{rcl} T(s,t_1) & = & \underset{t' \geq t_2}{argsup} \left\{ \int_{t_1}^{t'} K(s,\theta)d\theta + \overline{V}(s,t') \right\} \\ & = & \underset{t' \geq t_2}{argsup} \left\{ \int_{t_1}^{t_2} K(s,\theta)d\theta + \int_{t_2}^{t'} K(s,\theta)d\theta + \overline{V}(s,t') \right\} \end{array}$$

Or $\int_{t_1}^{t_2} K(s,\theta)d\theta$ est constante par rapport à t', elle n'affecte donc pas l'$argsup$. Et donc :

$T(s,t_1) = \underset{t' \geq t_2}{argsup} \left\{ \int_{t_2}^{t'} K(s,\theta)d\theta + \overline{V}(s,t') \right\}$.

Soit : $T(s,t_1) = T(s,t_2)$. □

On prouve ainsi que deux politiques optimales (selon le même critère) TMDP et SMDP+ sont équivalentes et que les deux formulations se valent. Afin d'étendre encore cette équivalence,

il faudrait effectuer cette preuve avec la fonction de dynamique d'état $W(s', t, t')$ mais ce cas plus général est pris en compte dans la discussion globale de la section 4.

1.4.3 Programmation linéaire

Une alternative à la résolution par programmation dynamique se situe dans les approches par programmation linéaire. On peut montrer [Put94], [HK04] qu'optimiser un MDP standard avec critère γ-pondéré se ramène à résoudre le problème linéaire :

$$min \sum_{s} V(s)$$
$$\forall s \in S, \forall a \in A, \quad V(s) - \gamma \cdot \sum_{s' \in S} P(s'|s,a)V(s') - r(s,a) \geq 0$$

Cette approche est particulièrement intéressante dans le cadre de l'approximation de la fonction de valeur d'un MDP factorisé. Dans la littérature, les travaux de [HK06, GHK04, FDMW04] utilisent la programmation linéaire approchée (ALP) pour résoudre facilement des problèmes de grande taille à variables d'état continues et discrètes. Etendre ces travaux au cadre temporel fait partie des objectifs futurs de la thèse.

1.4.4 Discrétisation par optimisation de l'erreur de Bellman

L'idée qui motive cette approche de résolution se formule de la façon suivante : on a vu au 1.4.2 que les formulation des politiques SMDP+ et TMDP étaient équivalentes, la politique optimale π^* présente donc des intervalles temporels sur lesquels l'action à entreprendre immédiatement est la même (éventuellement cette action peut être de ne rien faire). Peut-on alors chercher une discrétisation de la droite temporelle en intervalles distincts, puis résoudre notre problème dans un cadre discret, l'espace d'états étant alors constitué des états discrets auxquels on adjoint une variable "intervalle courant" prenant ses valeurs (en nombre fini) parmi les intervalles qu'on aura spécifiés ? La discrétisation a priori en une grille arbitraire entraîne une erreur qui peut être grossière sur la résolution du problème tout en augmentant inutilement la taille de l'espace d'état. La solution que l'on propose est de chercher les *dates de décision* optimales pour chaque état et de définir une variable d'intervalle dont on apprend les valeurs au fur et à mesure de l'exécution. Cet apprentissage se fait en évaluant l'erreur de Bellman temporelle pour chaque état (l'erreur de Bellman temporelle sera définie un peu plus loin). On définit ainsi exactement le nombre de points de discrétisation qui nous sont nécessaires et, par un processus de programmation dynamique, on cherche à faire converger en même temps ces points de discrétisation vers les bornes des intervalles de définition de la politique optimale et la politique courante vers la politique optimale.

Le fonctionnement général de l'algorithme suit le schéma présenté figure 1.6. Initialement, on considère une variable d'intervalles temporels discrète que l'on note $\tilde{T}$ et qui n'a qu'une valeur : l'intervalle $[0; +\infty[$ (la date 0 désignant la date de début d'exécution). On construit un modèle discret décrit par des fonctions $\tilde{P}$ et $\tilde{R}$ déduites du modèle continu fourni en entrée du système et on résout le MDP $\tilde{M}$ dont l'espace d'états correspond à l'espace d'état discret du système augmenté de la variable $\tilde{T}$. Puis on prolonge la politique $\tilde{\pi}$ obtenue sur la variable temporelle continue et on cherche, par état s, la date $\tilde{t}_s$ où l'erreur de Bellman (la quantité dont on peut améliorer la politique courante en optimisant sur un coup) est la plus grande selon le modèle continu. On augmente alors $\tilde{T}$ en insérant les $\tilde{t}_s$ dans les intervalles déjà définis et en fusionnant

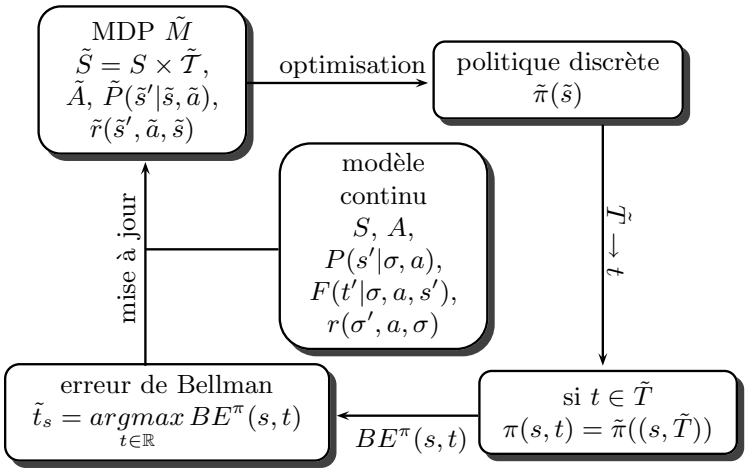

FIGURE 1.6 – Amélioration itérative de la politique

les intervalles consécutifs sur lesquels est définie la même action. On prolonge $\tilde{\pi}$ sur ce nouveau $\tilde{T}$ et on met à jour les fonctions $\tilde{P}$ et $\tilde{R}$. On recommence alors le processus : optimisation de $\tilde{M}$, définition de $\tilde{\pi}$ sur la variable t continue, recherche de la date où l'erreur de Bellman est la plus grande, etc.

Le détail du fonctionnement de cet algorithme a été introduit dans [RGT+06] et est présenté section 3.

On note que, parce que cet algorithme traite des variables d'état plutôt que des états énumérés, il se prête bien à un traitement sous forme factorisée ([BDG99]). Par ailleurs, son fonctionnement est "anytime" : on dispose à tout moment d'une politique dont la valeur s'accroît avec le temps.

La faiblesse de cet algorithme réside dans la difficulté d'évaluer la fonction de valeur continue de π. Cependant, c'est l'écueil des modèles trop génériques : en rajoutant des hypothèses sur la forme des fonctions de t on peut faciliter cette évaluation. Par ailleurs cette évaluation est surtout coûteuse à la première itération, la fonction de valeur de l'itération précédente peut être mémorisée et prolongée pour servir de base pour la recherche de l'erreur de Bellman maximale à l'itération suivante.

L'action "attendre" est introduite dans la résolution en la définissant uniquement dans le modèle discret qui fait passer d'une valeur de $\tilde{T}$ à la suivante. On peut raffiner le modèle en spécifiant une dynamique d'état pendant l'action "attendre". Cette dynamique est alors actualisée en même temps que le MDP $\tilde{M}$. On peut ainsi obtenir des politiques qui préconisent d'attendre dans un intervalle de temps donné, puis d'agir.

1.5 Description TMDP / SMDP+ de problèmes typiques : caractéristiques communes

A présent que l'on dispose des principes généraux des méthodes de résolution, on peut s'intéresser à caractériser les différentes fonctions utilisées dans les modèles (comme P_μ, par

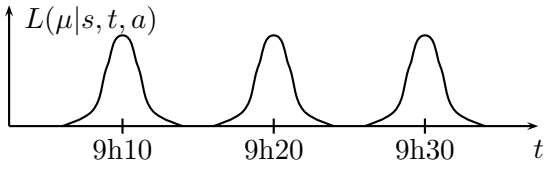

FIGURE 1.7 – Exemple de fonction $L(\mu|s,t,a)$

exemple) afin de spécialiser ces derniers pour améliorer l'efficacité des méthodes de résolution.

Afin de simplifier les notations, on se placera dans le cadre TMDP.

Considérons en premier la fonction $L(\mu|s,t,a)$. Cette fonction de t décrit la probabilité de réaliser μ sachant s et a en fonction du temps. Dans le cas des exemples présentés en section 1.1 on remarque que cette fonction est généralement régulière, définie par morceaux, la figure 1.7 illustre par exemple la probabilité de déclencher la réalisation "en route par le train vers l'ONERA" en fonction de la date courante, les trains passant approximativement toutes les 10 minutes entre 8h00 et 8h30. On note qu'une approximation sur la base de fonctions polynômiales définies par morceaux permet une représentation pertinente. [BL01] propose une modélisation via des fonctions constantes par morceaux, nous verrons à la section 2 comment nous étendons la résolution à toute fonction polynomiale par morceaux et nous ouvrirons sur une représentation sur d'autres bases de fonctions que des fonctions polynomiales.

La fonction P_μ décrit, elle la densité de probabilité de la durée d'une transition ou de sa date de fin (selon la valeur du paramètre T_μ). Dans [BL01], cette densité est nécessairement représentée comme une somme de fonctions de Dirac (distribution discrète). Nous avons souhaité pouvoir décrire des phénomènes dont la durée est incertaine et où l'incertitude porte sur un paramètre de durée continu. En effet, la durée du trajet en bus entre la maison et l'ONERA est décrite par un continuum de valeurs possibles, chacune associée à une densité de probabilité. De même, la prise de vue du satellite peut être gênée par une couverture nuageuse dont la durée d'effet est décrite par une densité de probabilité sur une variable continue. On a à notre disposition plusieurs types de distributions classiques pour décrire des densités de probabilité sur des variables continues, en fonction des problèmes, on peut souhaiter utiliser des distributions Gaussiennes, Bêta, Exponentielle, etc. Cependant il est important de considérer la question de savoir comment on va traiter ces distributions dans nos algorithmes. Notamment, il est important de voir comment on va calculer la convolution de P_μ avec R dans l'équation de programmation dynamique 1.13. Nous avons pris le parti d'approcher toute distribution continue par une distribution polynômiale définie par morceaux. Nous verrons au 2.1 comment cette orientation se justifie. Pour l'instant nous notons simplement que c'est un choix de modélisation peu contraignant et qu'il n'exclut pas l'extension éventuelle aux distributions classiques dans le futur.

La fonction $R(\mu,t,t')$ décrit la récompense qu'on obtient en réalisant μ entre t et t'. On peut assez souvent découpler les récompenses associées au début d'une transition de celles obtenues à la fin ou encore de celles dépendant de la durée. Nous prenons donc le parti, comme [BL01], de décomposer la fonction R en une somme de trois fonction :

$$R(\mu,t,t') = r_t(\mu,t) + r_{t'}(\mu,t') + r_\tau(\mu,t'-t) \tag{1.17}$$

Cette décomposition effectuée, on peut s'intéresser à la nature des fonctions r ainsi définies. Il apparait que pour r_t ou $r_{t'}$, les récompemses sont souvent des fonctions du temps constantes par morceaux. Cependant, pour définir la récompense associée à une arrivée à la maison le soir quand le match de football est déjà commencé, on peut avoir besoin d'utiliser une fonction linéaire de t', voire, si on considère que les premières minutes sont plus importantes, d'une fonction présentant plus de variations. Nous verrons en section 2 que l'hypothèse de [BL01] d'avoir des r linéaires par morceaux est extensible à toute fonction polynômiale par morceaux. Le même raisonnement est fait pour les fonctions r_τ.

Il reste enfin la dernière fonction continue du temps du modèle TMDP : $K(s, \theta)$. Cette fonction définit le coût instantané associé à l'action "ne rien faire" dans l'état s à la date t. La principale utilisation de K réside dans une fonction de coût de consommation (de carburant par exemple). On aura donc souvent des K constants par morceaux. Il est important de noter toutefois que dans la méthode que l'on introduira à la section 2, K peut être n'importe quelle fonction polynômiale définie par morceaux. Pour simplifier l'écriture, on la considérera constante par morceaux dans la suite.

Il y a une dernière fonction qui dépend du temps dans nos modèles, il s'agit de la fonction de dynamique d'état $W(s, t, t')$. Cependant cette fonction est en fait une fonction discrète : sur certains intervalles elle prendra la valeur discrète s'_1, sur d'autres, la valeur s'_2. Sa spécification n'est donc pas sujette à ambiguité.

1.6 Conclusion sur la spécialisation du modèle choisi et les approches de résolution

Dans cette première partie de modélisation, on s'est intéressés à des problèmes types de planification où l'agent doit se coordonner avec son environnement en fonction d'une variable temporelle explicite et observable (1.1). Une description des problèmes rencontrés a permis de mettre en évidence un manque dans les modèles classiques pour ce type de problème (1.2.1, 1.2.1 et 1.2.1). On a alors proposé un nouveau modèle (SMDP+, 1.2.2) et l'amélioration d'un modèle existant (TMDP, 1.2.3) afin de prendre en compte les particularités de la variable temporelle. On a démontré l'équivalence des deux formulations introduites (1.2.4). On s'est alors attachés, avant d'aller plus loin, à bien comprendre ce qui distingue le temps d'une autre variable continue (1.3). Cela a permis de définir des méthodes de résolution (1.4) inspirées des méthodes classiques et d'en proposer une spécifique (1.4.4). Enfin, sur la base de ces méthodes, on est revenu vers le cas pratique afin de définir plus précisément la forme des fonctions avec lesquelles on va travailler et pour vérifier que nos méthodes sont bien adaptées à une résolution d'un cas pratique (1.5).

Dans la dernière section, on a beaucoup insisté sur la modélisation par polynômes définis par morceaux. Cette approche — justifiée mathématiquement à la section 2 — nous permet d'évaluer et d'approcher, à moindre frais calculatoires, toute fonction continue par morceaux de $\mathbb{R}$ et nous assure donc une faisabilité de modélisation. Un aspect qui a été assez peu abordé concerne les intégrations possibles des techniques "classiques" MDP dans nos méthodes de résolution (décomposition, factorisation, hiérarchisation, approches heuristiques, ...). Il semble qu'il n'y ait pas de difficultés majeures à les utiliser ou les adapter, cependant ce travail relève encore des perspectives.

Enfin, pour conclure sur cette partie, il est intéressant de se replacer dans le cadre biagent (ou multiagent) de la thèse et de voir l'intérêt de disposer d'algorithmes de coordination en fonction d'une référence temporelle commune entre deux agents. En effet, nos deux agents devant coopérer en situation dangereuse (l'incendie par exemple) vont devoir coordonner leurs actions avec l'environnement dynamique et entre eux, on développera donc les détails des méthodes de résolution dans les sections suivantes en gardant à l'esprit que ces algorithmes sont destinés à être inclus dans un cadre coopératif biagent (ou multiagent) et qu'ils devront donc être prévus pour des contraintes de fonctionnement "en ligne" comme des possibles réinitialisations, des modifications du problème de départ, un aspect "anytime", etc.

Chapitre 2

Résolution d'un TMDP / SMDP+ par programmation dynamique

Dans ce chapitre, on choisit de se placer dans le cadre d'écritude TMDP et on cherche à trouver une politique optimale vis-à-vis du critère total par programmation dynamique. Pour cela, on étudie dans un premier temps la forme de l'équation de Bellman (2.1) pour déterminer dans quelle mesure on va pouvoir faire une résolution formelle et quand ce n'est pas possible, quelles hypothèses adopter afin de limiter l'erreur commise lors de la résolution. Puis on présentera la méthode de résolution de façon générale (2.2). Enfin, on détaillera cette méthode dans le cas d'une résolution exacte (2.3) et d'une résolution approchée (2.4).

2.1 Forme générale de la fonction de valeur, stabilité de certaines classes de fonctions par l'opérateur de Bellman

La question au centre de cette section est la suivante : on cherche une fonction de valeur $V(s,t)$ qui obéisse à l'équation de Bellman précédemment définie. Or les espaces de fonctions sont généralement difficiles à approcher car de dimension infinie. On cherche donc une forme de V qui soit aisément manipulable. L'approche par programmation dynamique utilise le fait que l'opérateur de Bellman est contractant et admet un point fixe. Pour effectuer une résolution exacte, il serait donc pratique d'avoir une fonction de valeur qui soit de forme stable par cette opérateur. On peut donc reformuler notre préoccupation : on cherche une écriture de V qui soit stable par L (opérateur de Bellman) soit, si $V \in \mathcal{C}$ (avec $\mathcal{C}$ une classe de fonctions) alors $LV \in \mathcal{C}$.

Cette recherche d'une classe de fonctions $\mathcal{C}$ stable par L doit cependant se faire en gardant à l'esprit le caractère utilisable de cette dernière. Par ailleurs, cette classe de fonctions sera sûrement liée aux classes des fonctions L, P_μ, K et R, il faut donc conserver leur sens physique et leur utilisatibilité pour la modélisation de notre problème.

Voyons ce que $V = LV$ implique : nous allons prendre les équations 1.10 à 1.13 et étudier les propriétés de V lorsqu'on lui applique ces équations.

