Optimisation en ligne pour la décision distribuée dans l'incertain

Emmanuel Rachelson
ONERA-DCSD, Toulouse
Supaéro, Toulouse
emmanuel.rachelson@onera.fr

Sous la direction de : Fréderick Garcia, Florent Teichteil, Patrick Fabiani, Jean-Loup Farges

Résumé

En partant d'un problème concret de planification de navigation dans un environnement instationnaire et incertain, on s'intéresse à la prise en compte d'une variable continue de "date courante" dans le modèle des Processus Décisionnels de Markov. On propose deux formalismes permettant de définir des politiques dépendant de l'état et de la date courants et les méthodes de résolution associées. Cette étude débouche sur deux perspectives distinctes : une généralisation aux problèmes à espace d'actions continu et une utilisation des méthodes de planification temporelle dans le cadre de la coordination temporelle décentralisée de deux agents coopératifs pour une mission similaire à celle présentée en introduction pour un seul agent.

1 Introduction

Imaginons un robot terrestre devant planifier sa navigation sur les routes d'une forêt en feu pour atteindre un lac où il pourra remplir son réservoir d'eau et partir s'attaquer à l'incendie. Ce problème présente plusieurs caractéristiques qui le différencient d'un problème de planification classique : d'une part l'environnement de l'agent est instationnaire, c'est-à-dire que le modèle que l'agent a de son environnement est sujet à modifications au fur et à mesure de l'exécution. D'autre part, les risques liés à l'exécution d'actions dans un environnement dangereux et hostile impliquent une certaine incertitude sur le résultat de ces actions. Ces deux caractéristiques peuvent être traitées en cherchant, pour notre agent pompier, un plan dans l'incertain avec une dépendance explicite au temps, typiquement "à 8h30, emprunter la route A pour se rendre au lac", "à 9h10, attendre 9h20 puis emprunter la route C", etc. Ce bref exemple illustre les deux idées suivantes. D'une part, le modèle présentant une dépendance explicite à la date courante (à 8h30 le feu a épargné la route A et à 9h10 les deux routes, A et B, sont bloquées), il est nécessaire de construire une stratégie dépendant directement de la date. D'autre part, le résultat des actions étant incertain, il est intéressant de rechercher le plan solution sous forme d'une politique qui à chaque état et à chaque date, associe une action optimale à entreprendre. Un exemple graphique du problème simplifié évoqué ci-dessus est présenté à la figure 1. On s'intéresse donc à la résolution d'un problème

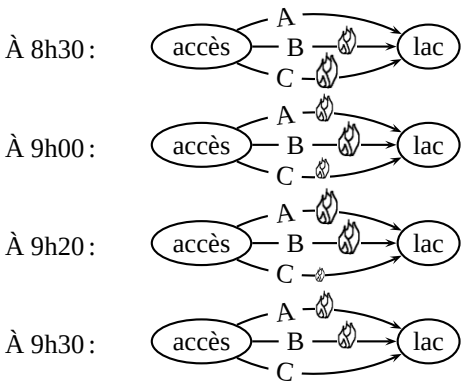

FIGURE 1 – Exemple d'évolution du problème

de décision dans l'incertain évoluant continûment avec le temps, à chaque instant les données du problème ont changé et il faut planifier selon un modèle instationnaire. Comme on recherche une solution dépendante du temps, on étend l'incertitude sur le résultat des actions à une incertitude sur leur durée. L'objectif est donc de considérer les variations du modèle pour trouver les intervalles temporels sur lesquels sont définies les actions optimales à entreprendre dans chaque état discret.

Cet article traite de la résolution de ces problèmes dépendant du temps et de leur insertion dans le cadre plus global de la planification biagents décentralisée.

2 Modèle et formalisation

Les Processus Décisionnels de Markov (PDM) ([Puterman, 1994], [Bellman, 1957]) constituent un modèle particulièrement adapté pour représenter les problèmes de décision séquentielle dans l'incertain. Dans cette section, nous présenterons brièvement les bases des PDM puis nous verrons comment les dépendances au temps s'expriment dans les différents modèles dérivés.

