Sorbonne Université
Master ANDROIDE

Ethical issues in multi-objective reinforcement learning

Stage de Master 2

Réalisé par :

Marius Le Chapelier

Encadré par :
Aurélie Beynier, Lip6, Sorbonne Université
Nicolas Maudet, LIP6, Sorbonne Université
Paolo Viappiani, LAMSADE, Université Paris Dauphine

Référent :
Tibaut Lust, LIP6, Sorbonne Université

2 septembre 2022

Table des matières

Chapitre 1

Introduction

Le stage s'est déroulé au Lip6, à Sorbonne Université, au sein de l'équipe Systèmes Multi-Agents, il a été supervisé par Aurélie Beynier, Nicolas Maudet et Paolo Viappiani. Ann Nowé, chercheuse du VUB, spécialiste de l'apprentissage par renforcement multi-objectif (MORL), était invitée au Lip6 pendant plusieurs semaines et a également suivi le stage pendant cette période.

Le stage porte sur l'apprentissage par renforcement multi-objectif (MORL), l'objectif est de développer des modèles MORL pour répondre à des problématiques éthiques. Je me suis posé des questions concernant les pistes d'amélioration d'un modèle existant, portant particulièrement sur la partie apprentissage par préférences du modèle.

1.1 Comment intégrer de l'éthique dans l'IA ?

Les systèmes de décisions basés sur l'IA font de plus en plus partie de nos vies, les laisser prendre des décisions importantes (justice, recrutement, voitures autonomes, prêts de banques, etc) à notre place va passer à un moment ou un autre par l'intégration de problématiques éthiques dans les domaines de l'IA.

La question de l'intégration de l'éthique dans l'IA est complexe, car il n'y a pas de définition objective de ce qu'est l'éthique : il existe différentes définitions philosophiques (déontologie, utilitarisme, etc). Comme ces différentes définitions peuvent être contradictoires ou incomparables (au sens d'une agrégation mathématique) [1], on peut se poser la question de comment, dans un problème de décision par exemple, valuer la composante éthique d'une action. La réponse n'est pas triviale et plusieurs approches ont été utilisées, chacune possédant ses avantages et inconvénients.

On a d'abord les approches dites **rule-based ou top-down** [2-6]. Dans ces approches, on inscrit les règles ou contraintes sociales, a priori, et explicitement, lors de la création du modèle, et l'apprentissage se fait dans le cadre de ces règles éthiques. Ces modèles ont les avantages d'assurer le respect des contraintes éthiques et l'explicabilité du comportement de l'agent. Mais le caractère explicite des contraintes les rend peu, voire pas flexibles. Leurs autres inconvénients sont qu'ils nécessitent l'énumération de toutes les contraintes a priori (tâche potentiellement impossible), et pour finir, que les contraintes définies a priori ne prennent pas en compte les différences de définitions ou de valeurs éthiques,

selon la culture, le contexte ou même le temps (le modèle reste le même et ne peut pas évoluer de lui-même).

On a ensuite les approches dites **data-driven ou bottom-up** [7-11], dans lesquelles on considère que les humains agissent en moyenne de manière éthique et donc que leurs choix peuvent être la base d'un apprentissage supervisé.
Ces modèles ont pour avantage d'être flexibles et adaptables, selon les environnements, contextes, cultures, car ils dépendent fortement des données en entrée. Certains peuvent également s'adapter à l'évolution des mœurs de la société en continuant son apprentissage durant son déploiement. Au contraire, dépendre fortement de ses données en entrées, induit de nombreux désavantages, comme la difficulté à expliquer le comportement général et à assurer que l'agent respecte effectivement les valeurs éthiques. Un autre inconvénient est la quantité de données réelles d'humains nécessaires pour obtenir des résultats réalistes. Les données réelles étant souvent collectées avec des questionnaires (Moral Machine, etc), si la quantité de données nécessaires est trop importante, la création du modèle se fait souvent avec des données simulées.

1.2 Utilisation de l'apprentissage par renforcement multi-objectif

La croissance de l'automatisation des tâches dans des domaines critiques grâce à l'IA rend primordial l'assurance que les agents autonomes agissent en suivant des valeurs proches des humains (value-aligned) [1, 9, 10]. Cette recherche d'agents value-aligned a eu pour conséquence la croissance de l'utilisation d'apprentissage par renforcement (RL) dans le domaine de l'IA éthique. En effet, le RL permet d'introduire de l'éthique lors de la modélisation de l'environnement, en introduisant des contraintes éthiques, ou de la modélisation de la fonction de récompense, en y incorporant des variables éthiques (par exemple, l'approche d'ethics-shaping présentée dans [7]). De plus, le RL permet de créer des environnements complexes, proches de la réalité qui peuvent facilement simuler des cadres de prises de décision éthiques.

L'utilisation de plusieurs objectifs lors de l'apprentissage nous permet plus de liberté et de complexité dans la modélisation de la prise de décision d'un agent. D'abord, pour modéliser l'automatisation d'une tâche suivant les valeurs éthiques humaines, plutôt que d'agréger les deux en un seul objectif, il est préférable de considérer d'un côté l'objectif correspondant à la tâche à accomplir, et de l'autre celui correspondant à la dimension éthique de la décision. Ensuite, le choix d'un humain dans une situation donnée, prend en compte plusieurs facteurs éthiques différents et l'utilisation d'un apprentissage multi-objectif nous permet de modéliser chacun indépendamment avec leur propre fonction objectif.

1.3 Direction du stage

Un axe important du stage a été la décision de développer un modèle avec une élicitation de préférence (ou apprentissage par préférences) pour extraire nos données relatives à l'éthique, plus précisément, pour la quantification des objectifs éthiques au niveau des récompenses de notre modèle MORL. Ainsi, la plupart des modèles choisis pour constituer

la base théorique du stage [10, 12-16] et l'ensemble des pistes suivies, pendant ce dernier, vont définir l'éthique en posant des questions à un "expert éthique", le décideur.

Nous faisons ici deux hypothèses, d'abord que ce décideur répond aux questions en considérant effectivement les conséquences éthiques de ses choix, et ensuite que l'on peut construire un modèle qui approxime ses choix en lui posant un nombre réduit de questions.

Au cours de ma recherche bibliographique, j'ai rencontré le modèle "Multi-Objective Reinforcement Active Learning" (MORAL) [10], un modèle MORL, data-driven, qui utilise de l'apprentissage par préférence pour répondre à des problèmes éthiques. De par sa proximité avec le modèle que nous voulions développer, nous avons décidé qu'il constituerait, en grande partie, la base du stage. Une section entière est consacrée à l'explication de ce modèle 3.1.

Assez rapidement, nous avons focalisé le stage sur une amélioration de la partie apprentissage par élicitation de préférences du modèle MORAL, notamment en réfléchissant aux caractéristiques des questions qui nous permettraient de collecter les meilleures informations, et en étudiant différents modèles de sélections de questions.

Chapitre 2

État de l'art

2.1 Positionnement du sujet par rapport à l'existant

2.1.1 Articles sur l'intégration de l'éthique dans l'IA

Pour effectuer un état de l'art, j'ai dans un premier temps lu plusieurs surveys [17, 18] sur l'intégration de l'éthique dans l'IA, présentant les différentes théories représentées et les technologies qui en découlent. Ces surveys classent l'ensemble des travaux à ce sujet en deux grandes catégories : les approches rule-based, et data-driven.

Les premières se basent sur une modélisation explicite des règles ou contraintes éthiques lors de la création du modèle et cela peut se traduire par une définition à la main des valeurs éthiques des actions [2]. Cela peut aussi se traduire par un "contexte éthique" auquel les actions choisies par l'agent doivent répondre [4]. Ou encore par une encapsulation totale du Processus de Décision Markovien (MDP) : on s'assure que les agents agissent de manière éthique en transformant un MDP multi-objectif prenant en compte les considérations éthiques, en un MDP éthique contraint mono-objectif, ne prenant plus en compte les composantes éthiques [6].

Les approches data-driven, quant à elles, se basent sur l'hypothèse que les humains agissent généralement de manière éthique. Collecter un ensemble de choix humains va permettre d'apprendre de manière supervisée à des agents à se comporter en suivant les normes éthiques. Certains modèles modifient directement la fonction de récompense pour y incorporer un terme d'"ethics shaping" prenant en compte les considérations morales de l'agent [7]. D'autres utilisent des algorithmes multi-armed bandit pour orchestrer les actions de l'agent, lui permettant d'agir selon une politique éthique lorsque c'est nécessaire, ou une politique ne prenant pas en compte ces considérations lorsqu'il n'y a pas de risque [8]. Une approche possible est d'apprendre les contraintes éthiques cachées derrière les comportements humains, pour ensuite apprendre à l'agent à remplir sa tâche en prenant en compte ces contraintes [11]. L'approche qui a le plus retenu mon attention est celle de [10], qui se base sur les ensembles de trajectoires de plusieurs agents experts des objectifs éthiques, pour ensuite approximer les fonctions objectifs de chacun d'entre eux, et finalement trouver un compromis entre ces fonctions objectifs en approximant les poids d'une combinaison linéaire avec un algorithme de preference-based learning.

2.1.2 Apprentissage par Renforcement Multi-Objectif (MORL)

Plusieurs modèles rule-based [3, 5, 6] ou data-driven [8, 10] ont choisi une approche Multi-Objectif pour modéliser leur environnement (Multi-Objective MDP). J'ai donc dû me documenter plus précisément sur le sujet, étudier quelles approches étaient les plus adaptées pour le stage. Un des articles que j'ai lu [19] est très complet, présentant les différentes architectures de MORL possibles, les avantages et inconvénients de chacune et discutant de l'évaluation de performance dans le cadre d'algorithmes de résolution de MDP. Par la suite, j'ai été en lien avec Mathieu Reymond, auteur de l'article [20], qui présente une approche MORL qui cherche à approximer le front de Pareto des vecteurs de récompenses possibles. Le réseau de neurones, soumis à l'apprentissage, est entrainé à mémoriser plusieurs comportements différents simultanément, afin de pouvoir reproduire chaque vecteur de récompense du front de Pareto approximé.

2.1.3 Apprentissage par élicitation de préférences (PBL)

Lorsque l'on cherche à approximer des poids ou des fonctions inconnus, comme c'est le cas dans les systèmes de décision éthiques, une piste intéressante est l'apprentissage par préférence. Le système demande sa préférence au décideur parmi deux actions ou entre deux trajectoires, sa réponse nous fournit des informations sur les poids ou fonction que l'on cherche à approximer, et au cours de l'élicitation de préférences, notre estimation devient de plus en plus précise. Selon les hypothèses de départ, une préférence peut retirer toute une partie de l'espace de recherche ou influer sur les probabilités des fonctions ou vecteurs de poids candidats. Les algorithmes PBL sont nombreux et ils divergent souvent dans l'heuristique de sélection des questions, et la façon d'extraire les informations des préférences.

Les hypothèses de départ influent beaucoup sur le modèle, mais de trop fortes hypothèses vont éloigner le modèle de la réalité. Deux hypothèses classiques sont de considérer que le décideur n'est jamais incertain et ne se trompe jamais dans ses préférences. De telles hypothèses ne s'appliquent pas en réalité à un décideur humain, mais permettent au modèle d'extraire davantage d'informations d'une préférence. Considérer que le décideur n'est jamais incertain permet d'induire un ordre total entre les actions ou les trajectoires, mais certains modèles [13] fonctionnent avec des ordres partiels, et permettent donc au décideur d'être indécis. L'heuristique de sélection de question à également un fort impact sur les performances des modèles. L'objectif est souvent de faire une dichotomie de l'espace de recherche, en passant par exemple par une approximation des hyper-espaces des questions potentielles [10], ou par un calcul d'une valeur d'information de chaque question (Approximated Expected Utility of Selection, AEUS) [12]. Certains articles récents [14] prouvent que ces modèles permettent de résoudre des problèmes classiques de Deep RL (DRL) avec des performances proches des algorithmes d'état de l'art du domaine.

2.1.4 IRL et GANs

Parmi les modèles de MORL dans un cadre éthique, les approches dites "data-driven" se basent sur des données en entrées, censées être des données issues de comportements humains (mais souvent simulées). Ces approches ont donc besoin d'un système pour transformer ces données en des outils utilisables par nos agents RL pour résoudre le MDP. Ce que vont faire la plupart des approches [8, 10, 11] c'est essayer d'approximer les contraintes

cachées ou fonctions de récompenses cachées des humains à partir d'ensembles de données de leur choix. On utilise ensuite ces paramètres approximés pour modéliser le MDP et ensuite apprendre à un agent RL à agir dans cet environnement. Les deux méthodes les plus utilisées pour réaliser cette tâche d'approximation de contraintes/fonctions de récompense cachées sont l'Inversed Reinforcement Learning (IRL) et les Generative Adversarial Networks (GANs).

La première consiste à estimer les paramètres d'une fonction de distribution de probabilité (responsable de la génération des données expertes) en résolvant un problème de maximum de vraisemblance (Maximum Likelihood Estimation, MLE) [21-23].

La seconde méthode ressemble à la première, et peut même parfois être équivalente [24, 25], à la différence près que l'on apprend à un autre agent à classifier les données expertes des données générées par le premier agent. Chaque agent apprend en fonction de l'autre, l'agent classifieur cherchant à reconnaître le type de données et l'agent générateur à faire déjouer le premier [22, 26].

2.1.5 Apprentissage par Renforcement Profond (Deep RL)

Bien que le sujet du stage ne porte pas directement sur le deep learning, les méthodes que j'ai rencontrées en réalisant l'état de l'art, notamment celles des approches data-driven (IRL, GANs et PBL), se basent sur des technologies de deep learning [10, 12-14, 23]. On peut également noter que l'article [19], présentant un état de l'art assez complet des modèles et architectures MORL indique que la plupart de ceux-ci utilisent du deep learning. Les approches data-driven étant celles qui ont le plus retenu mon attention, la direction du stage se dirige plutôt vers une utilisation de ces technologies.

2.1.6 Discussion générale sur l'éthique dans l'IA

Avec la croissance de l'IA dans beaucoup de domaines critiques, il est devenu important de comprendre précisément les comportements de nos modèles. La question de l'explicabilité ne pose pas de problème lorsque le modèle définit explicitement les contraintes éthiques (rule-based). Lorsque l'on s'intéresse à une approche basée sur les données, en revanche, l'étude du comportement se fait a posteriori et la compréhension des décisions n'est que partielle, particulièrement lorsque l'on utilise du deep learning [27].

L'article [1], discute de pourquoi raisonner avec des fonctions de coûts (ou de récompense), induisant un ordre total sur les actions, va à l'encontre même de la définition de l'éthique. Il présente certains axiomes des théorèmes de l'impossibilité ou de l'incertitude (proche du théorème d'Arrow pour les votes), et de comment créer un cadre réellement éthique pour notre modèle, notamment en raisonnant avec des ordres partiels plutôt que totaux, ou en introduisant de l'incertitude dans la valuation de nos actions (l'agent peut être incertain).

Chapitre 3

Cadre théorique du modèle

3.1 Multi-Objective Reinforcement Active Learning

Dans cette section, je vais présenter plus en détail le modèle qui constitue la base de mon stage, le modèle "Multi-Objective Reinforcement Active Learning" (MORAL) [10]. L'objectif de ce modèle est d'appliquer à un cadre éthique l'apprentissage par renforcement multi-objectif à partir de données expertes (données humaines) sous deux formes : des batchs de trajectoires, ainsi que des questions sur les préférences.

