Planification robuste pour la collaboration homme-robot

Yang You, Vincent Thomas, Rachid Alami, Francis Colas, Olivier Buffet

Planification robuste pour la collaboration homme-robot

Yang You[1], Vincent Thomas[1], Francis Colas[1], Rachid Alami[2], Olivier Buffet[1] *

[1] Université de Lorraine, INRIA, CNRS, LORIA, F-54000 Nancy, France
[2] LAAS-CNRS, Université de Toulouse, CNRS, F-31000 Toulouse, France

firstname.lastname@(loria|laas).fr

Résumé

Du point de vue du robot, une difficulté majeure de la collaboration homme-robot est d'être robuste face à des objectifs incertains de l'humain, et aux comportements incertains étant donné un objectif connu. Une question préliminaire clef est alors : Comment dériver des comportements humains réalistes étant donné un objectif connu ? En effet, pour rendre la collaboration possible, de tels comportements devraient aussi tenir compte du comportement du robot, alors que celui-ci n'est justement pas connu. Dans cet article, nous nous appuyons sur des modèles de décision markoviens, représentant l'incertitude sur l'objectif de l'humain par une distribution de probabilité sur un ensemble fini de fonctions de récompense (ce qui induira une distribution sur des comportements humains). Sur cette base, nous proposons deux contributions : 1. un algorithme de planification pour le robot qui est robuste au comportement incertain de l'humain et repose sur la résolution d'un POMDP obtenu en raisonnant sur la distribution sur les comportements humains ; et 2. une approche pour générer automatiquement un comportement humain incertain (une politique) pour chaque fonction de récompense fournie tout en tenant compte du comportement possible du robot. Un scénario de travail collaboratif permet de mener des expérimentations et de présenter des résultats qualitatifs et quantitatifs pour évaluer notre approche.

Mots-clés

Collaboration humain-robot, POMDP, planification robuste.

Abstract

From the robot's point of view, a major issue in human-robot collaboration is how to be robust against uncertain human objectives, and uncertain human behaviors given a known objective. A key preliminary question is then : How to derive realistic human behaviors given a known objective ? Indeed, to allow for collaboration, such behaviors should also account for the robot behavior, while it is not known in the first place. In this paper, we rely on Markov decision models, representing the uncertainty over the human objective as a probability distribution over a finite set of reward functions (what will induce a distribution over human behaviors). Based on this, we propose two contributions : 1. a robot planning algorithm that is robust to the uncertain human behavior and relies on solving a POMDP obtained by reasoning on the distribution over human behaviors ; and 2. an approach to automatically generate an uncertain human behavior (a policy) for each provided reward function while accounting for the possible robot behavior. A co-working scenario allows conducting experiments and presenting qualitative and quantitative results to evaluate our approach.

Keywords

Human-Robot Collaboration, POMDP, Robust Planning.

1 INTRODUCTION

Concevoir des robots intelligents pour assister un partenaire humain est un sujet d'actualité avec des applications dans l'industrie manufacturière [1]-[4], la santé [5], etc. Dans ces applications, le robot a souvent besoin de s'adapter à un objectif de l'humain fixé. Mais pour rendre ce robot assistant robuste, nous devons considérer comment ce robot pourrait s'adapter si l'objectif de l'humain et son comportement induit étaient incertains.

Pour contourner cette incertitude sur les objectifs de l'humain, HADFIELD-MENELL et al. [6] ont proposé le cadre CIRL (*Cooperative Inverse Reinforcement Learning*). Dans CIRL, l'humain et le robot doivent tous deux maximiser la récompense de l'humain. Mais la fonction de récompense de l'humain est paramétrée par une variable cachée au robot, ce qui conduit à un problème multi-agent sous observabilité partielle pour lequel une politique jointe optimale est calculée. CIRL a toutefois deux inconvénients : 1. CIRL calcule une politique jointe (pour l'humain et le robot) et fait l'hypothèse que l'humain doit suivre cette politique calculée. Cette hypothèse est souvent non-réaliste parce que la politique calculée pour l'humain peut être trop complexe pour être exécutée ou apprise par l'humain. 2. CIRL repose sur un environnement MDP avec une seule variable cachée θ correspondant à l'objectif de l'humain (paramètre de la fonction de récompense). Ce cadre assure que la croyance du robot sur θ est une statistique suffisante, faisant de CIRL un POMDP. Cependant, si l'état est partiellement observable, CIRL ne peut être réduit à un POMDP mais à un Dec-POMDP, problème qui est NEXP complet [7].

*This work was supported by the French National Research Agency (ANR) through the "Flying Coworker" Project under Grant 18-CE33-0001.

Dans cet article, nous considérons un humain indépendant, et devons donc tenir compte de l'incertitude sur son comportement, laquelle peut être due i. à l'incertitude sur son objectif (la fonction de récompense qu'il considère), et ii. aux différents comportements (politiques) qu'il pourrait adopter pour un objectif donné. Notre première contribution est ainsi un algorithme de planification pour le robot qui repose sur la résolution d'un POMDP où une des variables d'état cachées correspond au comportement humain actuel, chaque comportement étant représenté par un contrôleur à états fini (FSC). Ensuite, pour dériver des comportements humains réalistes pour un objectif donné, nous souhaiterions que l'humain tienne compte de la possibilité que le robot collabore, ce qui induit un paradoxe similaire au paradoxe de l'œuf et de la poule puisque nous ne disposons pas du comportement du robot à ce stade. Aussi, au lieu de dériver plusieurs comportements humains (FSC) pour un objectif donné, nous allons dériver un unique FSC représentant plusieurs comportements. Pour ce faire, notre seconde contribution consiste en 1. la résolution du problème en faisant l'hypothèse que le robot et l'humain partagent leurs observations (de sorte que le problème est un (M)POMDP, pas un Dec-POMDP), et 2. ensuite l'extraction d'un contrôleur à états fini stochastique (FSC) où une action a d'autant plus de chances d'être choisie qu'elle est prometteuse. En outre, comparé à CIRL, ni le robot ni l'humain n'ont accès à l'état réel de l'environnement.

