Sur le Gradient de la Politique pour les Systèmes Multi-Agents Coopératifs

Guillaume Bono, Jilles S Dibangoye, Laëtitia Matignon, Florian Pereyron, Olivier Simonin

Sur le Gradient de la Politique pour les Systèmes Multi-Agents Coopératifs

G. Bono[1] J. Dibangoye[1] L. Matignon[1,2] F. Pereyron[3] O. Simonin[1]

[1] Univ Lyon, INSA Lyon, INRIA, CITI, F-69621 Villeurbanne, France
[2] Univ Lyon, Université Lyon 1, LIRIS, CNRS, UMR5205, Villeurbanne, F-69622, France
[3] Volvo Group, Advanced Technology and Research

guillaume.bono@inria.fr

Résumé

L'apprentissage par renforcement (RL) pour les processus décisionnels de Markov partiellement observables décentralisés (Dec-POMDPs) accuse un certain retard par rapport aux progrès spectaculaires du RL mono-agent. Ceci s'explique en partie par un certain nombre d'hypothèses valables dans le cadre mono-agent, mais invalides dans les systèmes multi-agents. Pour combler ce retard, nous explorons les fondements mathématiques des méthodes par ascension du gradient de la politique dans le paradigme de l'entraînement centralisé pour un contrôle décentralisé (CTDC). Dans ce paradigme, l'apprentissage peut avoir lieu de façon centralisée tout en gardant la contrainte d'une exécution décentralisée. En partant de cette intuition, nous établissons dans ce document une extension multi-agents du théorème du gradient de la politique et du théorème de compatibilité des fonctions d'approximation de la valeur. Nous en tirons des méthodes « acteur critique » (AC) qui parviennent (i) à estimer le gradient de la politique à partir d'expériences collectives mais aussi (ii) à préserver le contrôle décentralisé du système à l'exécution. Nos expérimentations montrent que nos méthodes ne souffrent pas de la comparaison avec les techniques standard en RL sur un ensemble de bancs de test de la littérature.

Mots Clef

Contrôle décentralisé et stochastique – Processus Décisionnel de Markov Partiellement Observable – Systèmes Multi-Agents – Méthodes Acteur Critique

Abstract

Reinforcement Learning (RL) for decentralized partially observable Markov decision processes (Dec-POMDPs) is lagging behind the spectacular breakthroughs of single-agent RL. That is because assumptions that hold in single-agent settings are often obsolete in decentralized multi-agent systems. To tackle this issue, we investigate the foundations of policy gradient methods within the centralized training for decentralized control (CTDC) paradigm. In this paradigm, learning can be accomplished in a centralized manner while each agent can still execute its policy independently at deployment. Using this insight, we establish a new policy gradient theorem and compatible function approximations for decentralized multi-agent systems. Resulting actor critic methods preserve the decentralized control at the execution phase, but can also estimate the policy gradient from collective experiences guided by a centralized critic at the training phase. Experiments demonstrate our policy gradient methods compare favorably against standard RL techniques in benchmarks from the literature.

Keywords

Decentralized and Stochastic Control – Partially Observable Markov Decision Processes – Multi-Agent Systems – Actor Critic Methods.

1 Introduction

Ces dernières années, la capacité d'agents artificiels à apprendre des comportements par eux-mêmes en interagissant avec leur environnement s'est développée de façon spectaculaire [18, 19], promettant de grandes avancées dans la société et l'industrie. Une partie de ces progrès peut être imputée à l'apprentissage par renforcement (RL) mono-agent, particulièrement le RL profond. Il s'agit d'une branche de l'apprentissage automatique où le monde dans lequel évolue l'agent est décrit comme un processus décisionnel de Markov (MDP) [24]. Si d'autres agents interviennent, ils sont intégrés à cette description du monde, et les hypothèses considérées pendant l'apprentissage et l'exécution sont identiques. Dans ce modèle, les méthodes par ascension du gradient de la politique et les algorithmes *acteur critique* (AC) et *acteur critique naturel* (NAC) ont eu un certain succès, avec de bonnes garanties de convergence [1, 25, 14, 6]. Ces méthodes cherchent directement dans un espace de politiques stochastiques paramétrées en ajustant la valeur des paramètres dans la direction du gradient de la politique. Malheureusement, les adaptations de ces méthodes au cadre multi-agents se sont focalisées soit sur des agents indépendants [28, 21], soit sur des systèmes

où la connaissance du monde est commune [31]. Ces derniers peuvent en fait être ramenés à des systèmes mono-agent.

Au lieu de cela, nous considérons dans cet article un système multi-agents coopératif où l'apprentissage est centralisé, mais où l'exécution reste décentralisée. Grâce à ce paradigme, nous pouvons relâcher la contrainte de contrôle décentralisé pendant la phase d'apprentissage. Nous préservons néanmoins l'indépendance des politiques apprises pour leur exécution. Dans de nombreux cas d'application des systèmes multi-agents, les conditions lors de l'entraînement ne sont pas aussi strictes que les conditions à l'exécution : pendant les répétitions d'une pièce de théâtre, les acteurs peuvent lire le script, prendre des pauses ou échanger avec le metteur en scène ; pour gagner un match de football, les joueurs appliquent des tactiques développées avec leur entraîneur bien avant la rencontre, etc. Il est donc naturel de se demander si les méthodes d'ascension du gradient de la politique peuvent tirer profit de ce paradigme.

Le paradigme CTDC a été utilisé avec succès à des méthodes de planification pour les MDP partiellement observables décentralisés (Dec-POMDP), un modèle de choix pour la prise de décision séquentielle par un ensemble d'agents coopérant à la poursuite d'un objectif commun [4, 13, 26, 20, 10, 27, 8, 9]. Dans la littérature de la théorie des jeux, les Dec-POMDP sont des jeux stochastiques partiellement observables à gains identiques. Les Dec-POMDP généralisent d'autres modèles multi-agents collaboratifs, comme les MDP multi-agents [5], ou les jeux stochastiques à gains identiques [23] par exemple. L'hypothèse principale qui rend les Dec-POMDP fondamentalement différents et plus complexes que les MDP pourrait n'être pertinente que pendant la phase d'exécution : les agents n'ont pas accès au véritable état du monde, et ne peuvent pas transmettre aux autres agents leurs observations locales bruitées. Rien n'empêche les agents de partager leurs informations locales pendant l'entraînement à la condition qu'à l'exécution, ils ne basent leurs décisions que sur leurs expériences individuelles. De façon assez surprenante, cette intuition a été relativement peu exploitée, et le traitement formel du paradigme CTDC a fait l'objet de peu d'attention au sein de la communauté RL [16]. Quand cet entraînement centralisé a lieu en laboratoire ou se base sur un simulateur, il est possible d'exploiter l'information disponible plus riche que celle collectée lors de l'exécution, par exemple les états cachés, les informations locales d'autres agents, etc. Même si les travaux récents en RL multi-agents – et RL profond – partent de ce paradigme pour développer des méthodes spécifiques à certains domaines [12, 17, 11], les fondements théoriques du RL multi-agents décentralisé en sont encore à leurs balbutiements.

Cet article explore les fondements théoriques des méthodes du gradient de la politique dans le paradigme CTDC. Dans ce paradigme, les algorithmes acteur critique peuvent être adaptées pour entraîner plusieurs acteurs (ou politiques) indépendants guidés par un critique centralisé (c.à.d. une approximation de la fonction de valeur état-action) [11]. Les méthodes de cette famille diffèrent seulement par la façon dont est représenté et mis à jour ce critique centralisé. Le résultat principal de ce papier est une généralisation du théorème du gradient de la politique et du théorème de compatibilité des approximations de la valeur pour les Dec-POMDP en s'inspirant de leurs équivalents en MDP. Nous montrons en particulier que le critique centralisé est compatible avec le gradient de la politique s'il se décompose en une somme de critiques individuels, eux-mêmes combinaisons linéaires des caractéristiques de la politique individuelle correspondante. De plus, nous dérivons des règles de mise à jour qui ajustent les paramètres des critiques individuels dans la direction du gradient du critique centralisé. Nos expérimentations tendent à montrer que notre approche est compétitive par rapport aux techniques issues des paradigmes classiques du RL sur un jeu de bancs de test de la littérature.

Le reste de l'article est structuré de la façon suivante. La Section 2 revient sur la définition formelle des MDP partiellement observables (POMDP) et Dec-POMDP, et rappelle un ensemble de propriétés pertinentes. Dans la Section 3, nous passons en revue les méthodes du gradient de la politique pour les POMDP (mono-agent), puis nous poursuivons par une analyse des travaux les utilisant dans le cadre multi-agents coopératif en Section 4. La Section 5 présente nos résultats sur les fondements théoriques des méthodes du gradient de la politique pour les Dec-POMDP, et en dérive des algorithmes acteur-critique. Nous concluons par la présentation et l'analyse de nos résultats empiriques dans la Section 6

2 Contexte

2.1 MDP partiellement observables

Considérons un agent jouant le rôle de coordinateur central chargé de contrôler le comportement d'un processus décisionnel de Markov partiellement observable (POMDP) au cours de son évolution dans le temps. Ce cadre sert souvent à formaliser les systèmes multi-agents coopératifs dans lesquels tous les agents se transmettent explicitement et instantanément leurs observations bruitées.

Définition 1. *Soit* $M_1 \doteq (\mathcal{X}, \mathcal{U}, \mathcal{Z}, p, R, T, s_0, \gamma)$ *un POMDP. On note* X_t, U_t, Z_t *et* R_t *les variables aléatoires qui prennent respectivement leurs valeurs dans* $\mathcal{X}$, $\mathcal{U}$, $\mathcal{Z}$ *et* $\mathbb{R}$ *et qui décrivent : l'état de l'environnement, les commandes appliquées par l'agent, les observations et le signal de récompense qu'il a reçu à l'instant* $t = 0, 1, \ldots, T$. *La dynamique de l'environnement est modélisée par les probabilités de transition et d'observation* $p(x', z|x, u) \doteq \mathbb{P}(X_{t+1} = x', Z_{t+1} = z|X_t = x, U_t = u)$. $R(x, u) \doteq \mathbb{E}[R_{t+1}|X_t = x, U_t = u]$ *dénote l'espérance de la récompense immédiate,* $s_0(x) = \mathbb{P}(X_0 = x)$ *la distribution initiale sur les états, et* $\gamma \in [0, 1]$ *le facteur d'amortissement des récompenses futures.*

On définit récursivement l'historique $o_t \doteq (o_{t-1}, u_{t-1}, z_t)$

avec $o_0 \doteq \varnothing$, contenant la séquence de commandes et d'observations que l'agent a effectuées et perçues jusqu'au temps t. On note $\mathcal{O}_t$ l'ensemble des historiques potentiellement expérimentés par l'agent au temps t.