Pour adresser le problème de l'optimisation multi-objectif, on utilise un Processus de décision markovien partiellement observable (MOMDP), défini par le tuple $\langle \mathcal{S}, \mathcal{A}, p, r, \mu_0, \gamma \rangle$, où $\mathcal{S}$ et $\mathcal{A}$ sont les états et actions du système, p défini les probabilités de transition du système ($p(s'|s,a)$), $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^n$ est une fonction de récompense, μ_0 est la probabilité de l'état initial et γ est un facteur de réduction. Parmi les n objectifs, certains sont relatifs à l'éthique (secourir une personne, éviter des vases, etc.) et d'autres correspondent aux tâches basiques de l'agent (livrer un colis, etc.). On note respectivement $\mathcal{E}$ et $\mathcal{NE}$, les ensembles d'objectifs relatifs à l'éthique et non relatifs à l'éthique, tels que $\mathcal{E} \cap \mathcal{NE} = \emptyset$ et $||\mathcal{E}|| + ||\mathcal{NE}|| = n$. Dans la suite du rapport, les objectifs de $\mathcal{NE}$ sont appelés objectif non éthique par abus de langage. Pour la suite, on considère que $r(\tau_i) = \sum_{(s,a)\in\tau_i} r(s,a)$.

Le modèle est découpé en deux phases : la phase Adversarial IRL (AIRL) et la phase Active MORL (MORAL). Un schéma illustrant les deux phases est disponible ci-dessous 3.1.

La première phase a pour objectif de construire un vecteur de fonctions de récompense censées chacune approximer un comportement éthique distinct.

La première phase prend en entrée n datasets de trajectoires $\mathcal{D} = \{\tau_i\}_{i=1}^N$, chacun correspondant à un expert éthique. L'objectif est, pour chacun des experts, de construire une fonction de récompense f_θ (le discriminant AIRL) et une politique π_ϕ (le générateur AIRL) approximant au mieux le comportement de l'expert. La politique est entraînée en résolvant un problème de maximum de vraisemblance $\max_\phi \mathbb{E}_{\tau\sim\mathcal{D}}[\log p_\theta(\tau)]$. La fonction de récompense est optimisée pour différencier les trajectoires générées par π_ϕ des trajectoires du dataset expert, en utilisant une fonction de loss basée sur l'entropie croisée binaire. On note $\mathcal{F}$ l'ensemble des fonctions de récompenses construites, et Π l'ensemble des politiques d'imitation.

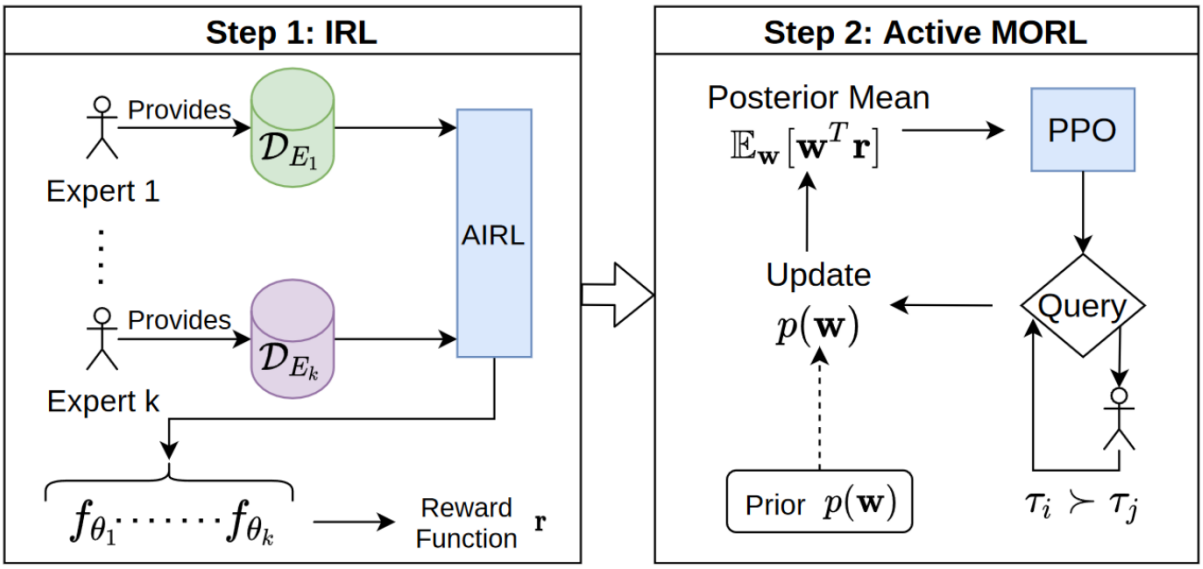

FIGURE 3.1 – Schéma du modèle MORAL [10]

La seconde phase utilise les fonctions de récompense des experts construites durant la première phase, pour modifier les récompenses de l'environnement en un nouveau vecteur de récompenses :

$r_{MORAL}(s,a) = \langle r_i(s,a),\ f_\theta(s,a)\rangle\ \forall i \in \mathcal{NE},\ \forall f_\theta \in \mathcal{F}$, où $\mathcal{NE}$ est l'ensemble des objectifs non éthiques, et $\mathcal{F}$ l'ensemble des fonctions de récompenses de la phase 1. Il est possible que certains objectifs éthiques ne soient pas considérés, ou certaines fonctions soient des agrégations de plusieurs objectifs éthiques, on a alors : $||\mathcal{F}|| < ||\mathcal{NE}||$. Cela induit que $||r_{MORAL}(s,a)|| < ||r_i(s,a)|| = n$. On note donc $n_{MORAL} = ||r_{MORAL}(s,a)||$.

Je vais présenter un exemple pour illustrer les $r_{MORAL}(s,a)$. Les valeurs que je vais utiliser au cours de l'exemple sont réalistes, mais arbitraires. Imaginons que le couple état-action (s,a) correspond à une livraison de colis, et produise donc une récompense objective de $r_i(s,a) = [1,0,0,0]$. Au début du modèle, on nous a fourni deux datasets de trajectoires, correspondant aux comportements de deux experts. On a donc construit deux discriminants AIRL, f_{θ_2} et f_{θ_2}, qui vont remplacer nos 3 objectifs éthiques dans les rewards $r_{MORAL}(s,a)$. Pour cela, on donne le couple (s,a) en entrée de nos deux discriminants AIRL : $f_{\theta_1}(s,a) = 0.3$ et $f_{\theta_2}(s,a) = 0.1$. On forme alors notre vecteur de récompense $r_{MORAL}(s,a) = [r_0(s,a), f_{\theta_1}(s,a), f_{\theta_2}(s,a)] = [1, 0.3, 0.1]$.

L'objectif de la seconde phase est de faire de l'optimisation multi-objectif de ces nouvelles récompenses. Pour cela, on va scalariser le vecteur de récompenses à l'aide d'une fonction d'agrégation.

La fonction d'agrégation de l'optimisation multi-objectif est une somme pondérée f_w, dont les poids sont estimés à partir de questions q_i posées à un décideur, avec un algorithme de type Markov Chain Monte Carlo (MCMC). C'est un apprentissage actif des préférences, c'est-à-dire que les questions sont posées au cours de l'optimisation multi-objectif. Lorsque l'on pose une question q_k au décideur, on lui donne $r(\tau_i)$ et $r(\tau_j)$, les récompenses de deux trajectoires, et il nous donne sa préférence : soit $\tau_i > \tau_j$, soit $\tau_j > \tau_i$. La façon dont le décideur choisit entre les deux vecteurs de récompenses est décrite en annexe E.2. Pour la suite, lorsque l'on évoque une question q_i, cela correspond au couple (τ_i, τ_j) tel que le décideur a jugé que $\tau_i > \tau_j$. On note l'ensemble de question posées $\mathcal{Q}$.

3.2 Explication de l'algorithme MCMC, pour l'estimation des préférences

Entrées *le nombre d'itérations n_iter, le vecteur de poids courant current_w, les questions posées au décideur* $\boldsymbol{q}$;
Résultat : liste des vecteurs de poids acceptés *posterior*

$i = 0$
$posterior = []$;
$posterior\_current\_w = \log(p(current\_w|q_1, ..., q_n))$;
for $i < n\_iter$ **do**
  $new\_w = propose\_w()$;
  $posterior\_new\_w = \log(p(new\_w|q_1, ..., q_n))$;
  if $posterior\_new\_w > posterior\_current\_w$ **then**
    $acceptance\_ratio = 1$;
  else
    $acceptance\_ratio = \exp(posterior\_new\_w - posterior\_current\_w)$;
  end
  if $acceptance\_ratio > random(0, 1)$ **then**
    $posterior.add(new\_w)$;
    $current\_w = new\_w$;
  else
    $posterior.add(current\_w)$;
  end
  $i = i + 1$;
end
return *posterior*;

Algorithme 1 : MCMC

Dans cette section, on va rentrer plus en détails dans l'explication de l'élicitation de préférences. Dans notre modèle, on utilise un algorithme de type Markov Chain Monte Carlo (MCMC) pour l'estimation des préférences du décideur. On va approximer ses préférences par une fonction d'agrégation à poids (somme pondérée ici). Le but de l'algorithme est d'estimer les vecteurs de poids les plus probables dans l'espace de recherche, à partir de n questions q_k posées au décideur.

Cet algorithme peut s'apparenter à une marche aléatoire dans l'espace de recherche, où l'on reste en place pour un temps proportionnel à la probabilité du vecteur de poids correspondant. On fait ensuite la moyenne de tous les poids visités et on obtient un vecteur de poids, paramètre d'une fonction qui approxime les préférences. Formellement, la vraisemblance d'une question, sachant un poids, est donnée par le modèle de Bradley-Terry :

$$p(\tau_i > \tau_j|w) = \left(\frac{\exp(w^T r(\tau_i))}{\exp(w^T r(\tau_i)) + \exp(w^T r(\tau_j))}\right) \tag{3.1}$$

À partir de la formule précédente, on peut calculer la vraisemblance d'un vecteur de poids sachant les questions posées au décideur de manière bayésienne, ou plutôt, son logarithme.

$$\log(p(w|q_1, ..., q_n)) = \log(p(w)) + \sum_{t=1}^{n} \log(p(q_t|w)) \tag{3.2}$$

On ajoute à la somme des vraisemblances des questions, la vraisemblance a priori du vecteur de poids ($\log(p(w))$). Ce terme correspond aux informations que l'on a a priori sur le vecteur de poids, les conditions qu'ils doivent remplir. Dans notre cas, les vecteurs de poids doivent être positifs, et de norme est inférieure à 1. Ces conditions a priori restreignent notre espace de recherche en un sous-espace que l'on va appeler espace a priori, par abus de langage.

Précisément, on va générer un vecteur de poids candidat à chaque pas de temps, la probabilité d'accepter ces nouveaux poids comme poids courants dépend des scores des poids courants et des nouveaux poids. Le score d'un vecteur de poids est donné par la formule (3.2). On retourne la moyenne de tous les poids acceptés.

L'algorithme complet peut être décrit comme ci-dessus 19.

3.3 Deep Learning from Human Preferences (DRLHP)

L'article MORAL fournit une étude comparative de résultats avec le modèle Deep Reinforcement Learning From Human Preferences (DRLHP) [14]. De manière similaire à MORAL, cet algorithme extrait des informations à partir de préférences d'un expert. Mais contrairement à MORAL, les préférences sont les seules informations disponibles. Plus précisément, on utilise les préférences expertes pour mettre à jour un unique réseau de neurones, le preference learner. Ce dernier est ensuite utilisé comme fonction de récompense du MDP. La mise à jour de ce réseau de neurones se fait avec une fonction de loss qui maximise la probabilité que le réseau préfère la bonne trajectoire τ_i pour chaque préférence experte (τ_i, τ_j) du batch.

3.4 Pareto Conditioned Network (PCN)

J'ai pris contact avec un doctorant d'Ann Nowé, Mathieu Reymond, travaillant sur un algorithme MORL, qu'il paraissait intéressant d'étudier dans le cadre du stage. L'algorithme est appelé Pareto Conditionned Network (PCN) [20], et a pour objectif d'approximer le front de Pareto des vecteurs de récompenses d'un environnement, dans un seul réseau de neurones, capable de reproduire les trajectoires qui correspondent à ce front de Pareto. L'algorithme est techniquement similaire à une classification. L'objectif est de classifier les tuples (états, horizon cible, vecteur de récompenses cible) selon les actions. Ainsi, à partir du tuple correspondant à l'état initial et aux horizons et récompenses cibles correspondant à une des trajectoires du front de Pareto, le réseau saura choisir les bonnes actions pour reproduire la trajectoire complète.

Il aurait été intéressant de faire une étude comparative de résultats entre PCN et MORAL, mais le temps m'a manqué pour faire les adaptations nécessaires : pour MORAL, il faudrait rendre fixe l'initialisation de l'environnement (les cases aux mêmes places de la grille), pour permettre la mémoire et reproduction des trajectoires passées. Pour PCN, il faudrait ajouter une phase d'apprentissage des préférences du décideur, pour choisir la trajectoire du front de Pareto qui correspond le mieux à ces préférences.

Chapitre 4

Contributions

Pendant toute la durée du stage, je me suis posé de nombreuses questions sur les pistes d'amélioration possibles concernant le modèle MORAL.

Je vais lister ci-dessous l'ensemble de mes contributions :

1. La première contribution a été l'amélioration de la normalisation des différents objectifs pendant l'optimisation multi-objectif. La normalisation est essentielle avant l'agrégation des objectifs.
2. J'ai suivi plusieurs pistes pour la simplification du modèle MORAL ou l'ajout d'une phase supplémentaire pour l'apprentissage de comportements plus précis que ceux à l'échelle d'une trajectoire complète. Ces pistes m'ont amené à deux nouveaux modèles, AFTER_MORAL 1 et 2.
3. La modification du système d'élicitation de préférences, pour donner la possibilité de poser au décideur des questions sur les actions plutôt que les trajectoires.
4. La contribution précédente m'a permis d'implémenter un nouveau modèle, MORAL_ACTIONS_2 où l'élicitation de préférences est effectuée entièrement avant la phase MORL.
5. J'ai effectué un travail sur les heuristiques de sélection de questions. Dans MORAL est utilisé une heuristique classique de calcul de retrait de volume (volume removal). Nous désirions comparer cette heuristique avec plusieurs autres de l'état de l'art, j'ai suivi plusieurs pistes, dont l'AEUS de l'article [12], qui calcule l'utilité d'une question vis-à-vis des deux réponses possibles du décideur.
6. J'ai également effectué un travail plus empirique d'amélioration de l'algorithme d'approximation des poids à partir des préférences expertes (MCMC) ainsi qu'une recherche de bons hyperparamètres pour ce dernier.
7. J'ai ensuite mis en place un système d'étude de qualité et de convergence de notre élicitation de préférence, pour analyser efficacement les performances de notre modèle en fonction des différents facteurs. J'ai voulu avoir une diversité dans l'étude de préférences (quatre indicateurs de qualité et deux de convergence) pour analyser les performances à plusieurs échelles.

