Patrouille multi-drones et observation de cibles mobiles

Jamy Chahal

▶ **To cite this version:**

Jamy Chahal. Patrouille multi-drones et observation de cibles mobiles. Intelligence artificielle [cs.AI]. Sorbonne Université, 2023. Français. NNT : 2023SORUS736 . tel-04590459

HAL Id: tel-04590459
https://theses.hal.science/tel-04590459v1

Submitted on 28 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse présentée pour l'obtention du grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Spécialité

Ingénierie / Systèmes Informatiques

École doctorale

Informatique, Télécommunication et Électronique Paris (ED130)

Patrouille multi-drones et observation de cibles mobiles

Jamy Chahal

Soutenue publiquement le : *30 novembre 2023*

Devant un jury composé de :

Prénom NOM, Poste, Lieu	*Role dans le jury*
Olivier SIMONIN, Professeur, INSA Lyon - INRIA	*Président du jury*
Gauthier PICARD, Directeur de Recherche, ONERA	*Rapporteur*
René MANDIAU, Professeur, Univ. Polytechnique Hauts de France - LAMIH	*Rapporteur*
Amal EL FALLAH SEGHROUCHNI, Professeur, Sorbonne Université - LIP6	*Directrice de thèse*
Assia BELBACHIR, Docteur, Sorbonne Université - LIP6	*Co-encadrante de thèse*

Remerciements

Ceux qui traversent actuellement cette expérience, qui l'ont traversé ou qui en ont été témoins, pourront attester que la réalisation d'une thèse peut être à la fois gratifiante et enrichissante, mais est également exigeante et par moments éreintante. La remise en question et le doute constituent des défis à ne pas sous-estimer, parfois déconcertants, mais souvent indispensables pour mener à bien des travaux de recherche. C'est dans ce contexte que je tiens à exprimer ma profonde gratitude envers mes encadrantes de thèse, dont le rôle a été essentiel pour que cette aventure se déroule au mieux. Amal, je tiens à te remercier tout particulièrement pour ta confiance, pour avoir partagé ton expertise, ainsi que pour ta bonhomie et ta bienveillance lors de mon immersion au sein du monde scientifique. Assia, cette thèse n'aurait pas pu également émerger sans ta vision ni ta perspicacité. Je te remercie pour ta patience, ton écoute, et d'avoir cru en moi lorsque tu étais encore mon enseignante en école d'ingénieurs.

Je tiens également à témoigner mon attachement à ma famille, mes amis, ceux qui ont veillé sur moi, qui ont été présents et notamment ceux qui ont apporté une aide tangible dans le cadre de cette thèse. Un grand merci à Claudia pour ton soutien précieux dans la relecture de l'ensemble de mes articles scientifiques en anglais. Je souhaite également exprimer ma sincère gratitude, Johvany, pour ton apport inestimable dans la réussite de nos expériences avec les drones. Par la même occasion, je désire remercier chaleureusement Maël, Marie-Anne, Arthur, Charlotte et Julia pour leur contribution à la correction et aux commentaires pertinents apportés à ce manuscrit de thèse. Enfin, je tiens à remercier mes collègues et amis de laboratoire, que ce soit du LIP6 ou du LS2A. Nos relations ont toujours été marquées par la solidarité et le partage. En guise de clôture de cette section, je tiens à partager une citation d'Isaac Asimov, en signe de reconnaissance envers ce scientifique et écrivain dont la perspective sur le monde, l'humanité et la robotique a profondément enrichi ma pensée :

« Qu'est-ce que la beauté, ou la charité, ou l'art, ou l'amour, ou Dieu ? Nous piétinerons éternellement aux frontières de l'Inconnu, cherchant à comprendre ce qui restera toujours incompréhensible. Et c'est précisément cela qui fait de nous des hommes. » [10]