Définition 2. *L'agent sélectionne à chaque instant une commande u_t selon une politique paramétrée $\pi \doteq (a_0, a_1, \ldots, a_T)$, où $a_t(u_t|o_t) \doteq \mathbb{P}(u_t|o_t;\theta_t)$ dénote la règle de décision appliquée au temps t, de paramètres $\theta_t \in \mathbb{R}^{\ell_t}$ avec $\ell_t \ll |\mathcal{O}_t|$, $\forall t \in \{0 \ldots T\}$.*

En pratique, la politique peut être construite comme par exemple un réseau de neurone profond, un contrôleur à états finis ou encore une simple structure linéaire normalisée par *softmax*. De telles représentations de la politique reposent sur différentes descriptions parfois incomplètes des historiques, appelées états internes de l'agent. Notons que si le modèle est connu ou estimé, il peut servir à calculer une forme d'états internes appelés croyances qui constituent une statistique exhaustive de l'historique [2]. Si on note $b^o \doteq \mathbb{P}(X_t|O_t = o)$ la croyance courante induit par l'historique o, avec une croyance initial $b^{\varnothing} \doteq s_0$, alors, après avoir effectué la commande $u \in \mathcal{U}$ et perçu l'observation $z \in \mathcal{Z}$, la nouvelle croyance se calcule ainsi : $\forall x' \in \mathcal{X}$,

$$\begin{aligned} b^{o,u,z}(x') &\doteq \mathbb{P}(X_{t+1} = x'|O_{t+1} = (o,u,z)) \\ &\propto \sum_{x\in\mathcal{X}} p(x', z|x, u) b^o(x). \end{aligned}$$

Par conséquent, utiliser les croyances au lieu des historiques dans la description des politiques préserve la capacité à agir de façon optimale, tout en réduisant significativement l'empreinte mémoire de la représentation de l'état interne. Cela permet aussi de se focaliser sur des politiques stationnaires, particulièrement intéressantes dans le cadre à horizon infini ($T = \infty$). Une politique π est dite stationnaire si $a_0 = a_1 = \ldots = a$, ou dit autrement $\theta_0 = \theta_1 = \ldots = \theta$; sinon, elle est non stationnaire.
En évoluant dans l'environnement guidé par la politique π, l'agent génère une trajectoire composée de récompenses, d'observations, de commandes et d'états $\omega_{t:T} \doteq (r_{t:T}, x_{t:T}, z_{t:T}, u_{t:T})$. Chaque trajectoire rapporte une récompense cumulée $R(\omega_{t:T}) \doteq \gamma^0 r_t + \cdots + \gamma^{T-t} r_T$. Les meilleures politiques sont celles qui produisent à partir de s_0 les meilleures récompense cumulée en espérance :

$$\begin{aligned} J(s_0;\theta_{0:T}) &\doteq \mathbb{E}_{\Omega_{0:T}\sim\mathbb{P}(\cdot|\pi,M_1)}[R(\Omega_{0:T})] \\ &= \sum_{\omega_{0:T}} \mathbb{P}(\omega_{0:T}|\pi, M_1) R(\omega_{0:T}) \end{aligned} \quad (1)$$

où on note $\mathbb{P}(\omega_{0:T}|\pi, M_1)$ la probabilité de générer la trajectoire $\omega_{0:T}$ en suivant π. Trouver le meilleur moyen pour l'agent d'interagir avec M_1 consiste à trouver le vecteur de paramètres $\theta^*_{0:T}$ solution de : $\theta^*_{0:T} \in \arg\max_{\theta_{0:T}} J(s_0;\theta_{0:T})$.
Il s'avère judicieux de subdiviser la performance d'une politique pour exploiter sa structure sous-jacente : en effet, les performances d'une politique π à partir du temps t ne dépendent des commandes antérieures à t qu'à travers les états et historiques courants. Cela nous conduit à définir les fonctions de valeur, Q-valeur et avantage de π.
La fonction de Q-valeur de π est défini par :

$$Q^\pi_t : (x,o,u) \mapsto \mathbb{E}[R(\Omega_{t:T})] \quad (2)$$

où $\Omega_{t:T} \sim \mathbb{P}(\cdot|X_t = x, O_t = o, U_t = u; \pi, M_1)$.
$Q^\pi_t(x,o,u)$ est l'espérance sur la récompense cumulée en exécutant au temps t la commande u à partir d'un état x et d'un historique o, puis en choisissant par π les futures commandes à partir de $t+1$. La fonction de valeur de π est donnée par :

$$V^\pi_t : (x,o) \mapsto \mathbb{E}_{U\sim a_t(\cdot|o)}[Q^\pi_t(x,o,U)] \quad (3)$$

où $V^\pi_t(x,o)$ est l'espérance sur la récompense cumulée en suivant la politique π à partir du temps t depuis un état x et un historique o. Enfin, la fonction avantage de π est simplement la différence des deux :

$$A^\pi_t : (x,o,u) \mapsto Q^\pi_t(x,o,u) - V^\pi_t(x,o) \quad (4)$$

où $A^\pi_t(x,o,u)$ est donc l'avantage relatif à exécuter u plutôt que de suivre π au temps t depuis un état x et un historique o, puis en revenant à la politique π par la suite. La propriété qui rend ces fonctions intéressantes est qu'elles vérifient la relation de récurrence suivante.

Lemme 1 (Équations de Bellman [3])**.** $\forall t = 0, 1, \ldots, T$, $\forall x \in \mathcal{X}, o \in \mathcal{O}_t, u \in \mathcal{U}$,

$$Q^\pi_t(x,o,u) = R(x,u) + \gamma \mathbb{E}[Q^\pi_{t+1}(X', O', U')] \quad (5)$$

où $(X', O', U') \sim \mathbb{P}(\cdot|X_t = x, O_t = o, U_t = u; a_{t+1}, M_1)$

(5) établit une relation temporelle entre les différents $V^\pi_{0:T}$, $Q^\pi_{0:T}$ et $A^\pi_{0:T}$, mais aussi $J(s_0;\theta_{0:T})$:

$$J(s_0;\theta_{0:T}) = \mathbb{E}_{X_0\sim\mathbb{P}(\cdot|s_0)}[V^\pi_0(X_0, o_0 = \varnothing)]. \quad (6)$$

Jusqu'à présent, nous nous sommes concentrés sur les systèmes contrôlés par un unique agent. La suite généralise aux systèmes décentralisés dans lesquels plusieurs agents doivent coopérer pour contrôler le même système.

2.2 POMDP Décentralisés

Considérons à présent un contexte où n agents coopèrent pour influencer l'évolution d'un système décrit comme un POMDP, mais où aucun d'eux ne peut percevoir l'état du monde, ni communiquer ses observations bruitées aux autres.

Définition 3. *Un Dec-POMDP $M_n \doteq (\mathcal{I}_n, \mathcal{X}, \mathcal{U}, \mathcal{Z}, p, R, T, \gamma, s_0)$ est défini tel que : $i \in \mathcal{I}_n$ dénote l'indice du $i^{ème}$ agent impliqué dans le processus ; $\mathcal{X}, \mathcal{U}, \mathcal{Z}, p, R, T, \gamma$ et s_0 sont définis tels que dans M_1, à la particularité que $\mathcal{U}$ et $\mathcal{Z}$ se décomposent en ensembles de commandes et d'observations individuelles $\mathcal{U} = \mathcal{U}^1 \times \cdots \times \mathcal{U}^n$ et $\mathcal{Z} = \mathcal{Z}^1 \times \cdots \times \mathcal{Z}^n$, avec $\mathcal{U}^i$ et $\mathcal{Z}^i$ l'ensemble des commandes et des observations propres au $i^{ème}$ agent.*

On appelle historique individuel de l'agent $i \in \mathcal{I}_n$ la séquence de commandes et d'observations individuelles exécutées et perçues jusqu'au temps $t = 0, 1, \ldots, T$, notée $o_t^i = (o_{t-1}^i, u_{t-1}^i, z_t^i)$ avec $o_0^i = \varnothing$. On note $\mathcal{O}_t^i$, l'ensemble des historiques individuels possible de l'agent i au temps t.

Définition 4. *L'agent $i \in \mathcal{I}_n$ sélectionne la commande u_t^i au temps t selon une politique paramétrée $\pi^i \doteq (a_0^i, a_1^i, \ldots, a_T^i)$ où $a_t^i(u_t^i|o_t^i) \doteq \mathbb{P}(u_t^i|o_t^i; \theta_t^i)$ est la règle de décision appliquée au temps t, de paramètres $\theta_t^i \in \mathbb{R}^{\ell_t^i}$, avec $\ell_t^i \ll |\mathcal{O}_t^i|$.*

De même que dans M_1, le nombre d'historiques individuels possibles grandit de façon exponentiel à chaque pas de temps. À ce jour, la seule statistique exhaustive connue pour les historiques individuels repose sur les états d'occupation définis par : $s_t(x, o) \doteq \mathbb{P}(x, o|\theta_{0:t-1}^{1:n}), \forall x \in \mathcal{X}, \forall o \in \mathcal{O}_t$. L'état d'occupation individuel induit par l'historique $o^i \in \mathcal{O}_t^i$ est une distribution de probabilité conditionnelle : $s_t^i(x, o^{-i}) \doteq \mathbb{P}(x, o^{-i}|o^i, s_t)$, où o^{-i} dénote l'historique joint des $n-1$ agents autres que i. Il est très difficile d'apprendre à projeter les historiques individuels vers des états internes proches des états d'occupation individuels, ce qui limite la capacité des algorithmes d'apprentissage à trouver des politiques optimales dans un temps raisonnable pour M_n. On peut alors se focaliser sur des politiques stationnaires pour lesquelles l'espace des historiques est projeté sur un ensemble fini d'états internes $\varsigma \doteq (\varsigma^1, \ldots, \varsigma^n)$, qui sont des représentations – souvent à pertes – des états d'occupations individuels. Les ς^i peuvent par exemple être les nœuds d'un contrôleur à états finis, ou encore les états internes d'un réseau de neurone récurrent (RNN). On note $\psi(\varsigma'|\varsigma, u, z)$ la probabilité de transition d'un état interne joint ς à l'état interne suivant ς' en fonction de la dernière commande effectuée u et de l'observation reçue z. Cette loi de transition est décomposable en lois individuelles $\psi^i(\varsigma'^{,i}|\varsigma^i, u^i, z^i)$. Dans la suite de cet article, nous considérerons la loi ψ fixée à priori, même si en général, elle fait partie de la politique et possède des paramètres à optimiser. Résoudre M_n revient à trouver une politique jointe $\pi \doteq (\pi^1, \ldots, \pi^n)$ – un tuple de n politiques individuelles – qui donne la meilleure récompense cumulée à partir d'une distribution d'états initiaux s_0. Autrement dit, $\theta_{0:T}^{*,1:n} \in \arg\max_{\theta_{0:T}^{1:n}} J(s_0; \theta_{0:T}^{1:n})$, avec

$$\begin{aligned} J(s_0; \theta_{0:T}^{1:n}) &\doteq \mathbb{E}_{\Omega_{0:T} \sim \mathbb{P}(\cdot|\pi, M_n)}[R(\Omega_{0:T})] \\ &= \sum_{\omega_{0:T}} \mathbb{P}(\omega_{0:T}|\pi, M_n) R(\omega_{0:T}) \end{aligned} \tag{7}$$

où $\mathbb{P}(\omega_{0:T}|\pi, M_n)$ est la probabilité de générer la trajectoire jointe $\omega_{0:T}$ en suivant la politique π. Étant donné une politique jointe π, M_n hérite des mêmes définitions que M_1, en particulier pour les fonctions $V_{0:T}^\pi$, $Q_{0:T}^\pi$ et $A_{0:T}^\pi$.