4.1 Normalisation du vecteur de récompenses

Notre modèle fait partie de la catégorie de modèles d'apprentissage multi-objectif qui utilisent une fonction d'agrégation pour se rapporter à un apprentissage mono-objectif. Dans cette catégorie de modèles, l'équilibre des échelles des récompenses de chaque objectif est capitale, car nous avons besoin de rendre comparable les différents objectifs et qu'aucun ne domine les autres. Si les échelles ne sont pas les mêmes, il faut passer par une phase de normalisation avant de les agréger.

Dans les environnements du modèle MORAL, en particulier, les échelles des objectifs éthiques ne sont pas les mêmes que celles des objectifs non éthiques. En effet, pour une trajectoire τ_i donnée, le récompense correspondant à un objectif non éthique $ne \in \mathcal{NE}$, $r_{ne}(\tau_i) \in [0, m]$, où m correspond au nombre de cellules correspondant à ne (12 cellules dans l'environnement 3). Contrairement à la récompense de l'objectif éthique $e \in \mathcal{E}$, $r_e(\tau_i) = \sum_{(s,a)\in\tau_i} f_{\theta_e}(s, a)$. $r_e(\tau_i)$ correspond à une somme de sorties de réseau de neurones et ne peut donc pas être borné théoriquement (cela dépend de l'apprentissage du réseau). Empiriquement, on observe qu'après la phase AIRL, $r_e(\tau_i) \in [-250, 100]$. On observe que les échelles $[0, 12]$ et $[-250, 100]$ ne sont pas comparables, et agréger les récompenses telles quelles n'aurait pas de sens, j'ai donc cherché plusieurs techniques pour les normaliser.

Dans MORAL, il n'y avait qu'une normalisation partielle des objectifs non éthiques, et pas de normalisation de l'objectif éthique. J'ai travaillé à améliorer le système avec plusieurs normalisations pour objectifs éthiques et non éthiques (respectivement 7 et 6). Elles fonctionnent de la manière suivante : Ajout au modèle MORAL d'une phase de pré-traitement durant laquelle sont calculées empiriquement des bornes inférieures et supérieures pour chacun des n objectifs. Ce sont les bornes supérieures et inférieures qui différencient les normalisations.

formule de la normalisation des n_{MORAL} objectifs du vecteur de récompense d'une trajectoire $r_{MORAL}(\tau_i)$:

$$r_{MORAL\_normalized}(\tau_i) = \left[\frac{r_{MORAL_k}(\tau_i) - lower\_bound_k}{upper\_bound_k - lower\_bound_k}\right]_{k\in[0,n_{MORAL}]} \tag{4.1}$$

Il y a, pour les normalisations, deux visions différentes possibles : normaliser la valeur des récompenses des trajectoires ou des actions. Dans le premier cas, l'objectif est qu'après avoir normalisé les récompenses de chacun des objectifs d'une trajectoire, chaque récompense soit comprise entre 0 et 1. $r_{MORAL\_normalized}(\tau_i) \in [0, 1]^{n_{MORAL}}$. Dans le second, la récompense de chaque action doit être comprise entre 0 et 1. $\forall(s, a) \in \tau_i, r_{MORAL\_normalized}((s, a)) \in [0, 1]^{n_{MORAL}}$. Pour vérifier ces caractéristiques, nous avons fait l'hypothèse que l'apprentissage MORL fait des concessions entre les objectifs et ne peut donc pas performer mieux sur un objectif que lorsqu'il apprend à n'optimiser que celui-ci. Nous nous sommes donc servi d'agents experts pour calculer les bornes supérieures des objectifs.

Pour certaines normalisations des objectifs éthiques, les bornes correspondent aux évaluations min/max de trajectoires issues de la politique du générateur AIRL (agent expert AIRL, section 3.1). Dans d'autres, on va plutôt calculer la moyenne de ces évaluations. On peut également calculer la borne inférieure avec un agent aléatoire, plutôt que l'évaluation minimum du générateur AIRL.

Pour les normalisations de l'objectif non éthique, la borne inférieure étant logiquement 0, il nous suffit de trouver une bonne borne supérieure. J'ai au départ pris la moyenne des récompenses des trajectoires d'un expert ("l'expert non éthique") n'ayant appris à ne satisfaire que l'objectif non éthique. Mais cet objectif étant souvent favorisé lors de l'apprentissage, j'ai essayé plusieurs normalisations pour le normaliser plus "sévèrement" (plus proche d'une valeur faible), en ajoutant un facteur au dénominateur, ou en prenant la récompense maximale parmi les trajectoires de l'expert, plutôt que la moyenne (borne supérieure plus importante).

Finalement, j'ai choisi de conserver une normalisation vis-à-vis de récompenses de trajectoires complètes, plutôt que de récompenses d'actions. Avec l'hypothèse faite en début de section, et pour favoriser la robustesse des valeurs, j'ai décidé pour la borne supérieure de calculer l'évaluation moyenne d'un agent ayant appris à optimiser uniquement cet objectif (le générateur AIRL pour un objectif éthique, l'agent expert non éthique pour l'objectif non éthique). Pour La borne inférieure des objectifs éthiques, j'ai choisi de considérer que le pire cas était une politique aléatoire. Elle correspond donc à l'évaluation moyenne d'une trajectoire produite par un agent avec une politique aléatoire.

Les normalisations que j'ai retenues permettent de mettre sur un pied d'égalité l'objectif avec les objectifs non éthiques, ce qui n'était pas le cas dans le modèle MORAL de base.

4.2 Modèles AFTER MORAL 1 et 2

Le modèle MORAL étant assez complexe, nous avions pour projet d'adapter (et possiblement complexifier) la partie élicitation de préférence tout en simplifiant la partie AIRL. Comme l'article se compare au modèle DRLHP, qui est une autre approche MORL, beaucoup plus simple que MORAL, je m'en suis inspiré pour créer deux nouveaux modèles.

Le premier modèle, AFTER_MORAL_1 est un modèle censé remplacer MORAL ou DRLHP, il reprend exactement le fonctionnement de DRLHP, en le simplifiant légèrement : Le preference learner qui donne les récompenses des actions, prend en entrée les vecteurs de récompenses plutôt que les couples états actions (baisse de complexité des entrées). Ce modèle n'a pas suffisamment bien fonctionné pour que l'on continue à le développer.

Le second modèle, AFTER_MORAL_2, est plus complexe et ne répond pas exactement aux mêmes besoins. Avant de développer les préférences sur les actions (section 4.3), nous pensions déjà ajouter une phase après MORAL, dont le but serait de développer des comportements plus "précis" que ceux à l'échelle d'une trajectoire. Ce modèle poserait des questions sur des actions ou des ensembles d'actions plus petits que les trajectoires. L'idée de AFTER_MORAL_2 est de remplacer la fonction d'agrégation de MORAL par un réseau de neurones, le preference learner, de DRLHP. Ce réseau de neurones prenant en entrée des actions, il répond à nos besoins. Le modèle se sert des discriminants AIRL calculés pendant la phase AIRL de MORAL pour former le vecteur de récompense airl, qui est donné en entrée du preference learner à la place du vecteur objectif de récompenses. Le preference learner est entrainé lors d'une première phase à partir des préférences expertes, puis utilisé pour donner la récompense des actions lors de la phase MORL (il remplace la somme pondérée comme fonction d'agrégation MORL).

J'ai testé ce deuxième algorithme et obtenu des résultats concluants (certaines solutions dominait même au sens de Pareto ce que j'obtenais avec MORAL). J'ai ensuite testé de lancer une exécution juste après la phase AIRL (c'est-à-dire à partir d'un agent aléatoire) et j'ai obtenu des résultats proches des exécutions où je partais d'un agent expert MORAL. Cela indique que le modèle AFTER_MORAL_2 remplace la deuxième phase de MORAL plutôt qu'elle la complète pour apprendre des comportements plus précis. De plus, l'utilisation d'un réseau de neurones supplémentaire comme fonction d'agrégation rendait l'explicabilité de l'optimisation multi-objective très compliquée. Nous avons donc décidé de ne pas poursuivre le développement de ce modèle.

4.3 Questions sur les actions plutôt que les trajectoires

Durant la phase MORAL, présentée dans la section 3.1, l'élicitation de préférences sert à équilibrer l'optimisation et l'orienter pour qu'elle corresponde aux préférences du décideur.

Une direction intéressante que j'ai voulue étudier est d'estimer les poids cachés du décideur à partir de questions sur les actions et non sur les trajectoires. Des questions sur les actions pourraient permettre intuitivement de mieux différencier la valeur de chaque objectif aux yeux du décideur. En effet, les récompenses des actions étant des vecteurs unitaires ou nuls, les différences sur les objectifs seront donc plus marquées que pour les trajectoires. Par exemple, avoir la préférence du décideur entre deux trajectoires $r(\tau_i) = [4, 6, 5, -1] > r(\tau_j) = [5, 5, 5, -1]$, nous donnera probablement une information moins marquée que la préférence $r((s, a)) = [0, 1, 0, 0] > r((s', a')) = [1, 0, 0, 0]$.

De plus, on retire la dimension "qualité globale" de la trajectoire. Il est probable que si l'on demande à un humain ses préférences, il considère la qualité globale de la trajectoire. Par exemple, soit deux trajectoires τ_i et τ_j, de récompenses $r(\tau_i) = [1, 8, 1, -2]$ et $r(\tau_j) = [6, 7, 5, 0]$, même si le décideur accorde plus d'importance à l'objectif 1, il est possible qu'il préfère τ_j, car sa qualité sur les autres objectifs est bien meilleure que τ_i. Cette dimension de "qualité globale" risque de réduire la précision de notre estimation de la valuation de chaque objectif aux yeux du décideur. La maximisation de la qualité globale étant déjà gérée avec la phase MORL, il n'est pas nécessaire de la prendre en compte ici. Les récompenses des actions étant des vecteurs unitaires, on élimine logiquement ce problème.

Les résultats de ce changement de modèle sont plutôt concluants, on obtient même parfois de meilleurs résultats qu'avec des préférences sur les trajectoires. Mais ce changement n'a pu fonctionner qu'avec une nouvelle heuristique de questions et l'ajout de température dans la formule de vraisemblance (sections 4.5 et 4.6). J'ai mis en annexe (chapitre A) des graphiques qui montrent une comparaison de 3 exécutions, une avec des préférences sur les trajectoires, une sur les actions, et une avec les paramètres présents dans la version de base de MORAL.

4.4 Nouveau modèle, MORAL_ACTIONS_2

Après avoir testé de poser des questions sur les actions plutôt que sur les trajectoires, j'ai constaté qu'une vingtaine de questions étaient nécessaires pour obtenir une bonne

estimation des préférences du décideur. Les 50 questions sont posées à intervalles réguliers pendant la phase MORL, la moitié de la phase est donc exécutée avec une mauvaise estimation des préférences. J'ai alors décidé de travailler sur un nouveau modèle, **MORAL_ACTIONS_2**, qui ferait la phase complète d'élicitation (sur des actions), avant la phase MORL. Cela permettrait de commencer directement avec une bonne estimation des préférences du décideur.

En effet, il est facile d'adapter l'élicitation des préférences sur les actions pour poser toutes les questions avant l'apprentissage. Car contrairement aux trajectoires, les récompenses des actions ne dépendent pas de la politique de l'agent. On peut donc, avec un batch d'actions suffisamment important, poser toutes les questions avant la phase de MORL.

Empiriquement, on a constaté que ce modèle fonctionne très bien et présente une convergence plus rapide que la version classique avec les préférences sur les actions. Ce qui peut être compris logiquement, car le vecteur de poids présente dès le départ une préférence marquée pour certains objectifs. Cependant, mise-à-part la vitesse de convergence, les résultats obtenus sont similaires, j'ai donc décidé de ne pas inclure de graphique dans le rapport.

On pourrait également adapter le modèle avec des préférences sur les trajectoires. Cela pourrait nous permettre, en comparaison avec MORAL, de sélectionner les questions dans un batch ayant une plus grande diversité que celui constitué uniquement par les trajectoires de la politique actuelle de l'agent. Cela pourrait nous amener à de meilleurs résultats sur la qualité globale de notre élicitation de préférences. Un inconvénient important est qu'il faudrait disposer d'un ensemble de trajectoires d'experts résultants du MORL, pour avoir un batch de trajectoires de qualité.

4.5 Sélection des questions

La vitesse de convergence ainsi que la qualité des solutions retournées par un système d'élicitation de préférences dépends fortement de la façon dont on sélectionne les questions. L'heuristique utilisée dans le modèle MORAL est basée sur le calcul, pour chaque question, du volume retiré de l'espace a priori (prior volume removal). Cette heuristique calcule comment un couple (τ_i, τ_j) découpe l'espace de recherche des poids. Soit $q_k^+ \Leftrightarrow \tau_i > \tau_j$ et $q_k^- \Leftrightarrow \tau_i < \tau_j$, pour faire une dichotomie de l'espace, le couple doit vérifier $\sum_{w_i \in w_{posterior}} p(q_k^+|w_i) = \sum_{w_i \in w_{posterior}} p(q_k^-|w_i)$. Dans le modèle de base, la formule de la vraisemblance a été modifiée pour fonctionner avec un delta entre les vecteurs de récompenses des deux trajectoires. Je ne rentrerai pas dans les détails, mais on appellera cette heuristique **delta_loglik**. On note w^+ et w^-, les vecteurs de poids qui résultent d'un run MCMC avec comme ensemble de questions posées respectivement $\mathcal{Q} \cup q_k^+$ et $\mathcal{Q} \cup q_k^-$.

J'ai ajouté au modèle plusieurs heuristiques de sélections de question, dans le but de faire une étude comparative de vitesse de convergence et de qualité de solution.

En enrichissant la formule de la vraisemblance avec une température (section 4.6), j'ai dû modifier le modèle pour pouvoir utiliser la formule de base de la vraisemblance et non la formule modifiée utilisant des deltas de récompenses. J'ai donc ajouté au modèle l'heuristique de sélection se basant sur le volume retiré, avec la formule de base de la vraisemblance, **basic_loglik**.

Je me suis basé sur l'article [12] pour ajouter au modèle une heuristique de sélection se

basant sur l'Expected Utility of Selection (EUS). L'idée ici, pour un couple (τ_i, τ_j), est de calculer les probabilités de w^+ et w^-, les poids résultants des runs de MCMC si l'on ajoute respectivement $q_k^+ = \tau_i > \tau_j$, ou $q_k^- = \tau_i < \tau_j$ au batch de préférences $\mathcal{Q}$.

$$EUS((\tau_i, \tau_j)) = p(w^+ \mid \mathcal{Q} \cup q_k^+) + p(w^-) \mid \mathcal{Q} \cup q_k^-) \tag{4.2}$$

Le système n'est pas exactement similaire, les vecteurs de poids des deux préférences possibles $w_{(\tau_i > \tau_j)}$ et $w_{(\tau_i < \tau_j)}$, sont estimés avec MCMC au lieu d'une résolution d'un problème quadratique comme dans l'article [12]. L'heuristique est appelée **EUS**.

Les questions sur les actions plutôt que sur les trajectoires ont impliqué plusieurs ajouts ou modifications du modèle, dont une nouvelle sélection de question. En effet, l'environnement étant lacunaire, la plupart des actions ont pour récompense un vecteur nul (les déplacements et actions dans le vide). Cela implique qu'avec une sélection aléatoire, la plupart des questions posées seront entre deux vecteurs nuls, et cela n'apportera pas d'information ou des fausses informations qui ne permettront pas de trouver les préférences du décideur. J'ai ajouté au modèle une heuristique de sélection qui ne peut pas poser une question où les deux actions ont le même vecteur de récompense ($r(s_i, a_i) = r(s_j, a_j)$), et qui a 90% de chances de poser une question sans vecteur nul ($r(s_i, a_i) \neq 0, r(s_j, a_j) \neq 0$). Cette heuristique est appelée **no_double_less_zeros**.