Résumé

Cette thèse s'intéresse à l'association de la problématique de l'observation avec celle de la patrouille dans un cadre multi-agents. La problématique de l'observation a pour objectif le suivi de cibles en maximisant au cours du temps le nombre de cibles mobiles observées par au moins un agent, tandis que la problématique de la patrouille a pour finalité de visiter le plus fréquemment possible un ensemble de lieu dans l'environnement, ou autrement dit, de couvrir régulièrement l'environnement. Nous proposons que les agents soient face à un dilemme d'exploration, via la recherche active de nouvelles cibles grâce à la patrouille de l'environnement, et d'exploitation, à travers la maximisation de l'observation des cibles. Nous nommons ce nouvel enjeu le Problème de l'Observation appuyée par la Patrouille (POP). En ne s'intéressant pas qu'au suivi de cibles, les agents réduisent également les risques de manipulation par des cibles intelligentes qui pourraient chercher à influencer leurs déplacements. Nous proposons un ensemble de méthodes de résolution du POP reposant soit sur l'utilisation de champs de potentiel (I-CMOMMT), soit sur des approches d'apprentissage, que ce soit par renforcement (FFRL, F2MARL) ou supervisé (MALOS). Ces méthodes sont comparées avec d'autres stratégies issues de la littérature. Les expériences sont réalisées dans un premier temps dans un environnement de simulation Gazebo/ROS2. Les agents y sont représentés par des drones et les cibles par des robots terrestres mobiles. Dans un second temps, les méthodes sont implémentées puis évaluées sur de vrais drones en volière. La thèse introduit deux autres contributions, sous forme d'outils, en complément des approches de résolution du POP. Le premier outil permet aux agents d'identifier efficacement les lieux ayant un potentiel intérêt à être visité, tandis que le second a pour objectif d'optimiser des paramètres d'une mission, tels que le nombre d'agents, tout en respectant une ou plusieurs performances prédéfinies par l'utilisateur. Dans un souci de reproductibilité, l'ensemble des codes, que ce soit pour l'environnement de simulation que les méthodes, sont open-source.

Abstract

This thesis focuses on the combination of the observation problem with the patrol problem within a multi-agent framework. The observation problem aims to track targets by maximizing the number of mobile targets observed by at least one agent over time. While the patrol problem aims to visit a set of locations in the environment as frequently as possible, essentially achieving regular coverage of the environment. We propose that the agents face an exploration-exploitation dilemma, achieved through actively searching for new targets via environmental patrol and maximizing target observation. We term this novel challenge the Patrolled Observation Problem (POP). By not solely concentrating on target tracking, the agents also mitigate the risk of manipulation by intelligent targets that might attempt to influence their movements. We present a set of solutions for the POP, relying on either potential fields (I-CMOMMT) or learning approaches, whether through reinforcement (FFRL, F2MARL) or supervised learning (MALOS). These methods are compared with other strategies from existing literature. The experiments are initially conducted within a Gazebo/ROS2 simulation environment, where agents are represented by drones and targets by mobile ground robots. Subsequently, these methods are implemented and evaluated on actual drones within an aviary. The thesis introduces two additional contributions in the form of tools, complementing the approaches for solving the POP. The first tool enables agents to efficiently identify locations with potential interest to be visited. The second tool aims to optimize mission parameters, such as the number of agents, while complying to one or more user-defined performance criteria. In the interest of reproducibility, all codes, whether for the simulation environment or the methods themselves, are open-source.

Publications de l'auteur

Certaines recherches, idées et illustrations présentées dans cette thèse sont contenues dans les publications suivantes :

Jamy CHAHAL, Assia BELBACHIR et AMAL EL FALLAH SEGHROUCHNI. "I-CMOMMT : A multiagent approach for patrolling and observation of mobile targets with a continuous environment representation". In : *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE* (juill. 2021). DOI : 10.18293/SEKE2021-135

Jamy CHAHAL, Amal El Fallah SEGHROUCHNI et Assia BELBACHIR. "A decision-making architecture for observation and patrolling problems using machine learning". In : *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*. Juill. 2021, p. 426-431. DOI : 10.1109/IIAI-AAI53430.2021.00074

Jamy CHAHAL, Amal El Fallah SEGHROUCHNI et Assia BELBACHIR. "A Force Field Reinforcement Learning Approach for the Observation Problem". In : *Intelligent Distributed Computing XIV, 14th International Symposium on Intelligent Distributed Computing, IDC 2021, Virtual Event, 16-18 September 2021*. T. 1026. Studies in Computational Intelligence. Springer, sept. 2021, p. 89-99. DOI : 10.1007/978-3-030-96627-0_9

Jamy CHAHAL, Assia BELBACHIR et Amal El Fallah SEGHROUCHNI. "Dynamic Interest Points : A Formalism to Identify Areas to Patrol within a Continuous Environment". In : *56th Hawaii International Conference on System Sciences, HICSS 2023, Maui, Hawaii, USA, January 3-6, 2023*. Sous la dir. de Tung X. BUI. ScholarSpace, jan. 2023, p. 6853-6862. URL : https://hdl.handle.net/10125/103464

Table des matières

Liste des algorithmes

Table des figures

Liste des tableaux

Acronymes

ACO Ant Colony Optimization. 37

AMTDS Adaptative Meta Target Decision Strategy. xxv, 51–53

AMTDS/EDC AMTDS with learning of event probabilities and enhancing divisional cooperation. 52

AMTDS/ESC AMTDS for energy saving and cleanliness. 53

AMTDS/RM AMTDS with relearning by self-monitoring. xxv, 52

AMTDS/RMLD AMTDS/RM and learning of dirt accumulation probability. 52

ANTE Average Number of Target Evasion. 24, 25

B-CMOMMT Behavior-CMOMMT. 18, 21, 73

BBLA Black-Box Learner Agent. 55, 56

BNPS Balanced neighbor-preferential selection. 51, 53

BRLP-CTO Cooperative Target Observation using Binary search for Randomization CTO. 24