3 Gradient de la politique

Dans cette section, nous passons en revue la littérature des méthodes du gradient de la politique pour le contrôle centralisé de systèmes mono-agent. Dans ce contexte, l'approche du gradient de la politique se base sur des algorithmes centralisés qui cherchent à optimiser la récompense cumulée directement dans l'espace des paramètres de la politique $\theta_{0:T}$. Bien que nous n'abordions ici que des politiques non stationnaires, les méthodes citées peuvent facilement être étendues au cas de politiques stationnaires pour lesquelles $a_t = a$ pour tout t. En considérant que la politique π est différentiable par rapport à ses paramètres $\theta_{0:T}$, l'algorithme met à jour $\theta_{0:T}$ dans la direction du gradient :

$$\Delta\theta_{0:T} = \alpha \frac{\partial J(s_0; \theta_{0:T})}{\partial \theta_{0:T}} \tag{8}$$

avec α un taux d'apprentissage. En itérant sur cette règle de mise à jour, pour peu que l'estimation du gradient soit correcte, $\theta_{0:T}$ converge vers un optima local. Malheureusement, estimer correctement le gradient peut se révéler impossible. Pour dépasser cette difficulté, le gradient peut être remplacé par une estimation non biaisée, ce qui revient à restreindre (8) à un gradient stochastique :

$$\Delta\theta_{0:T} = \alpha R(\omega_{0:T}) \frac{\partial \log \mathbb{P}(\omega_{0:T}|\pi, M_1)}{\partial \theta_{0:T}} \tag{9}$$

On calcule $\frac{\partial}{\partial \theta_{0:T}} \log \mathbb{P}(\omega_{0:T}|\pi, M_1)$ sans connaissance à priori de la distribution sur les trajectoires $\mathbb{P}(\omega_{0:T}|\pi, M_1)$. En effet, les propriétés d'indépendance conditionnelle du processus donnent :

$$\mathbb{P}(\omega_{0:T}|\pi, M_1) \doteq s_0(x_0) \prod_{t=0}^{T} p(x_{t+1}, z_{t+1}|x_t, u_t) a_t(u_t|o_t) \tag{10}$$

ce qui implique :

$$\frac{\partial \log \mathbb{P}(\omega_{0:T}|\pi, M_1)}{\partial \theta_{0:T}} = \sum_{t=0}^{T} \frac{\partial \log a_t(u_t|o_t)}{\partial \theta_t} \tag{11}$$

3.1 Méthode du rapport de vraisemblance

Les méthodes du rapport de vraisemblances, *e.g.,* `Reinforce` [29], exploitent la séparabilité du vecteur de paramètres $\theta_{0:T}$, qui mènent à la règle de mise à jour suivante : $\forall t = 0, 1, \ldots, T$

$$\Delta\theta_t = \alpha \mathbb{E}_{\mathcal{D}} \left[R(\omega_{0:T}) \frac{\partial \log a_t(u_t|o_t)}{\partial \theta_t} \right] \tag{12}$$

où $\mathbb{E}_{\mathcal{D}}[\cdot]$ dénote une approximation de l'espérance par une moyenne empirique prise sur un ensemble de trajectoires $\mathcal{D} = \{\omega_{0:T,j}\}_{1 \leqslant j \leqslant m}$ échantillonnées à partir de π et M_1. Le principal problème rencontré en utilisant cette règle de mise à jour centralisé est la variance importante de $R(\omega_{0:T})$, qui peut fortement ralentir la convergence. Pour compenser partiellement cette variance, deux constatations rapides permettent d'améliorer la règle de mise à jour. Premièrement, les futures actions ne dépendent pas des récompenses passées, *c.à.d.* $\mathbb{E}_{\mathcal{D}}[R(\omega_{0:t-1}) \frac{\partial}{\partial \theta_t} \log a_t(u_t|o_t)] =$

0, On peut donc utiliser $R(\omega_{t:T})$ au lieu de $R(\omega_{0:T})$ dans 12, ce qui réduit significativement la variance de notre estimation du gradient. Deuxièmement, l'estimation du gradient de la politique n'est pas biaisée par l'introduction d'une fonction de référence $\tilde{\beta}_t$ (en anglais : *baseline*) tant que celle-ci ne dépend pas de $\theta_{0:T}$. On peut ainsi remplacer $R(\omega_{t:T})$ par une récompense relative à cette référence $R(\omega_{t:T}) - \tilde{\beta}_t(x_t, o_t)$.

3.2 Acteur Critique

Pour améliorer la variance de l'estimation du gradient dans (12), le théorème du gradient de la politique [25] propose de remplacer $R(\omega_{t:T})$ par $Q_t^w(x_t, o_t, u_t)$, une approximation de la Q-valeur associée à la commande u_t dans l'état x_t et après un historique o_t puis en complétant la trajectoire en suivant la politique π : $Q_t^w(x_t, o_t, u_t) \approx Q_t^\pi(x_t, o_t, u_t)$, où $w_t \in \mathbb{R}^{l_t}$ est un vecteur de paramètres, avec $l_t \ll |\mathcal{X}||\mathcal{O}_t||\mathcal{U}|$. Ceci conduit au schéma algorithmique *acteur critique*, dans lequel un algorithme centralisé met à jour à la fois les paramètres $\theta_{0:T}$ de la politique et les paramètres $w_{0:T}$ de l'approximation de la Q-valeur :

$$\Delta w_t = \alpha \mathbb{E}_{\mathcal{D}} \left[\delta_t \frac{\partial \log a_t(u_t|o_t)}{\partial \theta_t} \right] \tag{13a}$$

$$\Delta \theta_t = \alpha \mathbb{E}\mathcal{D} \left[Q_t^w(x_t, o_t, u_t) \frac{\partial \log a_t(u_t|o_t)}{\partial \theta_t} \right] \tag{13b}$$

où on note $\delta_t \doteq \widehat{Q}_t^\pi(x_t, o_t, u_t) - Q_t^w(x_t, o_t, u_t)$, avec $\widehat{Q}_t^\pi(x_t, o_t, u_t)$ une estimation non-biaisée de la véritable Q-valeur $Q_t^\pi(x_t, o_t, u_t)$, utilisant par exemple une estimation Monte-Carlo (MC) de $R(\omega_{0:T})$ ou bien une différence temporelle (erreur TD) $r_t + \gamma Q_{t+1}^w(x_{t+1}, o_{t+1}, u_{t+1})$. Le choix des paramètres $w_{0:T}$ est crucial pour que l'estimation du gradient reste non-biaisé [25]. Il n'y a pas de biais si les approximations de Q-valeur $Q_{0:T}^w$ sont *compatibles* avec la politique π. Sans rentrer dans les détails formels, une approximation compatible $Q_{0:T}^w$ de la véritable fonction $Q_{0:T}^\pi$ peut être une combinaison linéaire des « caractéristiques » de la politique π, et ses paramètres $w_{0:T}$ constituent la solution du problème de régression linéaire qui estime $Q_{0:T}^\pi$ à partir de ces caractéristiques. En pratique, cette seconde condition n'est pas utilisée directement, et les paramètres $w_{0:T}$ sont mis à jour en utilisant par exemple des méthodes MC ou TD.

3.3 Acteur Critique Naturel

Suivre la direction du gradient n'est pas toujours le moyen le plus rapide pour converger. Le gradient naturel propose une mise à jour des paramètres $\theta_{0:T}$ selon la direction de plus rapide ascension par rapport à la métrique d'information de Fisher :

$$\mathbf{\Phi} \doteq \mathbb{E}_{\mathcal{D}} \left[\frac{\partial \log a_t(u_t|o_t)}{\partial \theta_t} \left(\frac{\partial \log a_t(u_t|o_t)}{\partial \theta_t} \right)^\top \right] \tag{14}$$

Cette métrique est invariante à une reparamétrisation de la politique. Combiner le théorème du gradient de la politique avec une approximation compatible de la Q-valeur, puis effectuer une mise à jour des paramètres dans la direction donnée par (14), donne naissance au schéma algorithmique *acteur-critique naturel*, qui remplace (13b) dans (13) par : $\Delta\theta_t = \alpha \mathbb{E}_{\mathcal{D}}[w_t]$.

4 Gradient de la politique pour les systèmes multi-agents

Dans cette section, nous passons en revue les extensions des méthodes du gradient de la politique mono-agent à des modèles multi-agents coopératifs. On distingue trois paradigmes, illustrés dans la Figure 1 : l'entraînement centralisé pour un contrôle centralisé (CTCC), l'entraînement distribué pour un contrôle décentralisé (DTDC), et l'entraînement centralisé pour un contrôle décentralisé (CTDC).

4.1 CTCC

Dans certaines applications des systèmes multi-agents coopératifs, on peut considérer des agents pouvant communiquer entre eux de façon parfaite, instantanée et illimitée. De telles applications peuvent être modélisées par des POMDP, ce qui rend possible d'utiliser les méthodes du gradient de la politique mono-agent (Section 3). Dans ce paradigme CTCC, illustré par la Figure 1 *(gauche)*, un seul acteur « coordinateur » guidé par un critique unique est utilisé. Le principal défaut de ce paradigme fait aussi sa force : ce besoin de communication parfaite, instantanée et illimitée entre les agents d'un bout à l'autre de l'exécution, à la fois pendant l'entraînement et le déploiement.