2.1 Le modèle PDM

Un problème représenté sous forme de PDM 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 probabilité à chaque transition (s,a,s'), et d'un modèle

de récompense R associant une valeur réelle à chaque transition. L'action à entreprendre à un moment donné de l'exécution dépend intuitivement de l'état courant, de l'état initial et de la chaîne des actions qui ont mené dans l'état courant. Sous les hypothèses de stationnarité du problème, on peut montrer que cette action ne dépend, pour un problème à horizon infini (on peut effectuer un nombre d'actions illimité), que de l'état courant. On définit ainsi des fonctions (politiques Markoviennes) qui, à chaque état courant, associent une action. L'objectif est alors d'optimiser ces politiques. Pour cela, on se dote d'un critère : on cherche à trouver la politique $\pi : S \rightarrow A$ qui maximise la fonction de valeur :

$$V_\gamma^\pi = E\left(\sum_{\delta=0}^{\infty} \gamma^{t_\delta} r_\delta^\pi | s_0\right) \quad (1)$$

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 $\gamma < 1$. Dans un PDM classique, on considère toutes les durées d'action comme unitaires et on a $t_\delta = \delta$. Ce critère 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 PDM $V^* = LV^*$:

$$V^*(s) = \max_{a \in A}\left(\sum_{s' \in S} P(s'|s,a)\left(R(s',a,s) + \gamma V^*(s')\right)\right) \quad (2)$$

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. On peut notamment résoudre les PDM par programmation dynamique, en construisant la suite des $V_{n+1} = LV_n$, qui converge vers V^*. À partir de cette formulation récursive, on va chercher à prendre en compte les durées incertaines des actions dans le modèle.

2.2 Le modèle PDSM

Le modèle des Processus Décisionnels Semi-Markoviens [Puterman, 1994] intègre la notion de coût de durée d'action. Il enrichit le modèle PDM en transformant le modèle de transition en une fonction $Q(\tau, s'|s,a)$ qui décrit la 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 PDSM considère en fait deux processus distincts, le processus réel qui comporte tous les états temporaires du système et le processus qu'on étudie (le PDSM lui-même). Ces deux processus concordent aux dates de décision. 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 $k(s,a)$ sous la forme :

$$k(s,a) = r(s,a) + \int_0^\infty \sum_{j \in S}\left[\int_0^u \gamma^t c(j,s,a) p(j|t,s,a) dt\right] F(du|s,a) \quad (3)$$

Et l'équation de Bellman devient :

$$V^\pi(s) = k_\pi(s) + \sum_{s' \in S} \int_0^\infty \gamma^\tau \cdot V^\pi(s') \cdot Q_\pi(d\tau, s'|s) \quad (4)$$

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

$$V^*(s) = \max_{a \in A}\{r(s,a) + \sum_{s' \in S} m(s'|s,a) V^*(s')\} \quad (5)$$

On est donc ramené à la résolution d'un PDM classique. On dispose donc avec le modèle PDSM de la possiblité de modéliser le coût de la durée de chaque action. Cependant, on ne peut toujours pas représenter notre problème initial. En effet, c'est le modèle lui-même qui doit dépendre explicitement d'un temps qui soit rendu observable pour qu'on puisse planifier en fonction de lui. Prendre en compte des durées d'action ne suffit pas, on cherche donc des modèles qui étendraient le cadre PDM aux problèmes instationnaires.

2.3 Les modèles avec temps explicite

Plusieurs écueils se présentent lorsqu'on entreprend de construire un modèle de décision dans l'incertain en fonction d'un temps explicite. 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, et la variable "date courante", continue, observable et qui croit 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.

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 PDM à espace d'état continu comme ceux présentés dans [Younes et Simmons, 2004], [Mausam *et al.*, 2005] ou [Guestrin *et al.*, 2004].

Cependant, on souhaite conserver à la variable temporelle sa place à part et mettre en évidence ses spécificités dans l'algorithme de planification. Le dernier écueil que l'on rencontre et qui est pris en compte par les modèles présentés plus bas est 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 PDM 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 4.1 comment cette réflexion peut s'étendre à d'autres variables continues. Les deux modèles suivants proposent une solution à cet écueil de modélisation de nos problèmes instationnaires à temps continu.