Pour comparer les différentes heuristiques de sélections, j'ai implémenté une heuristique qui sélectionne deux trajectoires aléatoires du batch courant, **random**.

Ce que j'ai constaté empiriquement est que la meilleure heuristique pour les préférences sur les actions était "selection_no_double_less_zeros", le plus important étant donc de comparer des actions avec des vecteurs de récompenses différents et non nuls. Pour les questions sur les trajectoires, l'heuristique "basic_loglik" est la meilleure (peu de vecteurs nuls). J'ai constaté également que l'heuristique de sélection était plus importante avec des préférences sur les actions que sur les trajectoires. L'étude complète est disponible en annexe (chapitre C).

4.6 Études des hyperparamètres du PBL

Dans cette section, je vais présenter le travail empirique d'amélioration du modèle d'élicitation de préférences que j'ai effectué. Afin de faire une présentation précise, je vais lister les modifications que j'ai effectuées au cours de mon étude. Dans certains cas, ces modifications sont simplement des changements de paramètres (nombre de questions, de candidats, etc), dans d'autres, ce sont différentes fonctions qui sont utilisées (gestion des poids en dehors de l'espace a priori, température de la vraisemblance, etc).

Ces paramètres impactent fortement la qualité des résultats obtenus et les meilleurs paramètres ne sont pas nécessairement les mêmes lorsque l'on utilise des préférences sur les trajectoires ou sur les actions.

1. Nombre de questions : la précision de notre estimation des préférences du décideur dépends logiquement de notre quantité d'information, c'est-à-dire du nombre de questions posées au décideur. Mais le cadre applicatif du modèle (poser ces questions à de vraies personnes sous forme de formulaires) implique un nombre réduit de

questions (moins de 100). Pour nos expériences, nous avons choisi de fixer le nombre de questions à 50 de modifier les autres hyperparamètres en fonction de ces 50 questions.

2. Nombre de poids candidats au cours du MCMC : J'ai présenté l'algorithme MCMC dans la section 3.2, ses résultats dépendent logiquement du nombre de poids candidats générés. On veut avoir un nombre suffisant de poids générés pour que le processus ait le temps de trouver un état stable, et que la moyenne soit représentative. J'ai par la suite fixé 10000 poids candidats.
3. Covariance des poids candidats au cours de MCMC : J'ai évoqué dans une section précédente que l'algorithme MCMC fonctionnait comme une marche aléatoire dans l'espace de recherche. On peut continuer l'analogie et parler de la taille des pas pour la covariance. En effet, la génération des poids fonctionnant avec une loi normale, la covariance influe sur la proximité entre le poids courant et les poids générés. J'ai par la suite fixé la covariance à 0,01.
4. Gestion des poids en dehors de l'espace a priori (façon de générer les w_new et/ou de calculer le prior) : J'ai évoqué l'espace a priori dans la section 3.2. Il y a plusieurs manières de gérer les poids qui sont en dehors de cet espace. On peut choisir de n'en proposer aucun comme candidat, ou de leur infliger une pénalité. Ces différentes manières de gérer l'espace a priori ont un impact sur le résultat du MCMC. Pour la plupart de nos tests, on ne propose que des poids présents dans l'espace a priori.
5. Température de la vraisemblance : La formule de la vraisemblance peut être enrichie, en ajoutant une "température", notée $\mathbb{T}$.

$$p(\tau_i > \tau_j|w) = \left(\frac{\exp(\mathbb{T}w^T r(\tau_i))}{\exp(\mathbb{T}w^T r(\tau_i)) + \exp(\mathbb{T}w^T r(\tau_j))}\right) \tag{4.3}$$

   Cette température peut renforcer ou diminuer l'impact de la préférence de notre vecteur de poids.
   Pour chaque question $q_k = (\tau_i, \tau_j)$ posée au décideur, on va calculer $p(\tau_i > \tau_j|w)$ et $p(\tau_j > \tau_i|w)$.
   Si $\mathbb{T} \to 0$ alors $p(\tau_i > \tau_j|w) - p(\tau_j > \tau_i|w) \to 0$, (on réduit l'impact de la préférence du vecteur de poids).
   Si $\mathbb{T} \to +\infty$ alors $\max(p(\tau_i > \tau_j|w), p(\tau_j > \tau_i|w)) - \min(p(\tau_i > \tau_j|w), p(\tau_j > \tau_i|w)) \to +\infty$ (on augmente l'impact de la préférence du vecteur de poids).
   La valeur de cette température impacte fortement la vraisemblance du vecteur de poids, au cours de l'élicitation, elle va impacter la vitesse de convergence de l'estimation des poids, mais aussi son instabilité.
6. MCMC en parallèles ou successifs : L'algorithme MCMC est généralement exécuté plusieurs fois en parallèle pour avoir une plus grande diversité d'estimation, et c'est alors la moyenne de ces estimations qui est prise comme résultat. J'ai constaté empiriquement que l'impact était négligeable dans notre cadre d'étude, on utilise donc un seul run d'MCMC pour notre estimation de poids.

Comme je l'ai dit dans une précédente section (4.3), l'élicitation de préférences sur les actions n'a été possible qu'avec l'ajout de la température dans la formule de la vraisemblance. En effet, avant l'ajout de température, les préférences n'avaient pas suffisamment d'impact sur l'évolution des poids résultants de MCMC. Mon étude des hyperparamètres m'a amené à la conclusion que la meilleure température était de 50 pour les préférences sur les actions, et 5 pour les trajectoires.

4.7 Étude de convergence et de qualité du PBL

L'objectif de l'élicitation de préférences est d'estimer les préférences d'un décideur, à travers des questions qu'on lui pose. La quantité d'information collectée augmente naturellement avec le nombre de questions, de telle sorte à ce qu'à partir d'un certain nombre de questions, notre estimation est suffisamment précise, et converge. Dans notre cas, on cherche à approximer les préférences du décideur avec une somme pondérée $f_{\mathbf{w}}$, et donc la valeur des objectifs aux yeux du décideur par un vecteur de poids $\mathbf{w}$. La première chose que l'on veut prouver est qu'au bout d'un certain nombre de questions posées au décideur, on a accumulé suffisamment d'information pour approximer convenablement ses préférences. On veut donc étudier la convergence de notre vecteur de poids $\mathbf{w}$. Mais pour prouver que notre modèle estime bien les préférences, prouver la convergence des paramètres n'est pas suffisant, il nous faut également prouver la qualité de notre estimation. Pour cela, il nous faut trouver des heuristiques qui estiment la qualité de l'approximation des préférences du décideur par $f_{\mathbf{w}}$. La mise en place d'un modèle passe également par l'analyse de résultats, et c'est pourquoi j'ai cherché à enregistrer des traces montrant la convergence des solutions ainsi que des heuristiques pour l'estimation de la qualité des solutions produites par nos différents modèles d'élicitation de préférences.

4.7.1 Étude de convergence

Pour l'étude de convergence, j'ai utilisé deux indicateurs principaux :

1. Le nombre de nouveaux poids acceptés durant l'algorithme MCMC : Si le nombre de nouveaux poids acceptés est strictement décroissant et converge, cela indique que l'estimation a convergé vers une sous zone de l'espace de recherche où les poids sont comparablement bons vis-à-vis des préférences. Les poids de cette zone sont préférés au reste de l'espace de recherche.
2. Convergence de la variance des poids rencontrés durant le MCMC : Si la variance des poids rencontrés durant le MCMC converge, cela indique que les poids acceptés sont de plus en plus proches et donc que l'estimation converge vers une zone de l'espace de recherche.

4.7.2 Étude de qualité

Pour l'étude de qualité, j'ai cherché des heuristiques qui quantifieraient la proximité entre la façon dont le décideur évalue les trajectoires et l'approximation que l'on en fait (avec $f_{\mathbf{w}}$). On va noter $D(\tau_i)$ l'évaluation que le décideur fait d'une trajectoire τ_i. Cette valeur est théorique pour les applications concrètes de notre modèle (on ne peut pas nécessairement demander à un humain comment il évalue quantitativement une situation), mais nous allons l'utiliser concrètement pour cette étude de qualité, car on utilise ici des données simulées (le décideur n'est pas un humain).

Dans un premier temps, j'ai utilisé deux heuristiques pour qualifier la qualité globale de notre élicitation de préférences (À quel point notre solution estime correctement les préférences du décideur), basées sur le tri d'un batch de 2000 trajectoires :

1. Heuristique 1, somme de l'évaluation du décideur : Le batch est trié par ordre croissant de somme pondérée $f_{\mathbf{w}}(\tau_i)$. On fait ensuite la moyenne de l'évaluation du décideur, des 100 premières trajectoires du batch trié : $\frac{1}{100}\sum_{i=1}^{100} D(\tau_i)$. Ainsi, si la

fonction d'évaluation du décideur est en minimisation, cette heuristique renvoie une valeur décroissante avec la proximité de retranscription des préférences du décideur.

2. Heuristique 2, nombre d'inversions : Le batch est trié par ordre croissant de l'évaluation du décideur $D(\tau_i)$. On prend ensuite les 100 premières trajectoires (les meilleures). Le batch est maintenant trié par ordre croissant de somme pondérée $f_{\mathbf{w}}(\tau_i)$. On compte le nombre d'inversions des 100 trajectoires dans le batch nouvellement trié. Ainsi, la valeur renvoyée par cette heuristique est décroissante avec la qualité de l'approximation des préférences du décideur par $\mathbf{w}$.

Notre modèle d'élicitation de préférences approxime avec un vecteur de poids, ce qui peut parfois restreindre la qualité de notre estimation. En effet, la qualité maximale de l'approximation de préférences par un vecteur de poids est variable selon la manière dont le décideur choisit sa solution préférée.

Des courbes comparatives des performances pour plusieurs vecteurs de poids cachés du décideur différents sont disponibles en annexe B.

Ceci étant dit, j'ai utilisé une nouvelle version des deux heuristiques précédentes, ces versions vont qualifier la qualité relative (à quel point $\mathbf{w}$ est une bonne solution en comparaison à tous les vecteurs de poids possibles) : Pour l'estimation de qualité relative, la qualité globale du poids (calculée précédemment) est normalisée par des estimations des bornes supérieures et inférieures de la qualité globale des poids. Ces bornes supérieures et inférieures sont respectivement les qualités globales maximum et minimum parmi 1000 vecteurs poids tirés aléatoirement dans l'espace a priori. Dans l'hypothèse où le nombre de poids généré est suffisant pour approximer correctement les bornes, si notre estimation relative est optimale (c'est la meilleure estimation possible parmi toutes les solutions atteignables par une approximation des préférences par un vecteur de poids), elle doit tendre vers 0.

Pour l'étude de qualité, on applique les deux versions de nos deux heuristiques à deux batchs de tests différents : Un batch constitué des trajectoires de la politique courante de l'agent et un batch de test, constitué de trajectoires de plusieurs agents différents (agents aléatoire, [1,3,1,1], [1,0,0,0], [0,1,0,1] et [0,0,1,1], 400 trajectoires par agent). Les deux batchs sont constitués de 2000 trajectoires, le premier est batch varie selon les exécutions de MORAL, car il dépend de la politique courante, alors que le deuxième est le même pour toutes les exécutions de MORAL. Le premier batch nous donne des informations sur la qualité de notre approximation des préférences, dans le cadre restreint des trajectoires de notre agent (ce sont sur ces trajectoires que les questions ont été posées). Le second nous donne des informations sur la qualité plus "globale" de l'approximation des préférences, c'est-à-dire si notre approximation fonctionne également sur des trajectoires différentes de celles sur lesquelles on a demandé des préférences au décideur.

Les deux versions de nos 2 heuristiques, appliquées à deux batchs de trajectoires différents, nous permettent de qualifier la qualité de notre élicitation de préférences, à plusieurs niveaux. Une estimation de la qualité globale, une estimation relative à l'ensemble des vecteurs de poids possibles, et cela, sur les trajectoires de la politique actuelle, ou plus globalement sur un batch de trajectoires diverses. Ces informations nous fournissent une idée précise de la portée de la qualité de notre estimation des préférences du décideur.

Chapitre 5

Conclusion

Une grande partie du stage a été portée à l'amélioration du modèle MORAL [10], portant sur plusieurs aspects comme la normalisation des objectifs éthiques et non éthiques 4.1, l'heuristique de sélection de questions 4.5, l'algorithme d'estimation du vecteur de poids traduisant les préférences du décideur (MCMC) 4.6, et pour finir une réelle étude de convergence et performance de l'élicitation de préférences 4.7.2.

Dans le but d'étudier l'intégration de l'éthique dans l'IA sous un autre angle, ou de comparer les performances avec celles de MORAL, j'ai implémenté plusieurs autres modèles MORL. Parmi ces modèles, on peut citer DRLHP [14] et PCN [20].

Je me suis basé sur MORAL pour l'améliorer, mais aussi tenté de m'en inspirer pour d'implémenter de nouveaux modèles : MORAL avec des préférences sur les actions plutôt que les trajectoires 4.3, MORAL_ACTIONS_2 avec la phase d'élicitation de préférence avant la phase d'apprentissage MORL 4.4, ou encore AFTER_MORAL_2, un modèle à mi-chemin entre MORAL et DRLHP qui remplace la somme pondérée par le preference-learner de DRLHP, comme fonction d'agrégation MORL 4.2.

Mes contributions ont globalement permis une amélioration du modèle MORAL, une diversification de ses utilisations possibles et une comparaison vis-à-vis des visions d'autres modèles MORL que l'on pourrait utiliser pour répondre à des problématiques éthiques.

Plusieurs pistes sont possibles pour poursuivre ce travail, et continuer d'améliorer le modèle. Une première direction intéressante serait d'adapter le modèle MORAL_2 avec des préférences sur les trajectoires, comme décrits dans la section 4.4. Une autre direction serait l'implémentation de l'heuristique de sélection EUS, telle qu'elle est décrite dans [12], sans utiliser un algorithme MCMC. On pourrait aussi réaliser, plus globalement, une réelle étude comparative de performance avec DRLHP, PCN ou d'autres algorithmes MORL. Par exemple, il serait possible d'ajouter une phase d'élicitation de préférences après PCN, à l'issue de laquelle on choisirait le vecteur de récompense du front de Pareto qui correspond le mieux aux préférences du décideur. On comparerait alors ce vecteur de récompenses avec le vecteur de récompenses moyen des trajectoires de notre agent expert MORAL. On pourrait également travailler à une compréhension plus précise de comment fonctionne l'apprentissage MORAL, dans le but d'expliquer précisément le comportement résultant du modèle en fonction des préférences du décideur (explicabilité).

Bibliographie

[1] Peter Eckersley. « Impossibility and Uncertainty Theorems in AI Value Alignment (or why your AGI should not have a utility function) ». In : *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*. Sous la dir. d'Huáscar Espinoza, Seán Ó hÉigeartaigh, Xiaowei Huang, José Hernández-Orallo et Mauricio Castillo-Effen. T. 2301. CEUR Workshop Proceedings. CEUR-WS.org, 2019. URL : http://ceur-ws.org/Vol-2301/paper%5C_7.pdf (pages 1, 2, 6).