4.2 DTDC

Assez étonnamment, il s'avère que la première adaptation des méthodes du gradient de la politique au cadre multi-agent visait à apprendre de façon distribuée les politiques à exécuter de façon décentralisée, comme par exemple l'algorithme `Reinforce` distribué [21]. Dans ce paradigme DTDC, les agents apprennent simultanément mais indépendamment leurs politiques individuelles en utilisant l'algorithme Reinforce avec chacun un critique et un acteur individuels, comme l'illustre la Figure 1 *(droite)*. L'indépendance des vecteurs de paramètres $\theta_{0:T}^1, \theta_{0:T}^2, \ldots, \theta_{0:T}^n$ donne lieu à la règle de mise à jour suivante : $\forall t = 0, 1, \ldots, T, \forall i \in \mathcal{I}_n$,

$$\Delta\theta_t^i = \alpha \mathbb{E}_{\mathcal{D}} \left[R(\omega_{0:T}) \frac{\partial \log a_t^i(u_t^i|o_t^i)}{\partial \theta_t^i} \right] \tag{15}$$

Il est à noter que la somme des gradients des politiques individuelles est un estimateur non-biaisé du gradient de la politique jointe. Néanmoins, la question reste ouverte de savoir comment tirer profit des méthodes acteur-critique (voir Section 3) pour réduire la variance de cet estimateur joint. De plus, l'algorithme `Reinforce` distribué est restreint à un apprentissage où l'échantillonnage des trajectoires est dicté par la politique apprise π (*on-policy*), quand des politiques d'exploration mieux choisies $\bar{\pi}$ peuvent for-

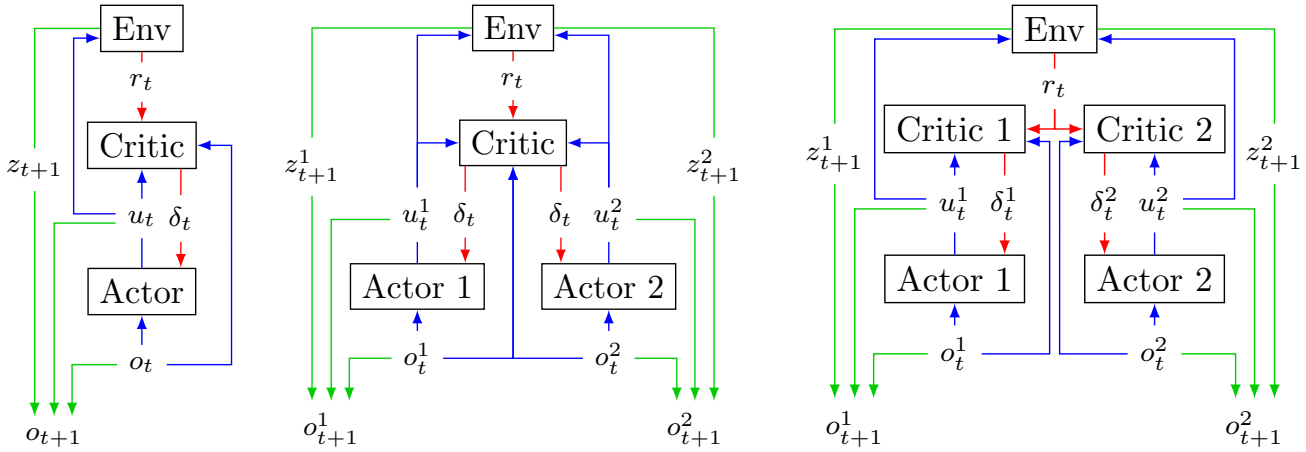

FIGURE 1 – Schéma algorithmique acteur-critique dans un modèle à deux agents pour les paradigmes : (*gauche*) CTCC, (*centre*) CTDC, et (*droite*) DTDC. Pour chaque figure, les flèches bleues représentent les commandes des agents sur l'environnement, les vertes montrent l'agrégation d'information pour le prochain pas de décision, et les rouges indiquent le signal de récompense propagé pour mettre à jour les différents paramètres.

tement améliorer la qualité de l'optimum local atteint à la convergence [6].

4.3 CTDC

Le paradigme CTDC a été employé avec succès pour résoudre des problèmes de planification de Dec-POMDP [4, 13, 26, 20, 10, 16, 27, 8, 9]. Dans un tel paradigme, un coordinateur central apprend pour l'ensemble des agents pendant la phase d'entraînement puis assigne les politiques individuelles apprises aux agents correspondants avant le début de la phase d'exécution. Comme le montre la Figure 1 *(centre)*, les algorithmes acteur critique dans ce paradigme CTDC mettent à jour un critique central mais optimisent un acteur par agent. Des travaux récents en apprentissage par renforcement (profond) [12, 17, 11] exploitent ce paradigme, en particulier l'approche par gradient de la politique multi-agents contrefactuel, et l'algorithme COMA qui en découle. Malheureusement, cette algorithme se focalise sur un critique central apportant une solution au problème de l'assignation des crédits. Pour un Dec-POMDP M_n (faiblement) séparable [7], il est possible d'apprendre une contribution relative de chaque agent à la valeur de la politique jointe, apportant ainsi une réponse au problème d'assignation du crédit. En revanche, en général, la dynamique et les récompenses d'un Dec-POMDP peuvent être fortement corrélées, ce qui rend difficile voire impossible d'accorder du crédit à tel ou tel agent. Contrairement aux méthodes de planification, les méthodes d'ascension du gradient de la politique dans ce paradigme CTDC manquent de fondements théoriques, qui permettraient de définir la forme du critique centralisé et les règles de mise à jour préservant certaines garanties formelles.

5 Gradient de la politique pour les Dec-POMDP

Dans cette section, nous apportons une réponse aux limitations des paradigmes CTCC et DTDC, et étendons les schémas algorithmiques acteur-critique et acteur-critique naturel de M_1 vers M_n.

5.1 Théorème du Gradient de la Politique

Notre principal résultat est une extension du théorème du Gradient de la politique [25] de M_1 vers M_n. Avant de poursuivre, nous commençons par exposer quelques résultats préliminaires qui nous permettrons d'établir notre résultat principal de la section. Toutes les preuves sont disponibles en annexes.

Lemme 2. *Pour toute fonction séparable* $f\colon (x^1,\dots,x^n) \mapsto f^1(x^1)\cdots f^n(x^n)$, *sa dérivée partielle par rapport à n'importe lequel de ses arguments* x^j, $j \in I_n$ *peut s'écrire :* $\frac{\partial}{\partial x^j} f(x) = f(x)\frac{\partial}{\partial x^j}\log f^j(x^j)$. *pour tout* $x = (x^1,\dots,x^n)$ *où* f *est différentiable et non nulle.*

Lemme 3. *Soit deux distributions* p *et* q *portant sur la même variable aléatoire* X. *Si le support de* p *est inclus dans celui de* q *alors :* $\mathbb{E}_{X\sim p}[f(X)] = \mathbb{E}_{X\sim q}[\frac{p(X)}{q(X)} f(X)]$.

Nous établissons à présent une expression des dérivées partielles des fonctions de valeurs $V^{\pi}_{0:T}$ par rapport aux vecteurs de paramètres $\theta^{1:n}_{0:T}$ dans le cas à horizon fini.

Lemme 4. *Soit un Dec-POMDP* M_n, *une politique jointe à évaluer* $\pi \doteq (a_0,\dots,a_T)$ *et une politique d'exploration* $\bar{\pi} \doteq (\bar{a}_0,\dots,\bar{a}_T)$. *Pour tout temps* $t = 0,1,\dots,T$, *agent*

$i \in \mathcal{I}_n$, *état caché* $x_t \in \mathcal{X}$, *et historique joint* $o_t \in \mathcal{O}_t$:

$$\frac{\partial V_t^\pi(x_t, o_t)}{\partial \theta_t^i} = \mathbb{E}_{U_t \sim \bar{a}_t(\cdot|o_t)}\Big[\frac{a_t(U_t|o_t)}{\bar{a}_t(U_t|o_t)} Q_t^\pi(x_t, o_t, U_t)\frac{\partial \log a_t^i(U_t^i|o_t^i)}{\partial \theta_t^i}\Big]. \quad (16)$$

Voici à présent le résultat principal de la section.

Théorème 1. *Soit un Dec-POMDP* M_n*, une politique à évaluer* $\pi \doteq (a_0, \dots, a_T)$ *et une politique d'exploration* $\bar{\pi} \doteq (\bar{a}_0, \dots, \bar{a}_T)$.

1. *Dans le cas à horizon fini* $T < \infty$*, pour tout temps* $t = 0, 1, \dots, T$ *et agent* $i \in \mathcal{I}_n$ *:*

$$\frac{\partial J(s_0; \theta_{0:T}^{1:n})}{\partial \theta_t^i} = \gamma^t \mathbb{E}_{(X_t, O_t, U_t) \sim \mathbb{P}(\cdot|\bar{a}_t, M_n)}\Big[\frac{a_t(U_t|O_t)}{\bar{a}_t(U_t|O_t)} Q_t^\pi(X_t, O_t, U_t)\frac{\partial \log a_t^i(U_t^i|O_t^i)}{\partial \theta_t^i}\Big]$$

2. *Dans le cas à horizon infini* $T = \infty$*, pour tout agent* $i \in \mathcal{I}_n$ *:*

$$\frac{\partial J(s_0; \theta^{1:n})}{\partial \theta^i} = \mathbb{E}_{(X, \Sigma, U) \sim \mathbb{P}(\cdot|\bar{s}, \bar{a})}\Big[\frac{a(U|\Sigma)}{\bar{a}(U|\Sigma)} Q^\pi(X, \Sigma, U)\frac{\partial \log a^i(U^i|\Sigma^i)}{\partial \theta^i}\Big]$$

où

$$\bar{s}(x, \varsigma) \doteq \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(X_t = x, \Sigma_t = \varsigma | M_n, \bar{a}, \psi, X_0, \varsigma_0)$$

Le théorème du gradient de la politique pour M_n (Théorème 1) est fondamentalement différent de celui pour M_1 [25]. Ce dernier part de l'hypothèse qu'un unique agent apprend à agir dans un (PO)MDP. À l'inverse, le Théorème 1 s'applique à un ensemble d'agents apprenant à contrôler l'évolution d'un POMDP de façon décentralisée. Les agents agissent indépendamment, mais leur estimation du gradient de la politique est guidée par une évaluation centralisée des fonctions $Q_{0:T}^\pi$. Pour utiliser cette propriété en pratique, il faut remplacer la véritable Q-valeur $Q_{0:T}^\pi$ par une approximation. Pour garantir que cette approximation est compatible – *c.à.d.* que le gradient calculé avec l'approximation est toujours dans la même direction que le véritable gradient – il faut lui donner une structure particulière que nous détaillons ci-dessous.

5.2 Approximations compatibles

Le résultat principal de cette section caractérise la forme des approximations compatibles $V_{0:T}^\sigma$ et $A_{0:T}^\nu$ respectivement pour les fonctions de valeur $V_{0:T}^\pi$ et les fonctions d'avantage $A_{0:T}^\pi$ pour un Dec-POMDP quelconque M_n. Ensemble, ces deux approximations permettent d'évaluer $Q_{0:T}^\pi(x_t, o_t, u_t) \doteq V_t^\pi(x_t, o_t) + A_t^\pi(x_t, o_t, u_t)$, pour tout temps $t = 0, 1, \dots, T$, historique joint $o_t \in \mathcal{O}_t$ et commande jointe $u_t \in \mathcal{U}$.