La section 2 discute des travaux connexes en collaboration homme-robot. La section 3 définit formellement les POMDP, Dec-POMDP et FSC. La section 4 décrit comment concevoir une politique robuste pour le robot afin qu'il s'adapte à plusieurs politiques humaines possibles. La section 5 explique comment générer un FSC stochastique pour l'humain de manière automatique pour une fonction de récompense donnée. Enfin, la section 6 présente des résultats empiriques obtenus sur une tâche de haut-niveau dans un environnement simulé, et les analyse avant de conclure.

2 Travaux connexes

On peut distinguer différentes approches pour la collaboration homme-robot selon le "modèle mental de l'humain" (dont dispose le robot), lequel, d'après TABREZ et al. [8], peut appartenir à l'une des trois catégories suivantes : modèle mental du premier ordre *(first-order mental model - 1oMM)* : le robot considère que l'humain se comporte de manière indépendante et ne tient pas compte des actions possibles du robot ; modèle mental du second ordre *(second-order mental model - 2oMM)* : le robot considère que l'humain tient compte du robot, ce qui induit une forme de modélisation récursive entre humain et robot jusqu'à une certaine profondeur ; modèle mental partagé *(shared-mental model - SMM)* : un SMM suppose que tous les agents dans le groupe ont des attentes communes, donc raisonnent de la même manière, ce qui assure une coordination optimale. Évidemment cette catégorisation s'applique symétriquement à la modélisation du robot par l'humain.

Nous nous concentrons ici sur des problèmes avec des dynamiques stochastiques et une observabilité partielle, ce qui nous conduit à considérer des modèles de décision markoviens. Dans ce contexte, un 1oMM correspond typiquement à un POMDP dont l'objectif est d'optimiser la politique d'un agent en particulier, alors que les politiques (a priori connues) de l'autre agent font partie de la dynamique du système. Par exemple, un POMDP pour le robot supposant un comportement humain connu est résolu dans [9]-[12]. Les SMM peuvent être formalisés comme des *POMDP décentralisés* (Dec-POMDPs), typiquement utilisés pour optimiser la politique jointe d'une équipe d'agents (partageant une même fonction de récompense). CIRL peut être vu comme un cas particulier où l'humain a une observabilité complète et la seule variable cachée du robot correspond à l'objectif réel de l'humain (sa fonction de récompense), ce qui autorise des techniques de résolution dédiées proches de la résolution d'un POMDP. Pour leur part, les 2oMM peuvent être formalisés comme des *POMDP interactifs* (I-POMDPs [13]), dans lesquels les agents se modélisent mutuellement de manière imbriquée. Ainsi, les I-POMDP soulèvent aussi le paradoxe de l'œuf et de la poule qu'il faut résoudre.

Les 1oMM échoueront dans de nombreuses tâches nécessitant un comportement collaboratif explicite de l'humain, et l'hypothèse principale des SMM est typiquement trop forte quand on collabore avec un humain. Nous allons donc doter le robot d'un modèle mental du 2nd ordre de l'humain. Notre approche de la planification robuste pour le robot pourrait être formalisée comme un I-POMDP, ce que nous évitons principalement pour simplifier les notations. [1] Toutefois, les 2oMM soulèvent toujours le paradoxe de l'œuf et de la poule puisque le comportement humain que nous recherchons nécessite de raisonner sur le comportement du robot que nous cherchons justement à dériver au départ. C'est par exemple vrai dans le cas 1. de la spécification manuelle d'une politique humaine [9], [10], où le concepteur humain doit raisonner sur le comportement du robot ; 2. de l'apprentissage d'une politique humaine [14], [15] ou d'une récompense humaine (par apprentissage par renforcement inverse) [16], [17], ce qui requiert un robot collaboratif pour faire la démonstration d'un comportement collaboratif ; ou 3. de la planification d'une politique humaine : le modèle doit inclure le comportement du robot.

Dans notre travail, nous employons une approche par planification pour générer des comportements humains plausibles, et traitons le paradoxe de l'œuf et de la poule en répondant à la question : "Et si l'humain pouvait aussi contrôler le robot ?" : 1. Nous supposons que l'humain peut contrôler le robot et a directement accès à ses observations passées, ce qui revient à faire adopter à l'humain un SMM (modèle mental partagé), et dérivons un problème mono-agent ; 2. en utilisant un solveur en-ligne, nous extrayons une politique humaine sous la forme d'un FSC stochastique.

1. Aussi, nous faisons finalement face à des POMDP, pour lesquels il est préférable d'employer un solveur de l'état de l'art.

3 État de l'art

3.1 Dec-POMDP

Par commodité, nous utilisons les Dec-POMDP pour formaliser le problème de collaboration, mais ne résolvons pas de Dec-POMDP pour obtenir une politique jointe pour l'humain et le robot.

Definition 3.1 *Un* Dec-POMDP *avec $|\mathcal{I}|$ agents est défini par un tuple $M \stackrel{\text{def}}{=} \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \Omega, T, O, R, b_0, \gamma \rangle$, où : $\mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$ est un ensemble d'*agents*; $\mathcal{S}$ est un ensemble d'*états*; $\mathcal{A} = \bigtimes_i \mathcal{A}^i$ est l'ensemble des actions jointes, avec $\mathcal{A}^i$ l'ensemble des* actions *de l'agent i; $\Omega = \bigtimes_i \Omega^i$ est l'ensemble des observations jointes, avec Ω^i l'ensemble des* observations *de i; $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ est la* fonction de transition*, avec $T(s, a, s')$ la probabilité de transiter de s à s' si a est effectuée; $O : \mathcal{A} \times \mathcal{S} \times \Omega \to \mathbb{R}$ est la* fonction d'observation*, avec $O(a, s', o)$ la probabilité d'observer o si a est effectuée et le prochain état est s'; $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ est la* fonction de récompense*, avec $R(s, a)$ la récompense immédiate pour l'exécution de a dans s; b_0 est la* distribution de probabilité initiale *sur les états; et $\gamma \in [0, 1)$ est le* facteur d'atténuation *appliqué aux récompenses futures.*