2.4 PDSM augmenté et TMDP

Le modèle PDSM augmenté a constitué notre première contribution dans [Rachelson *et al.*, 2006]. On augmente le modèle PDM d'une variable temporelle observable qui s'inscrit dans l'espace d'état et on décrit le modèle sur l'exemple du modèle PDSM, on a alors :

— Un état augmenté $\sigma \in \Sigma$ qui se décompose en :
  – Un espace d'état discrets S
  – Un axe du temps $t \in \mathbb{R}$
— Un espace d'actions discret A
— Une fonction de transition $P(\sigma'|\sigma, a)$ décomposable en $P(\sigma'|\sigma, a) = P(s'|\sigma, a) \cdot F(t'|\sigma, a)$
— Une fonction de récompense $r(\sigma', a, \sigma)$

On cherche alors à résoudre le problème $V^*(\sigma) = \max_\pi V^\pi(\sigma)$ avec :

$$V^\pi(\sigma) = \sum_{s' \in S} \int_0^\infty (r(s', t+\tau, \pi(\sigma), \sigma) + \gamma^\tau V^\pi(\sigma')) \cdot F(\tau|\sigma, \pi(\sigma)) P(s'|\sigma, \pi(\sigma)) d\tau = L^t_\pi(V^\pi)(\sigma) \quad (6)$$

Le second modèle proposé est une amélioration des Time-Dependent MDP (TMDP) introduits par [Boyan et Littman, 2001] et très proches du modèle PDSM augmenté ; il permet toutefois de décrire, en plus, des événements exogènes dont les dates d'occurrence ne dépendent pas de l'état du système. On décrit un TMDP 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.

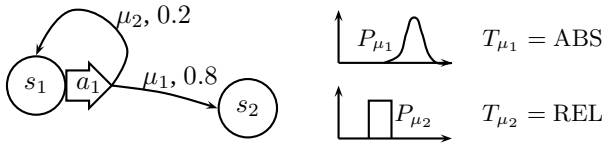

FIGURE 2 – TMDP - éléments de base

La figure 2 illustre la dynamique d'un TMDP. 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}.

On remarque toutefois que ces deux modèles n'intègrent pas d'action “attendre” par défaut (c'est une action implicite décrite par K). La méthode de résolution d'un PDSM par optimisation de l'erreur de Bellman permet de contourner ce problème et de spécifier des politiques au final du type $\pi(s, t)$. De même, l'algorithme de résolution par programmation dynamique des TMDP définit des politiques comme des applications $\pi(s, t) = (t', a)$ où la stratégie consiste à attendre jusqu'à t' puis à entreprendre l'action a.

3 Méthodes de résolution

En partant du problème initial de navigation, formalisé en PDSM augmenté ou en TMDP, on cherche une méthode pour trouver les politiques optimales du type $\pi(s, t)$. On présente ici l'algorithme d'optimisation par l'erreur de Bellman qui est au centre de [Rachelson *et al.*, 2006] et la méthode par programmation dynamique pour résoudre les TMDP présentée dans [Boyan et Littman, 2001] et que nous avons améliorée.

3.1 Optimisation de l'erreur de Bellman

Cet algorithme repose sur l'idée suivante. On va discrétiser l'axe des temps de façon à ne trouver que les dates de décision qui importent. L'amélioration itérative de la politique par optimisation de l'erreur de Bellman est illustrée à la figure 3. Initialement, on considère une variable temps 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 PDM $\tilde{M}$ dont l'espace d'état correspond à l'espace d'état discret du système augmenté de la variable

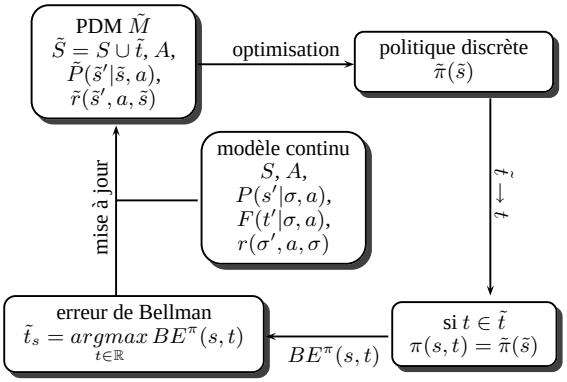

FIGURE 3 – Amélioration itérative de la politique

$\tilde{T}$. Puis on prolonge la politique $\tilde{\pi}$ obtenue sur la variable temporelle continue et on cherche, par état, 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 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.

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 ([Boutilier *et al.*, 1999]). Par ailleurs, son fonctionnement est "anytime" : on dispose à tout moment d'une politique dont la valeur s'accroit avec le temps. La faiblesse de cet algorithme réside dans la difficulté d'évaluer la fonction de valeur 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.

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 PDM $\tilde{M}$. On peut ainsi obtenir des politiques qui préconisent d'attendre dans un intervalle de temps donné, puis d'agir.