CBLS Concurrent Bayesian Learner Strategy. 57, 58

CC Conscientious Cognitive. xvii, 35, 42, 82, 88

CCPP Continuous Cooperative Patrolling Problem. xvii, 49–51, 54, 67

CGG Cyclic Algorithm for Generic Graphs. 37, 39, 40

CI Closest Idleness. 82, 87, 128, 131, 149

CMFMT Cooperative Multirobot Fair Multitarget Tracking. xvii, 25, 26

CMOMMT Multi-Robot Observation of Multiple Moving Targets. xvii, xxv–xxvii, 17–20, 22, 58, 63, 73, 163

CMUOMMT Cooperative Multi-UAV Observation of Multiple Moving Targets. xvii, xxvi, 20, 21, 25, 59, 60

CR Conscientious Reactive. xvii, 35, 38–42, 44–48, 56–58, 82, 88

CTO Cooperative Targets Observation. xxv, xxvi, 22, 24, 25

DIHI Double iterations among highest idleness. xv, 93, 96

DIP Dynamic Interest Points. xxiii, 89, 91, 97–100, 103, 173

DMST Distributed Multi-Target Search and Tracking. xxvii, 28, 29, 58

DRL-CMUOMMT Deep Reinforcement Learning - CMUOMMT. 21, 59

ENNC-QL Evolutionary Nearest Neighbor Classifier -Q-learning. 20

ER Expected Reactive. 48

F-CMOMMT Flexible-CMOMMT. 19

F2MARL Force Field MultiAgent Reinforcement Learning. xix, xx, xxiii, 105, 118, 119, 122–128, 131–133, 143, 170, 172

FBA Flexible Bidder Agent. xxvi, 39, 43, 44

FBAA FBA Asynchronous. 43

FCM Fuzzy C-means clustering. 23, 25

FCM-CTO Fuzzy C-means clustering CTO. 23

FDC Fonction de Distribution Cumulative. 147, 148

FDCE Fonction de Distribution Cumulative Empirique. 149–151

FFRL Force Field Reinforcement Learning. 105, 106, 114, 116–120, 124, 125, 128, 132, 170, 172

GBLA Gray-Box Learner Agent. 55, 56

GBS Greedy Bayesian Strategy. 57, 58

GPG Gradual Path Generation. 51, 52

GPS Global Positioning System. 158, 161

GRU Gated Recurrent Unit. 173

HCC Heuristic Conscientious Cognitive. 36, 48

HCR Heuristic Conscientious Reactive. 36, 38–40, 58

HI Highest Idleness. 82, 87, 128, 131

HPCC Heuristic Pathfinder Conscientious Cognitive. xvii, 36, 38–42, 44–47, 56, 58

HPMB Heuristic Pathfinder Mediated Trader Bidder. 39

HPTB Heuristic Pathfinder Two-shots Bidder. 39

I-CMOMMT Idleness-CMOMMT. xviii, xxi, xxiii, 63, 73–78, 81, 82, 87, 88, 114, 116, 117, 128, 136–138, 142, 143, 155, 157, 163, 167, 170, 173

IC Idleness Coordinator. 35, 36, 41

IFC Iteration among free cells. xv, xix, 92, 94–96

ILP Integer Linear Program. 25, 26

IPPO Independent Proximal Policy Optimization. xix, 111, 112, 119

KDE Kernel Density Estimation. 147, 149–151

LSTM Long Short-Term Memory. 46, 47, 173

MALOS MultiAgent Learning using Optimized Strategy. xx, 105, 132, 133, 135–137, 139–142, 170, 172

MAPPO Multi Agent Proximal Policy Optimization. xix, 119–121, 125, 131

MARL Multi-Agent Reinforcement Learning. 55

MI Mean Interval. xvii, 33, 45, 72

ML Machine Learning. 144

MPE Multi-robot Pursuit Evasion. 29, 30

MSI Mean Square Interval. 34, 72

MSP Multilevel Subgraph Patrolling. 39, 40, 58

MTBA Mediated Trader Bidder Agent. 38

P-CMOMMT Personality-CMOMMT. 19, 26, 73

PAMTS Profit-driven Adaptative Moving Targets Search. 20–22, 59

PB2 Population Based Bandits. 172

PBT Population Based Training algorithm. 112, 113, 117, 125, 173

PGS Probabilistic Greedy Selection. 50, 53, 76

PHD Probability Hypothesis Density. xxvii, 28, 29, 58

PHD-DMST PHD - DMST. 28

PID Proportional–integral–derivative controller. 162

PIP Predefined Interest Points. xxiii, 89, 98–100

POP Problème de l'Observation appuyée par la Patrouille. xviii, xix, 63, 64, 66–70, 72–74, 76–78, 80–82, 84, 86, 88–90, 92, 94, 96, 98–100, 102, 104, 105, 120, 128, 131, 132, 134, 143, 145, 146, 169, 170, 172