La *politique* d'action π^i d'un agent i associe à chacun de ses historiques d'action-observation une action à effectuer. L'objectif est alors de trouver une politique jointe $\pi = \langle \pi^1, \ldots, \pi^{|\mathcal{I}|} \rangle$ qui maximise l'espérance du retour atténué à partir de b_0 :

$$V^\pi(b_0) \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \mid S_0 \sim b_0, \pi\right].$$

Dans un POMDP, *c.-à-d.* quand $\mathcal{I} = \{1\}$, de nombreux solveurs reposent sur l'estimation de la fonction de valeur optimale $V^*(b)$, ou la fonction d'action-valeur $Q^*(b, a) \stackrel{\text{def}}{=} R(b, a) + \gamma \sum_o Pr(o|b, a) \cdot V^*(b_a^o)$, où b_a^o est le prochain état de croyance quand on effectue a et que l'on observe o.

Note : Dans notre contexte, nous allons considérer un problème à 2 agents (un humain et un robot) un peu différent puisqu'on remplace l'unique fonction de récompense r par une distribution sur des fonctions de récompense possibles, seul l'humain sachant quelle est la bonne fonction de récompense, et que l'on cherche à concevoir le comportement du robot, mais en raisonnant sur les comportements possibles de l'humain.

3.2 Contrôleurs à états finis

Dans notre travail, les politiques humaines sont représentées sous la forme de *contrôleurs à états finis* (FSC) (aussi appelés *graphes politiques* [18]), *c.-à-d.* des automates qui contiennent dans chaque état interne une distribution de probabilité à partir de laquelle une action est échantillonnée, et dont les transitions d'un état interne au suivant dépendent de l'action exécutée et de l'observation reçue.

Definition 3.2 *Pour des ensembles $\mathcal{A}$ et Ω d'un POMDP, un* FSC *stochastique est défini par un tuple $fsc \stackrel{\text{def}}{=} \langle N, \beta, \eta, \psi \rangle$, où :*

- *N est un ensemble fini de nœuds (internes);*
- *β est une distribution de probabilités à partir de laquelle échantillonner un nœud initial;*
- *$\eta : N \times \mathcal{A} \times \Omega \times N \to \mathbb{R}$, la fonction de transition, donne la probabilité $\eta(n, \langle a, o \rangle, n')$ de transiter du nœud n à n' si a est effectuée et o est observée; une transition déterministe peut être notée $n' = \eta(n, \langle a, o \rangle)$;*
- *$\psi : N \times \mathcal{A} \to \mathbb{R}$, la fonction de sélection d'action, donne la probabilité $\psi(n, a)$ de choisir $a \in \mathcal{A}$ depuis n.*

4 Planification de tâches robuste pour le robot

Nous voulons ici que le robot soit robuste aux différents objectifs (cachés) possibles de l'humain, chacun attaché à des comportements différents, et à même de suivre l'objectif réel de l'humain. Notre problème est formalisé comme un Dec-POMDP, si ce n'est que 1. la fonction de récompense exacte (parmi ρ candidates) n'est connue que de l'humain ; 2. pour chaque fonction de récompense possible r_i, il y a des comportements humains associés (sous la forme d'un FSC fsc_i) ; et 3. le robot connaît la distribution de probabilité sur les paires possibles récompense-FSC : $P(\{(r_1, fsc_1), \ldots, (r_\rho, fsc_\rho)\})$. Comme détaillé ci-dessous, ce comportement de robot robuste est obtenu d'abord en transformant cette distribution sur des FSC en un unique FSC, puis en utilisant ce FSC pour dériver un POMDP, et finalement en résolvant ce POMDP. Une solution candidate pour obtenir un FSC pour chaque fonction de récompense est présentée en section 5.

Pour convertir cette distribution de probabilité sur des FSC humains en un unique FSC, nous prenons "l'union" de ces FSC, la nouvelle distribution sur les nœuds initiaux revenant à 1. échantillonner un FSC fsc_i de la distribution $P(\{fsc_1, \ldots, fsc_\rho\})$, et 2. échantillonner un nœud de la distribution β_i. Plus formellement, en notant un FSC de base $fsc_i \stackrel{\text{def}}{=} \langle N_i, \beta_i, \eta_i, \psi_i \rangle$, le *FSC union* est défini comme :

$$N \stackrel{\text{def}}{=} \bigcup_{i=1}^{\rho} N_i,$$

$$\beta(n_i) \stackrel{\text{def}}{=} \beta_i(n_i) \cdot P(fsc_i),$$

$$\eta(n, \langle a, o \rangle, n') \stackrel{\text{def}}{=} \begin{cases} \eta_{i(n)}(n, \langle a, o \rangle, n') & \text{si } n' \in N_{i(n)}, \\ 0 & \text{sinon,} \end{cases}$$

(où $i(n) \stackrel{\text{def}}{=} i$ t.q. $n \in N_i$, *c.-à-d.*, $i(n)$ est l'identifiant du FSC auquel appartient n au départ), et

$$\psi(n, a) \stackrel{\text{def}}{=} \psi_{i(n)}(n, a).$$

Étant donné ce FSC, nous pouvons maintenant formaliser le problème de décision du robot comme un POMDP dans lequel chaque *état étendu* $e^t \in \mathcal{E}$ contient un état courant du monde s^t, une observation courante du robot o_R^t et le nœud courant n_H^t du FSC union de l'humain : $e^t = \langle s^t, n_H^t, o_R^t \rangle$.