Théorème 2. *Soit un Dec-POMDP* M_n*, des fonctions d'approximation* $V_{0:T}^\sigma$ *et* $A_{0:T}^\nu$*, de paramètres respectifs* $\sigma_{0:T}^{1:n}$ *et* $\nu_{0:T}^{1:n}$*. Ces approximations sont compatibles avec la politique jointe* $\pi \doteq (a_0, \dots, a_T)$ *paramétrée par* $\theta_{0:T}^{1:n}$ *si l'une des deux conditions suivantes est vérifiée* $\forall t = 0, 1, \dots, T$ *:*

1. $\forall i \in \mathcal{I}_n, \forall x_t \in \mathcal{X}, \forall o_t \in \mathcal{O}_t$,

$$\frac{\partial V_t^\sigma(x_t, o_t)}{\partial \sigma_t^i} = \mathbb{E}_{U_t^i \sim a_t^i(\cdot|o_t^i)}[\frac{\partial \log a_t^i(U_t^i|o_t^i)}{\partial \theta_t^i}] \quad (17)$$

et σ *minimise l'erreur quadratique moyenne (MSE)* $\mathbb{E}[\epsilon_t(X_t, O_t, U_t)^2]$

2. $\forall i \in \mathcal{I}_n, \forall x_t \in \mathcal{X}, \forall o_t \in \mathcal{O}_t, \forall u_t \in \mathcal{U}$,

$$\frac{\partial A_t^\nu(x_t, o_t, u_t)}{\partial \nu_t^i} = \frac{\partial \log a_t^i(u_t^i|o_t^i)}{\partial \theta_t^i} \quad (18)$$

et ν *minimise l'erreur quadratique moyenne (MSE)* $\mathbb{E}[\epsilon_t(X_t, O_t, U_t)^2]$

où $\epsilon_t(x_t, o_t, u_t) \doteq Q_t^\pi(x_t, o_t, u_t) - V_t^\sigma(x_t, o_t) - A_t^\nu(x_t, o_t, u_t)$.

Quand l'une de ces conditions est vérifiée, on a bien, pour toute politique d'exploration $\bar{\pi} \doteq (\bar{a}_0, \dots, \bar{a}_T)$:

$$\frac{\partial V_t^\pi(x_t, o_t)}{\partial \theta_t^i} = \mathbb{E}_{U_t \sim \bar{a}_t(\cdot|o_t)}\Big[\frac{a_t(U_t|o_t)}{\bar{a}_t(U_t|o_t)} \big(V_t^\sigma(x_t, o_t) + A_t^\nu(x_t, o_t, U_t)\big)\frac{\partial \log a_t^i(U_t^i|o_t^i)}{\partial \theta_t^i}\Big] \quad (19)$$

Le Théorème 2 est énoncé ci-dessus pour des politiques non-stationnaires et pour des problème à horizon fini $T < \infty$. Le résultat s'étend néanmoins très naturellement au cas à horizon infini et pour des politiques stationnaires où $a_t^i = a^i, \theta_t^i = \theta^i \ \forall t = 0, 1, \dots, \infty, \forall i \in \mathcal{I}_n$. Le théorème montre comment les conditions de compatibilité d'une approximation de la Q-valeur pour M_1 se généralisent dans le cas d'un Dec-POMDP M_n. Parmi les propriétés notables de ces approximations centralisées compatibles, il faut relever leur séparabilité :

$$V_t^\sigma : (x_t, o_t) \mapsto \sum_{i \in I_n} \mathbb{E}\left[\frac{\partial \log a_t^i(U_t^i|o_t^i)}{\partial \theta_t^i}\right]^\top \sigma_t^i \quad (20)$$

dans le premier cas, ou alors

$$A_t^\nu : (x_t, o_t, u_t) \mapsto \sum_{i \in I_n} \left(\frac{\partial \log a_t^i(u_t^i|o_t^i)}{\partial \theta_t^i}\right)^\top \nu_t^i \quad (21)$$

dans le second. Quel que soit le cas, à l'instar des résultats existants pour M_1, l'approximation A^ν ou V^σ dont la structure n'est pas contrainte constitue un degré de liberté très important pour tenter de réduire la variance de l'estimation du gradient. Dans notre paradigme CTDC, elle peut entre autre accéder aux état cachés x_t, ou encore combiner de façon plus intriquée l'information jointe disponible. Par

ailleurs, il est tout à fait envisageable d'exploiter la séparabilité des deux approximations compatibles :

$$Q_t^\pi(x_t, o_t, u_t) \approx \sum_{i \in I_n} \left(\frac{\partial \log a_t^i(u_t^i|o_t^i)}{\partial \theta_t^i} \right)^\top \nu_t^i + \sum_{i \in I_n} \mathbb{E}\left[\frac{\partial \log a_t^i(U_t^i|o_t^i)}{\partial \theta_t^i} \right]^\top \sigma_t^i + \tilde{\beta}_t(x_t, o_t, u_t) \quad (22)$$

avec $\tilde{\beta}_t$ une fonction de référence arbitraire, ne dépendant ni de ν, ni de σ, ni de θ et ne perturbant donc pas le gradient. Notons enfin que la séparabilité du critique centralisé ne nous permet pas de considérer les critiques individuels indépendamment l'un de l'autre, l'approximation du gradient est toujours jointe et guidée par cet unique critique centralisé.

5.3 Algorithmes Acteur Critique pour le Contrôle Décentralisé

Dans cette section, nous utilisons les résultats du Théorème 2 pour donner naissance à un schéma algorithmique acteur-critique pour M_n adapté au paradigme CTDC, que nous appelons *acteur critique pour le contrôle décentralisé* (ACDC). Cet algorithme ne nécessite pas de connaissance à priori du modèle, et l'échantillonnage des trajectoires peut se servir d'une politique d'exploration différente de la politique à évaluer. Il est centralisé [1] et itératif. Chaque itération consiste en une étape d'évaluation de la politique apprise, suivi d'une étape d'amélioration de celle-ci. L'étape d'évaluation crée une base d'échantillons de trajectoires (mini-batch) $\mathcal{D}$ à partir de $\mathbb{P}(\Omega_{0:T}|\pi, M_n)$ et stocke en mémoire les erreurs TD correspondantes (*voir* lignes 6–11). L'étape d'amélioration met à jour l'ensemble des paramètres θ, ν, et σ dans la direction du gradient estimé à partir de la moyenne empirique sur la base $\mathcal{D}$ en exploitant la séparabilité des approximations compatibles (*voir* lignes 12–16). Les taux d'apprentissage α_h^θ, α_h^ν et α_h^σ évoluent selon les conditions standard de Robbins et Monro pour les algorithmes d'approximation stochastiques [22], *c.à.d.* $\sum_{h=0}^\infty \alpha_h = \infty$, $\sum_{h=0}^\infty \alpha_h^2 < \infty$. De plus, si l'on suit les recommandations de [15], ces taux doivent être mis à jour à chaque itération tel que les paramètres des acteurs θ soient modifiés « plus lentement » que les paramètres ν and σ pour garantir la convergence. Pour faciliter la convergence d'une politique jointe pour une modification constante de ses paramètres, une méthode de choix est l'utilisation gradient naturel [1, 14]. L'algorithme ACDC naturel (NACDC) ne diffère de sa version initiale que par la formule de mise à jour des acteurs :

$$\theta_{t,h+1}^i \leftarrow \theta_{t,h}^i + \alpha_h^\theta \mathbb{E}_{\mathcal{D}_{t,h}}[\frac{a_t(u_t|o_t)}{\bar{a}_t(u_t|o_t)} \nu_{t,h}^i]$$

1. Il est néanmoins possible de le faire tourner de façon distribuée en donnant le moyen aux agents de collaborer pendant l'entraînement en se communiquant leur informations locales.

Algorithm 1: Actor-Critic for Decentralized Control (ACDC).

```
1  ACDC()
2      Initialize θ_0, ν_0, σ_0 arbitrarily and h ← 0.
3      while θ_h has not converged do
4          evaluation() and improvement()
5          h ← h + 1
6  evaluation()
7      Initialize D^h_{0:T} ← ∅
8      for j = 1...m and t = 0...T do
9          Create trajectories (x_{t:t+1}, o_{t:t+1}, u_t) ~ p(·|θ^{1:n}_{0:t})
10         Evaluate δ_t ← r_t + γV^σ_{t+1}(x_{t+1}, o_{t+1}) − V^σ_t(x_t, o_t)
11         Compose batch
           D_{t,h} ← {(o_t, u_t, δ_t, a_t(u_t|o_t)/ā_t(u_t|o_t))} ∪ D_{t,h}
12 improvement()
13     for i = 1...n do
14         Baseline update
           σ^i_{t,h+1} ← σ^i_{t,h} + α^σ_h E_{D_{t,h}}{δ_t (a_t(u_t|o_t)/ā_t(u_t|o_t)) φ^i_t(o^i_t)}
15         Critic update
           ν^i_{t,h+1} ← ν^i_{t,h} + α^ν_h E_{D_{t,h}}{δ_t (a_t(u_t|o_t)/ā_t(u_t|o_t)) φ^i_t(o^i_t, u^i_t)}
16         Actor update
           θ^i_{t,h+1} ← θ^i_{t,h} + α^θ_h E_{D_{t,h}}{(a_t(u_t|o_t)/ā_t(u_t|o_t)) φ^i_t(o^i_t, u^i_t) ν^i_{t,h}}
```

Pour conclure, cette section, quelques remarques sur les propriétés théoriques des algorithmes ACDC. D'une part, moyennant quelques conditions mineures, ils convergent avec probabilité 1 vers un optimum local puisque ce sont de véritables algorithmes d'ascension du gradient [6]. L'argument repose sur le fait qu'ils minimisent l'erreur de projection quadratique par descente de gradient stochastique, mais nous renvoyons le lecteur vers [6] pour plus de détails. D'autre part, ils terminent sur un optimum local qui est aussi un équilibre de Nash. En effet les dérivées partielles du critique centralisé par rapport à n'importe lequel de ses paramètres ne sont nulles qu'en un point d'équilibre, qui est aussi un optimum local.

6 Expérimentations

Dans cette section, nous montrons empiriquement les avantages du paradigme CTDC sur les paradigmes classiques CTCC et DTDC. Nos résultats semblent indiquer que les méthodes ACDC se comparent favorablement aux algorithmes existants sur de nombreux domaines multi-agents décentralisés de la littérature. Nous illustrons également les limitations de l'implémentation actuelle qui l'empêchent d'obtenir de meilleures performances.

6.1 Conditions expérimentales

Comme nous l'avons laisser transparaître tout au long de cet article, plusieurs facteurs clés peuvent affecter les performances des méthodes acteur-critique. Parmi ceux-ci, nous relèverons : le paradigme d'entraînement utilisé (CTCC *vs* DTDC *vs* CTDC) ; la représentation de la poli-

tique (stationnaire *vs* non-stationnaire) ; les structures d'approximation (linéaires *vs* réseaux de neurones récurrents (RNN)) ; la représentation des historiques (troncatures à mémoire finie *vs* états cachés d'un RNN). Nous implémentons trois variantes des méthodes acteur-critique qui combinent ces différents facteurs. Sauf mention contraire, nous ferons référence à chaque variante mise en œuvre par le nom du paradigme qu'elle implémente, par exemple CTDC pour notre algorithme `ACDC`, suivi de la structure de représentation interne utilisée quand cela est pertinent, par exemple : « *CTDC_trunc(*K*)* » pour un algorithme `ACDC` utilisant un historique tronqué des K dernières observations comme entrée d'une politique linéaire non-stationnaire, ou encore « *DTDC_rnn* » pour l'algorithme `Reinforce` distribué utilisant un réseau de neurone récurrent dans une politique stationnaire (voir Figure 2).