PPO Proximal Policy Optimization. 111, 112, 117, 119, 125, 172

RAMPAGER RAndom Multiagent PAtrollinG LSTM-Path-MakER. 47

RaS Random Selection. 50, 53

RLPM Random-Next-Neighbour-LSTM-Path-Maker. 46, 47

ROS Robot Operating System. 38, 153–155

RR Random Reactive. xvii, 35, 38, 41, 42, 50, 81

RS Repulsive Selection. 51

SA Simulated Annealing. 54

SC Single Cycle. xvii, 37, 39, 41, 42

SCAT Simultaneous Coverage and Tracking. xxviii, 27–29

SEBS State Exchange Bayesian Strategy. 48, 57, 58

SG Stratégie Gravitationnelle. 37

SIHI Single iteration among highest idleness. xv, 92, 93, 96

SR Stratégie Régionalisée. 37, 39

TSBA Two-Shot-Bidder Agent. 38

TVDF-SCAT Time Varying Density Function - SCAT. 28

Première partie

Introduction et état de l’art

Chapitre 1

Introduction

1.1 Contexte général

Le suivi de cibles consiste à obtenir, de manière continue et précise, la position et les mouvements d'objets ou d'entités spécifiques. Cette problématique, explicitée au sein de la section 2.1, trouve des applications concrètes dans de nombreux domaines, que ce soit dans le domaine militaire, la surveillance de l'environnement, la sécurité publique ou encore la recherche et le sauvetage. Selon les domaines d'application, l'emploi de drone pour effectuer la tâche du suivi de cibles offre de nombreux avantages :

1. Surveillance et sécurité : Les drones équipés de caméras et d'autres capteurs peuvent fournir une surveillance en temps réel de certaines zones, permettant ainsi de détecter et de suivre des cibles potentielles. Dans les domaines de la sécurité publique et de la défense, cela peut aider à la prévention des crimes, à la surveillance des frontières, à la détection des intrusions, à la surveillance des foules lors d'événements majeurs, ou encore à la collecte de renseignements pour des opérations militaires.
2. Recherche et sauvetage : Le suivi de cibles par drone est intéressant dans les opérations de recherche et de sauvetage, en particulier dans des zones difficiles d'accès pour les équipes de secours traditionnelles. Grâce à leur agilité et à leur capacité de vol stationnaire, les drones à voilure tournante peuvent repérer et maintenir un suivi continu des personnes perdues suite à un sinistre.
3. Gestion de l'environnement : Les drones peuvent être déployés pour surveiller l'état de l'environnement, comme la détection des incendies de forêt, la surveillance de la faune pour suivre le déplacement d'animaux sauvages à protéger, ou encore détecter la présence de braconnier pour ensuite les pister. Ces informations permettent une prise de décision plus rapide et plus précise en matière de protection de l'environnement.

Nous considérons l'utilisation de plusieurs drones au sein d'un même environnement comme un système multi-robots, ou encore plus largement comme un système multi-agents [42]. Un système multi-agents est un système composé de multiples entités autonomes, appelées agents, qui interagissent les uns avec les autres pour atteindre des objectifs communs ou individuels. Chaque agent possède ses propres capacités cognitives, perceptives et d'actions, ce qui lui permet de prendre des décisions indépendantes en fonction de son environnement et des informations qu'il perçoit. Dans le cas où les agents, ici par exemple les drones, collaborent afin d'atteindre un objectif commun d'observer un maximum de cibles mobiles au cours du temps, alors nous nous plaçons dans le contexte de la problématique de l'observation.

Les drones présentent également des avantages pour assurer des missions de patrouille. L'objectif, énoncé au sein de la section 2.2, est de couvrir continuellement l'environnement. Autrement dit, chaque lieu de l'environnement doit être visité aussi régulièrement que possible, et ce, avec un nombre limité d'agents. Les drones sont particulièrement prisés pour les raisons suivantes :