En partant du Dec-POMDP définissant la tâche et du FSC union associé, les fonctions de transition, d'observation et de récompense du problème de décision du robot peuvent s'écrire comme suit :

$$T_e(e^{t+1}, e^t, a_R^t) = Pr(e^{t+1}|e^t, a_R^t)$$
$$= \sum_{a_H^t} \sum_{o_H^{t+1}} T(s^t, \langle a_H^t, a_R^t\rangle, s^{t+1}) \cdot \eta(n_H^t, \langle a_H^t, o_H^{t+1}\rangle, n_H^{t+1}) \cdot$$
$$O(s^{t+1}, \langle a_H^t, a_R^t\rangle, \langle o_H^{t+1}, o_R^{t+1}\rangle) \cdot \psi(n_H^t, a_H^t),$$
$$O_e(e^{t+1}, a_R^t, o_R^{t+1}) = Pr(o_R^{t+1}|e^{t+1}, a_R^t) = \mathbf{1}_{o_R^{t+1}=\tilde{o}_R^{t+1}}$$

(où $\tilde{o}_R^{t+1}$ est l'observation dans e^{t+1}), et

$$r_e(e^t, a_R^t) = \sum_{a_H^t} r_{i(n_H^t)}(s^t, \langle a_H^t, a_R^t\rangle) \cdot \psi(n_H^t, a_H^t).$$

Résoudre ce POMDP associé au robot fournit une politique du robot robuste qui constitue une meilleure réponse à la distribution de probabilité initiale sur les politiques humaines données.

5 Générer des politiques humaines à partir d'objectifs

Nous décrivons maintenant comment générer un FSC stochastique pour l'humain en partant d'une fonction de récompense. Ce processus peut être réglé pour créer des comportements plus rationnels ou plus erratiques.

En effet, nous voulons traiter l'incertitude non seulement sur l'objectif réel de l'humain (sa fonction de récompense), comme dans CIRL, mais aussi concernant le comportement humain exhibé pour atteindre un tel objectif. Pour chaque objectif possible de l'humain, nous créons un Dec-POMDP pour le problème de collaboration. Ces Dec-POMDP ne diffèrent qu'en leurs fonctions de récompense, chacune modélisant un objectif spécifique (de l'humain). Pour chaque tâche associée à un tel Dec-POMDP, nous souhaiterions dériver automatiquement une politique stochastique humaine qui représente des comportements humains possibles guidés par ses propres objectifs, et qui tienne compte des interactions possibles avec le robot. Comme mentionné plus haut, une difficulté est que ces politiques humaines dépendent des politiques du robot dont nous ne disposons justement pas. Pour surmonter ce paradoxe de l'œuf et de la poule, nous supposons d'abord que l'humain peut contrôler les actions du robot et a accès aux observations du robot. Chaque Dec-POMDP est ainsi relâché en un MPOMDP (POMDP multi-agent) [19], *c.-à-d.* un problème qui devient mono-agent. [2] Aussi, pour tenir compte de l'incertitude sur les comportements de l'humain, nous laisserons l'humain faire un choix au hasard parmi plusieurs actions optimales ou quasi-optimales en employant une fonction softmax sur les valeurs d'action : $f(a_H|T, b) = e^{\frac{Q^*(b,a_H)}{T}} / \sum_{a'} e^{\frac{Q^*(b,a')}{T}}$, où $T > 0$ est un paramètre de température qui rend l'humain plus *rationnel* si la température T est basse, seules des actions optimales étant sélectionnées, et plus *erratique* si la température T est élevée, avec des distributions presque uniformes.

Suivant ces idées, nous extrayons des politiques humaines (représentées par des FSC) en utilisant l'algorithme 1, lequel est en partie inspiré par l'extraction de solutions POMDP de GRZEŚ et al. [20]. D'abord, nous calculons une distribution initiale sur les actions de l'humain P_{A_H} avec l'état de croyance initial b_0 (ligne 2). Un nœud de départ unique est créé (l. 3) avec P_{A_H}, b_0 et un poids initial $w = 1$. Ensuite n_0 est ajouté à la fois au nouveau FSC (N) et à une liste ouverte (G) (l. 4). En s'inspirant de LAO* [21], nous souhaitons étendre à chaque itération le nœud qui a la plus grande contribution à la valeur de b_0. Ainsi, tant que G n'est pas vide, nous sélectionnons le nœud $n \in G$ qui a la plus haute valeur estimée $V^*(n.b)$, pondérée par la probabilité d'atteindre ce nœud (estimée à travers $n.w$) (l. 6), puis nous traitons ce nœud avec les paires observation-action possibles sous l'état de croyance courant b (l. 7–9), les paires observation-action impossibles induisant des boucles (l. 22). Les lignes 10–12 calculent l'état de croyance mis à jour b', sa distribution sur les actions P_{A_H}, et son poids w'. Avant de créer un nouveau nœud n' avec ces composants nouvellement calculés, nous devons aussi vérifier : 1. si le nouvel état de croyance b' (ou un état de croyance proche) n'existe pas déjà dans le FSC (N) en considérant un seuil de distance ϵ en norme-1 ; 2. si le FSC ne contient pas déjà le nombre de nœuds maximum $N_{\max}$. Si les deux conditions sont satisfaites (l. 13), nous créons un nouveau nœud n' et l'ajoutons à N. Sinon, nous cherchons et traitons le nœud n' de N dont l'état de croyance est le plus proche du nouvel état de croyance b'. Avec cette approche, nous pouvons extraire des FSC stochastiques de taille bornée encodant plusieurs comportements humains.

Dans nos expérimentations, toutes les valeurs $Q^*(b, a)$ et $V^*(b)$ sont estimées en utilisant l'algorithme POMCP [22], de manière à obtenir de bonnes estimations rapidement, même pour des états de croyance non visités par une politique optimale.

6 Expérimentations

6.1 Cadre expérimental

Nos expérimentations ont été conduites sur un ordinateur doté d'un microprocesseur i9 à 2,3 GHz. Le code source sera rendu disponible sous licence MIT.