Considérons l'équation 1.10 :

$$V(s,t) = \sup_{t' \geq t} \left(\int_t^{t'} K(s,\theta) d\theta + \overline{V}(s,t') \right)$$

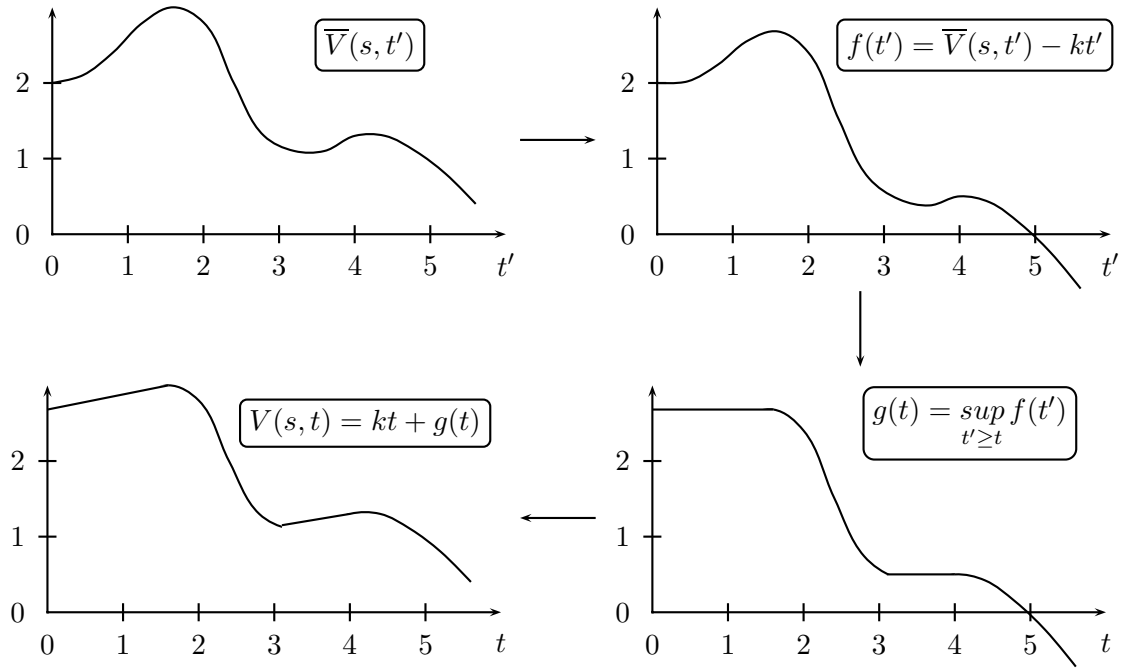

FIGURE 2.1 – Illustration de l'équation 1.10

D'après nos hypothèses, $K(s,\theta)$ est une fonction constante par morceaux donc $\int_t^{t'} K(s,\theta)d\theta$ est une fonction linéaire de t et de t'.

Prenons un exemple simple et observons quelle forme a $V(s,t)$ si on suppose connue $\overline{V}(s,t')$ et si on prend $K(s,\theta) = -k$. On a $V(s,t) = kt + \sup_{t' \geq t} \left(-kt' + \overline{V}(s,t')\right)$. La date $T(s,t)$ (équation 1.16) correspond donc à : $V(s,t) = \underset{t' \geq t}{argsup} \left(kt' + \overline{V}(s,t')\right)$. La figure 2.1 illustre alors comment on trouve $V(s,t)$ (on a pris un $\overline{V}$ quleconque et $k = 0.1$). On calcule d'abord $f(t')$, puis $g(t)$ et enfin $V(s,t)$.

On peut s'intéresser aux variations de $V(s,t)$ en fonction de t. Prenons deux instants t_1 et t_2 avec $t_1 < t_2$ et comparons $V(s,t_1)$ et $V(s,t_2)$. On distingue deux cas :

Premier cas : $T(s,t_1) \geq t_2$. On a vu à la section 1.4.2 que dans ce cas $T(s,t_1) = T(s,t_2)$. On a alors :

$$\begin{aligned} V(s,t_1) &= \sup_{t' \geq t_1} \left(-k(t'-t_1) + \overline{V}(s,t')\right) \\ &= \sup_{t' \geq t_1} \left(-k(t'-t_2) + \overline{V}(s,t')\right) - k(t_2 - t_1) \\ &= \sup_{t' \geq t_2} \left(-k(t'-t_2) + \overline{V}(s,t')\right) - k(t_2 - t_1) \\ &= V(s,t_2) - k(t_2 - t_1) \end{aligned}$$

$$\text{et donc}: \frac{V(s,t_2)-V(s,t_1)}{t_2-t_1} = k \tag{2.1}$$

La fonction $V(s,t)$ est donc croissante de pente k. Cela se comprend physiquement de la façon suivante : considérons un état du système dans lequel il faille attendre pour accéder à la récompense, l'esprérance du gain considérée à t_1 sera inférieure à celle considérée à t_2 car si la récompense est la même, le coût de l'attente est plus grand dans le premier cas. Cette situation est ilustrée par les segments croissants de la représentation de V à la figure 2.1.

Second cas : $T(s,t_1) < t_2$. Il y a donc une action à entreprendre en $T(s,t_1)$ et le problème en t_2 est totalement différent puisqu'on ne peut pas envisager d'entreprendre d'action dans le passé. On sait donc que :

$$\begin{aligned}
V(s,t_1) &= \sup_{t'\geq t_1}\left(-k(t'-t_1)+\overline{V}(s,t')\right)\\
&= \sup_{t'\in[t_1,t_2]}\left(-k(t'-t_1)+\overline{V}(s,t')\right) \text{ et donc :}\\
V(s,t_1) &\geq \sup_{t'\geq t_2}\left(-k(t'-t_1)+\overline{V}(s,t')\right)\\
&\geq \sup_{t'\geq t_2}\left(-k(t'-t_2)+\overline{V}(s,t')\right)-k(t_2-t_1)\\
&\geq V(s,t_2)-k(t_2-t_1)
\end{aligned}$$

$$\text{et donc}: \frac{V(s,t_2)-V(s,t_1)}{t_2-t_1} \leq k \tag{2.2}$$

Ce résultat nous garantit qu'il n'existe bien aucune autre durée d'attente qui nous permette de gagner plus, compte tenu du coût de l'attente. Dans le cas où t_1 et t_2 sont écartés d'une distance infinitésimale, ce résultat traduit le fait que $\overline{V}$ ne croit pas suffisamment avec t' pour compenser la perte due à k, il vaut mieux alors agir immédiatement plutôt que d'attendre. Ce sont donc les cas où $t' = t$ et ceux-ci correspondent sur la figure 2.1 aux zones où $V(s,t) = \overline{V}(s,t)$.

Au final, la pente de notre espérance de gain V en fonction du temps est bornée par l'opposé du taux de coût instantané : ce n'est pas un constat pessimiste, au contraire, c'est la preuve qu'on ne peut plus améliorer notre fonction de valeur.

Cette analyse des variations et de la forme de $V(s,t)$ — en plus de nous donner une idée de l'algorithme de résolution qui sera présenté plus loin — nous permet de tirer la conclusion suivante : sur certains intervalles, V est linéaire par morceaux (on conserve l'hypothèse d'un $K(s,\theta)$ constant par morceaux, au besoin on remplacera celle-ci par une hypothèse polynômiale par morceaux), sur les autres, elle est de la même classe de $\overline{V}$, que l'on note $\mathcal{D}$. En notant $\mathcal{P}_n$ l'ensemble des fonctions polynômiales de degré n définies par morceaux et à valeurs dans $\mathbb{R}$, on note que :

$$\mathcal{P}_1 \subset \mathcal{C} \text{ et } \mathcal{D} \subset \mathcal{C}$$

Continuons à présent à parcourir notre équation de Bellman et intéressons-nous à l'équation 1.13 :

$$U(\mu,t) = \begin{cases} \displaystyle\int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t')+V(s'_\mu,t')]dt' & \text{si } T_\mu = \text{ABS} \\ \displaystyle\int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t')+V(s'_\mu,t')]dt' & \text{si } T_\mu = \text{REL} \end{cases}$$

En utilisant la décomposition introduite à l'équation 1.17, on a :

$$U(\mu,t) = \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[r_t(\mu,t) + r_{t'}(\mu,t') + r_\tau(\mu,t'-t) + V(s'_\mu,t')]dt' & \text{si } T_\mu = \text{ABS} \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[r_t(\mu,t) + r_{t'}(\mu,t') + r_\tau(\mu,t'-t) + V(s'_\mu,t')]dt' & \text{si } T_\mu = \text{REL} \end{cases}$$

Et en posant $S_\mu(t') = P_\mu(-t')$ et en développant, on a :

$$U(\mu,t) = \begin{cases} \left(\int_{-\infty}^{\infty} P_\mu(t')dt'\right) r_t(\mu,t) + \int_{-\infty}^{\infty} P_\mu(t')r_{t'}(\mu,t')dt' + (S_\mu \otimes r_\tau(\mu,\cdot))(-t) + \\ \qquad\qquad \int_{-\infty}^{\infty} P_\mu(t')V(s'_\mu,t')dt' & \text{si } T_\mu = \text{ABS} \\ \left(\int_{-\infty}^{\infty} P_\mu(t'-t)d(t'-t)\right) r_t(\mu,t) + (S_\mu \otimes r_{t'}(\mu,\cdot))(t) + \\ \qquad\qquad \int_{-\infty}^{\infty} P_\mu(t'-t)r_\tau(\mu,t'-t)d(t'-t) + (S_\mu \otimes V(s_\mu,\cdot))(t) & \text{si } T_\mu = \text{REL} \end{cases}$$

Prenons le premier cas (T_μ =ABS) avec les hypothèses de modélisation introduites au 1.5 : le résultat est de la forme "$\mathcal{P}_n$ + constante + $\mathcal{E}(t)$ + constante".

La classe de fonctions $\mathcal{E}$ dépend de la forme de S_μ. Si P_μ est polynômiale alors $\mathcal{E} = \mathcal{P}_n$ (la valeur de n étant discutée plus loin), c'est le cas que nous utiliserons le plus par la suite. Si P_μ est gaussienne, exponentielle ou Bêta, le cas est traité dans les paragraphes qui suivent et le résultat à retenir est que la méthode par programmation dynamique n'est pas adaptée à la résolution dans le cas où à la fois P_μ est définie de façon implicite (cas des distributions gausienne et Bêta) et r ou V définies par morceaux. Enfin, le cas où P_μ est une distribution discrète sera abordé à la fin de cette section, c'est la base du cas où l'on peut effectuer une résolution exacte.

Les deux premiers termes du résultat ci-dessus impliquent l'existence des moments de P_μ calculés sur les intervalles de définition des fonctions r et, d'un point de vue pratique, leur calculabilité (ce qui exclut les distributions Gaussiennes de nos possiblités). Comme on le montre en annexe au A.2.1 de ce document, le calcul des convolutions dans le cas de fonctions r définies par morceaux impliquent une perte de la régularité du résultat qui rendent difficile le calcul de U si P_μ n'est pas dans $\mathcal{P}_n$.

Dans le second cas (T_μ = REL) avec les hypothèses de modélisation introduites au 1.5 : le résultat est de la forme "$\mathcal{P}_n$ + constante + $\mathcal{E}(t)$ + $S_\mu \otimes V$".

Les premiers termes nous permettent de tirer les mêmes conclusions que pour le cas T_μ = ABS. Le calcul du dernier terme est problématique. On sait que V est définie par morceaux

dans $\mathcal{C}$ mais on n'a pas de résultat sur sa stabilité par la convolution avec un polynôme défini par morceaux (ou une autre forme quelconque de P_μ). Ici, on décide de restreindre l'ensemble $\mathcal{C}$ selon les contraintes que l'on a en pratique afin de trouver une famille de fonctions stable par la convolution précédente et facilement manipulable. On note que chercher V sous forme polynômiale par morceaux peut être intéressant car alors toutes les contraintes évoquées jusqu'à présent ($\mathcal{P}_1 \subset \mathcal{C}$), stabilité par convolution avec P_μ, etc.) sont vérifiées. Etant donné qu'une telle représentation semble pertinente et facilement manipulable, on cherchera par la suite V dans $\mathcal{P}_n$.

On peut alors s'intéresser au degré de ce polynôme et à son calcul via l'équation de Bellman. Pour cela, on précise nos hypothèses. On définit $\mathcal{DP}_n$ l'ensemble des distributions à densité polynômiale par morceaux de degré inférieur à n et on écrit :

— $P_\mu \in \mathcal{DP}_a$
— $r_i \in \mathcal{P}_b$
— $L \in \mathcal{P}_c$

On prolonge l'espace $\mathcal{DP}_n$ pour $n = -1$ en écrivant que les éléments de $\mathcal{DP}_{-1}$ sont les distributions discrètes (cette prolongation est justifiée par la prolongation des propriétés de conservation du degré des convolutions entre polynômes). On note alors d le degré de V et on cherche à déterminer d en fonction de a, b et c quand on passe V "au travers" de l'opérateur de Bellman.

En notant $d^\circ()$ l'opérateur donnant le degré d'un polynôme, on a, par l'équation 1.13 :
— cas ABS : $d^\circ(U) = a + b + 1$ (on laisse le lecteur vérifier que $d^\circ(S_\mu \otimes r_\tau) = a + b + 1$)
— cas REL : $d^\circ(U) = max\{a + b + 1, a + d + 1\}$

On se place dans l'optique d'un algorithme type "itération de la valeur" que l'on initialise avec une fonction V de degré zéro. Cette commodité de notation nous permet d'écrire que $\forall \mu \in M, \quad U(\mu, \cdot) \in \mathcal{P}_{a+b+1}$. Il est aisé de reprendre les calculs en conservant l'opérateur max mais cela n'apporte rien au raisonnement.

Prenons à présent l'équation 1.12 :

$$Q(s, t, a) = \sum_{\mu \in M} L(\mu|s, t, a) \cdot U(\mu, t)$$

On obtient immédiatement que $\forall (s, a) \in S \times A, Q(s, \cdot, a) \in \mathcal{P}_{a+b+c+1}$.

L'équation 1.11 ne change pas le degré du polynôme (elle consitue un polynôme en aggrégeant des morceaux de polynômes de même degré). Donc $d^\circ(\overline{V}) = a + b + c + 1$.

Enfin, on a vu que l'équation 1.10 nous permettait de finir l'équation sur d et nous donne (en reprenant l'opérateur max qu'on a laissé par commodité plus tôt) :

$$d = max\{a + b + c + 1, a + d + c + 1\}$$

Ainsi, on peut tirer la conclusion suivante : si, initialement, d vaut zéro, alors, après une passe de programmation dynamique, d vaut $a + b + c + 1$, après deux itérations, $2a + b + 2c + 2$, etc. On en conclut que le degré du polynôme V explose, sauf si $a + c = -1$. Le seul cas possible de réalisation de cette condition est trouvé pour $a = -1$ et $c = 0$. Ce cas correspond à une situation où :

— P_μ est une distribution discrète
— L est une fonction constante par morceaux
— les r_i sont des polynômes de degré b quelconque.
Dans ce cas précis, on sait alors que V est de degré b.

A présent qu'on dispose d'un résultat sur la stabilité de V par l'opérateur de Bellman, on va chercher à savoir s'il est possible d'effectuer une résolution exacte. Dans le cas où le degré du polynôme augmente sans fin, on ne peut pas parler de résolution exacte puisque notre algorithme ne converge pas vers un unique polynôme. Une résolution exacte se fait donc dans le cadre $a + c = -1$. Il s'agit donc de déterminer les valeurs de b qui permettent ou non un calcul exact de coefficients du polynôme V défini par morceaux.

Quel que soit b, l'équation 1.13 se résout facilement car on sait calculer les coefficients de U sur ses différents morceaux sans approximation (voir annexe A.2.3). La même remarque s'applique à l'équation 1.12. Par contre, l'équation 1.11 implique de trouver, à s fixé, les points d'intersection des $|A|$ courbes $(Q(s,t,a))_{a\in A}$. Cela revient à trouver les intersections de polynômes de degré b et donc à trouver les racines d'un polynôme de degré b. On sait faire cette opération sans approximation dans les cas suivants :
— $b = 0$, cas trivial
— $b = 1$, intersection de droites
— $b = 2$, formule du binôme
— $b = 3$, fomule de Cardan ou de Sotta
— $b = 4$, Formule de Ferrari ou de Descartes

A partir de $b = 5$, Galois a prouvé que l'on ne pouvait pas trouver de méthode générale de résolution des racines réelles d'un polynôme. Une technique d'approximation intéressante qui sera abordée et utilisée plus tard est la méthode de Sturm (annexe B).

L'équation 1.10, enfin, impose de trouver les maximas de fonctions polynômiales de degré b. Si on a $b < 5$ on sait trouver les racines d'un polynôme de degré b et donc les extremas d'un polynôme de degré $b+1$ avec exactitude. La contrainte limitante est donc celle due à l'équation 1.11. On en déduit que :

La résolution exacte ne peut se faire que dans le cadre :
$$a = -1$$
$$b \in \{0, 1, 2, 3, 4\}$$
$$c = 0$$

On verra à la section suivante comment effectuer cette résolution exacte ainsi qu'une méthode de résolution dans le cas approché. On remarque que notre conclusion est plus générale que la conclusion formulée dans [BL01], en effet, nous avons montré les limites exactes d'une résolution dans le cadre polynomial par morceaux. Cette étude nous permet de bien cerner les difficultés associées à la résolution dans le cas approché et de proposer la méthode détaillée au 2.4

2.2 Méthode générale

La méthode générale de résolution d'un TMDP par programmation dynamique repose sur la construction de la suite de fonctions $(V_n(s,t))_{n\in\mathbb{N}}$ telle que :

$$
\begin{aligned}
V_{n+1}(s,t) &= \sup_{t' \geq t} \left(\int_t^{t'} K(s,\theta) d\theta + \overline{V}_n(s,t') \right) && (2.3)\\
\overline{V}_n(s,t) &= \max_{a \in A} Q_n(s,t,a) && (2.4)\\
Q_n(s,t,a) &= \sum_{\mu \in M} L(\mu|s,t,a) \cdot U_n(\mu,t) && (2.5)\\
U_n(\mu,t) &= \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t') + V_n(s'_\mu,t')]dt' & \text{si } T_\mu = \text{ABS} \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t') + V_(s'_\mu,t')]dt' & \text{si } T_\mu = \text{REL} \end{cases} && (2.6)
\end{aligned}
$$

Dans la section 2.3 on détaille la résolution dans le cas exact spécifié précédemment et on propose une méthode de résolution approchée pour le cas général à la section 2.4.

2.3 Résolution exacte d'un TMDP / SMDP+

On se place dans les hypothèses nécessaires à une résolution exacte vues au 2.1 et on a :
— P_μ une distribution discrète
— L une fonction constante par morceaux
— les $(r_i)_{i \in \{t,t',\tau\}}$ sont des polynômes de degré inférieur ou égal à 4.