1. Dynamique de vol : Les drones peuvent rapidement survoler de vastes zones, offrant une couverture aérienne bien supérieure à celle des patrouilles traditionnelles au sol. Cela permet aux forces de sécurité de surveiller des zones étendues, telles que les frontières, les zones urbaines denses ou les régions difficilement accessibles, avec une plus grande facilité et une efficacité accrue.
2. Flexibilité et adaptabilité : Les drones offrent une flexibilité opérationnelle considérable. Ils peuvent être déployés rapidement et facilement dans différentes situations et environnements. De plus, les drones peuvent être équipés de différentes charges utiles, telles que des capteurs infrarouges pour la vision nocturne, des détecteurs de gaz, des radars ou des équipements de communication, renforçant ainsi les capacités de surveillance et d'intervention.
3. Réduction des risques : Les drones peuvent être déployés dans des situations dangereuses, où l'environnement peut être nuisible, voire mortelle, pour des interventions humaines. Le scénario de patrouille peut inclure des désastres naturels ou industriels, mais aussi la présence de force hostile. Par conséquent, l'emploi de drone contribue à garantir la sécurité des équipes sur le terrain et à prévenir les incidents indésirables.
4. Coût-efficacité : L'utilisation de drones pour la patrouille peut offrir des avantages économiques significatifs. Comparés par exemple à l'emploi d'hélicoptère, les drones sont moins chers, plus économes en carburant et nécessitent moins de personnel.

Ainsi, les drones peuvent contribuer à réduire les coûts opérationnels d'une mission de patrouille.

Cette thèse aborde le défi de résoudre simultanément les questions de l'observation de cibles et de la patrouille de l'environnement. En d'autres termes, cela implique le développement de méthodes multi-agents combinant à la fois une couverture régulière de l'environnement, destinée à rechercher des cibles en mouvement, et du suivi des cibles qui ont été repérées. Nous considérons que les agents déployés sont des drones, lesquels, comme précédemment évoqué, offrent des atouts considérables pour accomplir ces tâches. L'intérêt de résoudre ces deux problématiques repose sur le constat que les approches visant à résoudre la problématique de l'observation se focalisent principalement sur la coordination entre les agents pour optimiser le suivi des cibles, sans expliciter les stratégies de recherche de nouvelles cibles, car considérées comme un niveau d'abstraction élevé.

Par ailleurs, l'incorporation de stratégies de recherche de cibles, en plus de leur suivi, fait émerger de nouveaux objectifs tels que la minimisation des disparités d'observation. En d'autres termes, d'équilibrer l'observation entre les cibles durant la mission. En effet, les agents peuvent choisir d'accomplir des tâches de recherche, au détriment du suivi d'une ou plusieurs cibles en cours. De plus, en adoptant cette approche, le risque de manipulation par les cibles intelligentes, qui sont conscientes de leur capacité à influencer le comportement des agents qui les suivent, est également réduit.

Associer l'observation avec la patrouille permet de matérialiser des missions variées. Par exemple, les missions de recherche et de sauvetage, où le but des agents est de trouver un compromis entre la recherche d'individus à secourir via la patrouille, et le suivi de ces individus pour communiquer en temps réel leurs positions aux secouristes. D'autres missions liées cette fois à la surveillance et la sécurité consistent à chercher et à suivre des intrus présents sur des zones dont l'accès est restreint. En dernier lieu, des missions similaires s'appliquent à la gestion de l'environnement, comme la préservation d'animaux protégés en réserve naturelle via le suivi de braconnier. Ou encore la surveillance d'animaux sauvages tels que les requins, tout en maintenant une patrouille ininterrompue pour détecter d'autres occurrences sur place, telles que la présence de nageurs aux alentours.

Néanmoins, les drones ont également leurs inconvénients. Leur autonomie énergétique constitue un défi majeur [15], ce qui suscite de nombreuses recherches visant à optimiser les trajectoires effectuées [2, 75, 63] ou prolonger leur durée de vol par de nouvelles sources d'alimentation [57]. À l'heure actuelle, un drone à voilure tournante est généralement limité à une autonomie de plusieurs dizaines de minutes. Il convient également de noter que la

dynamique des drones peut avoir des conséquences sur la détection des cibles en fonction des capteurs embarqués [119]. Au sein de cette thèse, nous ne considérons pas les enjeux liés à l'optimisation de la consommation énergétique. Par ailleurs, nous supposons que les capteurs sont parfaits, ne présentant pas de faux positifs ou faux négatifs lors de l'observation des cibles ou de la visite des lieux à patrouiller. Pour terminer, nous prenons l'hypothèse que les cibles sont mobiles et qu'elles ne possèdent pas la capacité de se dissimuler ou de se camoufler face aux agents. Ni également de riposter afin de nuire à l'intégrité des agents.

1.2 Plan de la recherche

1.2.1 Les questions de recherche (QR)

Au sein de cette section, nous formulons l'ensemble des questions auxquelles la thèse cherche à répondre :

QR1 : Quel est l'état actuel de la recherche concernant, dans un premier temps, la problématique de l'observation, et dans un second temps, la problématique de la patrouille ? Quel formalisme permettrait d'illustrer la double problématique de l'observation et de la patrouille ?

Contrainte 1 : Les cibles sont supposées mobiles, et n'ayant pas la capacité de se camoufler des agents.

Contrainte 2 : Afin d'être au plus proche de la dynamique des drones, nous souhaitons que les agents puissent se déplacer en espace libre, sans être restreint dans leurs mouvements.