Pour tester notre approche, nous avons conçu un scénario présenté figure 1. Dans ce scénario, un robot et un humain doivent réparer et maintenir plusieurs appareils situés dans un monde grille. Parmi les trois appareils présents sur la grille, un appareil doit être maintenu par le robot, qui doit se trouver sur la cellule de cet appareil pour effectuer l'opération ; deux autres appareils sont en panne, chacun requérant que l'humain et le robot accomplissent tous deux simultanément des actions de réparation à l'endroit où se trouve l'appareil. De plus, pour réparer un appareil, l'humain doit détenir un composent qu'il ne peut prendre que dans une

2. Ici, en effet, une seule entité reçoit toutes les observations et contrôle toutes les actions.

Algorithme 1 : Extraction d'une politique humaine sous la forme d'un FSC stochastique

[Input :] T : température softmax | $N_{\max}$: # max de nœuds | ϵ : distance max entre croyances
$P_{A_H} \leftarrow f(\cdot|T, b_0)$ // softmax dist. sur les actions
$n_0 \leftarrow node(\langle P_{A_H}, b_0, w = 1\rangle)$ // nœud de départ
$N \leftarrow \{n_0\}$; $G.pushback(n_0)$
while $|G| > 0$ **do**
  $G.sort()$ // tri des nœuds selon $w \cdot V^*(b)$
  $n = \langle b, P_{A_H}, w\rangle \leftarrow G.popfront()$
  forall $\langle o_h, a_h\rangle \in \Omega_h \times A_h$ **do**
    if $Pr(o_h, a_h|b, P_{A_H}) > 0$ **then**
      $b' \leftarrow BeliefUpdate(b, a_h, o_h)$
      $P'_{A_H} \leftarrow f(\cdot|T, b')$
      $w' \leftarrow w \cdot Pr(o_h, a_h|b, P_{A_H})$
      if $(b' \notin N(\epsilon)) \wedge (N.size() < N_{\max})$ **then**
        $n' \leftarrow node(P'_{A_H}, b', w')$
        $N \leftarrow N \cup \{n'\}$; $G.pushback(n')$
        $\eta(n, \langle a, o\rangle, n') \leftarrow 1$
      else
        $n' \leftarrow N.find(b')$
        $n'.w \leftarrow n'.w + w'$
        $\eta(n, \langle a, o\rangle, n') \leftarrow 1$
    else
      $\eta(n, \langle a, o\rangle, n) \leftarrow 1$
return $\langle N, \eta, \psi\rangle$

boîte à outil à une position spécifique. Notons aussi que l'humain comme le robot n'ont qu'une perception limitée de l'environnement.

Cette tâche est spécifiée à travers le Dec-POMDP suivant :

- États (S) : L'état $s \in S$ du problème est composé de : 1. la position de l'humain, 2. la position du robot, 3. le statut de chaque appareil, et 4. la détention ou non d'un composant par l'humain. Les positions de l'humain et du robot sont représentées par des coordonnées entières (x, y). Ils peuvent se trouver sur la même cellule. Le statut de chaque composant est soit "bon" (réparé), "en panne", ou "nécessitant maintenance".
- Observations de l'humain (Ω_H) : L'humain observe 1. sa position, 2. si le robot est sur la même cellule que lui ou pas, 3. le statut de l'appareil dans sa cellule (s'il y en a un).
- Observations du robot (Ω_R) : Le robot observe 1. sa position, 2. la position de l'humain, 3. le statut de l'appareil dans sa cellule (s'il y en a un).
- Actions et dynamique : L'humain et le robot disposent tous deux des actions suivantes : *Up, Down, Left, Right* (haut, bas, gauche, droite) qui sont les 4 actions de déplacement des agents ; *Attendre* : l'agent reste dans sa position courante ; *Réparer* : répare un appareil en panne si : 1. l'humain détient un nouveau composant ; 2. l'humain et le robot sont sur la même cellule que l'appareil en panne ; 3. l'humain et le robot effectuent tous deux l'action de réparation. En cas de succès, l'appareil en panne devient réparé et le composant de l'humain est consommé.
  L'humain peut *Prendre un composant* s'il se trouve dans la cellule de la boîte à outil et s'il n'en détient pas déjà un. Le robot peut *Maintenir* un appareil tout seul si l'appareil en a besoin et s'ils sont dans la même cellule. En cas de succès, le statut de l'appareil devient *bon*.
- Récompenses R : Une récompense de $+100$ est donnée quand tous les appareils ont été réparés et maintenus. Une récompense (pénalité) de -2 est donnée pour chaque action sauf pour l'action *Attendre* de l'humain, et une pénalité de -20 est donnée en cas d'action invalide. Une pénalité de -1 est donnée si l'humain attend alors que tous les appareils en panne n'ont pas encore été réparés. S'il n'y a plus d'appareils en panne, aucune pénalité n'est donnée en cas d'action *Attendre* de l'humain. La pénalité associée à l'action *Attendre* encourage le robot à repousser les actions de maintenance si cela permet à l'humain de fini les réparations tôt.

Ce grand Dec-POMDP a 2 304 états, 49 actions jointes (7 actions par agent), et 5 400 observations jointes (30 pour l'humain et 180 pour le robot). Note : Cet espace d'état croît de manière quadratique avec le nombre de cellules.

Pour représenter l'incertitude sur les objectifs de l'humain, nous remplaçons la fonction de récompense décrite précédemment par deux variantes. Dans la première, l'humain préfère réparer l'appareil de gauche en premier, recevant une récompense de $+10$ dans ce cas. Dans la seconde, symétriquement, l'humain préfère réparer l'appareil de droite en premier. Ces objectifs génèrent des politiques humaines différentes en utilisant l'algorithme 1. Initialement, le robot ne dispose que d'une distribution a priori sur les récompenses de l'humain et la politique associée (chacune avec probabilité $0,5$). Du fait de la coordination requise pour réparer les appareils, ces politiques humaines doivent tenir compte de l'aptitude du robot à aider l'humain quand c'est nécessaire.

6.2 Résultats qualitatifs

Nous utilisons la procédure décrite en section 5 pour calculer les FSC stochastiques de l'humain associés à chaque objectif de l'humain, et celle décrite en section 4 pour calculer la politique robuste du robot avec le solveur de POMDP SARSOP [23]. Pour économiser des ressources, un *seuil d'action* (ici toujours fixé à $0,1$) est utilisé pour élaguer les actions humaines de basse probabilité lors du calcul des FSC stochastiques de l'humain. Nous avons conduit des expérimentations avec 2 valeurs pour le paramètre de température T ($0,3$ et $0,5$) et 5 valeurs pour la taille maximale des FSC stochastiques $N_{\max}$ (voir table 1).