Pour les détails concernant l'algorithme et les différentes étapes de la résolution, on renvoie le lecteur à [Rachelson *et al.*, 2006]. Les preuves de convergence de cet algorithme ainsi que son implémentation font partie des objectifs futurs de la thèse, ceux-ci sont à mettre en lumière des résultats obtenus avec l'algorithme suivant.

3.2 Programmation Dynamique en TMDP

On étend l'équation de Bellman au modèle TMDP selon les équations suivantes :

$$V(s,t) = \sup_{t' \geq t} \left(\int_t^{t'} K(s,\theta)d\theta + \overline{V}(s,t') \right) \quad (7)$$

$$\overline{V}(s,t) = \max_{a \in A} Q(s,t,a) \quad (8)$$

$$Q(s,t,a) = \sum_{\mu \in M} L(\mu|s,t,a) \cdot U(\mu,t) \quad (9)$$

$$U(\mu,t) = \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t') + V(s'_\mu,t')]dt' \\ \qquad \text{si } T_\mu = \text{ABS} \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t') + V(s'_\mu,t')]dt' \\ \qquad \text{si } T_\mu = \text{REL} \end{cases} \quad (10)$$

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 :

$$r_\delta^\pi = \int_{t_\delta}^{t'_\pi} K(s_\delta,\theta)d\theta + R(s_\delta, t_\delta, t'_p i) \quad (11)$$

Le modèle TMDP tel que résolu dans [Boyan et Littman, 2001] était limité par les hypothèses suivantes :

- P_μ est une distribution à densité discrète (peigne de Dirac) ce qui ramène les durées à un nombre réduit de durées discrètes possibles.
- L est une fonction constante par morceaux.
- R est somme de fonctions linéaires par morceaux en t, t' et $t' - t$.
- K est une fonction constante par morceaux.

Sous ces hypothèses on peut montrer que la fonction de valeur est linéaire en t et le calcul de ses coefficients est aisé. Nous avons étendu les résultats de [Boyan et Littman, 2001] aux cas plus généraux suivants. De façon générale, on suppose que P_μ, L, R, K sont des fonctions polynomiales de t (et t' et $t' - t$ pour R) définies par morceaux (on considère par abus de notation un peigne de Dirac comme un polynôme de degré -1). Ce choix est motivé par la facilité que l'on a à approcher n'importe quelle fonction par une fonction polynomiale par morceaux. On a alors les résultats suivants :

- On peut effectuer une résolution exacte du problème TMDP par programmation dynamique si :
  - $d^\circ(P_\mu) = -1$ (d° = degré)
  - $d^\circ(L) = 0$
  - $d^\circ(R) \leq 4$

  Alors V un polynôme de degré $d^\circ(R)$.
- Dans le cas où $d^\circ(P_\mu) = -1$, $d^\circ(L) = 0$, $d^\circ(R) \geq 5$ la fonction de valeur est toujours un polynôme de degré

$d^\circ(R)$ et on en obtient une expression approchée grâce au théorème de Sturm ([Sturm, 1835]).

— Dans le cadre général, on peut effectuer une résolution approchée, l'approximation se faisant pour l'évaluation des $\overline{V}$ et V grâce au théorème de Sturm et à des techniques de splines ([J. H. Ahlberg, 1967]).

Par ailleurs, afin de pouvoir modéliser la dynamique du système pendant l'action "attendre", on peut réécrire la première équation de la façon suivante : on note $W(s,t,t') = s'$ la fonction qui décrit la dynamique du système discret si on attend en s entre t et t'. La première équation se réécrit alors :

$$V(s,t) = \underset{t' \geq t}{sup} \left(\int_t^{t'} K(s,\theta)d\theta + \overline{V}(W(s,t,t'),t') \right) \quad (12)$$

Ce dernier raffinement permet de modéliser et de résoudre les problèmes présentant une dynamique de leurs variables lors d'une action d'attente, comme, par exemple, le niveau de carburant de notre robot pompier ou la position orbitale d'un satellite d'observation de la Terre.

D'autres méthodes de résolution sont envisageables pour résoudre ce type de problèmes, notamment par programmation linéaire dans le cadre d'une représentation factorisée de l'espace d'état. A l'heure de rédaction de cet article, cet algorithme est en cours d'implémentation, il repose principalement sur les propriétés de stabilité des polynômes par convolution et par les opération usuelles, et par la capacité d'une fonction polynomiale définie par morceaux à approcher n'importe quelle fonction définie par morceaux.