QR2 : Comment développer une méthode cherchant un compromis entre la priorité de la patrouille et de celle de l'observation ?

QR3 : En quoi un lieu peut être considéré pertinent à visiter pour assurer une patrouille efficace ? Par quel(s) moyen(s) les agents peuvent-ils identifier ces lieux, en utilisant le formalisme mentionné à la question QR2 ?

QR4 : De quelle(s) manière(s) l'exploitation de l'apprentissage peut-elle mener au développement de stratégies cherchant à résoudre les défis liés à l'observation et à la patrouille ?

QR5 : Par quel(s) moyen(s) pouvons-nous optimiser le nombre d'agents, ainsi que d'autres paramètres de la mission, tout en s'assurant d'atteindre des objectifs préétablis ?

1.2.2 Méthode de résolution des questions de recherche

Résolution de QR1

- **Tâche 1 :** Réaliser une analyse méthodique de la littérature sur les recherches existantes concernant la problématique de l'observation et la problématique de la patrouille.
- **Tâche 2 :** Examiner en détail les limitations principales des formalismes actuels qui rendent complexe l'intégration de la patrouille dans le contexte de l'observation.
- **Tâche 3 :** Développer, en s'inspirant des formalismes déjà existants, un formalisme respectant les contraintes énoncées. Reconsidérer et réajuster les métriques d'évaluation afin de les redéfinir en conséquence. Réorganiser le fonctionnement des méthodes basées sur des formalismes anciens pour les adapter au nouveau contexte.
- **Tâche 4 :** Présenter les résultats des tâches précédentes en conférence [22].

Résolution de QR2

- **Tâche 1 :** Intégrer les objectifs de la patrouille dans une approche distribuée visant à résoudre la problématique de l'observation par un champ de force.
- **Tâche 2 :** Spécifier les informations échangées entre les agents et détailler les systèmes de collaboration qui en résultent.
- **Tâche 3 :** Comparer les résultats, ainsi que le compromis réalisé entre les deux objectifs, avec d'autres approches cherchant à résoudre soit la problématique de l'observation, soit la problématique de la patrouille.
- **Tâche 4 :** Présenter les résultats des tâches précédentes en conférence [22].

Résolution de QR3

- **Tâche 1 :** Établir les critères permettant de déterminer quelles sont les caractéristiques d'une zone intéressante à visiter pour les agents.
- **Tâche 2 :** Développer des algorithmes permettant d'identifier la localisation des zones qui correspondent aux critères énoncés précédemment.
- **Tâche 3 :** Effectuer une comparaison des temps d'exécution entre les différents algorithmes.
- **Tâche 4 :** Analyser la localisation des zones d'intérêt pour la patrouille par rapport à d'autres approches documentées dans la littérature.
- **Tâche 5 :** Présenter les résultats des tâches précédentes en conférence [23].

Résolution de QR4

- **Tâche 1 :** Répertorier les architectures existantes dans la littérature pour mettre en place un apprentissage multi-agents.
- **Tâche 2 :** Développer une approche reposant sur l'apprentissage par renforcement pour suivre des cibles aléatoires et évasives.
- **Tâche 3 :** Améliorer la collaboration entre les agents avec une architecture d'apprentissage centralisée pour une exécution distribuée.
- **Tâche 4 :** Développer une approche imitant le comportement d'une stratégie centralisée grâce à l'apprentissage supervisé.
- **Tâche 5 :** Comparer les différentes approches entre elles et avec les méthodes de la littérature.
- **Tâche 6 :** Présenter les résultats des tâches 1 et 2 en conférence [25].

Résolution de QR5

- **Tâche 1 :** Effectuer une étude approfondie de la littérature portant sur les outils permettant d'ajuster les paramètres d'une simulation en fonction des résultats obtenus, dans le but d'obtenir les paramètres optimaux qui respectent les contraintes définies par les utilisateurs.
- **Tâche 2 :** Concevoir une architecture pour l'outil d'aide à la décision qui intègre les paramètres à configurer en entrée, afin de générer en sortie la configuration optimale qui respecte les contraintes relatives aux critères d'évaluation.
- **Tâche 3 :** Expérimenter l'outil sur plusieurs scénarios, et analyser les résultats obtenus.
- **Tâche 4 :** Présenter les résultats des tâches précédentes en conférence [24].

1.2.3 Plan de la thèse pour répondre aux questions de recherche

Dans le but de répondre à la question QR1 de manière approfondie, une étude méthodique de la littérature est menée dans le chapitre 2, portant sur la problématique de l'observation, puis sur la problématique de la patrouille. Nous concentrons nos recherches sur les formalismes et les méthodes ayant une approche multi-agents.