Nous avons d'abord observé que le FSC stochastique extrait pour l'humain peut effectivement encoder des trajectoires possibles de l'humain permettant de résoudre la

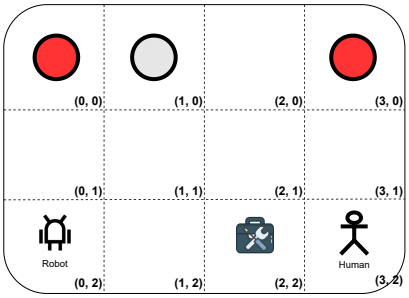

FIGURE 1 – Une tâche de collaboration : Un robot et un humain évoluent dans un monde grille de 4×3. Les cellules en haut à gauche $(0, 0)$ et en haut à droite $(3, 0)$ contiennent toutes deux des appareils en panne. Un appareil à maintenir est aussi situé en cellule $(1, 0)$. Une boîte à outil est située en cellule $(2, 2)$ où l'humain peut prendre des composants.

tâche si $N_{\max}$ est assez grand pour la température T courante. Par exemple, quand $T = 0,3$ et $N_{\max} = 100$, les FSC stochastiques de l'humain atteignent des profondeurs [3] de 22 et 18 (pour chaque objectif), lesquelles sont suffisantes pour que l'humain accomplisse la tâche si le robot collabore. Par contre, quand $N_{\max} = 50$, la profondeur atteinte par le FSC de l'objectif *répare-l'appareil-de-gauche-d'abord* est seulement de 13 (15 pour le 2nd objectif), ce qui ne permet pas à l'humain d'accomplir la tâche (une profondeur de 15 est requise). Nous avons aussi observé que, plus la température T est élevée, plus la valeur de $N_{\max}$ nécessaire pour qu'un FSC stochastique permettant à l'humain d'accomplir la tâche est élevée à son tour. Quand T croît, plus d'actions sont considérées dans chaque nœud, générant plus de branches et empêchant le processus d'expansion d'atteindre la fin de la tâche. Notons aussi que, même si nous fixons T à une valeur très basse (proche de 0), la distribution softmax peut contenir plusieurs actions. Cela permet d'encoder plusieurs politiques humaines optimales dans un unique FSC.

Quand $N_{\max}$ est assez grand pour la température T courante, nous observons que les politiques robustes obtenues pour le robot peuvent résoudre la tâche avec succès, tenant compte des incertitudes sur les objectifs et les comportements de l'humain. Il aide l'humain à réparer tous les appareils et effectue les opérations de maintenance requises. Par exemple, quand $T = 0,3$ et $N_{\max} = 100$, le robot, suivant la politique robuste obtenue, va d'abord à la cellule $(0, 0)$ et attend que l'humain prenne un composant à la cellule $(2, 2)$. Après, si le robot observe que l'humain s'approche, il continue de l'attendre et l'aide ensuite à réparer l'appareil de gauche quand l'humain atteint la cellule $(0, 0)$. Mais s'il observe que l'humain va vers le coin en haut à droite, alors le robot décide d'aller aussi vers la droite pour aider l'humain à réparer l'appareil de droite en premier. Le robot se déplace ensuite jusqu'à la cellule $(1, 0)$ pour effectuer l'opération de maintenance pendant que l'humain va prendre un second composant. Enfin, le robot aide l'humain à réparer l'autre appareil en panne pour finir la tâche.

Même si cela semble être un motif simple pour le robot, il requiert néanmoins que le robot raisonne sur de nombreuses trajectoires possibles de l'humain. S'il y a par exemple 2 actions optimales dans chaque état, même pour 5 pas de temps, il y a 2^5 trajectoires optimales (et ce nombre peut augmenter de manière spectaculaire si l'on considère des actions sous-optimales). De plus, dans ce scénario de collaboration complexe, les agents prennent des décisions en se reposant uniquement sur des observations partielles, et, comme illustré ci-dessus, le robot infère l'objectif de l'humain malgré son comportement incertain.

6.3 Analyse de la robustesse

Nous voulions aussi analyser quantitativement la robustesse de notre approche en comparant la méthode décrite en section 5 avec une méthode calculant la politique d'un robot reposant seulement sur des politiques humaines déterministes. Dans ce but, nous extrayons des paires de politiques humaines déterministes (une par objectif) ; puis, pour chaque paire, nous calculons la politique meilleure réponse du robot à une distribution uniforme sur ces deux politiques humaines. Pour extraire des politiques humaines déterministes, au lieu de garder la distribution softmax complète dans l'algorithme 1, nous n'échantillonnons qu'une action de cette distribution.

Nous avons collecté deux ensembles de paires de politiques humaines déterministes avec des températures de softmax T_{sample} différentes ($0,3$ et $0,5$), chaque ensemble contenant $Nb_{sample} = 50$ paires ($\Pi_H = \{\langle \pi^1_{H,l}, \pi^1_{H,r} \rangle, \ldots, \langle \pi^{50}_{H,l}, \pi^{50}_{H,r} \rangle\}$). Pour chaque paire de politiques humaines $\langle \pi^i_{H,l}, \pi^i_{H,r} \rangle$, nous notons π^i_R la meilleure réponse correspondante du robot à leur mixture (notée $\pi^i_{H,r+l}$). Nous estimons alors :

- la valeur moyenne (sur tous les $i \in [1, 50]$) de la politique du robot π^i_R faisant face à sa paire de politiques humaines correspondante, $V(\langle \pi^i_{H,l}, \pi^i_{H,r} \rangle, \pi^i_R)$;
- la valeur moyenne (sur tous les $i, j \in [1, 50]$) d'une politique de robot faisant face à une paire de politiques humaines, $V(\langle \pi^i_{H,l}, \pi^i_{H,r} \rangle, \pi^j_R)$, laquelle peut ne pas être celle utilisée pour générer la politique du robot (à moins qu'une seule politique optimale n'existe).