[2] David Abel, James MacGlashan et Michael L. Littman. « Reinforcement Learning as a Framework for Ethical Decision Making ». In : *AI, Ethics, and Society, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 13, 2016*. Sous la dir. de Blai Bonet, Sven Koenig, Benjamin Kuipers, Illah R. Nourbakhsh, Stuart Russell, Moshe Y. Vardi et Toby Walsh. T. WS-16-02. AAAI Technical Report. AAAI Press, 2016. URL : http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12582 (pages 1, 4).

[3] Adrien Ecoffet et Joel Lehman. « Reinforcement Learning Under Moral Uncertainty ». In : *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Sous la dir. de Marina Meila et Tong Zhang. T. 139. Proceedings of Machine Learning Research. PMLR, 2021, p. 2926-2936. URL : http://proceedings.mlr.press/v139/ecoffet21a.html (pages 1, 5).

[4] Justin Svegliato, Samer B. Nashed et Shlomo Zilberstein. « Ethically Compliant Sequential Decision Making ». In : *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, p. 11657-11665. URL : https://ojs.aaai.org/index.php/AAAI/article/view/17386 (pages 1, 4).

[5] Emery A. Neufeld. « Reinforcement Learning Guided by Provable Normative Compliance ». In : *Proceedings of the 14th International Conference on Agents and Artificial Intelligence, ICAART 2022, Volume 3, Online Streaming, February 3-5, 2022*. Sous la dir. d'Ana Paula Rocha, Luc Steels et H. Jaap van den Herik. SCITEPRESS, 2022, p. 444-453. DOI : 10.5220/0010835600003116. URL : https://doi.org/10.5220/0010835600003116 (pages 1, 5).

[6] Manel Rodriguez-Soto, Maite López-Sánchez et Juan A. Rodríguez-Aguilar. « Multi-Objective Reinforcement Learning for Designing Ethical Environments ». In : *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Sous la

dir. de Zhi-Hua Zhou. ijcai.org, 2021, p. 545-551. DOI : 10.24963/ijcai.2021/76. URL : https://doi.org/10.24963/ijcai.2021/76 (pages 1, 4, 5).

[7] Yueh-Hua Wu et Shou-De Lin. « A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents ». In : *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Sous la dir. de Sheila A. McIlraith et Kilian Q. Weinberger. AAAI Press, 2018, p. 1687-1694. URL : https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16195 (pages 2, 4).

[8] Ritesh Noothigattu, Djallel Bouneffouf, Nicholas Mattei, Rachita Chandra, Piyush Madan, Kush R. Varshney, Murray Campbell, Moninder Singh et Francesca Rossi. « Teaching AI Agents Ethical Values Using Reinforcement Learning and Policy Orchestration ». In : *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. Sous la dir. de Sarit Kraus. ijcai.org, 2019, p. 6377-6381. DOI : 10.24963/ijcai.2019/891. URL : https://doi.org/10.24963/ijcai.2019/891 (pages 2, 4, 5).

[9] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song et Jacob Steinhardt. « Aligning AI With Shared Human Values ». In : *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL : https://openreview.net/forum?id=dNy%5C_RKzJacY (page 2).

[10] Markus Peschl, Arkady Zgonnikov, Frans A. Oliehoek et Luciano Cavalcante Siebert. « MORAL : Aligning AI with Human Norms through Multi-Objective Reinforced Active Learning ». In : *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*. Sous la dir. de Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud et Matthew E. Taylor. International Foundation for Autonomous Agents et Multiagent Systems (IFAAMAS), 2022, p. 1038-1046. URL : https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p1038.pdf (pages 2-8, 20).

[11] Arie Glazier, Andrea Loreggia, Nicholas Mattei, Taher Rahgooy, Francesca Rossi et Kristen Brent Venable. « Learning Behavioral Soft Constraints from Demonstrations ». In : *CoRR* abs/2202.10407 (2022). arXiv : 2202.10407. URL : https://arxiv.org/abs/2202.10407 (pages 2, 4, 5).

[12] Riad Akrour, Marc Schoenauer et Michèle Sebag. « APRIL : Active Preference Learning-Based Reinforcement Learning ». In : *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*. Sous la dir. de Peter A. Flach, Tijl De Bie et Nello Cristianini. T. 7524. Lecture Notes in Computer Science. Springer, 2012, p. 116-131. DOI : 10.1007/978-3-642-33486-3_8. URL : https://doi.org/10.1007/978-3-642-33486-3%5C_8 (pages 3, 5, 6, 11, 15, 16, 20).

[13] Weiwei Cheng, Johannes Fürnkranz, Eyke Hüllermeier et Sang-Hyeun Park. « Preference-Based Policy Iteration : Leveraging Preference Learning for Reinforcement Learning ». In : *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I*. Sous la dir. de Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba et Michalis Vazirgiannis. T. 6911. Lecture Notes in Computer

Science. Springer, 2011, p. 312-327. DOI : 10.1007/978-3-642-23780-5_30. URL : https://doi.org/10.1007/978-3-642-23780-5%5C_30 (pages 3, 5, 6).

[14] Paul F. CHRISTIANO, Jan LEIKE, Tom B. BROWN, Miljan MARTIC, Shane LEGG et Dario AMODEI. « Deep Reinforcement Learning from Human Preferences ». In : *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Sous la dir. d'Isabelle GUYON, Ulrike von LUXBURG, Samy BENGIO, Hanna M. WALLACH, Rob FERGUS, S. V. N. VISHWANATHAN et Roman GARNETT. 2017, p. 4299-4307. URL : https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html (pages 3, 5, 6, 10, 20).

[15] Ioannis TSOCHANTARIDIS, Thorsten JOACHIMS, Thomas HOFMANN et Yasemin ALTUN. « Large Margin Methods for Structured and Interdependent Output Variables ». In : *Journal of Machine Learning Research* 6 (2005), p. 1453-1484. ISSN : 15337928. URL : http://jmlr.org/papers/v6/tsochantaridis05a.html (page 3).

[16] Thorsten JOACHIMS. « A support vector method for multivariate performance measures ». In : *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*. Sous la dir. de Luc De RAEDT et Stefan WROBEL. T. 119. ACM International Conference Proceeding Series. ACM, 2005, p. 377-384. DOI : 10.1145/1102351.1102399. URL : https://doi.org/10.1145/1102351.1102399 (page 3).

[17] Han YU, Zhiqi SHEN, Chunyan MIAO, Cyril LEUNG, Victor R. LESSER et Qiang YANG. « Building Ethics into Artificial Intelligence ». In : *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Sous la dir. de Jérôme LANG. ijcai.org, 2018, p. 5527-5533. DOI : 10.24963/ijcai.2018/779. URL : https://doi.org/10.24963/ijcai.2018/779 (page 4).

[18] Francesca ROSSI et Nicholas MATTEI. « Building Ethically Bounded AI ». In : *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, p. 9785-9789. DOI : 10.1609/aaai.v33i01.33019785. URL : https://doi.org/10.1609/aaai.v33i01.33019785 (page 4).

[19] Conor F. HAYES, Roxana RADULESCU, Eugenio BARGIACCHI, Johan KÄLLSTRÖM, Matthew MACFARLANE, Mathieu REYMOND, Timothy VERSTRAETEN, Luisa M. ZINTGRAF, Richard DAZELEY, Fredrik HEINTZ, Enda HOWLEY, Athirai A. IRISSAPPANE, Patrick MANNION, Ann NOWÉ, Gabriel de OLIVEIRA RAMOS, Marcello RESTELLI, Peter VAMPLEW et Diederik M. ROIJERS. « A practical guide to multi-objective reinforcement learning and planning ». In : *Auton. Agents Multi Agent Syst.* 36.1 (2022), p. 26. DOI : 10.1007/s10458-022-09552-y. URL : https://doi.org/10.1007/s10458-022-09552-y (pages 5, 6).

[20] Mathieu REYMOND, Eugenio BARGIACCHI et Ann NOWÉ. « Pareto Conditioned Networks ». In : *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*. Sous la dir. de Piotr FALISZEWSKI, Viviana MASCARDI, Catherine PELACHAUD et Matthew E. TAYLOR. International Foundation for Autonomous Agents et Multiagent Systems (IFAAMAS), 2022, p. 1110-1118. DOI : 10.5555/3535850.3535974. URL :

https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p1110.pdf (pages 5, 10, 20).

[21] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell et Anind K. Dey. « Maximum Entropy Inverse Reinforcement Learning. » In : *AAAI*. Sous la dir. de Dieter Fox et Carla P. Gomes. AAAI Press, 2008, p. 1433-1438. isbn : 978-1-57735-368-3. url : http://dblp.uni-trier.de/db/conf/aaai/aaai2008.html#ZiebartMBD08 (page 6).

[22] Firas Jarboui et Vianney Perchet. « Offline Inverse Reinforcement Learning ». In : *CoRR* abs/2106.05068 (2021). arXiv : 2106.05068. url : https://arxiv.org/abs/2106.05068 (page 6).

[23] Shehryar Malik, Usman Anwar, Alireza Aghasi et Ali Ahmed. « Inverse Constrained Reinforcement Learning ». In : *Proceedings of the 38th International Conference on Machine Learning*. Sous la dir. de Marina Meila et Tong Zhang. T. 139. Proceedings of Machine Learning Research. PMLR, 2021, p. 7390-7399. url : https://proceedings.mlr.press/v139/malik21a.html (page 6).

[24] Chelsea Finn, Paul F. Christiano, Pieter Abbeel et Sergey Levine. « A Connection between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models ». In : *CoRR* abs/1611.03852 (2016). arXiv : 1611.03852. url : http://arxiv.org/abs/1611.03852 (page 6).

[25] Justin Fu, Katie Luo et Sergey Levine. « Learning Robust Rewards with Adversarial Inverse Reinforcement Learning ». In : *CoRR* abs/1710.11248 (2017). arXiv : 1710.11248. url : http://arxiv.org/abs/1710.11248 (page 6).

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville et Yoshua Bengio. « Generative Adversarial Nets ». In : *Advances in Neural Information Processing Systems*. Sous la dir. de Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence et K.Q. Weinberger. T. 27. Curran Associates, Inc., 2014. url : https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf (page 6).

[27] Richard Meyes, Moritz Schneider et Tobias Meisen. « How Do You Act ? An Empirical Study to Understand Behavior of Deep Reinforcement Learning Agents ». In : *CoRR* abs/2004.03237 (2020). arXiv : 2004.03237. url : https://arxiv.org/abs/2004.03237 (page 6).

Annexe A

Comparaison entre nos nouveaux modèles et le modèle MORAL de base

A.1 Objectif de l'étude

Le but de cette section est d'étudier les performances de trois modèles : le premier correspond à un modèle dont les préférences portent sur les actions, celles du second sur les trajectoires et le troisième correspond lui au modèle basique de MORAL, c'est-à-dire sans toutes les modifications que j'ai apportées au modèle durant mon stage.

Pour faire l'étude rigoureuse d'un paramètre, ici sur quel élément portent les préférences, il faudrait fixer tous les autres et ne faire varier que celui-ci. Malheureusement, pour que chacun des modèles fonctionne, il faut fixer des hyperparamètres à des valeurs précises, notamment la température de la vraisemblance et l'heuristique de sélection de question. On va comparer dans cette étude deux modèles (trajectoires et actions) avec les meilleurs hyperparamètres que j'ai trouvés pour chacun. Il faut donc voir cette étude davantage comme une comparaison entre modèles, que comme une étude d'hyperparamètres (ici les préférences). J'ai pensé qu'il serait également intéressant de prouver l'amélioration de l'élicitation de préférences, en comparant les performances de nos deux modèles avec celles du modèle avec les paramètres tels qu'ils étaient avant toutes mes modifications.

A.2 Modalités de l'étude et résultats

Pour que l'étude soit plus représentative d'une performance globale du modèle, chaque courbe est la moyenne de 3 exécutions, avec des valeurs différentes des poids cachés du décideur. Les 3 poids cachés différents sont : [3,1,1], [1,3,1] et [1,2,3] (explication des poids cachés en annexe E.2).

Pour le modèle avec des préférences sur les actions, la température est de 50, et l'heuristique de sélection est **no_double_less_zeros**. Pour le modèle avec des préférences sur les trajectoires, la température est de 10, et l'heuristique de sélection est **basic_loglik**. Pour le modèle avec les paramètres de base de MORAL, la température est de 1, et l'heuristique de sélection de question est **delta_loglik**, décrite rapidement au début de la

section 4.5.

La courbe rouge correspond au modèle avec des préférences sur les actions, la verte à celui sur les trajectoires, et la bleue au modèle de base de MORAL.

Cette étude étant la première, je vais expliciter toutes les courbes et cela servira d'explication également pour les prochaines études, car les courbes seront les mêmes.

Évolution de la qualité de l'approximation :

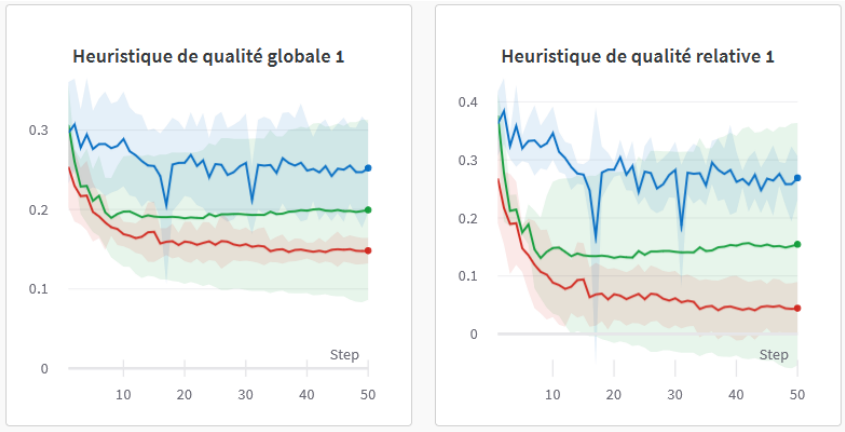

FIGURE A.1 – **Critère 1** : Évolution de l'heuristique 1 (globale ou relative) au cours de l'elicitation de questions

Ces deux premières courbes sont celles de la première heuristique de qualité, appliqué au 1er batch 4.7.2 (on y fera désormais référence comme le 1er critère de qualité). La première représente l'évolution de la somme des évaluations du décideur, pour les 100 meilleures trajectoires, selon notre estimation des préférences. La deuxième normalise cette évolution par rapport aux minimums et maximums estimés des vecteurs de poids, elle représente donc la qualité relative à l'espace de recherche de vecteurs de poids.