Par la suite, le chapitre 3 répond à la seconde partie de la question QR1 en définissant un formalisme associant à la fois la problématique de l'observation à la problématique de la patrouille, tout en assurant un déplacement libre des agents dans un environnement continue. Dans ce même chapitre, une approche distribuée basée sur un champ potentiel est présentée

pour aborder la question QR2, afin de trouver un compromis entre les objectifs parfois divergents de ce nouveau formalisme. Dans la continuité de ce chapitre, la question QR3 est abordée en menant une réflexion sur les facteurs rendant un lieu attrayant pour la patrouille. Par la suite, plusieurs algorithmes sont mis en place pour l'identification de ces lieux au sein d'un environnement.

Pour répondre à la question QR4, le chapitre 4 se focalise sur l'étude et la mise en œuvre d'algorithmes d'apprentissage multi-agents, visant à permettre une coordination distribuée entre les agents. Le chapitre 5 présente la création d'un outil pour répondre à la question QR5. Ce dernier cherche à comprendre l'impact des paramètres d'une mission, comme le nombre d'agents, sur les performances d'une méthode, et ce grâce à de l'apprentissage supervisé. Par la suite, à l'aide d'algorithmes d'optimisation, l'outil retourne un paramétrage optimal, maximisant ou minimisant les paramètres sélectionnés, tout en s'assurant de valider les performances désirées par l'utilisateur.

Le chapitre 6 aborde spécifiquement la mise en place d'expérimentations sur des drones pour évaluer les méthodes étudiées, à la fois en utilisant la simulation et en effectuant des vols en volière. Les cibles y sont représentées sous forme de robots au sol. Enfin, le chapitre 7 fournit une conclusion sur les travaux de cette thèse.

Chapitre 2

État de l'art : Formalismes et méthodes de résolution de la patrouille et du suivi de cibles mobiles

Cette partie présente un état de l'art sur deux sujets principaux : tout d'abord, la problématique de l'observation, puis celle de la patrouille. Chaque problématique est abordée en commençant par une définition générale, suivie des métriques d'évaluation qui la caractérisent, et enfin, une étude des approches, méthodes et formalismes associés à ces problématiques. La figure 2.1 illustre, à travers une classification sous forme de graphe, le positionnement de chaque méthode et formalisme en fonction de leurs caractéristiques.