Ces valeurs sont ensuite comparées avec les valeurs obtenues, dans les mêmes situations, par des politiques robustes du robot π^*_R générées à l'aide du FSC stochastique de l'humain (comme présenté en section 5).

Les résultats obtenus sont présentés en table 1. Dans la première partie de la table ($T_{sample} = 0,3$, $Nb_{sample} = 50$), nous pouvons observer que les politiques meilleures réponses π^i_R ont les meilleures valeurs moyennes en cas de confrontation avec les paires de politiques humaines déterministes à partir desquelles elles ont été calculées ($19,20$ et $15,90$ pour chaque objectif ; $17,55$ pour l'objectif incertain). C'est attendu puisque les politiques du robot sont faites pour être optimales dans ce cas. Toutefois, quand

3. La profondeur d'un FSC est la distance maximum entre le nœud initial (ici unique) et un nœud quelconque.

ces politiques π^i_R font face à d'autres paires de politiques humaines, le robot échoue la plupart du temps à aider l'humain (et les scores chutent respectivement à $-23,96$, $-22,57$ et $-23,72$), n'ayant pas de comportement approprié quand le comportement de l'humain dévie des attentes du robot. D'un autre côté, même si la politique π^*_R du robot a une valeur plus basse que dans le cas d'une meilleure réponse (π^i_R contre $\langle \pi^i_{H,l}, \pi^i_{H,r} \rangle$), elle est plus robuste aux incertitudes sur les politiques et objectifs de l'humain. Cette robustesse s'améliore même quand $N_{\max}$ croît.

Dans la seconde partie de la table 1 ($T_{sample} = 0,5$, $Nb_{sample} = 50$), les politiques π^i_R manquent encore de robustesse face aux paires de politiques humaines $j \neq i$. Aussi, quand T croît, le comportement de l'humain est plus erratique, et il devient plus difficile pour la politique robuste du robot π^*_R de collaborer avec l'humain, en particulier pour l'objectif 1. Même pour $N_{max} = 600$, la valeur moyenne contre $\pi^j_{H,l}$ est toujours bien plus basse que celle avec l'autre objectif de l'humain ($0,74$ contre $13,05$), alors que π^*_R était meilleure pour le même objectif dans la première partie de la table ($T = 0,3$). Nous supposons que les raisons sont les suivantes : D'abord, quand l'humain est plus erratique (T plus élevée) et suit l'objectif 1 (*réparer-l'appareil-de-gauche-d'abord*), il a plus de chances d'accumuler des pénalités, puisque la trajectoire pour prendre le second composant après la première réparation est plus longue et sujette à plus d'incertitudes. Ensuite, quand l'incertitude sur les comportements humains est élevée et que les deux objectifs sont également probables, le robot peut préférer se concentrer sur l'hypothèse la plus profitable, et décider d'accomplir en priorité l'objectif 2, pour lequel l'humain a moins de chances d'effectuer des actions erratiques.

La table 2 présente les temps de calcul enregistrés à chaque étape du processus. La première étape consiste à convertir chaque modèle de tâche Dec-POMDP en un MPOMDP (un par objectif de l'humain) ; la 2e étape à générer des politiques humaines stochastiques comme présenté en section 5 (processus le plus chronophage) ; la 3e étape en la construction du POMDP pour le robot en utilisant les modèles de tâche (Dec-POMDP) et les politiques stochastiques de l'humain (FSC) ; et l'étape finale en la résolution du POMDP pour le robot à l'aide de SARSOP pour obtenir une politique robuste.

TABLE 1 – Valeurs de diverses politiques du robot face à divers comportements de l'humain. $\pi^j_{H,x}$ représente un comportement humain échantillonné (parmi les paires précalculées) avec priorité à droite ($x = r$) ou à gauche ($x = l$), ou une mixture de ces comportements si $x = r + l$. j est remplacé par i quand le comportement du robot a été conçu précisément pour la même paire échantillonnée.

	$\pi^i_{H,l}$	$\pi^i_{H,r}$	$\pi^i_{H,l+r}$	$\pi^j_{H,l}$	$\pi^j_{H,r}$	$\pi^j_{H,l+r}$
($T_{sample} = 0.3$, $Nb_{sample} = 50$)						
π^i_R	19.20	15.90	17.55	-23.96	-22.57	-23.72
$\pi^*_R(N_{\max}, T)$						
$\pi^*_R(50, 0.3)$				-25.20	5.76	-9.77
$\pi^*_R(100, 0.3)$				14.07	11.60	12.09
$\pi^*_R(150, 0.3)$				17.17	11.60	13.81
($T_{sample} = 0.5$, $Nb_{sample} = 50$)						
π^i_R	12.51	16.47	14.49	-25.48	-25.23	-25.95
$\pi^*_R(N_{\max}, T)$						
$\pi^*_R(150, 0.5)$				-25.77	14.57	-8.34
$\pi^*_R(200, 0.5)$				-23.32	12.97	-11.30
$\pi^*_R(600, 0.5)$				0.74	13.05	6.29

TABLE 2 – Temps CPU (en secondes) pour différentes expérimentations

	$\pi^*_R(N_{\max}, T)$					
Step	$(50, 0.3)$	$(100, 0.3)$	$(150, 0.3)$	$(150, 0.5)$	$(200, 0.5)$	$(600, 0.5)$
Dec-POMDP→MPOMDP	13	13	13	13	13	12
Get Human Stoc. FSCs	130	246	435	487	581	2389
Build Robot POMDP	5	11	17	17	23	203
Solve Robot POMDP	3	6	14	11	16	102
Total time	151	276	479	528	633	2706