Plus précisément on écrit que :
— Dans le cas T_μ =ABS, $P_\mu(t') = \sum_{i=1}^{P} P_{a_i} \cdot \delta_{a_i}(t)$ avec δ_{a_i} la fonction de Dirac en a_i. On a donc $S_\mu(t') = \sum_{i=1}^{P} P_{a_i} \cdot \delta_{-a_i}(t')$
— Dans le cas T_μ =REL, $P_\mu(t'-t) = \sum_{i=1}^{P} P_{d_i} \cdot \delta_{d_i}(t'-t)$. Et on a donc $S_\mu(t'-t) = \sum_{i=1}^{P} P_{d_i} \cdot \delta_{-d_i}(t'-t)$.
— $r_\tau(\mu,\tau) = \sum_{j=0}^{B} b_t \tau^j$

On prend alors les équations dans l'ordre. Pour l'équation 2.6, cas ABS :

$$
\begin{aligned}
U_n(\mu,t) &= \int_{-\infty}^{\infty} P_\mu(t') \left[r_t(\mu,t) + r_{t'}(\mu,t') + r_\tau(\mu,t'-t) + V_n(s'_\mu,t') \right] dt' \\
&= r_t(\mu,t) \left(\int_{-\infty}^{\infty} P_\mu(t') dt' \right) + \int_{-\infty}^{\infty} P_\mu(t') r_{t'}(t') dt' + \int_{-\infty}^{\infty} P_\mu(t') r_\tau(t'-t) dt' + \\
&\qquad \int_{-\infty}^{\infty} P_\mu(t') V_n(,s'_\mu,t') dt' \\
&= r_t(\mu,t) + (r_{t'} \otimes S_\mu)(0) + (r_\tau \otimes S_\mu)(-t) + (V_n \otimes S_\mu)(0) \\
&= r_t(\mu,t) + \sum_{i=1}^{P} P_{a_i} \left(r_{t'}(\mu,a_i) + r_\tau(\mu,a_i - t) + V_n(s'_\mu,a_i) \right)
\end{aligned}
$$

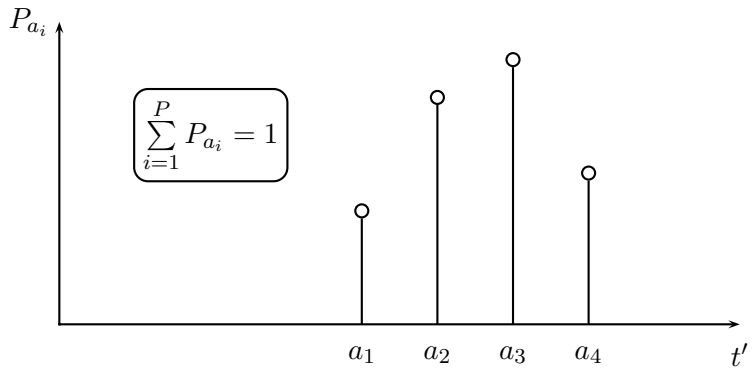

FIGURE 2.2 – Exemple de distribution discrète

On étudie séparément les quatre termes du membre de droite de cette équation. Le premier terme est un polynôme de degré B. Le second terme est constant. Le quatrième terme est constant. Le troisième terme se calcule de la façon suivante :

$$\begin{aligned}
r_\tau(\mu, a_i - t) &= \sum_{j=0}^{B} b_j (a_i - t)^j \\
&= \sum_{j=0}^{B} b_j \sum_{k=0}^{j} C_j^k {a_i}^{j-k} (-t)^k \\
&= t^B \left[b_B C_B^B (-1)^B \right] + \\
&\quad t^{B-1}(-1)^{B-1} \left[b_B C_B^{B-1} a_i + b_{B-1} C_{B-1}^{B-1} \right] \\
&\quad t^{B-2}(-1)^{B-2} \left[b_B C_B^{B-2} {a_i}^2 + b_{B-1} C_{B-1}^{B-2} a_i + b_{B-1} C_{B-2}^{B-2} \right] \\
&\quad \vdots \\
&\quad t^l (-1)^l \left[\sum_{k=l}^{B} b_k C_k^l {a_i}^{k-l} \right] \\
&\quad \vdots
\end{aligned}$$

Et donc :

$$\begin{aligned}
\sum_{i=1}^{P} P_{a_i} r_\tau(\mu, a_i - t) &= \sum_{i=1}^{P} P_{a_i} \sum_{l=0}^{B} t^l \left[(-1)^l \sum_{k=l}^{B} b_k C_k^l {a_i}^{k-l} \right] && (2.7) \\
&= \sum_{l=0}^{B} t^l \left[\sum_{i=1}^{P} P_{a_i} (-1)^l \sum_{k=l}^{B} b_k C_k^l {a_i}^{k-l} \right] && (2.8) \\
& && (2.9)
\end{aligned}$$

On a donc bien un polynôme de degré B dont on connait les coefficients. Pour l'équation 2.6, cas REL :

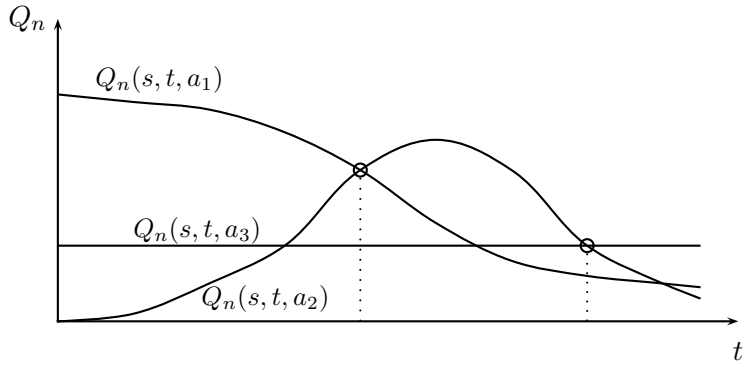

FIGURE 2.3 – Illustration de la recherche de $\overline{V}$

$$
\begin{aligned}
U_n(\mu,t) &= \int_{-\infty}^{\infty} P_\mu(t'-t)\left[r_t(\mu,t)+r_{t'}(\mu,t')+r_\tau(\mu,t'-t)+V_n(s'_\mu,t')\right]dt' \\
&= r_t(\mu,t)\left(\int_{-\infty}^{\infty} P_\mu(t'-t)dt'\right)+\int_{-\infty}^{\infty} P_\mu(t'-t)r_{t'}(t')dt'+\int_{-\infty}^{\infty} P_\mu(t'-t)r_\tau(t'-t)dt'+ \\
&\qquad \int_{-\infty}^{\infty} P_\mu(t'-t)V_n(,s'_\mu,t')dt' \\
&= r_t(\mu,t)+(r_{t'}\otimes S_\mu)(t)+(r_\tau\otimes S_\mu)(0)+(V_n\otimes S_\mu)(t) \\
&= r_t(\mu,t)+\sum_{i=1}^{P} P_{d_i}\left(r_{t'}(\mu,t+d_i)+r_\tau(\mu,d_i)+V_n(s'_\mu,t+d_i)\right)
\end{aligned}
$$

Le premier terme est un polynôme de degré B connu et le troisième terme est constant. Les second et quatrième termes s'écrivent en remplaçant les $(b_i)\, i \in \mathbb{N}$ par les coefficients de r_τ ou V_n (le calcul est similaire au cas précédent) :

$$\sum_{l=0}^{B} t^l \left[\sum_{i=1}^{P} P_{d_i} \sum_{k=l}^{B} b_k C_k^l {d_i}^{k-l}\right]$$

On sait donc calculer exactement les coefficients du polynôme $U_n(\mu,t)$ de degré B. Ce calcul a été effectué dans le cas d'une définition en un seul morceau des fonctions r_i, cependant, pour le cas général de fonctions définies par morceaux, on translate les intervalles de définition avec les a_i ou les d_i pour trouver les intervalles de définition de U_n et le calcul est presque le même.

Prenons à présent l'équation 2.5. Dans un premier temps, on trouve tous les intervalles de définition de Q_n. On pose δ_i les extrémités des intervalles de définition de L, et γ_i celles des intervalles de définition de U_n. On classe les δ_i et γ_i par ordre croissant et sur chacun des intervalles ainsi définis, comme L est constante, le polynôme Q_n est obtenu en multipliant simplement les coefficients de U_n par la valeur de L. On obtient ainsi les polynômes définis par morceaux $Q_n(s,t,a)$ de degré B.

L'équation 2.4 consiste à rechercher les intersections des polynômes. On fixe s et on prend $a_m = \underset{a}{argmax}\, Q_n(s,0,a)$. On va trouver le max pour tout t itérativement comme présenté à

l'algorithme 1 et sur la figure 2.3. On cherche la première intersection de $Q_n(s,t,a_m)$ avec la fonction Q_n d'une autre action, on prend l'intersection d'abscisse la plus petite. Cette intersection est racine du polynôme $Q_n(s,t,a_m) - Q_n(s,t,a)$ qui est au plus de degré B et on a $B < 4$, on peut donc trouver cette racine exactement. On vérifie qu'il s'agit bien d'une intersection et pas juste d'un point tangent en vérifiant le changement de signe autour de l'intersection. On redéfinit a_m, on stocke le Q_n courant dans $\overline{V}_n$ sur l'intervalle qu'on a trouvé et on recommence. L'algorithme s'arrête quand il n'y a plus d'intersection. On obtient ainsi un polynôme $\overline{V}_n(s,t)$ défini par morceaux et de degré B.

Algorithme 1 : Algorithme de construction de $\overline{V}$

$a_m \leftarrow \underset{a}{argmax}\, Q_n(s,0,a)$
$\overline{V}_n(s,t) \leftarrow Q_n(s,t,a_m)$
$t_0 \leftarrow 0$
$t_{\text{intersec}} \leftarrow 0$
tant que $t_{intersec} \neq \infty$ **faire**
  pour $a \in A \setminus \{a_m\}$ **faire**
    $t_{\text{intersec}} \leftarrow \infty$
    $t_{\text{new}} \leftarrow$ première racine de $Q_n(s,t,a_m) - Q_n(s,t,a)$ dans $[t_0, t_{\text{intersec}}]$
    (s'il n'y a pas de racines dans $[t_0, t_{\text{intersec}}]$, $t_{\text{new}} \leftarrow \infty$)
    si $t_{new} < t_{intersec}$ *et* $Q_n(s,t,a_m) - Q_n(s,t,a)$ *change de signe en* t_{new} **alors**
      $a_n \leftarrow a$
      $t_{\text{intersec}} \leftarrow t_{\text{new}}$
  $\overline{V}_n(s,t)_{|[t_0,t_{\text{intersec}}]} \leftarrow Q_n(s,t,a_m)$
  $a_m \leftarrow a_n$
  $t_0 \leftarrow t_{\text{intersec}}$

Enfin, dans l'équation 2.3 on cherche à connaître le *sup* de la fonction paramétrique :

$$f_t(t') = \int_t^{t'} K(s,\theta)d\theta + \overline{V}_n(s,t')$$

K est constante par morceaux donc $\displaystyle\int_t^{t'} K(s,\theta)d\theta$ est de la forme $K_1(t' - \alpha_1) + K_2(\alpha_1 - \alpha_2) + \ldots + K_n(\alpha_{p-1} - t)$. Donc :

$$\frac{df_t(t')}{dt'} = K(s,t') + \frac{\overline{V}_n(s,t')}{dt'}$$

Et donc annuler $\frac{df_t(t')}{dt'}$ revient à annuler des polynômes de degré $B-1$. On cherche donc les racines de ces polynômes sur chaque intervalle défini par les domaines de définition de K et $\overline{V}_n$. Comme la fonction peut être discontinue on ajoute à l'ensemble des abscisses ainsi trouvées les extrémités des intervalles de définition de K et $\overline{V}_n$. Parmi ces abscisses, on cherche le t' qui permet de maximiser $f_t(t')$. On trouve ainsi une fonction de la forme $K_i(\alpha_i - t) + \overline{V}_n(s,t)$ ou $\overline{V}_n(s,t)$. $V_n(s,t)$ est donc une fonction polynômiale par morceaux de degré au plus égal à B.

On dispose ainsi d'une méthode complète, utilisant les propriétés des polynômes de degré inférieur à 4 et distributions discrètes, permettant une résolution exacte de nos problèmes de décision dans l'incertain en fonction d'un temps explicite. Une bibliothèque d'opérations

mathématiques dédiées aux opérations décrites plus haut est en cours d'implémentation. Elle est destinée à être intégrée dans la boucle de programmation dynamique définie au 2.2.

Dans le cas général, cependant, il est très contraignant de devoir considérer des probabilités de transition dépendant du temps comme des fonctions constantes par morceaux et il est assez limitant de ne considérer que des distributions discrètes sur la durée des transitions. On souhaite donc pouvoir résoudre notre problème de programmation dynamique dans le cas général où :

$$\left.\begin{array}{r} P_\mu \in \mathcal{DP}_a \\ r_i \in \mathcal{P}_b \\ L \in \mathcal{P}_c \end{array}\right\} \text{ avec } a, b \text{ et } c \text{ quelconques.}$$

On peut remarquer immédiatement que la méthode exacte s'adapte facilement au cas "$a+c = -1$ mais $b \geq 5$". En effet, la méthode de Sturm nous donne une méthode simple pour trouver de façon approchée (avec la précision que l'on souhaite) les racines de n'importe quel polynôme. Comme on a $a + c = -1$, le degré de V reste bien stable au travers des itérations, la seule approximation étant faite sur les valeurs des bornes des intervalles de définition. Le réel problème de cas général vient du retrait de l'hypothèse $a+c = -1$. On propose une méthode de résolution approchée de ce problème à la section 2.4.

2.4 Résolution approchée d'un TMDP / SMDP+

La résolution approchée permet de couvrir tous les cas où $a + c \neq -1$ (2.1).On reste donc dans une approche "polynômiale par morceaux", cette approche semblant pertinente car on sait facilement approcher n'importe quelle fonction — même discontinue — par un polynôme défini par morceaux.

L'idée de la résolution approchée est de rajouter une étape simplificatrice entre deux itérations de programmation dynamique : on a vu que si V_n était de degré b, alors V_{n+1} est de degré $a + b + c + 1$. On veut se ramener à une fonction de degré b ou inférieur. L'idée est, sur chaque morceau de $V_{n+1}(s, t)$, d'approximer la fonction par un polynôme de degré b ou inférieur. Si l'erreur maximale commise lors de cette approximation est supérieure à un certain seuil, on coupe l'intervalle que l'on considère là où l'erreur est la plus grande et on effectue notre approximation de degré b ou inférieur sur les deux intervalles distincts résultants. On effectue ainsi une approximation de V par splines ([JHA67]), ce qui nous permet, à chaque itération, de ramener le degré du polynôme à un niveau plus bas. On obtient ainsi une approximation de V à ϵ près avec des polynômes de degré réduit moyennant un plus grand nombre de morceaux.

La question se pose alors de choisir intelligemment le degré de notre approximation. Il est alors intéressant de prendre en compte les remarques suivantes :

- Si $a + b + c + 1 \leq 4$, il est intéressant de choisir une approximation de degré inférieur ou égal à b pour pouvoir bénéficier des algorithmes exacts de récherche des racines des polynômes.
- Une approximation linéaire permet de relier les points en respectant leurs valeurs, une approximation cubique permet de respecter les valeurs des points et de leurs dérivées. Ce type d'approximation (B-splines) est d'usage assez répandu, notamment en imagerie.
- D'un point de vue intuitif on peut considérer que la fonction de valeur optimale va avoir une forme "ressemblant" aux fonctions de récompense. On peut donc supposer qu'une

approximation d'ordre supérieur à b sera plus fine mais n'apportera que peu d'information supplémentaire tout en alourdissant le processus de résolution par rapport à une approximation d'ordre b. Tandis qu'une approximation d'ordre inférieur à b risquera de manquer des aspects importants du problème. Par ailleurs, une approximation d'ordre b peut permettre de limiter le nombre de morceaux de la fonction V au final sans introduire de calculs superflus.

Ces considérations introduites, on peut s'intéresser à la résolution proprement dite.

Equation 2.6, cas ABS :

$$U_n(\mu,t) = r_t(\mu,t) + (r_{t'} \otimes S_m u)(0) + (r_\tau \otimes S_\mu)(-t) + (V_n \otimes S_\mu)(0)$$

Le premier terme est un polynôme connu de degré b. Le troisième terme utilise la formule de calcul des coefficients vue à l'annexe A.2.3. Les second et quatirème terme utilisent la formule de calcul de $\int f(x)g(-x)dx$ que l'on explicite ici :

$$(r_{t'} \otimes S_\mu)(0) = \int_{-\infty}^{\infty} r_{t'}(\mu,t')S_\mu(-t')dt'$$

On pose : $r_{t'}(\mu,t') = \sum_{j=0}^{B} b_j t^j$ et $S_\mu(-t) = P_\mu(t) = \sum_{i=0}^{A} a_i t^i$. Soit alors $C(\mu,t')$ le polynôme de coefficients $\vec{a} \otimes \vec{b}$. Comme $\vec{a}$ et $\vec{b}$ sont les coefficients des polynômes ci-dessus, leur valeur change selon le point du domaine de définition, le polynôme $C(\mu,t')$ est donc défini par morceaux également. Et donc :

$$(r_{t'} \otimes S_\mu)(0) = \int_{-\infty}^{\infty} C(\mu,t')dt'$$

Equation 2.6, cas REL :

$$U_n(\mu,t) = r_t(\mu,t) + (r_{t'} \otimes S_\mu)(t) + (r_\tau \otimes S_\mu)(0) + (V_n \otimes S_\mu)(t)$$

Le premier terme est un polynôme connu. Les second et quatrième termes sont calculés comme indiqué à l'annexe A.2.3. Le troisième terme, enfin, est calculé comme présenté au paragraphe précédent. On obtient donc les coefficients d'un polynôme de degré $a+b+1$ défini par morceaux.

L'équation 2.5 est résolue comme dans le cas exact : dans un premier temps on trouve les intervalles de définition de Q_n puis, sur chaque intervalle, on effectue la convolution des vecteurs de coefficients de L et U_n. On obtient ainsi les coefficients de Q_n, polynôme de degré $a+b+c+1$.