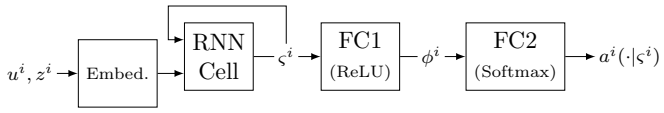

FIGURE 2 – Réseau de neurones récurrent utilisé comme structure « acteur » de chaque agent $i \in \mathcal{I}_n$. Une couche LSTM met à jour un état caché interne à partir du précédent et d'une encapsulation (embedding) d'une paire commande-observation. Elle est suivie d'un perceptron dont les activations sont rectifiées linéairement (ReLU) qui génère un vecteur de caractéristiques (features) ϕ^i, qui sont combinées linéairement par un second perceptron dont la sortie est normalisée par un softmax pour donner la règle de décision conditionnelle $a^i(\cdot|\varsigma^i)$.

Nous avons mené nos expérimentations sur une station de calcul *Dell Precision Tower 7910* équipé d'un CPU *Intel Xeon* à 16 cœurs cadencés à 3GHz, de 16Go de RAM et d'une carte graphique *nVIDIA Quadro K620* munie de 2Go de mémoire vidéo. Nous avons simulé plusieurs bancs de test standard de la littérature des Dec-POMDP, – *Dec. Tiger*, *Broadcast Channel*, *Mars*, *Box Pushing*, *Meeting in a Grid*, et *Recycling Robots* pour les citer – tel qu'ils sont défini sur `http://masplan.org`. Le détail des meta-paramètres utilisés est donné dans le Tableau 1 en annexe.

6.2 Importance de la représentation de l'état interne

Dans cette section, nous mettons en place des expériences visant à comprendre la façon dont la représentation des historiques affectent les performances des méthodes `ACDC`. La Figure 3 comparent les récompenses cumulées moyennes obtenues avec des historiques tronqués de taille 1 et 3, celles obtenues avec des RNN, et la performance ϵ-optimal donnée par l'algorithme de planification centralisé `FB-HSVI` [10]. Pour des horizons courts, ici $T = 10$, CTDC rnn converge rapidement vers de bonnes solutions par rapport à *CTDC_trunc(*1*)* et *CTDC_trunc(*3*)*. Ceci laisse penser que *CTDC_rnn* apprends des représentations des historiques plus concises et discriminantes que la représentation tronquée. Il semblerait néanmoins que cette différence de performances s'amenuise quand l'horizon de planification augmente (non illustré ici). On notera néanmoins une certaine instabilité de la récompense cumulée empirique pour les RNN sur les tâches plus complexe comme *Dec. Tiger*, un banc de test mettant en avant l'importance de la collecte d'information avant de prendre une action décisive, où les pénalités en cas d'erreur sont, en valeur absolues, bien plus élevées que les récompenses en cas de succès.

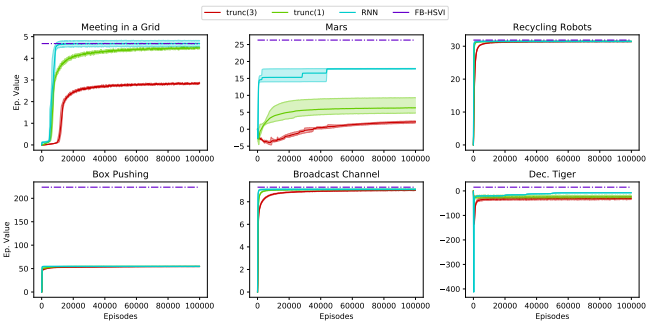

FIGURE 3 – Comparaison des différentes structures de représentation des historiques à $T = 10$.

Dans l'ensemble, nos expériences montrent un fort potentiel des RNN pour représenter les historiques. Ils ont l'avantage par rapport aux historiques tronqués d'apprendre automatiquement des classes d'équivalence et une représentation compacte de l'état interne, en se basant seulement sur la retro-propagation du gradient issu du signal de récompense. Il est aussi important de noter ici l'importance du choix des méta-paramètres, en particulier le taux d'apprentissage, qui, s'il est trop élevé, encourage des convergences prématurées vers des optima locaux insatisfaisants, sans laisser le temps à la politique d'explorer le reste des trajectoires possibles. Attention également à certaines propriétés spécifiques des modèles considérés, pour lesquelles les méthodes par ascension de gradient peinent à échapper au piège d'optima locaux. Nous n'avons pas pu identifier avec certitude quelles caractéristiques particulières avaient le plus d'impact négatif sur les performances, et nous laissons à de futures études l'analyse et l'exploration de méthodes plus fiables pour entraîner ces architectures.

6.3 Comparaison des différents paradigmes

Dans cette section, nous comparons les paradigmes CTCC, DTDC et CTDC en utilisant indifféremment les représentations ayant données les meilleurs résultats dans chaque cas. Nous incluons également à la comparaison les résultats issus de deux autres algorithmes de la littérature Dec-POMDP : un algorithme de planification ϵ-optimal appelé `FB-HSVI` [10], et un algorithme de planification par échantillonnage appelé Espérance-Maximisation Monte-Carlo (`MCEM`) [30] ; un choix justifié par de nombreuses similarités avec les méthodes acteur critique. Nous souli-

gnons le fait que nous ne cherchons pas à égaler les performances de FB−HSVI qui est un algorithme de planification centralisé nécessitant la connaissance à priori du modèle. Il nous fournit des performances de référence très proches de l'optimal global. Quant à MCEM, les résultats sont repris de [30] [2].

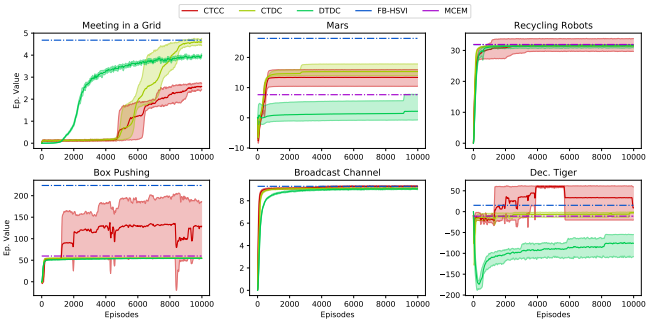

FIGURE 4 – Comparaison des paradigmes pour $T = 10$.

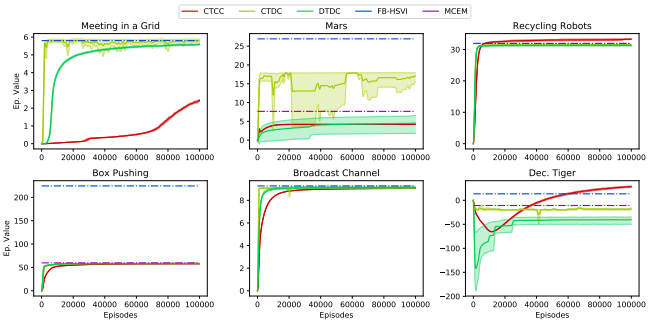

FIGURE 5 – Comparaison des paradigmes pour $T = \infty$.

Pour presque tous les bancs de test, CTDC semble prendre le meilleur sur les deux autres paradigmes, que ce soit à horizon fini ou infini. CTCC pâtit probablement du fléau de la dimension dans l'espace des historiques joints, et ne parvient pas à l'explorer de façon suffisamment efficace avant que les taux d'apprentissage rendent les mises à jour des paramètres négligeables, ou que le nombre maximum d'itérations fixé au départ ne soit atteint. Le fait d'utiliser la politique apprise comme politique d'échantillonnage (on-policy) amplifie très certainement cet effet. Ayant un espace d'historiques individuels bien plus réduit à explorer, CTDC donne de meilleur résultats que CTCC dans ces expérimentations. Comparé à DTDC qui explore lui-aussi des espaces d'historiques de dimensions plus raisonnables, le paradigme CTDC a un net avantage, et l'utilisation d'un critique centralisé compatible donnent de meilleures performances sur la vitesse de convergence et la qualité de l'optimum local atteint. Bien que CTDC donnent des résultats favorables – ou au moins comparables – par rapport à l'algorithme de l'état de l'art MCEM, il y a encore une large marge d'amélioration pour atteindre les optima globaux donnés par FB−HSVI pour tous les bancs de test. Comme mentionné précédemment, c'est en partie dû à une compression encore imparfaite des historiques, mais aussi à des limitations intrinsèques des méthodes par ascension de gradient, qui ne peuvent garantir qu'un optimum local.

2. Sauf pour *Meeting in a Grid* et *Broadcast Channel* pour lesquels les valeurs indiquées dans [30] étaient bien au delà de l'optimal, laissant penser à une anomalie.

7 Conclusion

Cet article pose les fondations théoriques des méthodes acteur-critique pour les Dec-POMDP dans le paradigme CTDC. Dans ce paradigme, un algorithme acteur-critique centralisé apprend des politiques indépendantes, une par agent, guidé par un unique critique joint. Nous montrons qu'un critique centralisé compatible peut s'écrire comme la somme de critiques individuels, qui sont chacun des combinaisons linéaires des « caractéristiques » de la politique individuelle correspondante. Nos expérimentations montrent que nos méthodes acteur-critique appelées ACDC se démarquent favorablement des approches standard du RL pour un certain nombre de bancs de test de la littérature. L'implémentation actuelle de ACDC soulève deux problèmes bien connu du domaine : le compromis exploitation-exploration et la représentation des états internes individuels. En particulier pour ce dernier point, apprendre à projeter les historiques individuels vers des états internes proches des états d'occupation individuels est un défi de taille, auquel nous comptons continuer à contribuer dans le futur. Outre ce problème de représentation des historiques individuels, ACDC peut exploiter la séparabilité du critique joint compatible pour passer à l'échelle d'un assez grand nombre d'agents. Nous nous intéressons à présent à une application multi-agents décentralisée à large échelle, pour laquelle nous cherchons à exploiter cette propriété.

Références

[1] Shun-Ichi AMARI. "Natural Gradient Works Efficiently in Learning". In : *Neural Comput.* 10.2 (1998). ISSN : 0899-7667.

[2] Karl J ASTRÖM. "Optimal Control of Markov Decision Processes with Incomplete State Estimation". In : *Journal of Mathematical Analysis and Applications* 10 (1965).

[3] Richard E BELLMAN. "The Theory of Dynamic Programming". In : *Bulletin of the American Mathematical Society* 60.6 (1954).

[4] Daniel S BERNSTEIN et al. "The Complexity of Decentralized Control of Markov Decision Processes". In : *Mathematics of Operations Research* 27.4 (2002).