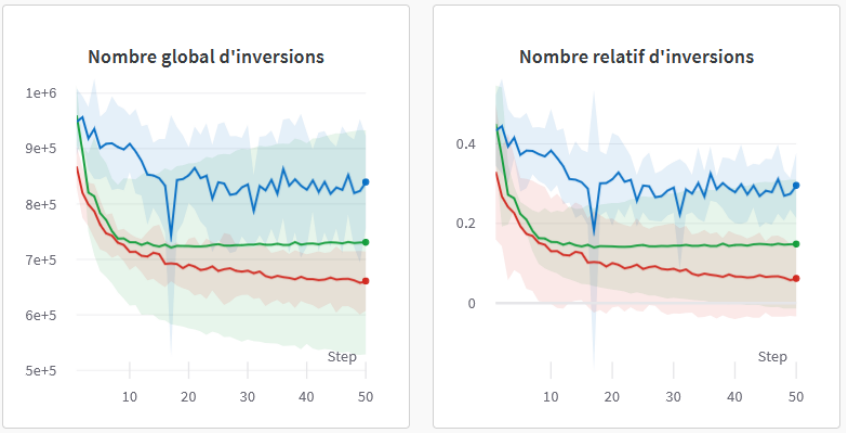

FIGURE A.2 – **Critère 2** : Évolution du nombre d'inversion (global ou relatif) au cours de l'elicitation de questions

Ces deux courbes sont celles de la deuxième heuristique de qualité, appliqué au 1er batch 4.7.2 (on y fera désormais référence comme le 2ème critère de qualité). La première représente l'évolution du nombre d'inversions par rapport au tri du décideur. La deuxième

normalise cette évolution par rapport aux minimums et maximums estimés des vecteurs de poids, elle représente donc la qualité relative à l'espace de recherche de vecteurs de poids.

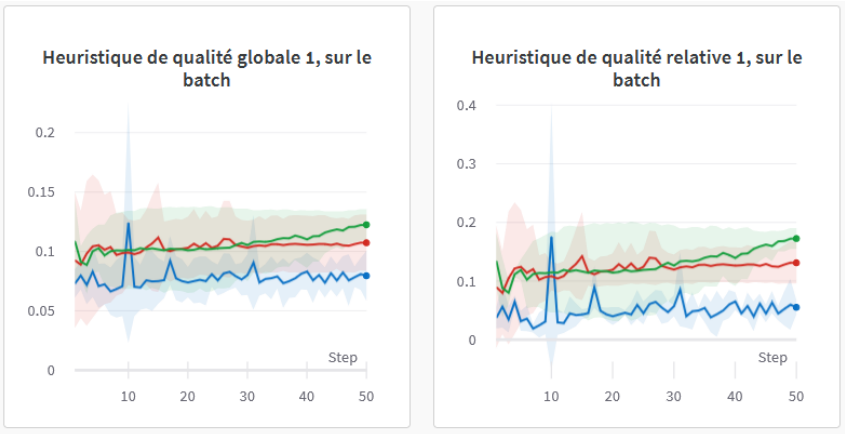

FIGURE A.3 – **Critère 3** : Évolution de l'heuristique 1 (globale ou relative) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

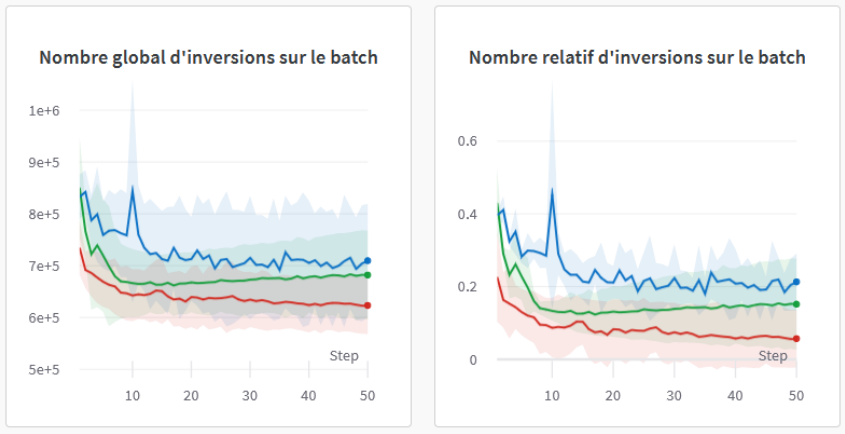

FIGURE A.4 – **Critère 4** : Évolution du nombre d'inversion (global ou relatif) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

Ces quatre courbes représentent les mêmes critères de qualité que les courbes précédentes, mais cette fois-ci appliqués à un batch de trajectoires différent de celui sur lequel on a posé les questions au décideur (batch de test différent du batch d'apprentissage). Ces courbes peuvent donner une idée de la portée de la qualité de l'estimation (si elle se transpose ou non à d'autres données). On fera référence aux deux premières comme le 3ème critère de qualité, et deux dernières comme le 4ème critère.

Étude de la convergence au cours de l'élicitation :

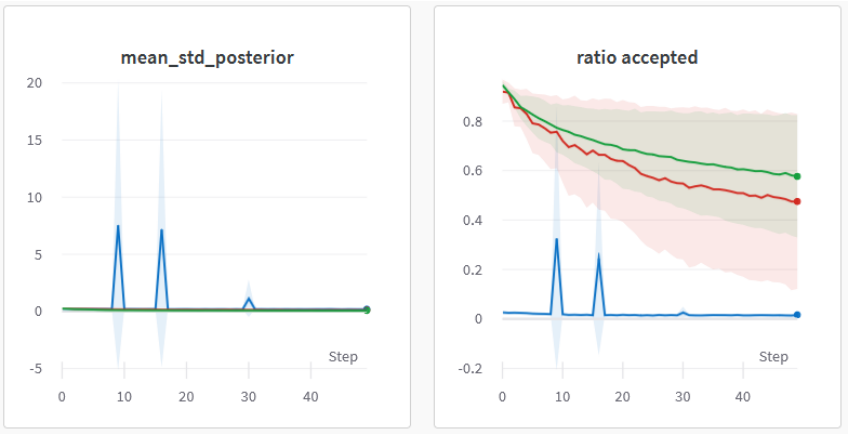

FIGURE A.5 – Étude de la convergence au cours de l'elicitation de questions

Ces deux courbes sont celles des deux heuristiques d'estimation de convergence 4.7.1

A.3 Analyse des résultats

On peut d'abord constater une nette amélioration des résultats pour 3 des 4 critères de performances de nos deux modèles par rapport au modèle de base (nombre d'inversions, pour les deux batchs de trajectoires différents, et l'heuristique n°1 uniquement pour le batch de trajectoire similaire au batch d'apprentissage). Pour le 4ème critère de performance, au contraire, le modèle de base performe mieux que les deux autres, en se plaçant dans le top 10% des poids aléatoires, là où ils sont entre le top 10% et 20%. On peut en conclure que le modèle de base fait plus d'erreurs (un plus grand nombre d'inversions) mais des erreurs moins graves (somme des évaluations du décideur plus faible). Cela peut notamment s'expliquer par le fait que le modèle de base calcule des vecteurs de poids très équilibrés (chaque poids entre 0.4 et 0.6), contrairement à nos nouveaux modèles qui traduisent les préférences de manière plus extrême, avec des vecteurs assez déséquilibrés. Les erreurs peuvent donc être plus importantes.

Pour comparer nos deux nouveaux modèles, on peut constater que les performances sur les actions performent mieux, en moyenne, que celles sur les trajectoires. On peut notamment voir que les élicitations sur les actions permettent de converger vers 0, pour nos 2 premiers critères de performance. Contrairement aux élicitations sur les trajectoires qui semblent pallier entre le top 10% et 20% des poids possibles (entre 0.1 et 0.2 pour les deux premiers critères). On peut également noter que la variance des résultats est plus grande pour les préférences sur les trajectoires, en comparaison aux préférences sur les actions. Encore un point plutôt positif pour le modèle sur les actions.

Annexe B

Étude des performances, en fonction des poids cachés du décideur

B.1 Objectif de l'étude

Le but de cette section est d'étudier les performances de notre modèle PBL, en fonction de plusieurs vecteurs de poids cachés du décideur. Une explication des préférences en fonction des poids cachés du décideur, illustrée d'un exemple, est disponible en annexe E.2.

B.2 Modalités de l'étude et résultats

On a exécuté le modèle pour les poids cachés du décideur suivants : [1,2,3] (courbe rouge), [3,1,1] (courbe verte) et [1,3,1] (courbe bleue). Son heuristique de choix est une minimisation de l'entropie entre ses poids cachés et la récompense de la trajectoire/de l'action.

Pour toutes les exécutions, on a fixé tous les paramètres du PBL :

1. Les préférences sont sur des actions et pas des trajectoires.
2. l'heuristique de sélection des questions est **no_double_less_zeros** (pas de comparaisons entre mêmes vecteurs de récompense et 90% de chance de comparer deux vecteurs non nuls) 4.5.
3. Température de 50.

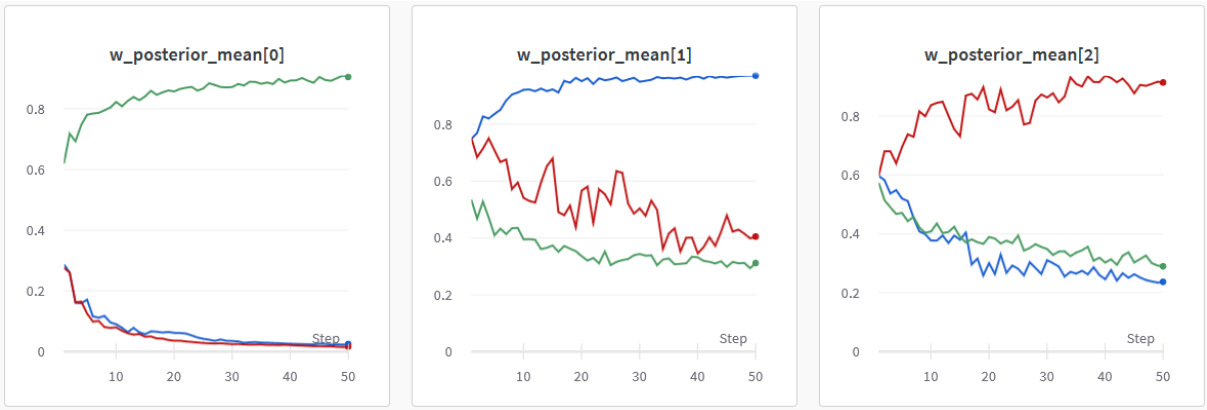

FIGURE B.1 – Évolution des poids au cours de l'elicitation de questions

Les 3 courbes ci-dessus n'étaient pas présentes lors de la première étude, car nous avions fait des moyennes de plusieurs courbes. Elles représentent l'évolution du vecteur de poids retourné par MCMC en fonction du nombre de questions posées au décideur.

Évolution de la qualité de l'approximation :

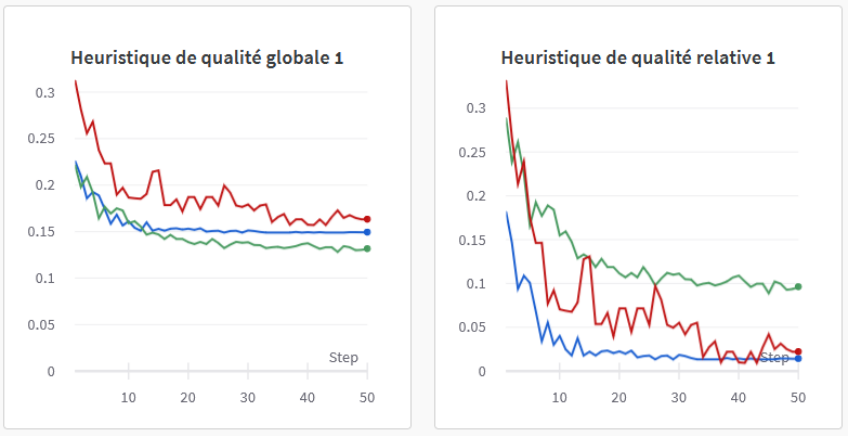

FIGURE B.2 – **Critère 1** : Évolution de l'heuristique 1 (globale ou relative) au cours de l'elicitation de questions

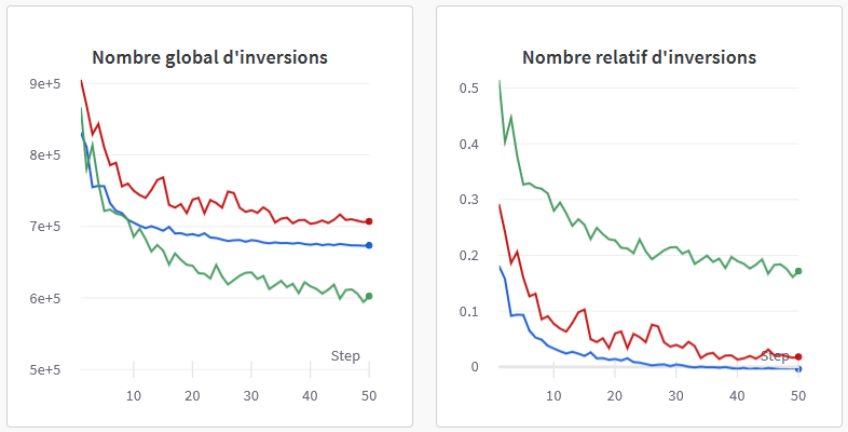

FIGURE B.3 – **Critère 2** : Évolution du nombre d'inversion (global ou relatif) au cours de l'elicitation de questions

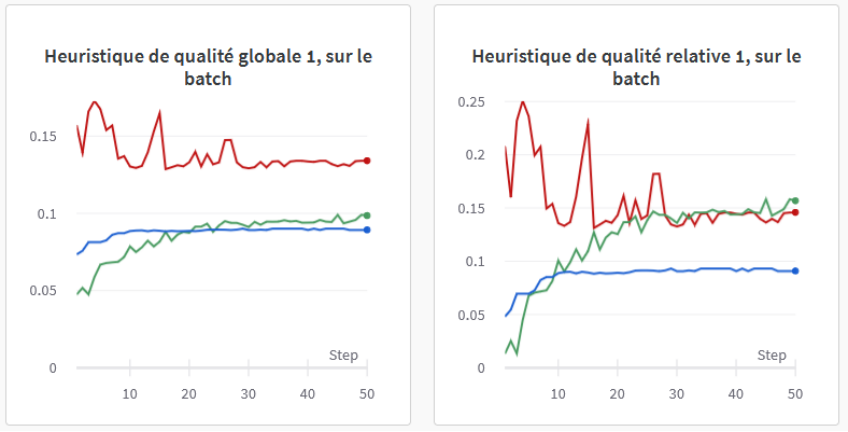

FIGURE B.4 – **Critère 3** : Évolution de l'heuristique 1 (globale ou relative) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

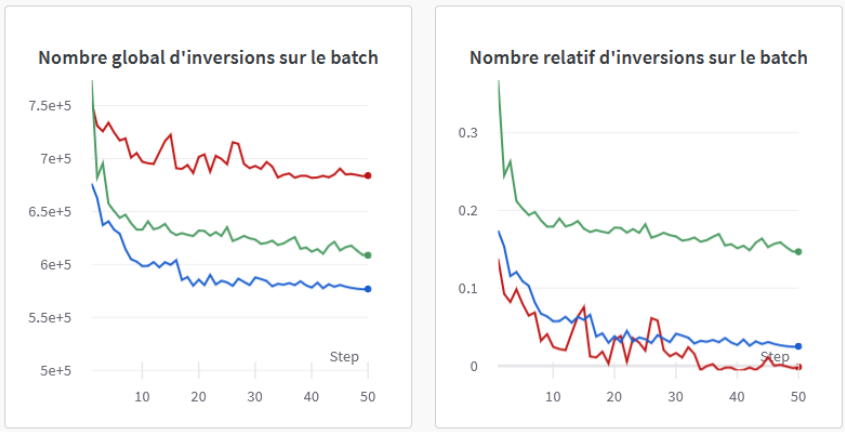

FIGURE B.5 – **Critère 4** : Évolution du nombre d'inversion (global ou relatif) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