Pour l'équation 2.4, on adapte la résolution à la valeur de $a+b+c+1$:

— Si $a+b+c+1 \leq 4$, on procède comme dans le cas exact (voir 2.3 et B)
— Si $a+b+c+1 \geq 5$, on trouve la première intersection par dichotomie en utilisant la méthode de Sturm (voir annexe B)([Stu35]). On trouve ainsi les intervalles de définition

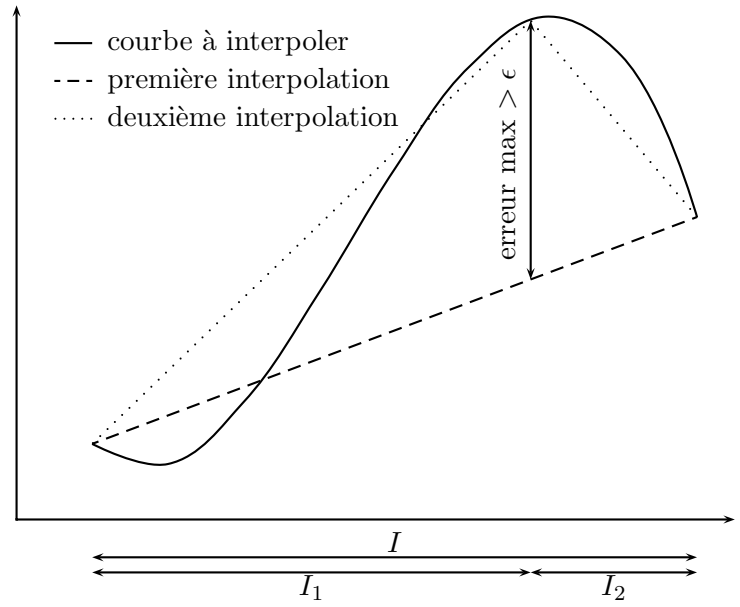

FIGURE 2.4 – Illustration de la réduction de degré du polynôme final

de $\overline{V}$ à un ϵ près.

Enfin, pour l'équation 2.3 :

— Si $a+b+c+1 \leq 5$, on sait résoudre exactement $\frac{d\overline{V}(s,t')}{dt'} + K(s,t') = 0$, on procède donc comme dans le cas d'une résolution exacte (2.3).

— Si $a+b+c+1 \geq 6$, on procède de nouveau par la méhode de Sturm ([Stu35]).

On obtient ainsi les coefficients d'un polynôme de degré $a+b+c+1$ défini par morceaux. On s'attache alors à en réduire le degré. La méthode est illustrée à la figure 2.4 et à l'algorithme 2. On note que la manière de choisir les points de réduction est arbitraire et que cette heuristique est sujette à amélioration.

Algorithme 2 : Algorithme d'approximation et de réduction du degré d'un polynôme

```
input : deg, le degré de l'approximation
input : P le polynôme à approximer
input : meth, méthode d'interpolation
pour I = intervalle de définition de P faire
    poly_approché ← interpolation(deg, P, I, meth)
    err ← sup_{t∈I}{|P(t) − poly_approché(t)|}
    tant que err ≥ ϵ faire
        t_err ← argsup_{t∈I}{|P(t) − poly_approché(t)|}
        Insérer t_err dans I
retourner poly_approché
```

On dispose donc d'une méthode complète permettant de calculer la fonction de valeur optimale V^* (pour le critère total) correspondant à notre problème TMDP / SMDP+. Cette

méthode est actuellement en cours d'implémentation dans un planificateur utilisant les algorithmes de la bibliothèque d'opérations sur les polynômes précédemment codée.

Chapitre 3

Recherche des dates de décision pour les TMDP / SMDP+ : une méthode de discrétisation par l'optimisation de l'erreur de Bellman

3.1 L'idée générale

On se place dans tout ce chapitre dans la formulation SMDP+. Comme présenté au 1.4.4, l'idée de base de la résolution par recherche des dates de décision optimales peut être formulée de la façon suivante : la politique que l'on recherche est de la forme "en s, quelque soit la date entre t_1 et t_2, la meilleure action à entreprendre est a_5, puis entre t_2 et t_3, il vaut mieux ne rien faire, ...". Partant de cette idée, on va chercher à déterminer les dates t_1, t_2, etc. par état et on va considérer les différents intervalles temporels ainsi définis comme autant de valeurs d'une variable d'état. On intégrera alors les fonctions du modèle continu sur chacun de ces intervalles et on pourra alors résoudre un problème discret que l'on suppose équivalent (ou suffisamment proche). Nous avons proposé et détaillé cette idée dans [RGT+06], nous présentons ici une version améliorée.

On risque de se retrouver confronté à de nombreuses valeurs pour les t_i, cependant cet aspect n'affecte que peu notre raisonnement, en effet, par état, on ne stocke que les dates où la décision optimale change. Imaginons qu'en un état s on définisse trois intervalles différents, alors, par rapport à un problème discret sans aucun aspect instationnaire, on ne stockera que trois états là où on en stockait un pour la spécification de la politique. On échappe donc au *curse of dimensionnality* de Bellman car on ne fait pas de produit cartésien des ensembles associés aux variables d'état. On verra plus clairement à la section 3.2 comment on met en oeuvre une telle approche factorisée.

Pour expliciter le fonctionnement de l'algorithme développé à partir de cette idée, on introduit les concepts suivants :

Définition (Critère γ-pondéré en SMDP+)**.** *On optimise notre politique vis-à-vis du critère γ-pondéré étendu au cadre SMDP. Ce critère s'exprime comme :*

$$V_\gamma^\pi = E\left(\sum_{\delta=0}^{\infty} \gamma^{t_\delta} r_\delta^\pi | s_0\right)$$

avec : $r_\delta^\pi = R((s_\delta, t_\delta, \pi(s_\delta, t_\delta)))$ *et chaque date de transition* t_δ *est décrite par la densité de probabilité* $F(t_{\delta+1}|s_\delta, t_\delta, \pi(s_\delta, t_\delta), s_{\delta+1})$.

L'équation de Bellman s'écrit alors (equation 1.6) :

$$V^\pi(\sigma) = \sum_{s' \in S} \int_0^\infty \left(r(s', t+\tau, \pi(\sigma), \sigma) + \gamma^\tau V^\pi(\sigma')\right) \cdot F(\tau|\sigma, \pi(\sigma), s') P(s'|\sigma, \pi(\sigma)) d\tau = L_\pi^t(V^\pi)(\sigma)$$

On rappelle qu'on cherche une politique $\pi(s,t)$ avec $s \in S$ et $t \in \mathbb{R}$.

Définition (Période de décision)**.** *Une* période de décision *est un intervalle temporel sur lequel, pour un état discret donné, l'action à entreprendre spécifiée par la politique est constante.*

Définition (Période de décision optimale)**.** *Une période de décision optimale est un intervalle temporel sur lequel, pour un état discret donné, l'action à entreprendre spécifiée par la politique en cet état est toujours l'action optimale.*

Définition (Ensemble $\tilde{\mathcal{T}}$)**.** *On note* $\tilde{\mathcal{T}}$ *l'ensemble des périodes de décision d'une politique. Les éléments de* $\tilde{\mathcal{T}}$ *sont notés* $\tilde{T}$ *et on notera* $\tilde{t}_i$ *et* $\tilde{t}_{i+1}$ *les bornes de* $\tilde{T}_i$.

Définition (Politique SMDP+)**.** *Une politique SMDP+* $\tilde{\pi}$ *est une application associant un intervalle de décision* $\tilde{T}_i$ *de* $\tilde{\mathcal{T}}$ *et un état discret* $s \in S$ *à une action* $a \in A$.

L'idée centrale de la résolution va être de peupler, dépeupler et corriger l'ensemble $\tilde{\mathcal{T}}$ sur lequel on définit nos politiques SMDP+ au fur et à mesure que l'on optimise les différentes politiques que l'on définit sur $\tilde{\mathcal{T}} \times S$.

Initialement, on met dans $\tilde{\mathcal{T}}$ l'unique intervalle $]-\infty; +\infty[$. En fait, on définit des ensembles $\tilde{\mathcal{T}}_s$ qui sont le résultat de la factorisation de l'espace d'état : dans $\tilde{\mathcal{T}}_s$, on ne stocke que les intervalles de décision utiles à la spécification de la politique $\tilde{\pi}$ en s. Cet aspect, bien qu'au coeur de l'efficacité de l'algorithme, sera occulté par la suite car il alourdit les notations. On se rappelera toutefois que quand on considère qu'on explore l'espace $\tilde{\mathcal{T}}$, on explore en fait un espace factorisé dans lequel, pour un unique s, il y a un nombre réduit de $\tilde{t}_i$ servant à définir les intervalles de $\tilde{\mathcal{T}}_s$ (cet aspect est abordé plus en détail dans [RGT+06]).

L'action "attendre" est alors facile à définir : pour conserver la similarité avec le modèle TMDP et le déterminisme de l'action "attendre" on considère que celle-ci associe un probabilité 1 à la transition qui amène de $s, \tilde{T}_i$ dans $W(s), \tilde{T}_{i+1}$. On peut alors définir les politiques SMDP+ dans l'espace d'actions $\tilde{A} = A \cup \{attendre\}$.

L'idée est donc de trouver la politique $\tilde{\pi}$ telle que son équivalent continu π soit ϵ-optimale.

3.2 La méthode

L'optimisation de $\tilde{\pi}$ suit les grandes étapes décrites ci-après et présentées à la figure 3.1.

Initiatialisation : génération du MDP $\tilde{M}$**.** On construit un MDP $\tilde{M}$ constitué de :

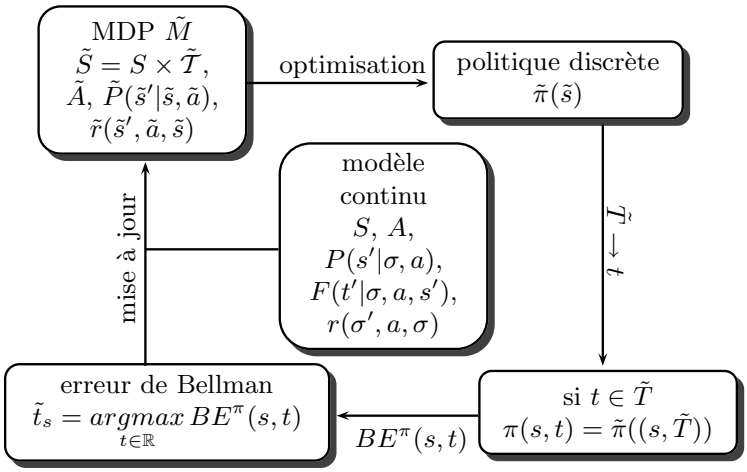

FIGURE 3.1 – Amélioration itérative $\tilde{\pi}$

— un espace d'états $\tilde{S} = S \times \tilde{\mathcal{T}}$
— une espace d'actions $\tilde{A} = A \cup \{attendre\}$
— une fonction de transition $\tilde{P}(\tilde{s}', \tilde{a}, \tilde{s})$
— un modèle de récompense $\tilde{r}(\tilde{s}', \tilde{a})$

La fonction $\tilde{P}((s', \tilde{T}'), \tilde{a}, (s, \tilde{T}))$ décrit la probabilité que l'action $\tilde{a}$, entreprise en s pendant la période de décision $\tilde{T}$ mène l'agent en s', pendant la période $\tilde{T}'$. De même, $\tilde{r}$ représente l'expérance des récompenses associées à la transition $(\tilde{s}', \tilde{a}, \tilde{s})$. On calcule ces deux grandeurs de la façon suivante :

$$\tilde{P}((s', \tilde{T}'), \tilde{a}, (s, \tilde{T}) = E^{t\in\tilde{T}}\left(E^{t'\in\tilde{T}'}\left(Q((t', s'|\tilde{a}, (s,t)))\right)\right)$$

Et pour $\tilde{r}$:

$$\tilde{r}(\tilde{s}', \tilde{a}, \tilde{s}) = E^{t\in\tilde{T}}\left(E^{t'\in\tilde{T}'}\left(r((s', t'), a, (s,t))\right)\right)$$

$$\tilde{r}(\tilde{a}, \tilde{s}) = E^{\tilde{T}'\in\tilde{\mathcal{T}}}\left(E^{s'\in S}\left(E^{t\in\tilde{T}}\left(E^{t'\in\tilde{T}'}\left(r((s', t'), a, (s,t))\right)\right)\right)\right)$$

On remarque que c'est ici que s'insère la fonction définissant le coût d'une attente (la fonction $K(s, \theta)$ du modèle TMDP, que l'on considère incluse dans r par abus de notation dans le modèle SMDP+), dans la définition de $\tilde{r}$.

Plus en détail on a :

$$\begin{aligned}\tilde{P}((s',\tilde{T}'_k),\tilde{a},(s,\tilde{T}_j)) &= E^{t\in\tilde{T}_j}\left(E^{t'\in\tilde{T}'_k}\left(P(s'|s,t,a)F(t'|s,t,a,s')\right)\right)\\ &= \frac{1}{t_{j+1}-t_j}\int_{\tilde{t}_j}^{\tilde{t}_{j+1}} P(s'|s,t,a)F(t'|s,t,a,s')dt\\ &= \frac{1}{t_{j+1}-t_j}\int_{\tilde{t}_j}^{\tilde{t}_{j+1}} P(s'|s,t,a)\int_{\tilde{t}_k}^{\tilde{t}_{k+1}} F(t'|s,t,a,s')dt'dt\end{aligned}$$

en notant G la fonction de répartition de F :

$$G(v|s,t,a,s') = Pr(t'<v|s,t,a,s') = \int_{-\infty}^{v} F(t'|s,t,a,s')dt'$$

$$\tilde{P}((s',\tilde{T}'_k),\tilde{a},(s,\tilde{T}_j)) = \frac{1}{t_{j+1}-t_j}\int_{\tilde{t}_j}^{\tilde{t}_{j+1}} P(s'|s,t,a)\left[G(\tilde{t}_{k+1}|s,t,a,s') - G(\tilde{t}_k|s,t,a,s')\right]dt \quad (3.1)$$

On peut remarquer que si on reprend l'hypothèse SMDP d'indépendance de t' et s', cette intégrale se simplifie en le produit de deux termes facilement calculables. Cependant, cette hypothèse étant très restrictive, on l'évitera pour l'instant.

Pour $\tilde{r}$ cela s'écrit :

$$\begin{aligned}\tilde{r}((s,\tilde{T}_j),\tilde{a}) &= E^{t\in\tilde{T}}\left(r(a,(s,t))\right)\\ &= \frac{1}{\tilde{t}_{j+1}-\tilde{t}_j}\int_{\tilde{t}_j}^{\tilde{t}_{j+1}} r(a,(s,t))dt \qquad (3.2)\end{aligned}$$

avec

$$\begin{aligned}r(a,s,t) &= E^{s'\in S}\left(E^{t'\in\tilde{T}'}\left(r((s',t'),a,(s,t))\right)\right)\\ &= \sum_{s'\in S}\int_{-\infty}^{\infty} P(s'|s,t,a)F(t'|s,t,a,s')r((s',t'),a,(s,t))dt'\end{aligned}$$

On définit ainsi complètement le MDP $\tilde{M}$.

Première étape : optimisation de $\tilde{M}$. On résout l'équation de Bellman avec un critère γ-pondéré pour $\tilde{M}$ par une méthode classique (itération de la valeur, de la politique, programmation linéaire exacte ou approchée, ...). On obtient ainsi une politique SMDP+ $\tilde{\pi}(s,\tilde{T})$ et la fonction de valeur associée $V^{\tilde{\pi}}$. On rappelle que pour calculer cette fonction de valeur, on n'a pas évalué $V^{\tilde{\pi}}$ en chaque élément de $\tilde{\mathcal{T}}$ en chaque état discret s mais uniquement dans les quelques $\tilde{T}_j$ pertinents pour s. On a donc rajouté $Card(\tilde{\mathcal{T}}_s)$ là où il n'y en avait qu'un (cela implique en fait une représentation de F et r un peu plus détaillée que celle qu'on a présentée ci-dessus où l'on sépare les $\tilde{T}$ pré- et post-action). On évite ainsi l'explosion combinatoire de l'espace d'états, limitant sa taille à $E^{s\in S}(Card(\tilde{\mathcal{T}}_s))\cdot Card(S)$. Sur un exemple simple présentant une récompense et deux échéances (une apparition et une disparition de la récompense), on s'aperçoit que $Card(\tilde{\mathcal{T}}_s)$ vaut au maximum 3 et que les trois valeurs sont utiles à la spécification de la politique, on utilise donc bien un espace d'états dont la taille est minimale au regard de l'expressivité de notre solution.

Deuxième étape : recherche des dates de décision optimales. L'idée de la troisième étape est de chercher, étant donné la politique courante considérée dans le cadre continu, les dates en lesquelles on pourrait le mieux améliorer cette politique. Pour cela on utilise l'*erreur de Bellman* en s.

On considère, pour tout $(s,t) \in S \times \mathbb{R}$, la fonction $V^\pi(s,t)$, evaluation de la prolongation π sur $S \times \mathbb{R}$ de la politique $\tilde{\pi}$ définie sur $S \times \tilde{T}$. On définit l'erreur de Bellman et la t-erreur de Bellman comme :

Définition (Erreur de Bellman)**.** *Soit une politique π définie sur un MDP classique $< S, A, P, r >$. L'erreur de Bellman représente la quantité dont on peut améliorer la fonction de valeur V^π en chaque état, en une passe de programmation dynamique. On l'écrit :*

$$BE(V^\pi(s)) = \max_{a \in A} \left(r(s,a) + \gamma \sum_{s' \in S} P(s,a,s') V^\pi(s') \right) - V^\pi(s) \tag{3.3}$$

Définition (t-erreur de Bellman)**.** *La t-erreur de Bellman représente la quantité dont on peut améliorer la valeur de la politique SMDP+ prolongée sur $\mathbb{R}$ en (s,t), en une passe de programmation dynamique. On l'écrit :*

$$\begin{aligned} BE_s(t) &= BE(V^\pi(s,t)) \\ &= \max_{a \in A} \left(r(s,a,t) + \sum_{s' \in S} \int_{-\infty}^{\infty} \gamma^{t'-t} V^\pi(s',t') F(t'|s,t,a,s') P(s'|s,a,t) dt' \right) - V^\pi(s,t) \end{aligned} \tag{3.4}$$

Comme dans le cas d'un MDP discret, on prend la t-erreur de Bellman comme une mesure de l'écart à l'optimum et on l'utilise comme une heuristique pour rechercher les dates de décision optimales. Pour chaque état s, on cherche les triplets (s, t_s, ϵ_s) où ϵ_s est le maximum de $BE_s(t)$ atteint en t_s. On a ainsi, par état s, l'instant où la décision courante est la plus "optimisable".

Pour pouvoir évaluer $BE_s(t)$, il faut disposer de V^π et donc l'évaluer. Cette évaluation passe par les étapes suivantes :

— Prolongation de la politique $\tilde{\pi}$ sur $S \times \mathbb{R}$: $\forall t \in \tilde{T}_j, \pi(s,t) = \tilde{\pi}(s, \tilde{T}_j)$.