[5] Craig BOUTILIER. "Planning, Learning and Coordination in Multiagent Decision Processes". In : *Proc. of the Sixth Conf. on Theoretical Aspects of Rationality and Knowledge*. 1996.

[6] Thomas DEGRIS, Martha WHITE et Richard S. SUTTON. “Linear Off-Policy Actor-Critic”. In : *Proc. of the 29th Int. Conf. on ML, ICML 2012, Edinburgh, Scotland, UK*. 2012.

[7] Jilles Steeve DIBANGOYE et al. “Exploiting Separability in Multi-Agent Planning with Continuous-State MDPs”. In : *Proc. of the Thirteenth Int. Conf. on Autonomous Agents and Multiagent Systems*. 2014.

[8] Jilles Steeve DIBANGOYE et al. “Optimally Solving Dec-POMDPs As Continuous-state MDPs”. In : *Proc. of the Twenty-Fourth Int. Joint Conf. on AI*. 2013.

[9] Jilles Steeve DIBANGOYE et al. *Optimally solving Dec-POMDPs as Continuous-State MDPs : Theory and Algorithms*. Research Report RR-8517. 2014.

[10] Jilles S DIBANGOYE et al. “Optimally Solving Dec-POMDPs as Continuous-State MDPs”. In : *Journal of AI Research* 55 (2016).

[11] Jakob N. FOERSTER et al. “Counterfactual Multi-Agent Policy Gradients”. In : *CoRR* (2017).

[12] Jayesh K. GUPTA, Maxim EGOROV et Mykel KOCHENDERFER. “Cooperative Multi-agent Control Using Deep Reinforcement Learning”. In : *Autonomous Agents and Multiagent Systems*. 2017.

[13] Eric A HANSEN, Daniel S BERNSTEIN et Shlomo ZILBERSTEIN. “Dynamic Programming for Partially Observable Stochastic Games”. In : *Proc. of the Nineteenth National Conf. on AI*. 2004.

[14] Sham KAKADE. “A Natural Policy Gradient”. In : *Advances in Neural Information Processing Systems 14 (NIPS 2001)*. 2001.

[15] Vijay R. KONDA et John N. TSITSIKLIS. “Actor-Critic Algorithms”. In : *Advances in Neural Information Processing Systems 12*. 2000.

[16] Landon KRAEMER et Bikramjit BANERJEE. “Multi-agent reinforcement learning as a rehearsal for decentralized planning”. In : *Neurocomputing* 190 (2016).

[17] Ryan LOWE et al. “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments”. In : *Advances in Neural Information Processing Systems 30*. 2017.

[18] Volodymyr MNIH et al. “Human-level control through deep reinforcement learning”. In : *Nature* 518.7540 (fév. 2015). ISSN : 0028-0836.

[19] Matej MORAVČÍK et al. “DeepStack : Expert-level artificial intelligence in heads-up no-limit poker”. In : *Science* 356.6337 (2017).

[20] Frans A OLIEHOEK et al. “Incremental Clustering and Expansion for Faster Optimal Planning in Dec-POMDPs”. In : *Journal of AI Research* 46 (2013).

[21] Leonid PESHKIN et al. “Learning to Cooperate via Policy Search”. In : *Sixteenth Conf. on Uncertainty in Artificial Intelligence (UAI-2000)*. 2000.

[22] H ROBBINS et S MONRO. “A stochastic approximation method”. In : *The annals of mathematical statistics* 22.3 (1951).

[23] Yoav SHOHAM et Kevin LEYTON-BROWN. *Multiagent Systems : Algorithmic, Game-Theoretic, and Logical Foundations*. New York, NY, USA, 2008. ISBN : 0521899435.

[24] Richard S SUTTON et Andrew G BARTO. *Introduction to Reinforcement Learning*. 2nd. Cambridge, MA, USA, 2016. ISBN : 0262193981.

[25] Richard S SUTTON et al. “Policy Gradient Methods for Reinforcement Learning with Function Approximation”. In : *Proc. of the 12th Int. Conf. on Neural Information Processing Systems*. Cambridge, MA, USA, 1999.

[26] Daniel SZER et François CHARPILLET. “An Optimal Best-First Search Algorithm for Solving Infinite Horizon DEC-POMDPs”. In : *Proc. of the Fifteenth European Conf. on ML*. 2005.

[27] Daniel SZER, François CHARPILLET et Shlomo ZILBERSTEIN. “MAA* : A Heuristic Search Algorithm for Solving Decentralized POMDPs”. In : *Proc. of the Twenty-First Conf. on Uncertainty in AI*. 2005.

[28] Ming TAN. “Multi-agent Reinforcement Learning : Independent vs. Cooperative Agents”. In : *Readings in Agents*. San Francisco, CA, USA, 1998.

[29] Ronald J WILLIAMS. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In : *ML* 8.3 (1992).

[30] Feng WU, Shlomo ZILBERSTEIN et Nicholas R JENNINGS. “Monte-Carlo Expectation Maximization for Decentralized POMDPs”. In : *Proc. of the Twenty-Fourth Int. Joint Conf. on AI*. 2013.

[31] Xinhua ZHANG, Douglas ABERDEEN et S. V. N. VISHWANATHAN. “Conditional Random Fields for Multi-agent Reinforcement Learning”. In : *Proc. of the 24th international conference on Machine learning* (2007).

A Meta-paramètres pour ACDC

Représentation	RNN	trunc(K)
Horizon	10 / ∞^1	10 / ∞^1
Escompte	1 / 0.9	1 / 0.9
Nombre de tests	3	3
Itérations	100000	100000
Mini-batch	256	32
Taux d'apprentissage initial α_0	0.05	1.0
Taux d'apprentissage minimal $\alpha_{\min}$	0	0
Amortissement du taux λ_α	0.96	0.37
Pas d'amortissement κ_α	40000	80000
Type d'erreur	Monte-Carlo	Time-Difference
Saturation du gradient	20	Aucune
Échantillonage	On-policy	On-policy

TABLE 1 – Meta-paramètres pour les méthodes ACDC. - Note[1] : Les algorithmes ont échantillonné des trajectoires de taille T telle que $\frac{1}{(1-\gamma^T)} \max_{x,u} R(x,u) \ll \epsilon$

Formule de mise à jour du taux d'apprentissage :

$$\alpha_h = \alpha_{\min} + (\alpha_0 - \alpha_{\min}) \times \exp\left(\frac{h}{\lambda_\alpha} \log \kappa_\alpha\right)$$

B Preuves

B.1 Preuve du Lemme 2

Démonstration.

$$\begin{aligned}
\frac{\partial f(x)}{\partial x^j} &= f(x)\frac{\partial \log f(x)}{\partial x^j} \text{ (dérivée de la composition } \log \circ f) \\
&= f(x)\frac{\partial}{\partial x^j}\left(\log \prod_{i=1}^{n} f^i(x^i)\right) \text{ (séparabilité)} \\
&= f(x)\frac{\partial}{\partial x^j}\sum_{i=1}^{n} \log f^i(x^i) \text{ (propriétés du log)} \\
&= f(x)\sum_{i=1}^{n}\frac{\partial \log f^i(x^i)}{\partial x^j} \\
&= f(x)\frac{\partial \log f^j(x^j)}{\partial x^j} \text{ (termes nuls } \forall i \neq j)
\end{aligned}$$

□

B.2 Preuve du Lemme 3

Démonstration. Soit $\mathcal{X}$ le domaine de la variable aléatoire X.

$$\begin{aligned}
\mathbb{E}_{X\sim p}[f(X)] &= \int_{x\in\mathcal{X}} p(x)f(x)dx \\
&= \int_{x\in\mathcal{X}} \frac{q(x)}{q(x)}p(x)f(x)dx \\
&= \int_{x\in\mathcal{X}} q(x)\frac{p(x)}{q(x)}f(x)dx \\
&= \mathbb{E}_{X\sim q}\left[\frac{p(X)}{q(X)}f(X)\right]
\end{aligned}$$

□

B.3 Preuve du Lemme 4

Démonstration. Calculons la dérivée partielle de $V_t^\pi(x_t, o_t)$ par rapport à θ_t^i :

$$\frac{\partial V_t^\pi(x_t,o_t)}{\partial \theta_t^i} \doteq \frac{\partial}{\partial \theta_t^i}\sum_{u_t\in U} a_t(u_t|o_t)Q_t^\pi(x_t,o_t,u_t) \quad (23)$$

$$= \sum_{u_t\in U}\Big[\frac{\partial a_t(u_t|o_t)}{\partial \theta_t^i}Q_t^\pi(x_t,o_t,u_t) + a_t(u_t|o_t)\frac{\partial Q_t^\pi(x_t,o_t,u_t)}{\partial \theta_t^i}\Big] \quad (24)$$

$$= \sum_{u_t\in U}\left[\frac{\partial a_t(u_t|o_t)}{\partial \theta_t^i}Q_t^\pi(x_t,o_t,u_t) + 0\right] \quad (25)$$

Dans l'équation 25, on utilise le fait que $\frac{\partial Q_t^\pi(x_t,o_t,u_t)}{\partial \theta_t^i} = 0$. En effet, si l'on se réfère à l'équation de Bellman du Lemme 1, ni la récompense immédiate, ni l'espérance sur la Q-valeur à $t+1$ ne dépend de θ_t^i. La preuve se conclut aisément en appliquant le Lemme 2 sur $a_t(u_t|o_t)$, puis en introduisant $\bar{a}$ par le Lemme 3. □

B.4 Preuve du Théorème 1

Cas à horizon fini.