Étude de la convergence au cours de l'élicitation :

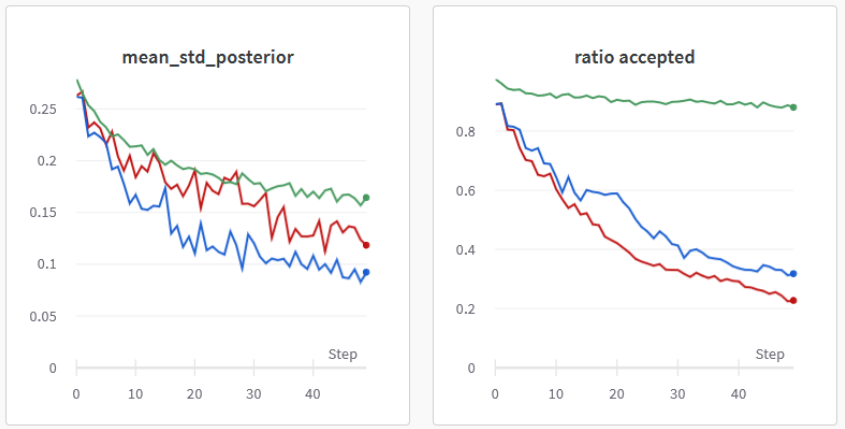

FIGURE B.6 – Étude de la convergence au cours de l'elicitation de questions

B.3 Analyse des résultats

On peut voir que certains poids cachés du décideur sont plus difficiles à approximer que d'autres. Pour une élicitation sur des actions, les poids [1,3,1] et [1,2,3] sont plus faciles à approximer que les poids [3,1,1]. On observe de meilleurs résultats chez les deux premiers pour les 4 indicateurs de qualité. Pour le vecteur [3,1,1], les indicateurs de qualités relatifs au poids le placent entre le top 10% et le top 20%, sur 1000 poids tirés aléatoirement. Ces performances ne paraissent pas très bonnes, pour une élicitation de 50 questions. On peut voir que le modèle accepte quasiment toujours autant de poids au cours du MCMC (critère de convergence 2, ratio_accepted), au bout de 50 questions, ce qui pourrait s'expliquer par la mauvaise qualité des poids trouvés. On peut également noter que pour [1,3,1] et [1,2,3], notre élicitation nous permet d'atteindre une solution quasiment optimale (convergence vers 0) pour 3 des 4 indicateurs de qualités (nb_inv, weight_eval, nb_inv_batch).

Si l'on voulait faire une étude plus rigoureuse de la performance de l'élicitation, il faudrait tirer un grand nombre de vecteurs de poids cachés et faire la moyenne des résultats de notre modèle.

Annexe C

Étude sur les sélections de questions, sur les actions

C.1 Objectif de l'étude

Le but de cette section est d'étudier les performances de l'élicitation sur les actions, en fonction des différentes heuristiques de sélection.

C.2 Modalités de l'étude et résultats

J'ai choisi de comparer trois heuristiques de sélections, où l'on a 2 versions de chaque heuristique. On compare donc 6 exécutions en tout.

Les 3 heuristiques sont :

1. sélection **random** (courbe rouge)
2. sélection **basic_loglik** (courbe bleue)
3. sélection **EUS** (courbe verte)

chaque heuristique possède une deuxième version, où j'ai combiné l'heuristique avec **no_double_less_zeros**. Le principe étant que la sélection choisit de deux vecteurs de récompenses différents, et 90% de chances de deux vecteurs non nuls. Pour **basic_loglik**, et **EUS**, toutes les paires d'actions comparées sont sélectionnées au préalable selon les principes de **no_double_less_zeros**. Les courbes des secondes versions des 3 heuristiques sont de la même couleur que les versions précédentes, mais en pointillé.

Pour toutes les exécutions, on a fixé tous les paramètres du PBL :

1. Les préférences sont sur des actions et pas des trajectoires.
2. Température de 50.
3. Les poids cachés du décideur sont [1,3,1] (explications en annexe E.2).

J'ai choisi ce vecteur de poids cachés du décideur, car je trouvais plus intéressant de comparer les heuristiques de sélections pour des poids cachés où l'heuristique de référence (**no_double_less_zeros**) obtient de très bons résultats.

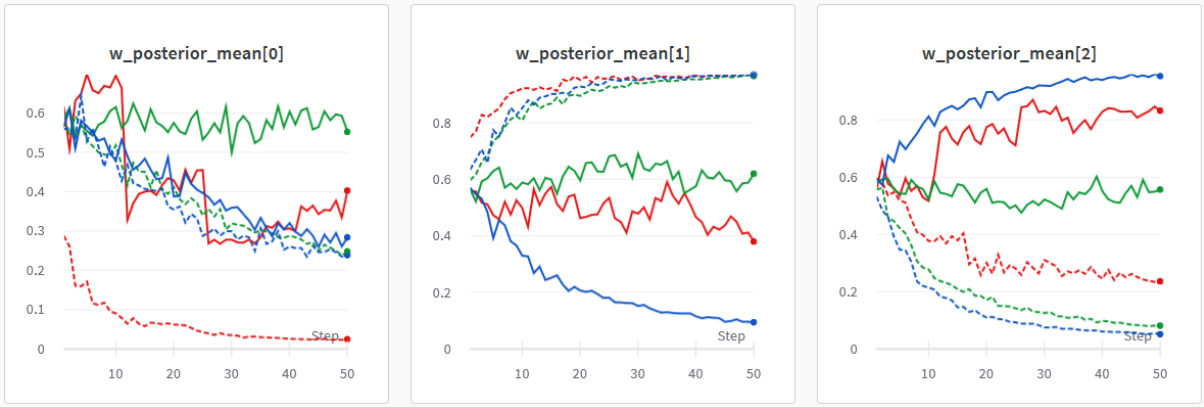

FIGURE C.1 – Évolution des poids au cours de l'elicitation de questions

Évolution de la qualité de l'approximation :

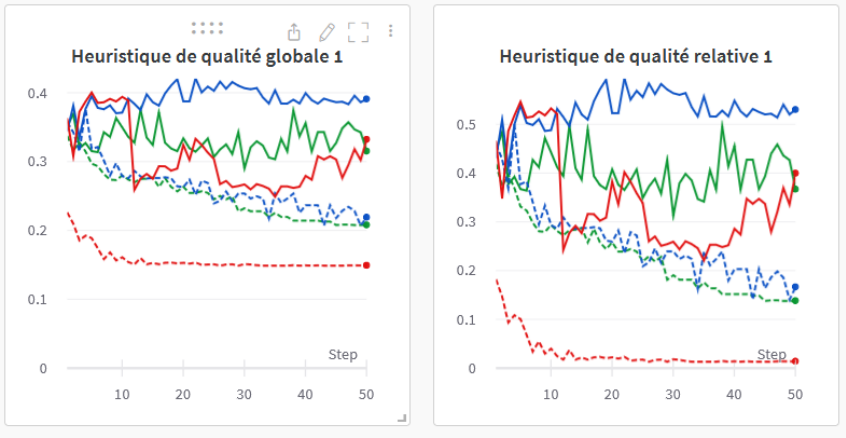

FIGURE C.2 – **Critère 1** : Évolution de l'heuristique 1 (globale ou relative) au cours de l'elicitation de questions

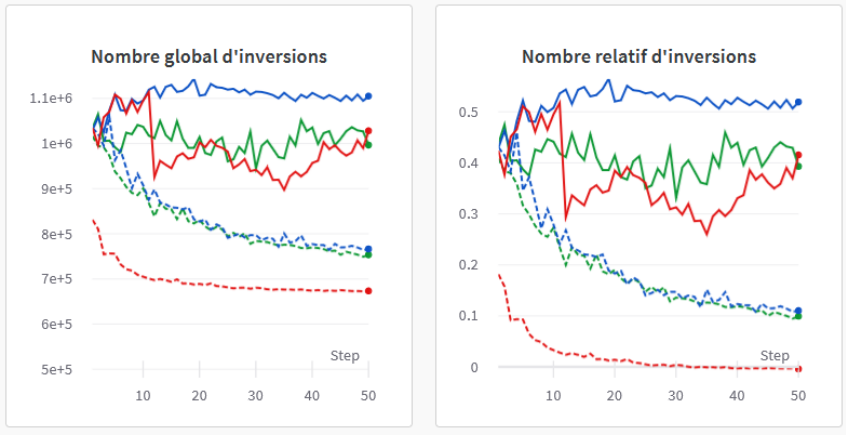

FIGURE C.3 – **Critère 2** : Évolution du nombre d'inversion (global ou relatif) au cours de l'elicitation de questions

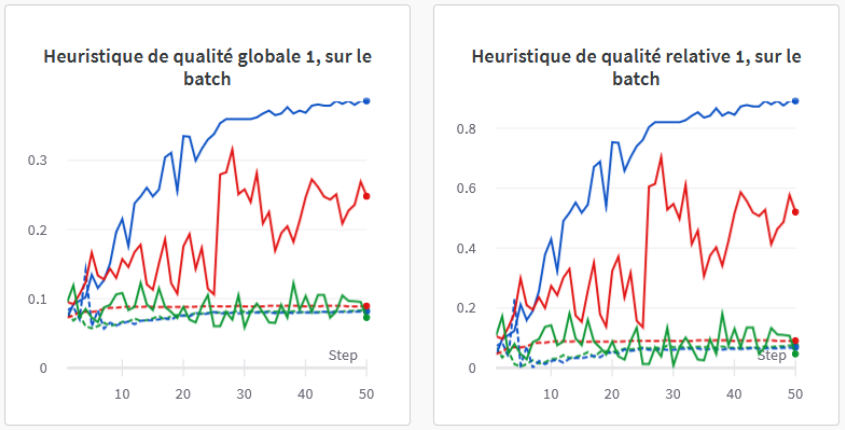

FIGURE C.4 – **Critère 3** : Évolution de l'heuristique 1 (globale ou relative) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

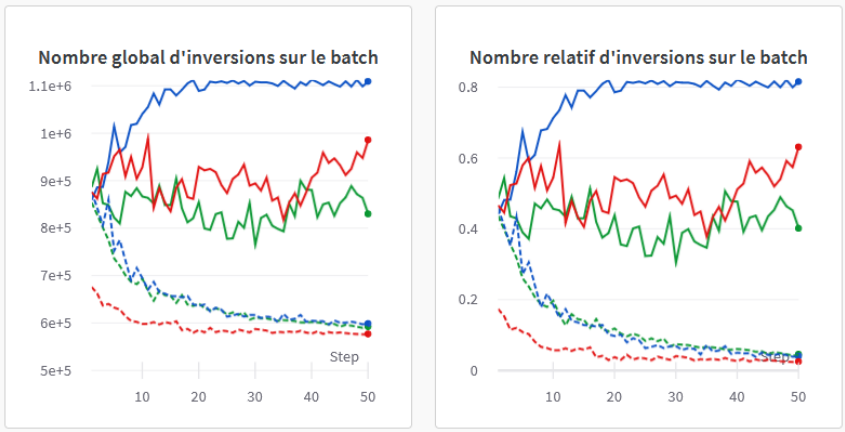

FIGURE C.5 – **Critère 4** : Évolution du nombre d'inversion (global ou relatif) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

Étude de la convergence au cours de l'élicitation :

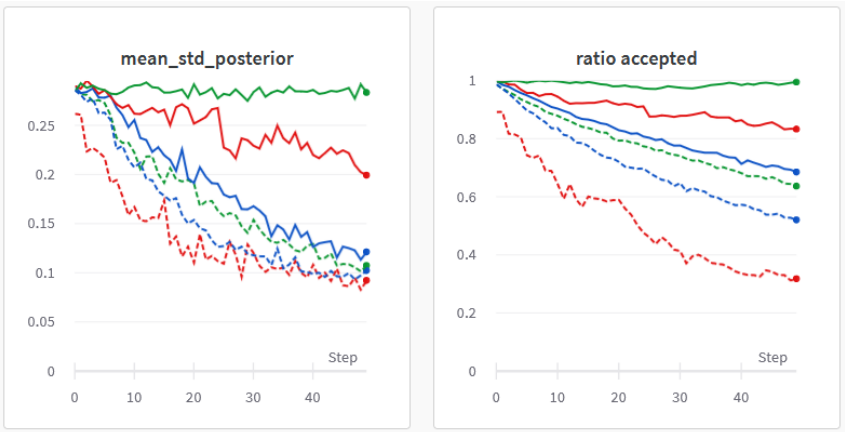

FIGURE C.6 – Étude de la convergence au cours de l'elicitation de questions

C.3 Analyse des résultats

On peut constater le fort impact des heuristiques de sélection de questions pour les élicitations sur des actions. En effet, certaines heuristiques (**EUS** ou **basic_loglik**) obtiennent de moins bons résultats que 40% des poids aléatoires, sur 3 critères de qualité sur 4. Alors que l'heuristique **random** combinée à **no_double_less_zeros** fait partie du top <10%, sur les 4 critères de qualités (donc à toutes les échelles).

Une seconde chose que l'on peut noter est l'amélioration des 3 heuristiques, lorsqu'elles sont combinées à **no_double_less_zeros**. Combinées à **no_double_less_zeros**, elles convergent toutes vers un top 10% pour les 4 critères de qualité. Pour des préférences sur les actions, le plus important est donc de poser au décideur des questions sur lesquelles on pourra tirer de réelles conclusions (vis-à-vis des vecteurs de récompenses).

On peut également noter l'important écart de qualité entre les deux versions de l'heuristique **basic_loglik**. Sa première version obtient des résultats catastrophiques : top 20% des pires poids, parmi 1000 poids aléatoires, pour les critères 3 et 4. Top 50% pour les critères 1 et 2. Au contraire, sa deuxième version se place top 20% pour les critères 1 et 2, et top 10% pour les critères 3 et 4. L'heuristique **basic_loglik** essaye de faire une dichotomie de l'espace de recherche (des poids) en sélectionnant les questions pour lesquelles notre estimation des préférences du décideur est la plus indécise. Il est donc probable qu'il ne sélectionne des questions qu'avec des vecteurs de récompenses objectives en double ou nuls. Cela pousserait notre estimation à tirer de mauvaises conclusions des réponses du décideur.

On peut voir que la deuxième version de l'heuristique **random** est nettement meilleure que toutes les autres sur tous les critères excepté le 3ème. On peut donc penser que les heuristiques **basic_loglik** et **EUS** ne sont pas adaptées à des préférences sur les actions.

Annexe D

Étude sur les sélections de questions, sur les trajectoires

D.1 Objectif de l'étude

Le but de cette section est d'étudier les performances de l'élicitation sur les trajectoires, en fonction des différentes heuristiques de sélection.

D.2 Modalités de l'étude et résultats

Pour toutes les exécutions, on a fixé tous les paramètres du PBL :

1. Les préférences sont sur des trajectoires.
2. Température de 10.
3. Les poids cachés du décideur sont [1,3,1] (explications en annexe E.2).

J'ai choisi ce vecteur de poids cachés du décideur, car je trouvais plus intéressant de comparer les heuristiques de sélections pour des poids cachés où l'heuristique de référence (**basic_loglik**) obtient de très bons résultats.