— Evaluation de $V^\pi(s,t) = L^t_\pi(V^\pi)(s,t)$

$$V^\pi(s,t) = \sum_{s' \in S} \int_{-\infty}^{\infty} \left(r(\sigma', \pi(\sigma), \sigma) + \gamma^{t'-t} V^\pi(\sigma') \right) \cdot F(t'|\sigma, \pi(\sigma), s') P(s'|s,t,\pi(\sigma)) dt'$$

(on note $\sigma = (s,t)$ pour alléger la notation).

— Calcul de $BE_s(t)$ (equation 3.4).

— Maximisation de $BE_s(t)$.

La première étape est relativement simple. La difficulté se pose à la seconde. En effet, l'évaluation de la politique π est un processus de même complexité que son optimisation. Cependant, on a déjà une estimation de la valeur de V^π au travers de $V^{\tilde{\pi}}$. On peut, par ailleurs utiliser, comme cela a été fait pour la formulation TMDP, les propriétés des fonctions du modèle. On peut alors, par exemple, chercher V^π de façon approchée sous la forme d'un polynôme de degré fixe. Pour cela, on utilise les propriétés des polynômes vues en annexe ainsi que les propriétés des fonctions exp-poly introduites dans [RGT+06], en particulier les propriétés de projection sur l'espace des polynômes. On rappelle simplement :

Définition (Famille de fonctions exp-poly)**.** *Une application $f : \mathbb{R} \to \mathbb{R}$ est dite exp-poly de degré n et de coefficient α s'il existe une application $p \in \mathcal{P}_n$ (polynômiale par morceaux de degré n) et un nombre $\alpha \in \mathbb{R}$ tels que :*

$$\forall x \in \mathbb{R}, f(x) = e^{\alpha x} \cdot p(x) \tag{3.5}$$

En utilisant l'algorithme d'approximation vu à la fin de la section 2.4, illustré par l'algorithme 2 et la figure 2.4 et en procédant par itérations de la valeur avec comme initialisation la fonction continue par morceaux qui prolonge $V^{\tilde{\pi}}$ dans $\mathbb{R}$, on parvient à approcher V^{π}.

Enfin, pour la recherche des triplets (s, t_s, ϵ_s), qui constitue le problème clé de notre algorithme, on cherche à maximiser la t-erreur de Bellman. On pose l'opérateur de programmation dynamique L_a^t :

$$L_a^t(V^{\pi})(s,t) = r(s,a,t) + \sum_{s'} \int_{-\infty}^{\infty} \gamma^{(t'-t)} V^{\pi}(s',t') F(t'|s,t,a,s') P(s'|s,t,a) dt'$$

On a alors :

$$BE_s(t) = \max_{a \in A} \left\{ L_a^t(V^{\pi})(s,t) \right\} - V^{\pi}(s,t)$$

On cherche alors :

$$\begin{aligned} \max_{t \in \mathbb{R}} BE_s(t) &= \max_{t \in \mathbb{R}} \max_{a \in A} \left\{ L_a^t(V^{\pi})(s,t) - V^{\pi}(s,t) \right\} \\ &= \max_{a \in A} \max_{t \in \mathbb{R}} \left\{ L_a^t(V^{\pi})(s,t) - V^{\pi}(s,t) \right\} \end{aligned}$$

Pour un s donné, on va donc chercher le maximum de $L_a^t(V^{\pi})(s,t) - V^{\pi}(s,t)$ pour tous les a on prendra alors le maximum sur cette famille de maxima. Le problème se ramène donc à trouver le maximum sur t, à s et a fixés, de $L_a^t(V^{\pi})(s,t) - V^{\pi}(s,t)$. En supposant cette expression dérivable par morceaux et en se plaçant sur chaque morceau, on cherche donc à résoudre l'équation :

$$\frac{\partial \left(L_a^t(V^{\pi})(s,t) - V^{\pi}(s,t) \right)}{\partial t} = 0 \tag{3.6}$$

On dispose, avec le cadre polynômial précédemment défini, d'une expression littérale polynômiale pour $BE_s(t)$, la valeur du maximum se calcule alors aisément par une des méthodes présentées en annexe B. Une autre option que l'on peut envisager pour trouver ces maxima consiste à effectuer une résolution numérique, en utilisant des algorithmes évolutionnaires (type algorithmes d'essaim) qui recherchent directement le minimum de $BE_s(t)$ sans passer par sa dérivée.

On dispose donc, selon la représentation que l'on choisit, de plusieurs méthodes pour trouver les triplets (s, t_s, ϵ_s).

Troisième étape : peuplement de $\tilde{\mathcal{T}}$. Si ϵ_s est supérieur à un certain ϵ que l'on s'est fixé initialement, alors on effectue les opérations suivantes (en supposant que $\tilde{t}_j < t_s < \tilde{t}_{j+1}$) :

— On ajoute $[\tilde{t}_j, t_s]$ et $[t_s, \tilde{t}_{j+1}]$ à $\tilde{\mathcal{T}}_s$.
— On retire $[\tilde{t}_j, \tilde{t}_{j+1}]$ de $\tilde{\mathcal{T}}_s$.
— On prolonge $\tilde{\pi}$ sur le $S \times \tilde{\mathcal{T}}$ nouvellement défini.

Si tous les ϵ_s sont inférieurs à ϵ on s'arrête.

Quatrième étape : mise à jour de $\tilde{M}$. Pour les valeurs de $\tilde{\mathcal{T}}$ qui ont disparu ou qui sont apparues, on redéfinit $\tilde{P}$ et $\tilde{r}$ à partir des $\tilde{P}$ et $\tilde{r}$ de l'itération précédente en modifiant les valeurs à partir des valeurs déjà calculées et des équations 3.1 et 3.2.

On obtient ainsi un nouveau MDP $\tilde{M}$. On reprend alors à la première étape et on initialise la politique que l'on recherche avec la politique $\tilde{\pi}$ que l'on a adaptée au nouvel espace d'état $\tilde{S}$ à l'étape 4 de l'itération précédente.

De cette manière, on obtient une politique SMDP+ que l'on prolonge sur $\mathbb{R}$. Les périodes de décision sur lesquelles sont définies la politique SMDP+ tendent à minimiser la t-erreur de Bellman et donc tendent vers les périodes où cette erreur est nulle, on converge ainsi vers des périodes de décision où l'action à effectuer est constante et correspond à tout instant à l'action optimale à entreprendre.

Cet algorithme n'a pas encore été implémenté et testé : les objectifs d'implémentation sont pour l'instant de finir le planificateur dans le cadre TMDP, puis d'y adjoindre cette méthode de résolution, et enfin d'étendre à des fonctions autres que polynômiales cette méthode.

Chapitre 4

La généralisation aux espaces d'actions continus (actions paramétriques)

4.1 Espaces d'actions continus / actions paramétriques

Dans le traitement des MDP de façon générale, certains travaux se sont attachés à intégrer des variables continues dans la résoution des MDP. On peut citer à ce sujet les approches type POMDP ([KLC98]) et les approches par programmation linéaire approchée (ALP) [HK04] ou par factorisation pour la discrétisation de l'espace d'état ([FDMW04]). Ces approches permettent de traiter des variables comme l'espace, les ressources ou le temps à horizon fini du point de vue continu, autorisant une planification plus fine et évitant l'explosion combinatoire de l'espace d'états dûe à la discrétisation, en contrepartie d'algorithmes adaptés au traitement de variables et fonctions continues. Cependant, les actions définies dans tous ces modèles restent discrètes et parfois limitatives de la liberté d'action de l'agent ; on dispose par exemple d'une action qui a pour effet telle distribution sur la position d'arrivée, ou telle distribution sur l'état des ressources après l'action. Or, quand on définit une action "avancer" par exemple, la première chose qui vient à l'esprit pour pouvoir définir l'action précisément est "avancer de combien ?". Chacune des actions considérées dans les cadres précédents associe une unique valeur au paramètre "longueur d'un pas" et définit alors l'action associée par ses effets (probabilistes) sur l'état. De la même manière que nous sommes passés d'un espace d'état dénombrable dans le cadre MDP classique à un espace d'états non-dénombrable en y introduisant les variables d'état continues, on se propose ici de définir un cadre pour définir des *actions continues* ou *actions paramétriques* dans le contexte de la planification temporelle en environnement instationnaire.

Cette démarche découle naturellement des travaux précédents, en effet, les difficultés que l'on a éprouvées tout au long de la modélisation et de la résolution vis-à-vis de l'action "attendre" viennent du fait que si on considère un temps continu observable, alors on doit considérer "attendre" comme une action paramétrique de paramètre "durée". Cette démarche peut alors se généraliser à de nombreuses autres actions, on citera pour exemple les actions "pivoter d'un angle θ" pour le satellite agile en orbite, "avancer d'une distance L" pour le robot démineur dans un champ de mines, "investir une somme X" pour l'agent en bourse, "injecter une dose d du produit A" pour le système d'aide à la décision médicale, et bien sûr "attendre une durée τ" pour notre agent pompier perdu au milieu des flammes.

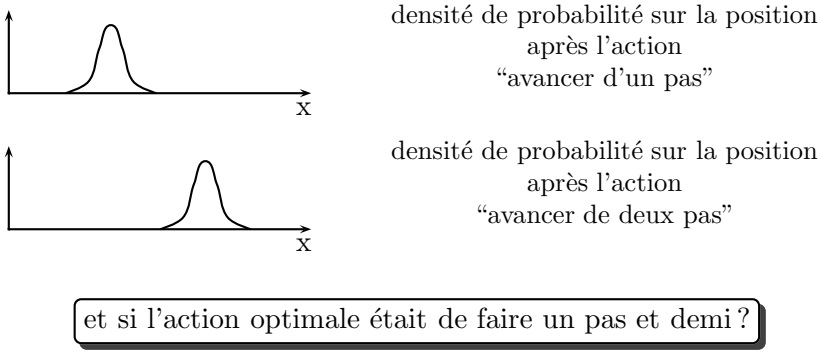

FIGURE 4.1 – Problématique des actions continues

La définition d'actions continues dans des problèmes de décision contraints à été introduite dans le formalisme des Controlled Markov Chains (CMC) dans [Alt99, AS93] et poursuivies dans le cadre de la programmation dynamique dans [Ber95]. Dans les sections suivantes, on s'attache à définir formellenent le cadre d'étude des actions paramétriques pour des problèmes instationnaires et à explorer les possibilités de résolution. En particulier, à la section 4.2, on introduit le vocabulaire et le cadre de représentation des actions continues dépendantes du temps, inspiré du formalisme des chaînes de Markov contrôllées. Puis à la section 4.3 on présente les grandes lignes d'une première méthode envisagée pour la résolution de ces problèmes à actions paramétriques. Enfin à la section 4.4 on reprend le problème de l'action "attendre" précédemment abordé et on montre qu'il s'agit bien d'un problème à action paramétrique et on expose comment on retrouve la méthode de résolution proposée au 2.2 quand on effectue une résolution du problème par programmation dynamique. On conclut alors sur cette première proposition dans le cadre des actions continues à la section 4.6.

4.2 Cadre formel de modélisation

On définit un MDP instationnaire à actions paramétriques comme :

Définition (MDP instationnaire à actions paramétriques)**.** *Un MDP instationnaire à actions paramétriques est décrit par :*

- *un espace d'états S composé de variables continues ou discrètes, incluant une variable temporelle observable t,*
- *un espace d'actions paramétriques $a(x) \in A$ avec x le vecteur des paramètres, prenant ses valeurs dans un ensemble X qu'on appelle espace des paramètres,*
- *un modèle de transition $P(s'|s, a, x)$,*
- *un modèle de récompense $r(s, a, x)$.*

On considère que P est une distribution à densité p (densité discrète pour les variables discrètes, continue pour les variables continues), on écrit alors $dP(s'|s, a, x) = p(s'|s, a, x)ds'$.

On définit également une politique pour ce problème :

Définition (politique)**.** *Une politique d'un MDP instationnaire à actions paramétriques est une application :*

$$\pi : \left\{ \begin{array}{ccc} S & \rightarrow & A \times X \\ s & \mapsto & (a, x) \end{array} \right.$$

On redéfinit également le critère γ-pondéré (et le critère total par la même occasion) :

Définition (critère γ-pondéré)**.** *On cherche à optimiser la politique en maximisant le critère :*

$$V_\gamma^\pi = E(\sum_{\delta=0}^{\infty} \gamma^{t_\delta} r_\delta^\pi | s_0) \tag{4.1}$$

avec :

$$r_\delta^\pi = r(s_\delta, \pi(s_\delta))$$

Ce critère se traduit sous la forme d'une équation de Bellman généralisée :

Définition (equation de Bellman)**.**

$$Q(s, a, x) = r(s, a, x) + \int_{s' \in S} \gamma^{\tau(s', s)} V(s') dP(s' | s, a, x) \tag{4.2}$$

$$Q(s, a) = \sup_{x \in X} Q(s, a, x) \tag{4.3}$$

$$V(s) = \max_{a \in A} Q(s, a) \tag{4.4}$$

Cette équation traduit la propriété suivante : en un état, le fait d'entreprendre l'action $a(x)$ permet de gagner $r(s, a, x)$ dans un premier temps, puis toutes les récompenses accessibles à partir de s' dans un second temps (equation 4.2). Optimiser une politique π sur un coup (par programmation dynamique) revient alors à choisir, pour chaque action séparément, le vecteur de paramètres x qui maximise le revenu de l'action (equation 4.3), puis, parmi tous ces revenus d'action, choisir l'action qui a le plus fort gain (equation 4.4). C'est ainsi que l'on peut imaginer une première méthode de résolution par programmation dynamique.

On peut résumer l'équation de Bellman en écrivant que la fonction de valeur optimale est le point fixe de l'opérateur L :

$$V^*(s) = \max_{a \in A} \left\{ \sup_{x \in X} \left(r(s, a, x) + \int_{s' \in S} \gamma^{\tau(s', s)} V^*(s') dP(s' | s, a, x) \right) \right\} = LV^*(s) \tag{4.5}$$

Dans les équations 4.2 et 4.5, la grandeur $\tau(s', s)$ correspond au $t_{\delta+1} - t_\delta$ du critère. Si le temps est observable (c'est-à-dire s'il y a une variable temporelle dans l'espace d'état), alors $\tau(s', s) = t(s') - t(s)$. Si ce n'est pas le cas, alors on est dans un cadre où le chaque durée d'action est considérée unitaire (comme dans un MDP classique) et où $\tau(s', s) = 1$.

A présent que nous avons posé notre problème, nous allons tenter d'y apporter des méthodes de résolution.

4.3 Méthode de résolution

La première méthode de résolution que l'on se propose de mettre en place est une méthode par programmation dynamique, découlant directement du cadre classique, des chapitres précédents et de la formulation qui vient d'être introduite.

On définit donc la suite de fonctions $(V_n(s))_{n\in\mathbb{N}}$ que l'on fait converger vers V^* en utilisant la propriété du point fixe de L. La suite $(V_n(s))_{n\in\mathbb{N}}$ vérifie :

Définition (équation de programmation dynamique)**.**

$$\forall s \in S, V_{n+1}(s) = \max_{a\in A}\left\{\sup_{x\in X}\left(r(s,a,x) + \int_{s'\in S}\gamma^{\tau(s',s)}V_n(s')dP(s'|s,a,x)\right)\right\} = LV_n(s) \quad (4.6)$$

On définit ainsi aisément des algorithmes type "itération de la valeur".

4.4 Retour sur le cas précédent : l'action "attendre" est une action paramétrique

On peut réécrire le problème TMDP dane le cadre des actions paramétriques. On identifie alors les éléments des deux formalismes :

— L'état est consitué des variables $(s,t) \in S \times \mathbb{R}$ définies à la section 1.2.3. On va redéfinir les notations et on va noter s_d l'état discret (et S_d son ensemble), s_t la variable temporelle (et S_t son ensemble) et on renomme $s = (s_d, s_t)$ ainsi que S l'ensemble des (s_d, s_t).
— l'espace d'action est constitué de toutes les actions discrètes, indépendantes du vecteur des paramètres. Pour chaque action discrète a_i on a en fait $a_i(x) = a_i$. On ajoute à cet espace l'action $attendre(\tau)$ où τ est le paramètre de durée. L'espace des paramètres ne décrit donc qu'un unique paramètre (x=τ) positif, on a donc $X = \mathbb{R}^+$.
— Pour les actions discrètes, on a :

$$p(s'|s,a,\tau) = \sum_{\mu_{s'_d}} L(\mu_{s'_d}|s_d,a,s_t)\cdot P_{\mu_{s'_d}}(s'_t - s_t)$$

$\mu_{s'_d}$ désignant l'ensemble des μ accessibles à partir de s et réalisant s'_d. On note qu'ici, les cas ABS et REL n'ont qu'une importance de modélisation : quel que soit le cas, c'est bien la durée de la transition que l'on définit (au besoin, on peut remplacer $P_{\mu_{s'_d}}(s'_t - s_t)$ par $P_{\mu_{s'_d}}(s'_t)$ pour passer dans le cas ABS). Et, comme on a considéré l'action attendre comme déterministe et stationnaire au 2, on peut écrire :

$$P(s'|s, attendre, \tau) = \begin{cases} 1 & \text{si } s'_d = s_d \text{ et } s'_t = s_t. \\ 0 & \text{sinon} \end{cases}$$

Soit encore :

$$p(s'|s, attendre, \tau) = \delta_{(s_d, s_t+\tau)}(s')$$

— Enfin le modèle de récompense est défini comme précédemment pour les actions discrètes :

$$r(s,a,\tau) = r_t(\mu_{s'_d}, s_t) + \int_{s' \in S} p(s'|s,a,\tau) \left[r_{t'}(\mu_{s'_d}, s'_t) + r_\tau(\mu_{s'_d}, s'_t - s_t) \right] ds' \qquad (4.7)$$

Et pour l'action paramétrique attendre, on a en fait $r_\tau(\mu_{s'_d}, y) = \int_{s_t}^{s_t+y} K(s_d, \theta) d\theta$, $r_t(\mu_{s'_d}, s_t) = 0$ et $r_{t'}(\mu_{s'_d}, s'_t) = 0$ et donc :

$$r(s, attendre, \tau) = \int_{s_t}^{s_t+\tau} K(s_d, \theta) d\theta$$

On remarque ici que pour l'écriture de l'espace d'états, on a assez naturellement partitionné ce dernier en trois parties : l'ensemble des variables discrètes S_d qui engendreront des distributions de probabilité discrètes, l'ensemble des variables d'état continues (implicites dans l'écriture ci-dessus car il n'y en a pas) S_c, et la variable temporelle qui prend ses valeurs dans S_t et qui alimente la fonction spéciale $\tau(s'_t - s'_t)$. Cette décomposition naturelle résulte de l'abstraction des différents aspects : discret, continu et temporel du problème à actions paramétriques.