Démonstration. Soit $t = 1, \ldots, T$. Séparons d'abord la mesure de performance $J(s_0; \theta_{0:T}^{1:n})$ en deux termes : une récompense passée du temps 0 au temps $t-1$ et une récompense future de t à T.

$$\begin{aligned}
&J(s_0;\theta_{0:T}^{1:n}) \doteq \mathbb{E}_{(X_0,O_0)\sim\mathbb{P}(\cdot|M_n)}[V_0^\pi(X_0,O_0)] \\
&= \mathbb{E}_{(x_0,o_0)\sim\mathbb{P}(\cdot|M_n)}[\mathbb{E}_{u_0\sim a_0(\cdot|o_0)}[Q_0^\pi(x_0,o_0,u_0)]] \\
&= \mathbb{E}_{(X_0,O_0,U_0)\sim\mathbb{P}(\cdot|a_0,M_n)}[Q_0^\pi(X_0,O_0,U_0)] \\
&= \mathbb{E}_{(X_{0:1},O_{0:1},U_{0:1})\sim\mathbb{P}(\cdot|a_{0:1},M_n)}[R(X_0,U_0)+\gamma Q_1^\pi(X_1,O_1,U_1)] \\
&= \mathbb{E}_{(X_{0:t},O_{0:t},U_{0:t})\sim\mathbb{P}(\cdot|a_{0:t},M_n)}[R(\omega_{0:t-1})+\gamma^t Q_t^\pi(X_t,O_t,U_t)] \\
&= J(s_0;\theta_{0:t-1}^{1:n}) + \gamma^t J(s_t;\theta_{t:T}^{1:n})
\end{aligned}$$

où $J(s_0;\theta_{0:t-1}^{1:n})$ dénote la mesure de performance selon la politique jointe partielle $a_{0:t-1}$ en partant de l'état d'occupation initial s_0 :

$$J(s_0;\theta_{0:t-1}^{1:n}) \doteq \mathbb{E}_{(X_{0:t-1},O_{0:t-1},U_{0:t-1})\sim\mathbb{P}(\cdot|a_{0:t-1},M_n)}[R(\omega_{0:t-1})]$$

et $J(s_t;\theta_{t:T}^{1:n})$ dénote la mesure de performance selon la politique jointe partielle $a_{t:T}$ en partant de l'état d'occupation s_t définit par $s_t(X_t,O_t) \doteq \mathbb{P}(X_t,O_t|a_{0:t-1},M_n)$:

$$\begin{aligned}
&J(s_t;\theta_{t:T}^{1:n}) \doteq \mathbb{E}_{(X_{0:t},O_{0:t},U_{0:t-1})\sim\mathbb{P}(\cdot|s_t,M_n)}[V_t^\pi(X_t,O_t)] \\
&= \mathbb{E}_{(X_{0:t},O_{0:t},U_{0:t-1})\sim\mathbb{P}(\cdot|s_t,M_n)}[\mathbb{E}_{U_t\sim a_t(\cdot|O_t)}[Q_t^\pi(X_t,O_t,U_t)]] \\
&= \mathbb{E}_{(X_{0:t},O_{0:t},U_{0:t})\sim\mathbb{P}(\cdot|s_t,a_t,M_n)}[Q_t^\pi(X_t,O_t,U_t)]
\end{aligned}$$

Calculons ensuite les dérivées partielles de $J(s_0;\theta_{0:t-1}^{1:n})$ et $\gamma^t J(s_t;\theta_{t:T}^{1:n})$ par rapport à θ_t^i pour n'importe quel $i \in I_n$:

$$\frac{\partial J(s_1;\theta_{0:T}^{1:n})}{\partial \theta_t^i} = \gamma^t \mathbb{E}_{(X_{0:t},O_{0:t},U_{0:t-1})\sim\mathbb{P}(\cdot|s_t,M_n)}\left[\frac{\partial V_t^\pi(X_t,O_t)}{\partial \theta_t^i}\right] \quad (26)$$

en remarquant que $\partial J(s_0;\theta^{1:n}_{0:t-1})/\partial\theta^i_t = 0$. On applique ensuite le Lemme 4 à partir de (26) pour conclure la preuve. □

Cas à horizon infini.

Démonstration. Développons d'abord l'expression de la dérivée partielle de la fonction de valeur : $\forall x \in \mathcal{X}, \forall \varsigma$,

$$\begin{aligned}\frac{\partial V^\pi(x,\varsigma)}{\partial\theta^i} &= \frac{\partial}{\partial\theta^i}\left(\mathbb{E}_{U\sim a(\cdot|\varsigma)}\left[Q^\pi(x,\varsigma,U)\right]\right)\\ &= \frac{\partial}{\partial\theta^i}\left(\sum_{u\in\mathcal{U}} a(u|\varsigma)Q^\pi(x,\varsigma,u)\right)\\ &= \sum_{u\in\mathcal{U}}\Big(\frac{\partial a(u|\varsigma)}{\partial\theta^i}Q^\pi(x,\varsigma,u) + a(u|\varsigma)\frac{\partial Q^\pi(x,\varsigma,u)}{\partial\theta^i}\Big)\\ &= \sum_{u\in\mathcal{U}} a(u|\varsigma)\left[\frac{\partial\log a^i(u^i|\varsigma^i)}{\partial\theta^i}Q^\pi(x,\varsigma,u) + \frac{\partial Q^\pi(x,\varsigma,u)}{\partial\theta^i}\right]\\ &= \mathbb{E}_{U\sim a(\cdot|\varsigma)}\left[\frac{\partial\log a^i(U^i|\varsigma^i)}{\partial\theta^i}Q^\pi(x,\varsigma,U) + \frac{\partial Q^\pi(x,\varsigma,U)}{\partial\theta^i}\right]\end{aligned}$$

Détaillons à présent la dérivée partielle de $Q^\pi(x,\varsigma,u)$:

$$\begin{aligned}\frac{\partial Q^\pi(x,\varsigma,u)}{\partial\theta^i} &= \frac{\partial}{\partial\theta^i}\left(R(x,u) + \gamma\mathbb{E}_{\substack{X',Z\sim p(\cdot|x,u)\\ \Sigma'\sim\psi(\cdot|\varsigma,u,Z)}}\left[V^\pi(X',\Sigma')\right]\right)\\ &= \gamma\mathbb{E}_{\substack{X',Z\sim p(\cdot|x,u)\\ \Sigma'\sim\psi(\cdot|\varsigma,u,Z)}}\left[\frac{\partial V^\pi(X',\Sigma')}{\partial\theta^i}\right]\end{aligned}$$

On ré-injecte ce résultat :

$$\begin{aligned}\frac{\partial V^\pi(x,\varsigma)}{\partial\theta^i} &= \mathbb{E}_{U\sim a(\cdot|\varsigma)}\left[\frac{\partial\log a^i(U^i|\varsigma^i)}{\partial\theta^i}Q^\pi(x,\varsigma,U)\right]\\ &\quad+\gamma\mathbb{E}_{\substack{U\sim a(\cdot|\varsigma)\\ X',Z\sim p(\cdot|x,U)\\ \Sigma'\sim\psi(\cdot|\varsigma,U,Z)}}\left[\frac{\partial V^\pi(X',\Sigma')}{\partial\theta^i}\right]\end{aligned}$$

Répétons à nouveau ce développement avec $\frac{\partial V^\pi(x',\Sigma')}{\partial\theta^i}$ dans l'expression précédente pour faire apparaître un schéma récurrent :

$$\begin{aligned}\frac{\partial V^\pi(x,\varsigma)}{\partial\theta^i} &= \mathbb{E}_{U\sim a(\cdot|\varsigma)}\left[\frac{\partial\log a^i(U^i|\varsigma^i)}{\partial\theta^i}Q^\pi(x,\varsigma,U)\right]\\ &+\gamma\mathbb{E}_{\substack{U\sim a(\cdot|\varsigma)\\ X',Z\sim p(\cdot|x,U)\\ \Sigma'\sim\psi(\cdot|\varsigma,U,Z)\\ U'\sim a(\cdot|\Sigma')}}\left[\frac{\partial\log a^i(U'^{,i}|\Sigma'^{,i})}{\partial\theta^i}Q^\pi(X',\Sigma',U')\right]\\ &+\gamma^2\mathbb{E}_{\substack{U\sim a(\cdot|\varsigma)\\ X',Z\sim p(\cdot|x,U)\\ \Sigma'\sim\psi(\cdot|\varsigma,U,Z)}}\left[\mathbb{E}_{\substack{U'\sim a(\cdot|\Sigma')\\ X'',Z'\sim p(\cdot|X',U')\\ \Sigma''\sim\psi(\cdot|\Sigma',U',Z')}}\left[\frac{\partial V^\pi(X'',\Sigma'')}{\partial\theta^i}\right]\right]\end{aligned}$$

Et ainsi de suite jusqu'à obtenir :

$$\begin{aligned}\frac{\partial V^\pi(x,\varsigma)}{\partial\theta^i} = \lim_{t\to\infty}\sum_{k=0}^{t}\gamma^k\mathbb{E}_{U,X',\Sigma'\sim\mathbb{P}^k(\cdot|M_n,a,\psi,x,\varsigma)}\Big[\\ \frac{\partial\log a^i(U^i|\Sigma'^{,i})}{\partial\theta^i}Q^\pi(X',\Sigma',U)\Big]\end{aligned}$$

où $\mathbb{P}^k(x',\varsigma'|M_n,a,\psi,x_0,o_0)$ dénote la probabilité de rencontrer l'état x' et la représentation interne ς' après avoir suivi la politique a pendant k pas de temps, dans un environnement M_n en partant de l'état x et de la représentation interne ς. La formule finale s'obtient en prenant l'espérance de l'expression précédente sur la distribution initiale des états s_0. □

B.5 Preuve du Théorème 2

Démonstration. La preuve montre que l'estimation du gradient basée sur les approximations $V^\sigma_{0:T}$ et $A^\nu_{0:T}$, telles que $\sigma^{1:n}_{0:T}$ et $\nu^{1:n}_{0:T}$ satisfassent l'une des deux conditions, préserve la direction du véritable gradient de la politique. Nous dériverons cette preuve dans le cas où la seconde condition est vérifiée, car un raisonnement tout à fait similaire tient dans le premier cas. Dans ce second cas, $\sigma^{1:n}_{0:T}$ doit minimiser l'erreur quadratique moyenne (MSE) $\mathbb{E}[\epsilon^2_t(X_t,O_t,U_t)]$. Pour tout temps $t = 0,1,\dots,T$, pour tout agent $i \in I_n$:

$$\frac{\partial\mathbb{E}[\epsilon^2_t(X_t,O_t,U_t)]}{\partial\nu^i_t} = 2\mathbb{E}[\epsilon_t(X_t,O_t,U_t)\frac{\partial A^\nu_t(X_t,O_t,U_t)}{\partial\nu^i_t}].$$

car la distribution de X_t, O_t, U_t ne dépend pas de ν. Comme $\nu^{1:n}_{0:T}$ satisfait la condition (18), on a :

$$\frac{\partial\mathbb{E}[\epsilon^2_t(X_t,O_t,U_t)]}{\partial\nu^i_t} = \mathbb{E}[\epsilon_t(X_t,O_t,U_t)\frac{\partial\log a^i_t(U^i_t|O^i_t)}{\partial\theta^i_t}].$$

En développant l'expression de $\epsilon_t(X_t,O_t,U_t)$, et en réarrangeant les différents termes, on obtient :

$$\begin{aligned}&\mathbb{E}\left[Q^\pi_t(x_t,o_t,u_t)\frac{\partial\log a^i_t(u^i_t|o^i_t)}{\partial\theta^i_t}\right]\\ &= \mathbb{E}\left[\big(V^\sigma_t(x_t,o_t) + A^\nu_t(x_t,o_t,u_t)\big)\frac{\partial\log a^i_t(u^i_t|o^i_t)}{\partial\theta^i_t}\right]\end{aligned}$$

Grâce au Lemme 4 l'expression devient :

$$\begin{aligned}&\frac{\partial V^\pi_t(x_t,o_t)}{\partial\theta^i_t}\\ &= \mathbb{E}\left[\big(V^\sigma_t(x_t,o_t) + A^\nu_t(x_t,o_t,U_t)\big)\frac{\partial\log a^i_t(U^i_t|o^i_t)}{\partial\theta^i_t}\right]\end{aligned}$$

Puis en utilisant le Lemme 3 pour introduire la politique d'exploration $\bar\pi$, on déduit l'expression final (19). □