La courbe rouge correspond l'heuristique de sélection **basic_loglik**, la verte à **EUS** et la bleue à l'heuristique **random** 4.5.

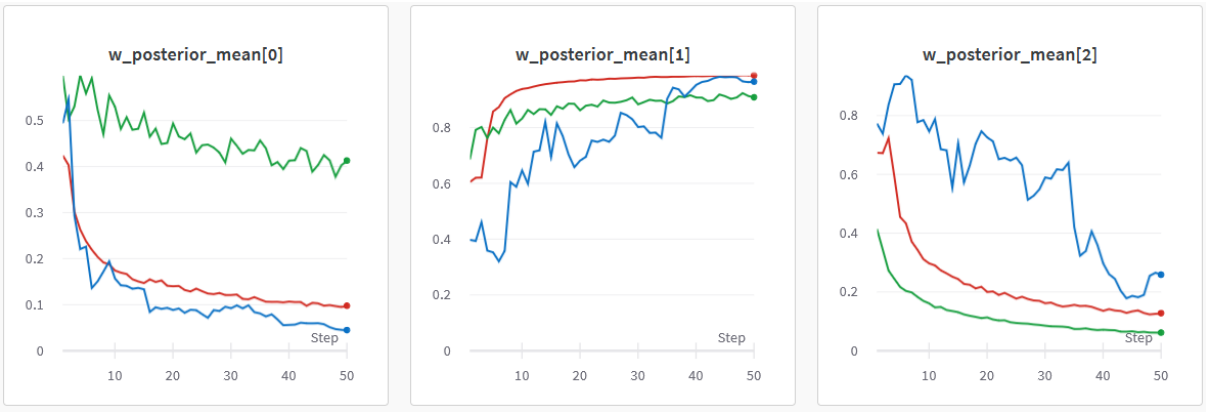

FIGURE D.1 – Évolution des poids au cours de l'elicitation de questions

Évolution de la qualité de l'approximation :

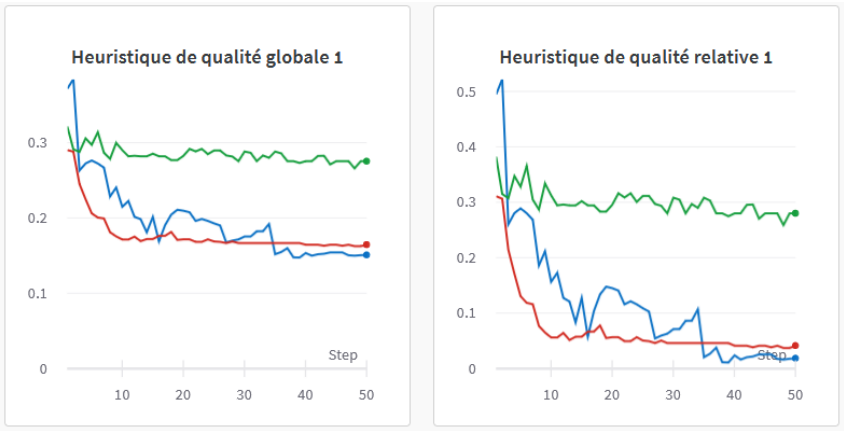

FIGURE D.2 – **Critère 1** : Évolution de l'heuristique 1 (globale ou relative) au cours de l'elicitation de questions

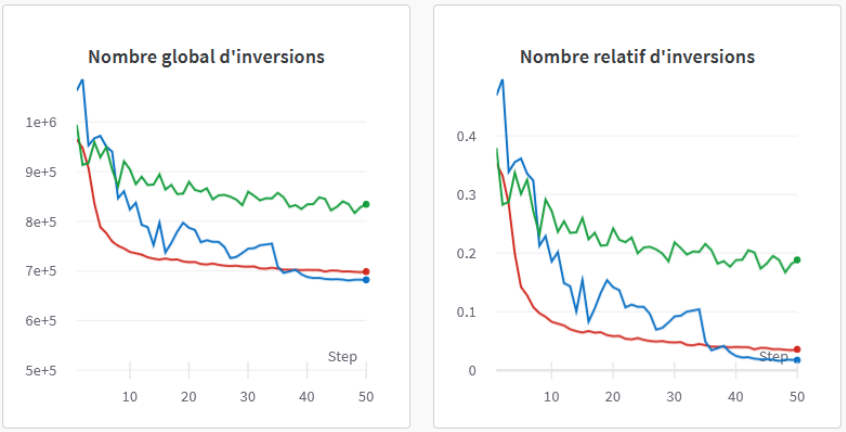

FIGURE D.3 – **Critère 2** : Évolution du nombre d'inversion (global ou relatif) au cours de l'elicitation de questions

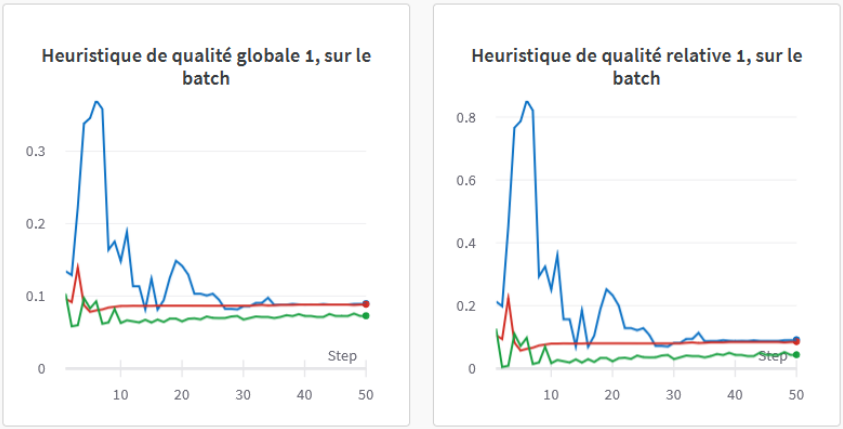

FIGURE D.4 – **Critère 3** : Évolution de l'heuristique 1 (globale ou relative) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

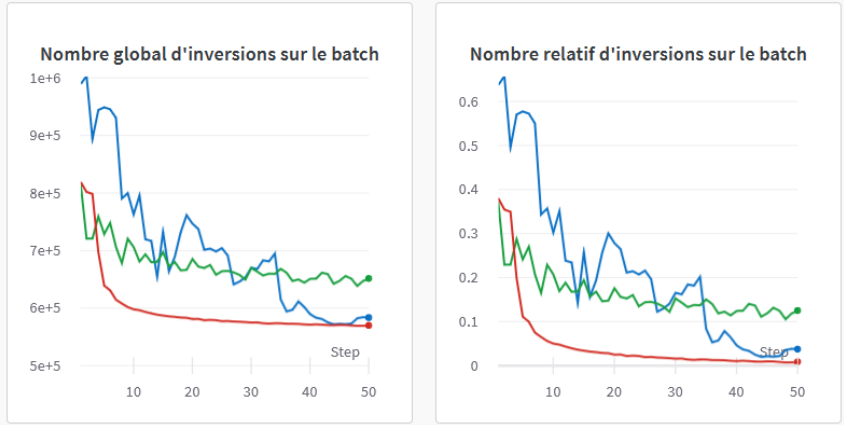

FIGURE D.5 – **Critère 4** : Évolution du nombre d'inversion (global ou relatif) avec le batch de trajectoires exterieures au cours de l'elicitation de questions

Étude de la convergence au cours de l'élicitation :

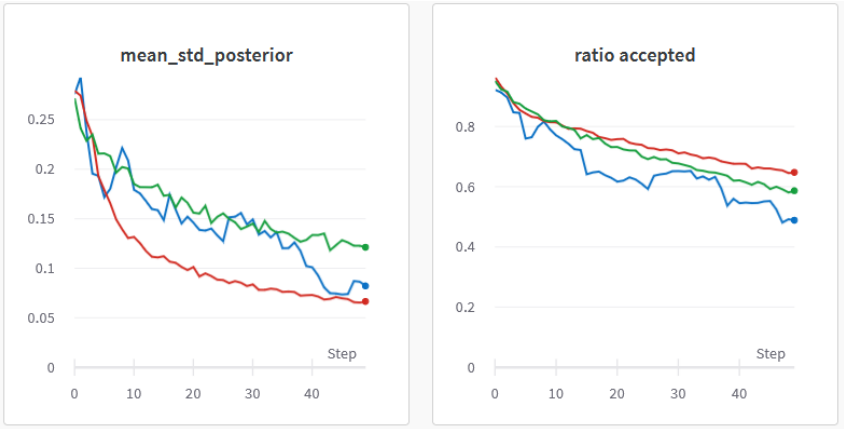

FIGURE D.6 – Étude de la convergence au cours de l'elicitation de questions

D.3 Analyse des résultats

On peut constater que l'impact des heuristiques de sélection de questions est moins important que pour les préférences sur les actions. En effet, la sélection aléatoire (**random**) obtient de très bons résultats, ce qui indique que l'écart de quantité d'information apportée entre deux questions sur des trajectoires est en moyenne assez faible. Mais on peut néanmoins noter que l'heuristique **EUS** obtient de moins résultats que la sélection aléatoire, ce qui indique qu'avoir une mauvaise heuristique peut impacter les résultats. De l'autre côté, l'heuristique **basic_loglik** obtient des résultats comparables à l'heuristique aléatoire, mais plus rapidement et de manière plus stable que celle-ci. En effet, on constate qu'avec la sélection **basic_loglik**, les résultats finaux sont approchés en à peine une dizaine de questions, là où l'heuristique aléatoire n'est "stable" qu'à partir de 40 questions.

Annexe E

Environnement du modèle

E.1 Présentation de l'environnement

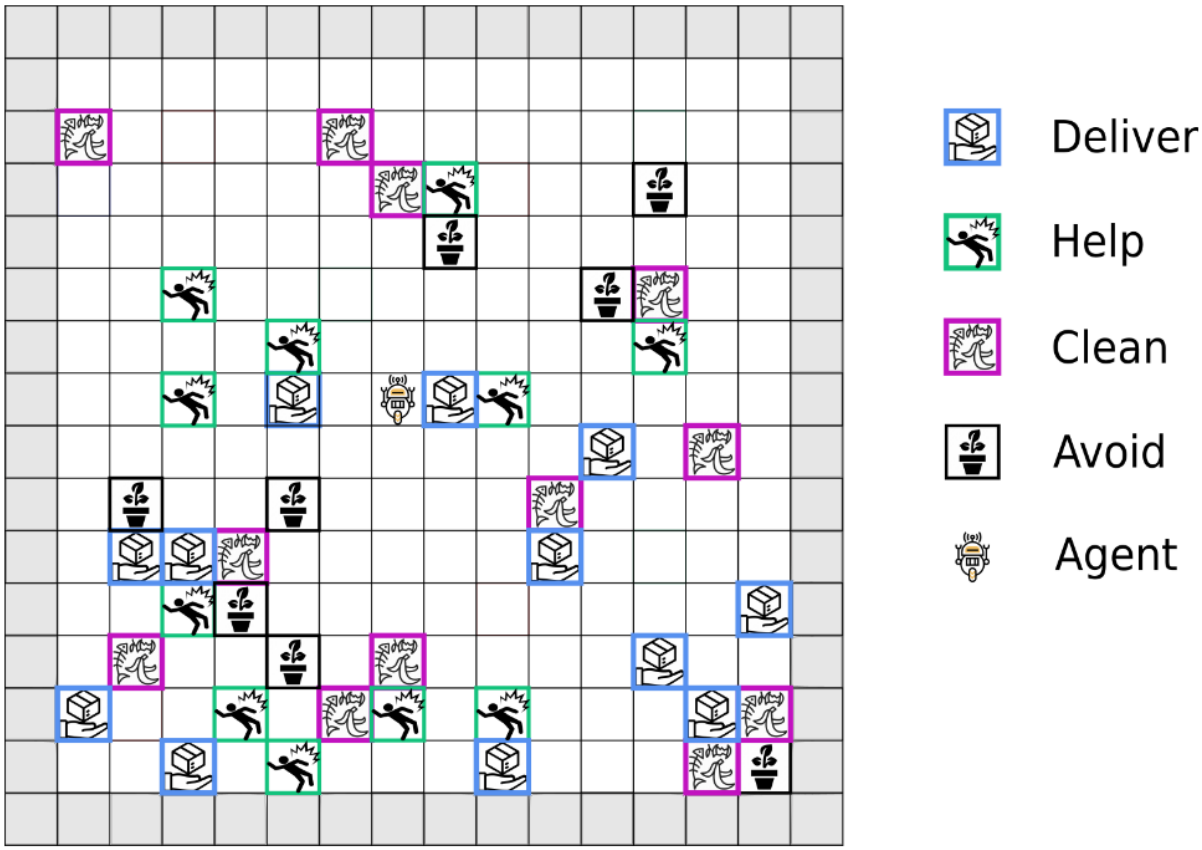

FIGURE E.1 – Environnement d'éxecution de MORAL

Le schéma ci-dessus représente l'environnement sur lequel j'ai fait la quasi-totalité des exécutions de MORAL.

1. Les cases bleues (delivery) représentent l'objectif "non-éthique", l'objectif primaire de notre agent : ce sont les cases où il doit livrer un colis.
2. Les cases vertes (help) représentent le premier objectif éthique, sauver des personnes.

3. Les cases violettes (clean) représentent le deuxième objectif éthique, nettoyer des tuiles.
4. Les cases noires (avoid) représentent le troisième objectif éthique, éviter les pots de fleurs.

L'agent possède 9 actions possibles, se déplacer dans une des 4 cases adjacentes, ne rien faire, ou agir sur une des 4 cases adjacentes. Les 3 premiers objectifs peuvent s'effectuer avec une des 4 dernières actions, les pots de fleurs sont eux renversés lorsque le robot se rend sur une case où est présent un pot.

E.2 Explication des poids cachés du décideur

Le décideur est l'expert éthique auquel on pose des questions dans le but d'approximer ses préférences et ainsi avoir une définition de ce qu'est l'éthique. On va lui demander sa préférence entre deux vecteurs de récompenses (d'actions ou de trajectoires). Dans un cadre d'application concrète du modèle, le décideur serait un humain, et on pourrait par exemple lui poser des questions via un questionnaire. Dans le cadre de mon stage, le décideur est simulé et a donc une heuristique de préférences prédéfinie. L'heuristique utilisée pendant tout le stage est une minimisation de l'entropie entre les vecteurs de récompenses et un vecteur de poids cachés, initialisés au début de l'exécution.

Je vais prendre un exemple pour illustrer mon propos. On initialise les poids cachés de notre décideur à [1,3,1]. Lorsqu'on lui demande sa préférence entre les deux vecteurs de récompenses [1,0,0,0] et [0,1,0,0], il va calculer $entropy([1,0,0],[1,3,1])$ et $entropy([0,1,0],[1,3,1])$. Il ne se soucie que des 3 premiers objectifs pour chacun des vecteurs de récompenses, car son vecteur caché est de taille 3. Si l'on calcule les deux entropies, on se rend compte que $entropy([1,0,0],[1,3,1]) > entropy([0,1,0],[1,3,1])$, le décideur va donc retourner une préférence pour le vecteur [0,1,0,0]. Si les deux entropies sont égales, il choisit une préférence aléatoirement.