Ainsi on a bien pu traduire entièrement le modèle TMDP dans le cadre des actions paramétriques. Nous allons maintenant vérifier que l'équation de programmation dynamique proposée par [BL01] et que nous avons étendue correspond bien à l'écriture de l'équation de Bellman généralisée que nous avons introduite au 4.3.

On cherche à résoudre l'équation 4.5, soit :

$$\begin{aligned}
V^*(s) &= \max_{a \in A} \left\{ \sup_{\tau \in \mathbb{R}^+} \left(r(s,a,\tau) + \int_{s' \in S} \gamma^{s'_t - s_t} V^*(s') p(s'|s,a,\tau) ds' \right) \right\} \\
&= \max_{a \in A} \left\{ \sup_{\tau \in \mathbb{R}^+} \left(r(s,a,\tau) + \int_{s' \in S} \gamma^{s'_t - s_t} V^*(s') \sum_{\mu_{s'_d}} L(\mu_{s'_d}|s_d,a,s_t) \cdot P_{\mu_{s'_d}}(s'_t - s_t) ds' \right) \right\} \\
&= \max_{a \in A} \left\{ \sup_{\tau \in \mathbb{R}^+} \left(r(s,a,\tau) + \iint_{\substack{s'_d \in S_d \\ s'_t \in S_t}} \gamma^{s'_t - s_t} V^*(s') \sum_{\mu_{s'_d}} L(\mu_{s'_d}|s_d,a,s_t) \cdot P_{\mu_{s'_d}}(s'_t - s_t) ds'_d ds'_t \right) \right\} \\
&= \max_{a \in A} \left\{ \sup_{\tau \in \mathbb{R}^+} \left(r(s,a,\tau) + \sum_{s'_d \in S_d} L(\mu_{s'_d}|s_d,a,s_t) \int_{s'_t \in S_t} \gamma^{s'_t - s_t} V^*(s') \cdot P_{\mu_{s'_d}}(s'_t - s_t) ds'_t \right) \right\}
\end{aligned}$$

Dans le cadre TMDP, on avait $\gamma = 1$, donc (comme mentionné plus haut, on considère la différence entre cas REL et ABS comme implicite ; on pourrait, le cas échéant, remplacer $P_{\mu_{s'_d}}(s'_t - s_t)$ par $P_{\mu_{s'_d}}(s'_t)$) :

$$V^*(s) = \max_{a \in A} \left\{ \sup_{\tau \in \mathbb{R}^+} \left(r(s,a,\tau) + \sum_{s'_d \in S_d} L(\mu_{s'_d}|s_d,a,s_t) \int_{s'_t \in S_t} P_{\mu_{s'_d}}(s'_t - s_t) V^*(s') ds'_t \right) \right\}$$

On sépare l'action *attendre* des actions discrètes :

$$
\begin{aligned}
V^*(s) = max \Bigg\{ & \max_{a \in A \setminus \{attendre\}} \left\{ \sup_{\tau \in \mathbb{R}^+} \left(r(s,a,\tau) + \sum_{s'_d \in S_d} L(\mu_{s'_d} | s_d, a, s_t) \int_{s'_t \in S_t} P_{\mu_{s'_d}}(s'_t - s_t) V^*(s') ds'_t \right) \right\}, \\
& \sup_{\tau \in \mathbb{R}^+} \left(r(s, attendre, \tau) + \sum_{s'_d \in S_d} L(\mu_{s'_d} | s_d, attendre, s_t) \int_{s'_t \in S_t} P_{\mu_{s'_d}}(s'_t - s_t) V^*(s') ds'_t \right) \Bigg\} \\
= max \Bigg\{ & \max_{a \in A \setminus \{attendre\}} \left\{ \sup_{\tau \in \mathbb{R}^+} \left(r(s,a,\tau) + \sum_{s'_d \in S_d} L(\mu_{s'_d} | s_d, a, s_t) \int_{s'_t \in S_t} P_{\mu_{s'_d}}(s'_t - s_t) V^*(s') ds'_t \right) \right\}, \\
& \sup_{\tau \in \mathbb{R}^+} \Big(r(s, attendre, \tau) + V^*(s_d, s_t + \tau) \Big) \Bigg\}
\end{aligned}
$$

La nature statique de l'action *attendre* (on ne change pas d'état discret s_d) fait qu'on ne peut avoir deux actions *attendre* successives (on le montre par l'absurde en considérant le $\sup_{\tau \in \mathbb{R}^+}(\ldots)$ et en montrant que si deux actions successives sont $attendre(\tau_1)$ et $attendre(\tau_2)$ avec τ_1 et τ_2 non nuls, alors il existe une action $attendre(\tau_1 + \tau_2)$ qui vaut plus que $attendre(\tau_1)$ en premier lieu et donc que l'action $attendre(\tau_1)$ ne correspond pas au $\sup_{\tau \in \mathbb{R}^+}(\ldots)$). La récompense associée à une attente nulle est celle associée à l'action discrète suivante car $r(s, attendre, 0) = 0$. On peut donc considérer que notre séquence d'actions est une alternance de séquences attente-action. Cela nous permet d'écrire que :

$$
\begin{aligned}
V^*(s) = \sup_{\tau \in \mathbb{R}^+} \Bigg(r(s, attendre, \tau) + \max_{a \in A \setminus \{attendre\}} \Bigg\{ & r(s,a,\tau) + \sum_{s'_d \in S_d} L(\mu_{s'_d} | s_d, a, s_t) \cdot \\
& \int_{s'_t \in S_t} P_{\mu_{s'_d}}(s'_t - s_t) V^*(s') ds'_t \Bigg\} \Bigg)
\end{aligned}
\tag{4.8}
$$

Or on a (equation 4.7) :

$$
\begin{aligned}
r(s,a,\tau) &= r_t(\mu_{s'_d}, s_t) + \int_{s' \in S} p(s'|s,a,\tau) \left[r_{t'}(\mu_{s'_d}, s'_t) + r_\tau(\mu_{s'_d}, s'_t - s_t) \right] ds' \\
&= \int_{s' \in S} p(s'|s,a,\tau) \left[r_t(\mu_{s'_d}, s_t) + r_{t'}(\mu_{s'_d}, s'_t) + r_\tau(\mu_{s'_d}, s'_t - s_t) \right] ds' \\
&= \sum_{s'_d \in S_d} L(\mu_{s'_d} | s_d, a, s_t) \int_{s'_t \in S_t} P_{\mu_{s'_d}}(s'_t - s_t) \left[r_t(\mu_{s'_d}, s_t) + r_{t'}(\mu_{s'_d}, s'_t) + r_\tau(\mu_{s'_d}, s'_t - s_t) \right] ds'_t
\end{aligned}
$$

Donc l'équation 4.8 se réécrit :

$$V^*(s) = \sup_{\tau \in \mathbb{R}^+} \left(r(s, attendre, \tau) + \max_{a \in A \setminus \{attendre\}} \left\{ \sum_{s'_d \in S_d} L(\mu_{s'_d} | s_d, a, s_t) \cdot \right. \right.$$
$$\left. \left. \int_{s'_t \in S_t} P_{\mu_{s'_d}}(s'_t - s_t) \left[r_t(\mu_{s'_d}, s_t) + r_{t'}(\mu_{s'_d}, s'_t) + r_\tau(\mu_{s'_d}, s'_t - s_t) + V^*(s') \right] ds'_t \right\} \right) \quad (4.9)$$

L'équation 4.9 correspond exactement aux équations 1.10 à 1.13 ; la politique que l'on exprime avec le formalisme d'actions continues est donc bien la même que celle qu'on a calculée dans le modèle TMDP. On peut donc conclure sur la question initiale : le problème TMDP est un problème instationnaire à action paramétrique et sa résolution par programmation dynamique est bien équivalente à la méthode de programmation dynamique que l'on a décrite au 4.3. Dans la méthode proposée par [BL01], l'aspect paramétrique est masqué par le fait qu'une action d'une politique TMDP comporte en fait deux actions réelles : une action $attendre(\tau)$ et une action discrète. La séparation de ces deux actions et le fait que l'action $attendre(0)$ n'ait aucune influence permet de mettre en évidence le processus de la chaîne de Markov contrôllée sous-jacent.

On peut également mentionner ici que si on avait conservé le terme en γ, l'équation 4.9 s'écrirait :

$$V^*(s) = \sup_{\tau \in \mathbb{R}^+} \left(r(s, attendre, \tau) + \gamma^\tau \max_{a \in A \setminus \{attendre\}} \left\{ \sum_{s'_d \in S_d} L(\mu_{s'_d} | s_d, a, s_t) \cdot \right. \right.$$
$$\left. \left. \int_{s'_t \in S_t} \gamma^{s'_t - s_t} P_{\mu_{s'_d}}(s'_t - s_t) \left[r_t(\mu_{s'_d}, s_t) + r_{t'}(\mu_{s'_d}, s'_t) + r_\tau(\mu_{s'_d}, s'_t - s_t) + V^*(s') \right] ds'_t \right\} \right) \quad (4.10)$$

Cette équation représente l'équation de programmation dynamique pour un TMDP γ-pondéré.

4.5 Mais alors pourquoi une étude du temps dans le cadre MDP ?

A présent que l'on a défini le formalisme d'actions paramétriques (ou continues), et qu'on a montré qu'il englobait le travail effectué sur le modèle TMDP et le généralisait à plusieurs actions et à des effets probabilistes plus généraux, on peut se demander l'intérêt de rendre particulière la variable temporelle.

Il y a plusieurs réponses à ce problème. La première réponse est d'ordre mathématique, nous la présentons brièvement avant de présenter la réelle raison physique qui fait qu'on ne peut considérer la variable temporelle comme une variable continue comme les autres.

La première raison vient du critère que l'on utilise. Le critère γ-pondéré effectue une pondération de la récompense en fonction de sa date d'obtention dans le futur, il met, par cela, en avant la durée d'une transition et la particularise via la fonction $\tau(s', s)$ vue à l'équation 4.5. Ce faisant, la résolution d'un MDP à actions paramétriques "standard" diffère de celle d'un MDP à actions paramétriques pour lequel le temps est un paramètre de l'action car alors la résolution doit mettre en oeuvre des techniques permettant de traiter le terme $\gamma^{\tau(s',s}$ qui apparait dans les équations 4.2, 4.5 et 4.6 (l'apparition de ce terme vient simplement du fait qu'on a rendu observable et contrôlable la durée comme paramètre d'action). Ainsi, il y a bien un sens à chercher des méthodes de résolution aux problèmes dépendant explicitement du temps, en plus du cadre des actions paramétriques, car même dans le cadre des actions paramétriques la variable temporelle conserve une place à part (grace à sa présence particulière dans le critère γ-pondéré).

La seconde raison est plus d'ordre physique (ou philosophique) : il s'agit de bien comprendre le sens de chacune des variables que l'on manipule. Cette considération rejoint les trois sens de la variable temporelle dont on a discuté en section 1.3. En effet — et cela rejoint la considération mathématique précédente — le principal aspect du problème temporel qui le différencie d'un problème à actions continues standard réside dans le fait que le temps a une triple signification :

— il représente le *paramètre* de l'action ("attendre(durée)"),
— il représente une *variable d'état* ("date courante"),
— il représente le *temps de la chaîne* de Markov (t_δ) qui représente le problème ($\gamma^{t_{\delta+1}-t_\delta}$).

La variable temporelle couple ainsi des aspects non-contrôlables (le temps de la chaîne) et des aspects contrôlables (l'état du système). On peut retrouver cette caractéristique dans une moindre mesure sur d'autres variables d'action continues comme la position dans le cas d'actions de déplacement, mais sans affecter la dynamique de la chaîne.

4.6 Conclusion sur l'utilisation d'espaces d'actions continus

L'introduction du formalisme des MDP instationnaires à actions paramétriques est le résultat de la maturation des travaux entrepris dans le cadre SMDP+ d'abord, puis dans le cadre TMDP. La définition d'un MDP instationnaire à actions continues englobe l'étude des deux cadres précédents sans les rendre obsolètes puisqu'ils contituent des méthodes de résolution à part entière pour certaines classes de CMC : les problèmes temporels explicites.

Toutefois, les MDP instationnaires à actions continues constituent un cadre plus général qu'il faut à présent mettre en lumière des travaux effectués sur les MDP à variables continues et discrètes. En effet, dans [KP99], [GKP01], [SP01], [dFR01], [HK04], [GHK04], [HK06], [KH06a] puis [KH06b] (par ordre de progression d'idées), ou dans [FDMW04], les auteurs développent un cadre très complet de traitement des problèmes à variables d'état continues et discrètes ; l'introduction de variables continues implique l'introduction d'effets continus et l'étape suivante réside dans l'optimisation continue des actions engendrant ces effets continus. De façon plus terre-à-terre, on a introduit, sur le modèle des CMC, une variable "distance" continue qu'on a rendue continue plutôt que de la discrétiser. On a alors défini les effets continus de l'action

"faire un pas" sur cette variable continue. On a étendu cette méthode aux particularités de la variable temporelle. L'étape naturelle suivante à laquelle répond le cadre des MDP instationnaire à actions paramétriques est de définir une action continue "faire x pas" où x est un nombre réel, dans un univers dépendant du temps. Ainsi, le cadre des actions continues constitue un prolongement naturel des travaux cités ci-dessus et ouvre des perspectives de traitement de nouvelles classes de problèmes à espaces d'états, d'action et temps continus.

Ainsi, en abstrayant la démarche générale que l'on a mis en place pour les SMDP+ et les TMDP, on est parvenus à un cadre formel général de représentation des actions dans le cadre MDP. Ce cadre permet d'une part de traiter les problèmes à variables continues en explicitant le sens des différentes variables du modèle, et d'autre part, il permet de traiter des types de problèmes qui constituent un prolongement logique des travaux effectués sur les variables d'état discrètes et continues dans le cadre MDP.

Chapitre 5

L'insertion dans le cadre "en ligne" et biagent décentralisé

Reprenons le problème inital que l'on s'est posé en introduction. On a abstrait de ce problème des grandes caractéristiques, notamment :

- l'aspect "en ligne" : à savoir le besoin de réactivité, de capacité à corriger des plans pendant l'exécution,
- l'aspect décentralisé de la prise de décision pour les deux agents : cette option étant en partie arbitraire, on la discutera au début de cette section,
- le besoin de coordination temporelle : que ce soit d'un agent avec l'univers ou de deux agents entre eux,
- le caractère instationnaire du problème qui implique la planification en fonction d'une variable de temps explicite,
- et l'aspect incertain des transitions entre différents états, des durées d'action et des dates de réalisation d'évènements exogènes.

Puis on a décomposé le problème d'un point de vue hiérarchique en un problème de coordination biagent au niveau mission et un problème de planification monoagent au niveau planification et exécution.

On a alors défini les grandes lignes d'une méthode de coordination biagent, dont on s'est aperçu qu'elle nécessitait l'existence d'un algorithme de planification temporelle monoagent dans l'incertain. C'est cette première proposition de méthode de coordination que l'on se propose de présenter ici. Une étape préliminaire à toute validation de cette méthode qui est — somme toute — assez intuitive, consiste en sa comparaison avec un état de l'art complet sur les problèmes de la planification bi et multiagent dans l'incertain (notamment les travaux de [Mou04, BM04], [Bou96] ou [GAZ07, BLZ05, GZ04]) et de la coordination temporelle des systèmes multiagents (par exemple via [VdWW05] ou [JFL$^+$01]). Cette étape bibliographique de validation et/ou de modification de cette première méthode de coordination proposée fait partie des travaux prévus dans un futur proche.

5.1 Le problème-type auquel on s'intéresse

Reprenons le problème biagent présenté en introduction. Notre agent au sol et l'hélicoptère doivent coordonner leurs actions pour parvenir à remplir la mission, de façon au moins meilleure que s'ils avaient été seuls. Pour parvenir à cette coordination sans définir de supérieur hiérarchique

entre les deux agents ou sans faire intervenir d'agent-tiers qui jouerait implicitement le rôle de super-agent ou de médiateur, il apparait nécessaire - dans le cadre de la planification - de définir un protocole d'échange d'informations pour que chaque agent puisse informer l'autre des points forts de sa stratégie à des fins de coordination. On va définir ce protocole de communication en section 5.2. On verra alors comment on peut mettre en place une méthode de coordination décentralisée dans un premier cas simple où les agents sont relativements indépendants (section 5.3). Puis on s'intéressera à un cas plus complexe où apparaissent des conflits et des incohérences dues à l'existence d'actions communes (section 5.4) dans le cadre décentralisé. On verra alors comment on met en place une seconde couche au protocole de coordination pour résoudre ces conflits (section 5.4).

Avant de présenter plus avant la méthode que l'on propose, on va justifier certaines options et certaines hypothèses que l'on a choisies.

Pourquoi une coordination temporelle explicite des agents ? Cette question a déjà été partiellement traitée dans les sections précédentes, le premier besoin de coordination temporelle explicite concerne la coordination de l'agent et de son environnement. Cette coordination peut se faire par le biais d'évènements plutôt que par référence à un temps explicite (par exemple : "si un évènement x se produit, ie. si on est dans un état donné correspondant, alors entreprendre l'action a"). Cependant on se trouve dans un cadre où l'évolution du problème est continue (et éventuellement rapide) et où l'aspect de "timing" est critique. Prenons un exemple simple pour illustrer ce fait : imaginons que notre agent pompier se trouve en un point de la forêt et se base sur l'occurence de l'observation d'un état "voie libre" pour se mettre en route. Le déclenchement de son action sera alors le même, quelle que soit sa distance à la voie qui était précédemment bloquée. Or, en procédant de cette manière, on perd du temps à l'exécution car on ne prend pas en compte la durée du trajet de l'agent jusqu'à la voie en question, ou, du point de vue du problème, on ne prend pas en compte la propagation des échéances issues de l'état "voie libre" sur tout le modèle. Propager ces échéances dans un modèle discret revient en fait à propager les durées de trajet de l'agent vers la voie et donc à planifier explicitement en fonction du temps. Quitte à propager beaucoup de durées (éventuellement incertaines) dans le modèle, on a pris le parti de traiter une variable temporelle continue. Le deuxième cas de coordination concerne l'interaction des deux agents entre eux : on ramène ce cas au cas précédent par la remarque suivante : la décision étant décentralisée, chaque agent est non contrôlable par son homologue, on peut donc le considérer comme faisant partie de l'univers de ce dernier. On voit ici se profiler l'idée de la résolution que l'on propose dans les sections suivantes : l'agent A, en communiquant ses intentions à l'agent B, lui permet de modifier son modèle d'univers en conséquence et de planifier en fonction des échéances introduites par les actions de A dans l'univers.