7 CONCLUSION

Dans cet article, nous traitons le problème de l'incertitude sur les objectifs de l'humain dans la collaboration homme-robot. La contribution est double : 1. Premièrement, nous proposons un algorithme de planification robuste pour le robot qui repose sur un POMDP avec des incertitudes sur l'objectif réel de l'humain et sur son comportement. Nous détaillons formellement comment concevoir ce POMDP du robot en partant de modèles de tâche (Dec-POMDP) et d'une distribution sur les politiques humaines stochastiques (FSC), une par objectif possible. 2. Deuxièmement, nous discutons du paradoxe de l'œuf et de la poule dans les modèles mentaux du second ordre, et proposons une méthode pour surmonter cet obstacle et générer automatiquement un FSC humain qui contient des comportements humains possibles pour chaque objectif. Des paramètres peuvent être réglés pour ajuster la diversité des comportements humains générés. Notons que l'algorithme de planification robuste pour le robot (1) ne dépend pas de la façon de dériver les politiques de l'humain (2). À travers des expérimentations, nous montrons que notre approche peut s'adapter à des comportements humains incertains avec différents objectifs. Nous pensons que ce travail est important pour les situations de collaboration dans lesquelles le robot et l'humain ont besoin de raisonner sur les actions possibles de l'autre, et où considérer des politiques de l'humain myopes ou déterministes n'est pas suffisant pour générer des politiques de robot robustes. En outre, notre approche ne requiert qu'un Dec-POMDP décrivant la tâche de collaboration humain-robot, et pas de comportement humain donné a priori. Cela rend notre méthode générique pour aborder divers problèmes de collaboration homme-robot, la principale difficulté étant le passage à l'échelle face à de grands problèmes. Dans le futur, nous prévoyons d'étudier comment cette politique robuste du robot se comporte face à de vrais humains dans des environnements simulés, et comment tirer parti de ces simulations pour améliorer le comportement du robot, maintenant que le paradoxe de l'œuf et de la poule a été surmonté. Nous prévoyons aussi d'implémenter notre approche sur une situation réelle où un drone

doit aider un humain à réparer des appareils (dans le cadre du projet Flying Co-Worker).

Références

[1] A. M. ZANCHETTIN, N. M. CERIANI, P. ROCCO, H. DING et B. MATTHIAS, "Safety in human-robot collaborative manufacturing environments : Metrics and control," *IEEE Trans. on Automation Science and Engineering*, t. 13, n° 2, 2016.

[2] K. R. GUERIN, C. LEA, C. PAXTON et G. D. HAGER, "A framework for end-user instruction of a robot assistant for manufacturing," in *ICRA*, 2015.

[3] X. V. WANG, Z. KEMÉNY, J. VÁNCZA et L. WANG, "Human–robot collaborative assembly in cyber-physical production : Classification framework and implementation," *CIRP Annals*, t. 66, n° 1, p. 5-8, 2017.

[4] A. M. ZANCHETTIN et P. ROCCO, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *IROS*, 2013.

[5] M. G. JACOB, Y.-T. LI, G. A. AKINGBA et J. P. WACHS, "Collaboration with a Robotic Scrub Nurse," *Com. ACM*, t. 56, n° 5, p. 68-75, mai 2013.

[6] D. HADFIELD-MENELL, S. J. RUSSELL, P. ABBEEL et A. DRAGAN, "Cooperative Inverse Reinforcement Learning," in *NIPS*, 2016.

[7] D. S. BERNSTEIN, S. ZILBERSTEIN et N. IMMERMAN, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, 2002.

[8] A. TABREZ, M. LUEBBERS et B. HAYES, "A Survey of Mental Modeling Techniques in Human–Robot Teaming," *Current Robotics Reports*, t. 1, déc. 2020.

[9] V. V. UNHELKAR, S. LI et J. A. SHAH, "Decision-Making for Bidirectional Communication in Sequential Human-Robot Collaborative Tasks," in *HRI*, 2020.

[10] S. NIKOLAIDIS, Y. X. ZHU, D. HSU et S. SRINIVASA, "Human-Robot Mutual Adaptation in Shared Autonomy," in *HRI*, 2017.

[11] M. CHEN, S. NIKOLAIDIS, H. SOH, D. HSU et S. SRINIVASA, "Planning with Trust for Human-Robot Collaboration," in *HRI*, 2018.

[12] ——, "Trust-Aware Decision Making for Human-Robot Collaboration : Model Learning and Planning," *J. Hum.-Robot Interact.*, t. 9, n° 2, jan. 2020. DOI : 10.1145/3359616.

[13] P. DOSHI et P. GMYTRASIEWICZ, "A Framework for Sequential Planning in Multi-Agent Settings," *JAIR*, t. 24, juil. 2005.

[14] W. ZHENG, B. WU et H. LIN, "POMDP Model Learning for Human Robot Collaboration," in *CDC*, 2018.

[15] V. V. UNHELKAR et J. A. SHAH, "Learning Models of Sequential Decision-Making with Partial Specification of Agent Behavior," in *AAAI*, 2019.

[16] S. RUSSELL, "Learning Agents for Uncertain Environments (Extended Abstract)," in *COLT*, 1998.

[17] A. Y. NG et S. RUSSELL, "Algorithms for Inverse Reinforcement Learning," in *ICML*, 2000.

[18] N. MEULEAU, K.-E. KIM, L. KAELBLING et A. CASSANDRA, "Solving POMDPs by searching the space of finite policies," in *UAI*, 1999.

[19] D. V. PYNADATH et M. TAMBE, "The Communicative Multiagent Team Decision Problem : Analyzing Teamwork Theories and Models," *JAIR*, t. 16, juin 2002.

[20] M. GRZEŚ, P. POUPART, X. YANG et J. HOEY, "Energy Efficient Execution of POMDP Policies," *IEEE Trans. on Cybernetics*, t. 45, 2015. DOI : 10.1109/TCYB.2014.2375817.

[21] E. A. HANSEN et S. ZILBERSTEIN, "LAO* : A heuristic search algorithm that finds solutions with loops," *Artificial Intelligence*, t. 129, n° 1, p. 35-62, 2001.

[22] D. SILVER et J. VENESS, "Monte-Carlo Planning in Large POMDPs," in *NIPS*, 2010.

[23] H. KURNIAWATI, D. HSU et W. S. LEE, "SARSOP : Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *RSS*, 2008.