4 Deux ouvertures pour la suite de la thèse

Le travail effectué sur la variable temporelle met en évidence un type d'action peu abordé dans le cadre PDM : les actions paramétriques (ou les espaces d'action continus). Le passage d'une unique action paramétrique (comme "attendre τ unités de temps") à plusieurs est évoqué à la section 4.1.
Par ailleurs c'est un travail plus global qui a motivé le développement et l'amélioration des deux formalismes présentés précédemment. En effet, le problème général auquel on s'intéresse concerne deux agents, un drone aérien et un robot terrestre, dans une situation similaire à celle présentée en introduction, qui travaillent en équipe et doivent coopérer pour atteindre leur but. Ce type de problème nous pousse à rechercher une solution de type décentralisée ; nous abordons cette problématique dans la section 4.2.

4.1 Actions paramétriques

L'idée de traiter l'action "attendre" comme une action paramétrique "attendre(durée)" nous incite à considérer la possibilité de considérer des actions paramétriques de façon générale comme "avancer(distance)", "tourner(angle)" etc. On définit pour cela le formalisme suivant. On considère un PDM à actions paramétriques ou à espace d'action continu comme :

— un espace d'états S continu ou discret.
— un espace d'actions paramétriques $a_i(x) \in A$ avec le vecteur des paramètres $x \in X$
— un modèle de transition $P(s'|s,a_i,x)$
— un modèle de récompense $r(s,a_i,x)$

Une politique s'écrit alors comme : $\pi : s \mapsto (a_i,x)$ et on cherche à optimiser le critère : $V_\gamma^\pi = E\left(\sum_{\delta=0}^{\infty} \gamma^{t_\delta} r_\delta^\pi | s_0\right)$, avec :

$$r_\delta^\pi = r(s_\delta, \pi(s_\delta), x_\delta^\pi) \quad (13)$$

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

$$Q(s,a_i,x) = r(s,a_i,x) + \int_{s'} \gamma^{\tau(s,s')} V(s') dP(s'|s,a_i,x) \quad (14)$$

$$Q(s,a_i) = \underset{x \in X}{sup}\, Q(s,a_i,x) \quad (15)$$

$$V(s) = \underset{a_i \in A}{max}\, Q(s,a_i) \quad (16)$$

On peut réécrire le problème TMDP dans ce formalisme. On reprend quelque peu les notations pour cela : on a $X = \mathbb{R}$ (le seul paramètre est τ), on note A l'ensemble des actions ne dépendant pas du paramètre, l'espace d'actions est donc constitué de $A \cup \{attendre(\tau)\}$. On distingue l'état complet $\sigma \in \Sigma$ incluant le temps de l'état discret $s \in S$. On a $dP(s',t'|s,t,attendre,\tau) = \delta_s(s')\delta_{t+\tau}(t')dt'$. Enfin, pour simplifier l'écriture, on ne considère que des cas "$T_\mu = ABS$", le cas général étant similaire. Ainsi :

$$\begin{aligned}
V(s) &= \underset{a}{max}\, \underset{\tau \geq 0}{sup} \left\{ r(\sigma,a,\tau) + \int_{\sigma'} \gamma^{\tau(\sigma',\sigma)} V(s') dP(\sigma'|\sigma,a,\tau) \right\} \\
&= \underset{a}{max}\, \underset{\tau \geq 0}{sup} \Big\{ r(\sigma,a,\tau) + \sum_{s'} P(s'|\sigma,a,\tau) \cdot \\
&\qquad \int_{t'} \gamma^{t'-t} P_{s'}(t') V(s',t') dt' \Big\} \\
&= max \Big\{ \underset{a}{max} \left(r(s,t,a) + \sum_{s'} P(s'|\sigma,a) \int_{t'} \dots dt' \right), \\
&\qquad \underset{\tau > 0}{sup} \left(r(s,attendre,\tau) + \gamma^\tau V(s) \right) \Big\} \\
&= \underset{\tau \geq 0}{sup} \Big\{ r(s,attendre,\tau) + \\
&\qquad \gamma^\tau \cdot \underset{a}{max} \left(r(s,t,a) + \sum_{s'} P(s'|s,t,a) \int_{t'} \dots dt' \right) \Big\}
\end{aligned}$$

En prenant $\gamma = 1$ on retrouve l'expression de $V(s,t)$ présentée à l'équation 7.