Pourquoi une décision décentralisée ? On est dans un cadre où l'on pourrait mettre en oeuvre une décision centralisée entre les deux agents, avec des garanties de convergence et d'optimalité sûrement meilleures que celles en décentralisé. On s'intéresse à l'approche décentralisée pour trois raisons :

- Robustesse de la solution : en laissant de l'autonomie décisionnelle aux deux agents, on s'assure qu'en cas de perte du premier, le second peut réagir et adapter sa stratégie. Cela permet d'éviter le pire cas où on perd les deux agents parce que le premier est détruit.
- Distribution et utilisation des ressources embarquées : En procédant de façon décentralisée on utilise les capacités de calcul des deux agents directemnt dans le contexte qui leur est propre : chacun calcule une stratégie sans avoir à communiquer des informations sur son environnement à un super-agent qui centraliserait l'information et la décision. Par

ailleurs, le calcul d'une politique biagent ou multiagent centralisée conduit très vite à des problèmes intraitables pour des raisons de mémoire et de taille du problème (on a déjà remarqué ce phénomène dans le cadre des MMDP [Bou96] ou des Dec-MDP [BZI02]). La décentralisation permet donc une distribution du problème sur ses différentes constituantes et la recherche d'une solution sous-optimale intéressante sous la forme d'une aggrégation de solutions locales.

— Gestion de l'hétérogénéité et généricité des problèmes : Les agents que l'on souhaite coordonner sont fortement différents par leur nature physique, leurs capacités et leurs missions. Coordonner des agents identiques masque un aspect de "compréhension" du plan de l'agent homologue. Entre un hélicoptère et un rover terrestre — et de façon générale entre deux agents hétérogènes — il est nécessaire de définir un langage de communication pour la coordination. Notre approche décentralisée définit ce langage dans un premier temps, hors ligne, avant la mission. En traitant le problème de façon décentralisée, on laisse les agents traiter des problèmes de planification qui leur sont propres et limiter la communication à des messages uniquement utilies à la coordination. On peut donc simplement traiter un groupe d'agents hétérogènes.

Certains exemples simples de coordination décentralisée (problème des deux taxis, jeu de cartes) illustrent le fait que la décentralisation peut affecter l'optimalité de la solution trouvée au profit de la performance de calcul. S'il est possible que la solution optimale centralisée pour l'équipe soit l'union de deux politiques très sous-optimales individuellement, notre approche telle qu'on va la détailler risque de mettre longtemps avant de trouver cette solution. Dès lors, il nous faut distinguer deux cas de planification-exécution :

— Si la mission est prévue à l'avance et que l'on peut effectuer une planification et une coordination centralisée avant la mission, alors on suppose qu'on a trouvé, hors-ligne, de façon centralisée, un plan quasi-optimal dont on a distribué les composantes sur les agents. On suppose également qu'on a défini un protocole de communication comme celui qu'on verra en section 5.4 adapté à la mission. Le problème que l'on se pose alors est celui de la correction des plans et de la coordination de ces derniers en fonction de l'évolution du problème réel à l'exécution.

— Si en revanche les agents sont pris dans une situation d'urgence et n'ont pas le temps et les moyens de mettre en place une première passe de planification centralisée, alors le plan est à construire de façon décentralisée dès le début et on doit alors anticiper le fait que si l'on cherche à obtenir une solution meilleure que les solutions individuelles des agents, cette solution peut très bien être sous-optimale. Dans ce cadre, on cherchera à prendre en compte — de façon qualitative dans un premier temps — les contraintes sur la communication des agents et on cherchera à considérer des messages utile de taille raisonnable. Sur ces aspects de communication, il reste beaucoup à faire, notamment en relation avec les travaux de [GAZ07] et [SGV06].

5.2 Communication interagents - définition de variables communes

Prenons un exemple simpliste qu'utilisera comme fil rouge pour illustrer les idées principales des sections suivantes. Imaginons un problème où un robot au sol R est situé sur une grille de navigation à trois noeuds : un noeud de départ, un noeud de transition et un noeud où se situe un objectif à photographier. Par ailleurs un drone aérien H a également pour mission de prendre une photo de cet objet. La photo peut être prise n'importe quand car il fait jour, mais

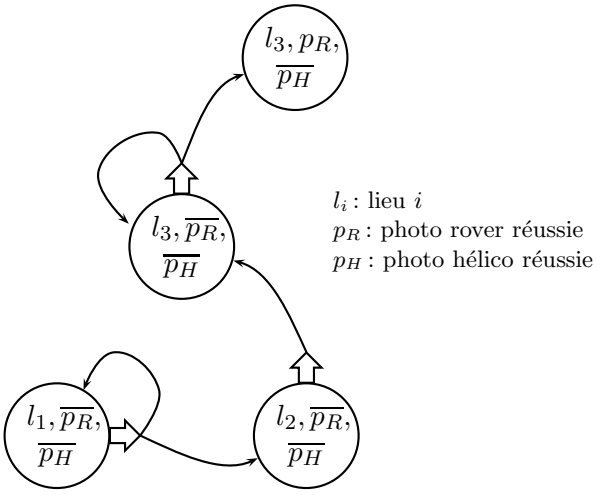

FIGURE 5.1 – Représentation simplifiée du graphe états-transitions initial pour le rover

la récompense est plus grande si les deux photos sont prises avec moins de 20 secondes d'écart pour pouvoir faire une image stéréo de la scène. Une contrainte qui nous permet de simplifier le problème réside dans le fait qu'une fois une bonne photo prise, on ne reprend plus de photo. Ce problème est représenté à la figure 5.1. En chaque état, le choix d'action est simplifié à l'extrême et réduit à un choix entre agir et attendre. On décrit simplement l'espace d'état d'un agent par les cinq variables :

— l_i, variable de lieu, décrivant la position de l'agent,
— p_R, vraie si la photo du rover est un succès,
— p_H, vraie si la photo de l'hélicoptère est un succès,
— t, la date courante, continue.

Supposons que chaque agent est capable d'élaborer un plan et qu'on ne se soucie pas du processus de coordination lui-même, mais uniquement de la phase de communication. La question qui se pose est de définir le contenu d'un message d'un agent à l'autre. On a besoin pour cela de définir le sens de ce message et sa forme. On commence par se poser la question naïve suivante :

Que veut dire l'agent R à l'agent H ? Sachant que, justement, l'agent H ne sait rien de la nature de l'agent R (et réciproquement), R est dans le besoin d'expliquer à H ce qu'il compte faire sans lui détailler son plan. Il est donc hors de question de transférer à H la stratégie de R. De fait, si R a un message à communiquer à H, c'est que leurs problèmes se recoupent et sont interconnectés. L'idée principale est donc de définir sur quelles variables porte cette connection. Dans notre exemple, la connection porte sur la variable booléenne "photo prise". En effet, le seul message que R puisse envoyer à H qui changerait le problème de H est le message "la photo terrestre sera prise à telle date". Réciproquement, le seul message de H affectant le problème de R est un message "la photo aérienne sera prise à telle date". On remarque que ce message porte sur une variable d'état décrivant une récompense qui existe dans les deux problèmes (sous des noms différents) et qui correspond à une variable commune globale de problème. On énonce donc qu'un message de R à H porte sur l'impact (temporel) de la stratégie de R sur les variables communes. On peut alors se poser la question de la forme du message.

Quel message R envoie-t-il à H ? Il est quasi-immédiat, après la remarque précédente, de déduire qu'un message de R à H porte sur les variations temporelles des variables communes du problème. Plus formellement, on définit :

Définition (Problème de coopération biagent décentralisé dans l'incertain)**.** *On définit un problème de coopération biagent décentralisé dans l'incertain comme la donnée, d'un triplet* $< \mathcal{P}_1, \mathcal{P}_2, \mathcal{V}_c >$.
$\mathcal{P}_1$ *et* $\mathcal{P}_2$ *sont deux problèmes mono-agent de décision temporelle dans l'incertain dans lesquels l'espace d'états (discret ou continu) est factorisé en deux ensembles de variables d'état : les* variables propres *(*$\mathcal{V}_{p1}$ *et* $\mathcal{V}_{p2}$*) et les* variables communes $\mathcal{V}_c$.
Les variables communes sont des variables booléennes de description de récompenses ou d'aspects du problème qui sont communs aux deux agents. Ces variables sont déclarées hors ligne ou en ligne au début du problème.
Les variables propres sont définies comme les variables décrivant un problème mono-agent et propres à l'agent. Le produit cartésien des supports de $\mathcal{V}_{p1}$ *et ce* $\mathcal{V}_c$ *forme l'espace d'état de* $\mathcal{P}_1$.

Dans notre exemple précédent, les variables communes sont les variables p_R et p_H. On définit alors le contenu d'une communication :

Définition (Communication)**.** *On définit une communication — dans le cadre d'un problème de coordination biagent décentralisé dans l'incertain — comme une vecteur de densités de probabilité* $p_v(t)$ *décrivant la probabilité qu'une variable commune* v *soit vraie à la date* t.

Ainsi, chaque agent est capable de communiquer à son collègue l'impact de sa stratégie sur l'univers : par exemple en simulant une exécution de son plan, ou en intégrant les probabilitées d'occurence de l'évènement v sur tout son problème en fonction du temps, l'agent H peut définir la fonction $p_v(t)$ et la communiquer à R. On note qu'on peut effectuer une approche simplifiée en approximant les densités de probabilités : on peut définir un seuil de plausibilité pour la réussite d'un évènement et ne communiquer que la date de réalisation de l'évènement, ainsi, si H considère sa politique et détermine qu'entre les dates $t = 26$ et $t = 28$ il a une probabilité de prendre la photo supérieure à 0.9, il peut, plutôt que d'envoyer l'évolution continue de cette probabilité avec le temps, envoyer un message simplifié à B : "$(v = VRAI, t \in [26, 28])$".

Ce cadre de communication nous permet de propager à moindre coût de communication les conséquences des actions d'un agent dans l'univers de l'autre. Cette première proposition de méthode est à confronter à d'autres approches pour déterminer si elle est efficace et pour la raffiner ou la changer. On s'en sert dans la suite pour définir les différents aspects de l'approche de coordination que l'on propose.

5.3 Coordination sans conflits

Prenons notre problème jouet précédent et observons ce que l'on peut tirer d'une communication entre agents. Nous nous plaçons dans le cadre où il n'y a pas eu de première passe de coordination centralisée, chaque agent construit donc son premier plan san pouvoir supposer quoique ce soit sur l'action de son collègue. Chaque agent planifie ses actions seul et, initialement, nous supposerons qu'il n'y a aucune action "attendre" dans les premiers plans solutions

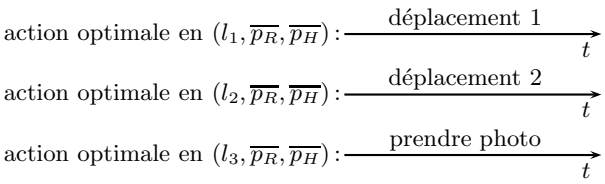

FIGURE 5.2 – Politiques initiales

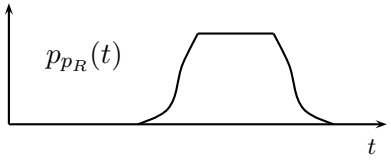

FIGURE 5.3 – Densité de probabilité sur la date de prise de la photo pour le rover

qui se représentent de façon simple à la figure 5.2.

Les variables communes sont les booléens p_R et p_H. L'agent R considère sa politique et ses transitions et réalise que selon cette politique, il a une densité de probabilité $p_{p_R}(t)$ (représentée à la figure 5.3) de prendre la photo à la date t.

Il émet donc le message "$(p_R, p_{p_R}(t))$" à destination de H. H, entre temps, a fait de même et si sa stratégie est logiquement la même que celle de R, son estimation de la date de prise de photo est différente car la dynamique des deux agents n'est pas la même (H se déplace plus vite mais de façon plus aléatoire que R par exemple), la fonction $p_{p_H}(t)$ est donc différente de $p_{p_R}(t)$.

H et R intègrent alors le sens du message reçu dans leur modèle. La méthode de mise à jour du modèle individuel d'univers dépend de la spécification du problème, on peut noter que pour un modèle factorisé par variables d'état, la mise à jour est facilitée, en effet, dans un tel modèle les fonctions de récompense du modèle sont additives et les probabilités de transition sont décomposées en produits de probabilités sur des groupes de variables. Dans le cas de notre exemple, l'intégration du message de R dans le graphe d'états-transitions de H rend possibles certaines transitions qui n'avaient pas été représentées précédemment car leur probabilité d'occurence était nulle. On représente ce nouveau graphe états-transitions pour H à la figure 5.4.

H stocke alors sa politique initiale et résout ce nouveau problème proposé par R. Le message qu'il enverra à destination de R à l'issue de cette résolution sera une réponse à la proposition de R. H stockera alors ce problème et reprendra le problème initial dans lequel il intégrera la réponse de R, il étendra sa politique initiale à son problème corrigé par R et l'optimisera de nouveau. On remarque qu'ainsi, on effectue en permanence l'optimisation de deux stratégies distinctes et que chaque agent considère alternativement de chaque stratégie. Chacune de ces stratégies sont issues d'une stratégie "graine" initiale proposée par un des agents qui a orienté la recherche dès le début puisqu'on n'a pas pu faire de passe de planification centralisée initialement.

Le problème présenté figure 5.4 permet à H de planifier "en fonction du plan de R" sans avoir eu à communiquer la totalité de ce plan. Il permet notamment à H de décider s'il est

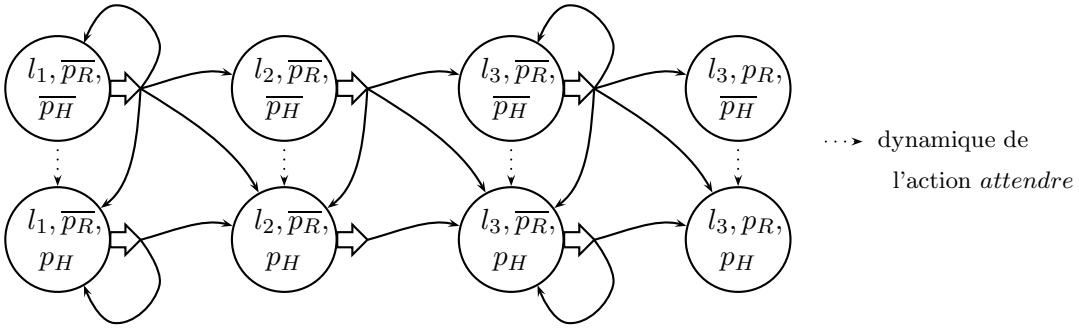

FIGURE 5.4 – Mise à jour du problème pour l'hélicoptère

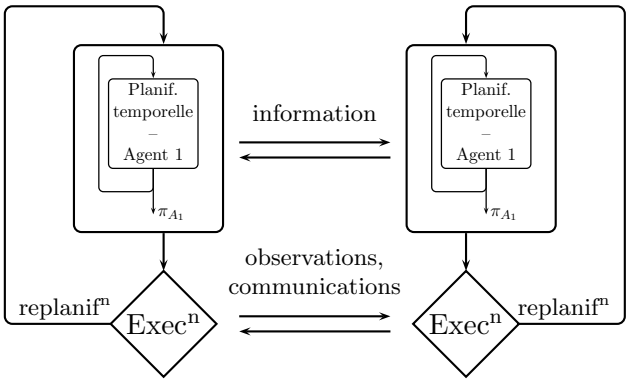

FIGURE 5.5 – Illustration du fonctionnement en ligne de la méthode de coordination

plus intéressant pour lui de se coordonner avec R pour prendre la photo en même temps ou de prendre la photo à un autre moment, plus tôt par exemple pour limiter les risques de panne. On parvient ainsi à coordonner les stratégies des deux agents.

De façon plus générale, pendant l'exécution, si un agent détecte que la réalité s'écarte de ce que son modèle prévoit — ou bien que l'execution de sa stratégie se déroule mal —, et que cela influe sur le message qu'il a envoyé à son homologue, alors il peut demander une replanification en corrigeant le plan courant de son côté puis en renvoyant simplement un nouveau message contenant une variation temporelle des variables communes. Ce fonctionnement global est illustré à la figure 5.5.

De cette manière, on propose une solution décentralisée pour construire des stratégies de coopération et de coordination biagent. Cette approche avait été proposée en version simplifiée dans le rapport[Rac05]. Afin de comparer les deux stratégies obtenues au final, on rajoute un élément aux messages émis afin de quantifier l'apport personnel de chaque agent à l'espérance de gain global de la mission, ce qui permet de trancher au final et de choisir une des deux stratégies à mettre en oeuvre.

Un problème de taille se pose cependant : on n'a aucune garantie de trouver une solution optimale et surtout, on n'a aucune garantie de trouver de solution du tout (dans le cas où il y

en a une). On cherche donc à mettre en évidence dans un premier temps les causes qui font que l'on peut se retrouver avec un algorithme qui renvoie une politique qui n'atteint pas le but.

5.4 Le problème des actions communes

Une politique permet d'atteindre le but si, par exemple, dans le processus de programmation dynamique qui a permis de la trouver, il existe un chemin de probabilité non-nulle qui relie l'état initial et le (ou les) but(s). Dans un problème monoagent, on définit ainsi l'espace atteignable par programmation dynamique directe et l'espace des états qui peuvent mener à une solution par programmation dynamique inverse. Dans le cas de notre problème, il n'y a pas de solution si on n'arrive pas à initialiser le problème et donc si on n'arrive pas à construire de stratégie monoagent initiale (on note que si on peut trouver deux stratégies initiales et qu'elles sont destructrices entre elles ou non-compatibles il suffit de ne rien faire pour un agent et d'appliquer la politique pour l'autre : la solution n'est pas optimale mais il en existe au moins une). On peut ne pas avoir de stratégie initiale si le problème biagent ne comporte pas de composante resolvable par un des agents seul. Donc trouver les caractéristiques qui impliquent que la planification monoagent ne trouve pas de solution implique de trouver les aspects du problème biagent que l'on n'a pas conservé dans la problème monoagent lors du passage centralisé $\rightarrow$ décentralisé. En d'autre mots, il s'agit d'isoler les objectifs qui ne sont pas atteignables par un agent seul mais uniquement par les deux de façon conjointe : il s'agit d'isoler les actions communes. Ces actions se distinguent parmi les actions coordonnées que l'on a vues jusque là, en effet, toutes les actions avaient jusqu'ici, toujours un intérêt individuel pour l'agent, pour une action commune, il n'y a aucun gain individuel, simplement un gain d'équipe.

Nous pouvons illustrer notre propos en modifiant l'exemple de l'hélicoptère et du rover des sections précédentes. Imaginons à présent que la mission se déroule de nuit, que l'hélicoptère soit toujours équippé d'un appareil photo, mais que le rover dispose, à la place de son appareil photo, d'un dispositif d'éclairage. La mission est toujours de prendre une photo de la zone l_3. Un bref retour sur l'algorithme précédent nous montre que l'on est dans un cas de blocage. En effet, du point de vue des modèles individuels, le rover n'a rien à gagner à éclairer la zone et l'hélico n'a rien à gagner à prendre une photo de nuit. L'ensemble des récompenses du problème sont donc nulles ou négatives (négative si on considère par exemple un coût en carburant) et la planification monoagent ne fournit aucune politique individuelle qui permette de résoudre le problème. En effet, même s'il existe des états à récompenses positives dans le problème, ces dernières sont situées dans une partie de l'état non-atteignable : par exemple, il faudrait que l'hélicoptère puisse atteindre l'état $(l_3, p_R, \overline{p_H})$ (où p_R désigne maintenant l'éclairage de la zone par le rover) pour pouvoir y entreprendre l'action de prendre la photo, or initialement, il ne peut pas puisqu'il résout le problème seul. Inversement, il faudrait que le rover soit au courant de l'intention de l'hélicoptère de prendre une photo pour pouvoir considérer une récompense associée au fait d'éclairer la zone à un moment donné. Dans cette solution de blocage, biaiser le modèle pour initier une première politique qui permettra d'élaborer une solution n'est pas acceptable car d'une part elle brise l'autonomie des agents en introduisant un acteur humain, et d'autre part parce que le modèle représente la réalité et que le modifier revient à résoudre un autre problème, même si la solution de ce nouveau problème nous conviendrait pour le premier. Une première passe de planification centralisée est par contre envisageable si on en a l'occasion. Cette passe trouvera une stratégie solution s'il y en a une et l'algorithme en ligne pourra démarrer sur cette base.

Toutefois, si on ne peut effectuer cette passe de planification centralisée, le problème reste bloqué. Nous nous sommes attachés à comprendre ce que l'on perd en passant de l'aspect centralisé au discret, en particulier pour les actions communes. C'est la définition même de l'action commune qu'on perd en fait : en voulant que chaque agent planifie uniquement pour lui-même et n'impose rien à son homologue, on s'interdit implicitement de faire figurer dans les espaces d'actions toutes les actions qui nécessitent l'action conjointe des deux agents. Les exemples de ces actions sont pourtant nombreux dans les situations réelles, le cas typique étant illustré par l'exemple des deux robots élévateurs devant soulever une lourde palette : aucun n'est capable de la lever seul (ni d'en soulever un coin) et la seule action qui permette de remplir l'objectif est une action d'équipe, une action commune qui fait lever la palette à un agent virtuel "binôme d'agents". Or si on considère un binôme d'agents qui décide, alors on est revenu dans un cadre de décision centralisée et on ne remplit pas un des premiers objectifs que l'on s'est fixé. Le compromis entre centralisation et décentralisation est généralement trouvé dans les aspects de délibération et/ou de négociation ([JFL+01]). On décide donc d'adjoindre au processus de communication présenté à la section 5.3, un seconde couche qui permet une négociation des agents autour des actions communes.

On modifie alors la formalisation du problème et on inclut, dans les espaces d'actions, des actions communes qui ont le sens suivant :

Définition (Action commune dans les problèmes de décision décentralisée dans l'incertain)**.** *Si a est une action commune, alors elle est présente dans les espaces d'actions des problèmes $\mathcal{P}_1$ et $\mathcal{P}_2$ et elle est alors considérée comme une action individuelle de l'agent ne nécessitant aucune autre ressource que les ressources nécessaires à l'agent pour effectuer sa part de l'action.*

Introduire les actions communes permet alors aux agents d'atteindre de nouveau le même espace d'états que dans le problème biagent centralisé. Cependant, il faut à présent assurer que :

1. chaque action commune planifiée par un agent a bien valeur de proposition et non d'ordre,
2. chaque action commune présente dans le plan final ne rentre pas en conflit avec le reste des plans des agents et avec les transitions effectivement réalisables dans le problème.

Afin de vérifier ces conditions, la seconde couche du protocole de négociation adjoint à tout message d'un agent H vers un agent R, une liste de paires "action commune - intervalle temporel" pendant laquelle le plan de H utilise le second agent comme une ressource. L'agent R reçoit alors le message de l'agent H, intègre comme précédemment les variations temporelles des variables communes dans son modèle d'univers puis cherche une politique contrainte par les actions communes imposées par H. Si aucune solution n'apparait, alors R rejette la contrainte en question et envoie à H un message de refus (on peut raffiner la méthode ici en proposant l'usage de contraintes souples). Dans le cas contraire, R optimise sa politique contrainte et renvoie alors un message similaire à celui qu'il a reçu, portant donc sur les variables communes et les actions communes. On parvient ainsi à générer des solutions nécessitant des actions communes. Comme précédemment, cette méthode est symétrique, on dispose de deux stratégies améliorées alternativement par chaque agent. Lorsque H reçoit un message de refus venant de R, c'est à lui de proposer une nouvelle stratégie avec des nouvelles contraintes sur les actions communes. S'il n'existe pas d'autres combinaisons d'actions communes pour H non plus, alors c'est que H a (ou va) refusé(er) la proposition de R également et donc que l'objectif n'est pas atteignable même avec des actions communes (puisques celles-ci ne sont pas admissibles) et donc que l'objectif n'est pas atteignable dans le problème centralisé non plus.

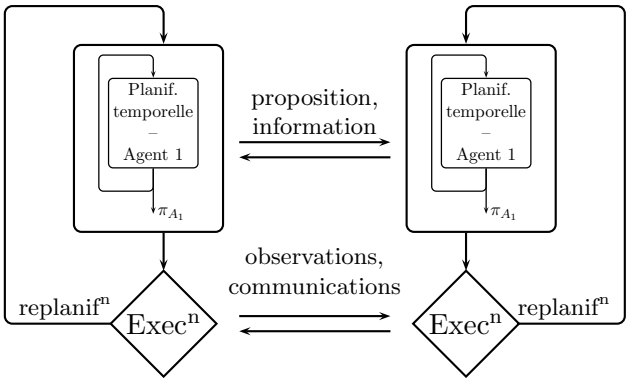

FIGURE 5.6 – La méthode de coordination incluant la seconde couche du protocole de communication

La méthode générale est illustrée figure 5.6.

Afin de pouvoir comparer ces deux stratégies distinctes quand vient le moment d'exécuter, on rajoute à chaque message une valeur de contribution à la valeur de l'état initial de la stratégie globale : les récompenses associées aux actions communes ne sont comptabilisées que par les agents initiateurs de la stratégie, et chacun indique pour combien il contribue ; la stratégie de plus grande valeur est alors choisie et exécutée.

5.5 Conclusion sur l'aspect biagent et ouvertures

On note qu'on ne dispose ni de garantie d'optimalité, ni de critère de terminaison. Cette méthode étant une première proposition, elle n'a pas été explorée dans tous ses détails. Elle a principalement été imaginée afin de définir les caractéristiques du planificateur temporel sur lequel l'attention s'estvite portée. Cependant, la méthode a continué à évoluer en parallèle, notamment par la définition de la seconde couche du protocole de communication.

Afin d'être comparée à l'existant, maintenant que le planificateur temporel dans l'incertain est en cours de réalisation, cette méthode nécessite la confrontation aux solutions proposées dans la littérature. Cela permettra de la confirmer, de l'invalider de la spécialiser ou de la modifier dans son principe et son application.

Conclusion

Pour conclure ce rapport, on peut reprendre la démarche qui a suscité l'évolution des idées présentées précédemment. Partant d'un problème de coordination, on a mis en évidence plusieurs caractéristiques qui ont orienté nos recherches, notamment l'aspect décentralisé, la décision dans l'incertain, la planification temporelle ... On a alors imaginé une première méthode de coordination dérivée de notre problème, et on en a déduit les caractéristiques nécessaires du planificateur monoagent qu'on s'est donné pour tâche de réaliser. On a alors commencé l'étude détaillée à partir du plus bas niveau de notre architecture (appliquant ainsi une sorte de cycle de conception en V) : le planificateur temporel. On a conservé à l'esprit, lors de son développement théorique et pratique, que ce planificateur deva être intégré dans le cadre biagent. Plusieurs possibilités nouvelles d'évolution de la thèse sont nées de cette étude de la planification temporelle sans qu'on se soit lancé dans une recherche particulière de la complexité. Au fur et à mesure que le travail sur l'aspect de planification temporelle avançait, plusieurs raffinements ont été apportés au premier jet de la méthode de coordination biagent. Au final, aujourd'hui le travail est mené sur plusieurs fronts :

- Planification temporelle :
  - implantation et test des méthodes de planification SMDP+ par programmation dynamique et par recherche des dates de décision optimales.
  - preuves d'équivalence des modèles et formalisation d'un tout cohérent intégrant l'action paramétrique d'attente.
  - Extension au traitement d'espaces d'actions continus (similarité avec les travaux en contrôle optimal des processus continus) et aux méthodes approchées de résolution avec espaces d'état, notamment via les techniques de décomposition, factorisation et résolution par programmation linéaire approchée.
- Coordination multiagent dans l'incertain : travail bibliographique de confrontation du premier modèle proposé à des solutions existantes.
- Planificateur final : implantation des solutions retenues, tests et comparaison avec d'autres solutions de la littérature.

L'ensemble de ces directions de recherche est lié par la thématique globale de notre problème applicatif de coordination de notre binôme de pompiers. L'intégration des solutions obtenues et à venir sur un cadre réel de coopération : drone hélicoptère - robot terrestre, satellite - station sol, etc. est un débouché intéressant et très motivant.

FIGURE 5.7 – Une application possible ?

Annexe A

Propriétés des convolutions utilisées

A.1 Cas où f est un polynôme

A.2 Cas où f est un polynôme défini par morceaux

A.2.1 Perte de la régularité du résultat

A.2.2 Forme générale des convolutions de polynômes définis par morceaux

A.2.3 Algorithme de calcul

Calcul du domaine de $h(t) = (f \otimes g)(t)$

Expression de $h(t)$ sur chaque intervalle du domaine

Cas $\int_{\gamma}^{\delta} f(x)g(t-x)dx$

Cas $\int_{\gamma}^{(} t-\delta)f(x)g(t-x)dx$

Cas $\int_{(}^{} t-\gamma)^{(}t-\delta)f(x)g(t-x)dx$

Annexe B

Racines de polynômes

B.1 degré deux, formule du binôme

B.2 degré trois, formule de Cardan

B.3 degré quatre, formule de Ferrari

B.4 degré cinq et plus, méthode de Sturm

Table des figures

Bibliographie

[Alt99] Eitan Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, London, 1999.

[AS93] Eitan Altman and Adam Shwartz. Time-sharing policies for controlled Markov chains. *Operations Research*, 41(6) :1116–1124, nov-dec 1993.

[BDG99] Craig Boutilier, Richard Dearden, and Moises Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1-2) :49–107, 1999.

[Bel57] R.E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[Ber95] Dimitri Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[BL01] J. A. Boyan and M. L. Littman. Exact solutions to time dependent MDPs. *Advances in Neural Information Processing Systems*, 13 :1026–1032, 2001.

[BLZ05] R. Becker, V. Lesser, and S. Zilberstein. Analyzing myopic approaches for multi-agent communication. In *Intelligent Agent Technology, Compiègne, France*, 2005.

[BM04] Aurélie Beynier and Abdel-Illah Mouaddib. Decentralized Markov decision processes for handling temporal and resource constraints in a multiple robot system. In *7th International Symposium on Distributed Autonomous Robotic System*, 2004.

[Bou96] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Theoretical Aspects of Rationality and Knowledge*, pages 195–201, 1996.

[BZI02] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4) :819–840, 2002.

[dFR01] D. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. In *IEEE Transactions on Automatic Control*, 2001.

[DHW94] Denise Draper, Steve Hanks, and Daniel Weld. Probabilistic planning with information gathering and contingent execution. In K. Hammond, editor, *Proceedings of the Second International Conference on AI Planning Systems*, pages 31–36, Menlo Park, California, 1994. American Association for Artificial Intelligence.

[Die98] Thomas G. Dietterich. The MAXQ method for hierarchical reinforcement learning. In *Proc. 15th International Conf. on Machine Learning*, pages 118–126. Morgan Kaufmann, San Francisco, CA, 1998.

[Die00] Thomas G. Dietterich. State abstraction in MAXQ hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems 12*, pages 994–1000, 2000.

[DL95] Thomas Dean and Shieu-Hong Lin. Decomposition techniques for planning in stochastic domains. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, 1995.

[FDMW04] Z. Feng, R. Dearden, N. Meuleau, and R. Washington. Dynamic programming for structured continuous Markov decision problems. In *20th Conference on Uncertainty in Artificial Intelligence*, pages 154–161, 2004.

[GAZ07] C. V. Goldman, M. Allen, and S. Zilberstein. Learning to communicate in a decentralized environment. In *Autonomous Agents and Multi-Agent Systems*, 2007.

[GHK04] Carlos Guestrin, Milos Hauskrecht, and Branislav Kveton. Solving factored MDPs with continuous and discrete variables. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004.

[GKP01] Carlos Guestrin, Daphne Koller, and Ronald Parr. Max-norm projections for factored MDPs. In *International Joint Conference on Artificial Intelligence*, pages 673–682, 2001.

[GZ04] C.V. Goldman and S. Zilberstein. Decentralized control of cooperative systems : Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 2004.

[HK04] M. Hauskrecht and B. Kveton. Linear program approximations for factored continuous-state Markov decision processes. *Advances in Neural Information Processing Systems*, 16 :895–902, 2004.

[HK06] M. Hauskrecht and B. Kveton. Approximate linear programming for solving hybrid factored MDPs. In *9th International Symposium on Artificial Intelligence and Mathematics*, 2006.

[HMK+98] Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Hierarchical solution of Markov decision processes using macro-actions. In *Uncertainty in Artificial Intelligence*, pages 220–229, 1998.

[HSHB00] Jesse Hoey, Robert St.Aubin, Alan Hu, and Craig. Boutilier. Optimal and approximate stochastic planning using decision diagrams. Technical Report TR-2000-05, University of British Columbia - Vancouver, BC, Canada, oct. 2000.

[JFL+01] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation : Prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, 10 :199–215, 2001.

[JHA67] J. L. Walsh J. H. Ahlberg, E. N. Nielson. *The Theory of Spline Functions and Their Applications.* Academic Press, New York, 1967.

[KH06a] B. Kveton and M. Hauskrecht. Learning basis functions in hybrid domains. In *AAAI*, 2006.

[KH06b] B. Kveton and M. Hauskrecht. Solving factored MDPs with exponential-family transition models. In *16th International Conference on Planning and Scheduling*, 2006.

[KHW95] Nicholas Kushmerick, Steve Hanks, and Daniel S. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76(1-2) :239–286, 1995.

[KLC98] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101 :99–134, 1998.

[KP99] Daphne Koller and Ronald Parr. Computing factored value functions for policies in structured MDPs. In *International Joint Conference on Artificial Intelligence*, pages 1332–1339, 1999.

[MBB+05] Mausam, E. Benazera, R. Brafman, N. Meuleau, and E. A. Hansen. Planning with continuous resources in stochastic domains. In *Proc. of the 19th International Joint Conf. on Artificial Intelligence*, pages 1244–1251, 2005.

[ML98] Stephen M. Majercik and Michael L. Littman. MAXPLAN : A new approach to probabilistic planning. In *Artificial Intelligence Planning Systems*, pages 86–93, 1998.

[Mou04] Abdel-Illah Mouaddib. Co-operative scheduling for a resource bounded multiagent planning system. *Journal of Experimental & Theoretical Artificial Intelligence*, 16(2) :57–71, 2004.

[Par98] R. Parr. Flexible decomposition algorithms for weakly coupled Markov decision problems. In *Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, 1998.

[Put94] M. L. Puterman. *Markov Decision Processes*. John Wiley & Sons, Inc, 1994.

[Rac05] E. Rachelson. Coordination multi-robots terrestre et aérien - rapport de M2R. Technical report, ONERA-DCSD Toulouse, 2005.

[RGT+06] E. Rachelson, F. Garcia, F. Teichteil, P. Fabiani, and J.-L. Farges. Une approche du traitement du temps dans le cadre MDP : trois méthodes de découpage de la droite temporelle. In *Journées Françaises Planification, Décision, Apprentissage*, 2006.

[Sab02] Régis Sabbadin. Graph partitioning techniques for Markov decision processes decomposition. In *ECAI*, pages 670–674, 2002.

[SGV06] Matthijs T. J. Spaan, Geoffrey J. Gordon, and Nikos Vlassis. Decentralized planning under uncertainty for teams of communicating agents. In *Proc. of Int. Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 249–256, 2006.

[SP01] D. Schuurmans and R. Patrascu. Direct value-approximation for factored MDPs. In *Proc. NIPS-14*, 2001.

[Stu35] C. Sturm. *Mémoire sur la résolution des équations numériques*. Ins. France Sc. Math. Phys., t. 6, 1835.

[VdWW05] Jeroen M. Valk, Mathijs M. de Weerdt, and Cees Witteveen. Algorithms for coordination in multi-agent planning. In Ioannis Vlahavas and Dimitris Vrakas, editors, *Intelligent Techniques for Planning*, pages 194–224. Idea Group Inc., London, 2005.

[WFL95] Michael Wellman, Matthew Ford, and Kenneth Larson. Path planning under time-dependent uncertainty. In *Proc. of the 11th Conf. on Uncertainty in Artificial Intelligence*, pages 532–539, 1995.

[YS04] Hakan L. S. Younes and Reid G. Simmons. Solving generalized semi-Markov processes using continuous phase-type distributions. In *Proc. of the 19th National Conf. on Artificial Intelligence*, pages 742–747, 2004.