Les principaux aspects du problème TMDP qui le différencient d'un problème à actions continues standard sont :

— 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 à temps continu qui représente le problème (γ^t)

  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.
— Le domaine de définition de t est potentiellement non-borné, ce qui implique des difficultés de formalisation pour écrire le modèle du problème.

En conclusion, il est intéressant - du point de vue de la planification monoagent - d'aborder les espaces d'action continus pour généraliser le travail déjà effectué. On s'intéresse par ailleurs au fonctionnement en ligne de ces algorithmes dans le cadre d'une planification biagent décentralisée.

4.2 Planification et coordination biagents décentralisée

Imaginons à présent qu'un drone hélicoptère et notre robot pompier coopèrent pour mener la même mission à terme. Notre objectif est maintenant de trouver un plan qui utilise au mieux les capacités des deux agents pour obtenir un résultat plus efficace que dans le cadre monoagent. Il faut donc parvenir à un plan qui permet aux agents de *coopérer* et de *coordonner* leurs actions. Dans la mesure où on cherche à rendre les agents autonomes et où on s'intéresse au fonctionnement distribué du système biagent, on se place dans un cadre de prise de décision décentralisée et on cherche une méthode pour que chaque agent construise sa stratégie propre en dialoguant avec son homologue pour optimiser les gains de la mission. Un protocole de coordination avant planification a été proposé dans [Rachelson, 2005] où chaque agent inscrit sa planification dans une boucle de niveau supérieur d'échange d'informations et de coordination. Les algorithmes précédents permettent aux agents d'effectuer, en plus d'un planification en univers instationnaire, une coordination temporelle.

La méthode de planification décentralisée fonctionne de la manière suivante : chaque agent effectue une première planification compte tenu de son modèle. Il propose alors un certain nombre d'actions communes et informe de l'impact de ses actions sur les variables de description de l'univers communes aux deux agents. L'autre agent intègre ces informations et planifie en fonction en cherchant d'abord à s'accomoder des contraintes dues aux propositions du premier agent puis en les rejetant s'il ne trouve pas de solution, puis il informe et propose en retour. On dispose donc de deux initialisations de stratégies différentes qui sont échangées entre les agents. Cette méthode s'intègre bien dans le cadre d'une planification en ligne par les aspects de coordination temporelle et par l'aspect itératif d'élaboration et de correction de la stratégie commune. Ce fonctionnement est illustré figure 4. Des alternatives à ce fonctionnement issues de la littérature bi- et multiagent ainsi que l'implémentation de cette méthode constituent les perspectives actuelles pour l'aspect biagent.

5 Conclusion

L'implémentation des planificateurs temporels étant en cours, l'étape suivante est d'intégrer ces planificateurs dans un contexte biagent. Les ouvertures évoquées permettront alors d'améliorer les performances et les capacités du système. L'objectif final de porter ces algorithmes de planification sur un vecteur réel sera alors envisageable. Les aspects d'intégration de techniques PDM décomposés et factorisés, d'apprentissage et les comparaisons des modèles entre eux pourront alors constituer une dernière étape d'amélioration.

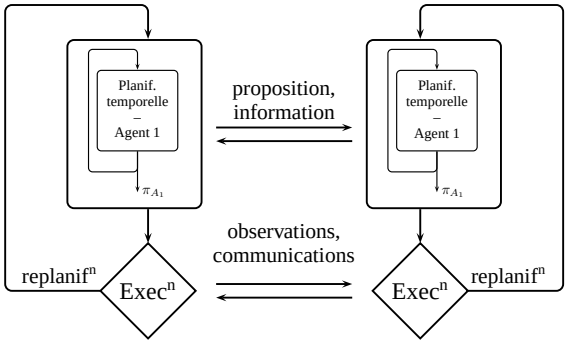

FIGURE 4 – Coordination décentralisée

Références

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

[Boutilier *et al.*, 1999] C. Boutilier, R. Dearden, et M. Goldszmidt. Stochastic dynamic programming with factored representations, 1999.

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

[Guestrin *et al.*, 2004] C. Guestrin, M. Hauskrecht, et B. Kveton. Solving factored MDPs with continuous and discrete variables. Dans *Proc. of the 20th Conf. on Uncertainty in Artificial Intelligence*, 2004.

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

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

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

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

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

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

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