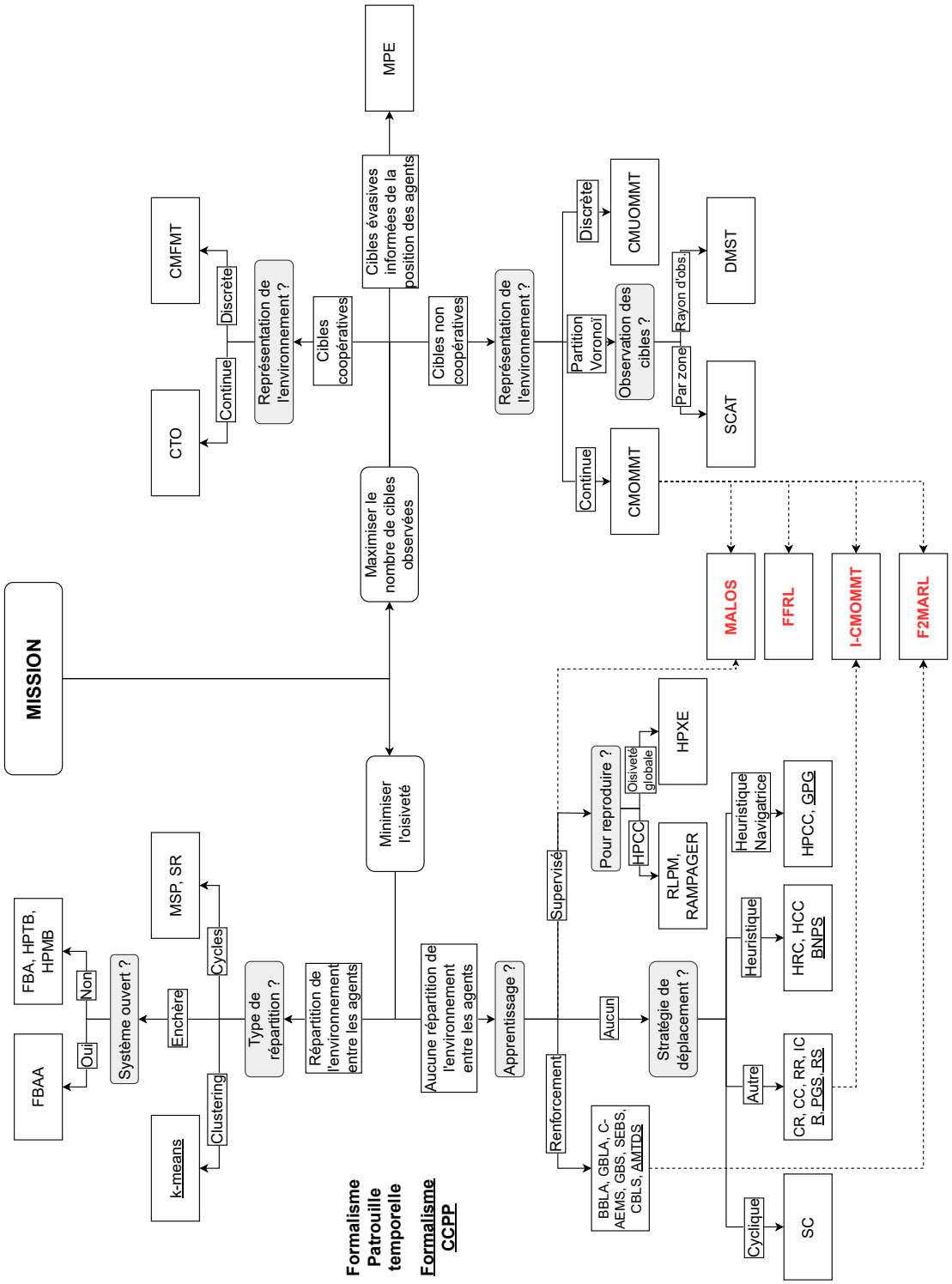


FIGURE 2.1 Illustration, sous forme d'arbre, de l'état de l'art des problématiques de l'observation et de la patrouille. Les contributions de la thèse sont indiquées en rouge.

2.1 État de l'art sur la problématique de l'observation

2.1.1 Contexte et définition

La problématique de l'observation se définit comme la maximisation au cours du temps du nombre de cibles mobiles observées par au moins un agent. Pour ce faire, chaque agent possède une capacité d'observation, pouvant par exemple être matérialisée par un rayon d'observation. L'objectif est ainsi de déplacer les agents pour observer simultanément l'ensemble des cibles, ou à défaut de minimiser le temps pendant lequel une cible n'est pas observée par au moins un agent. Les agents travaillent en collaboration, c'est-à-dire en équipe. Ils échangent ainsi des informations et cherchent à se coordonner pour maximiser l'utilité globale, qui est ici le nombre de cibles observées par au moins un agent.

Les travaux de KHAN, RINNER et CAVALLARO [62] répertorient deux types de coopération des cibles avec les agents. Les cibles dites coopératives partagent continuellement leurs positions à l'ensemble des agents. Dans ce cadre d'étude, répertorié au sein de la section 2.1.5, les agents cherchent à maximiser l'observation des cibles en connaissant leurs positions à chaque instant dans l'environnement. Tandis que les cibles dites neutres, ou non-coopératives, doivent être préalablement observées par un agent pour que ce dernier puisse connaître leurs positions.

Par ailleurs, les cibles peuvent adopter plusieurs comportements vis-à-vis des agents. Le comportement le plus étudié au sein de la littérature est le comportement aléatoire ou naïf. Dans ce cas, les cibles se déplacent de position en position, sélectionnées aléatoirement [11], ou adoptent un déplacement linéaire avec une probabilité à chaque pas de temps de s'orienter vers un nouveau cap, dont l'orientation est aléatoire [123, 33]. Un second comportement, le comportement évasif, suppose que les cibles ont la capacité de détecter les agents environnants, à travers par exemple un rayon de détection, et de fuir en conséquence. Les cibles évasives réactives [92, 66] vont dans le sens opposé des agents afin de minimiser le temps durant lequel elles sont observées, tandis que les cibles évasives rationnelles [31] adoptent une stratégie d'évasion s'adaptant au comportement des agents.

Le nombre de cibles présent au sein de l'environnement peut évoluer selon les caractéristiques du scénario étudié. Ainsi, un système dit fermé considère que les frontières de l'environnement sont hermétiques, avec un nombre de cibles fixé et constant. Tandis qu'un système dit ouvert permet aux cibles d'entrer et de sortir de l'environnement. Cette ouverture du système peut se faire via des points d'entrées et de sorties spécifiques dont la localisation

est connue des agents, ou sans contraintes particulières, où les cibles peuvent rejoindre ou quitter l'environnement à n'importe quel endroit de la frontière.

Les scénarios s'inscrivant au sein de la problématique de l'observation sont nombreux. Afin de les conceptualiser, les formalismes permettent de définir un cadre spécifique d'étude, en y précisant par exemple la représentation de l'environnement, les capacités d'observation et de communication des agents, ou encore le type de coopération des cibles. L'objectif de cet état de l'art est d'étudier l'évolution de ces formalismes, et pour chacun d'entre eux, répertorier les méthodes de résolution s'y inscrivant.

2.1.2 Les métriques d'évaluation

Au sein de la littérature, plusieurs métriques d'évaluation permettent de mesurer et de comparer les performances des méthodes résolvant la problématique de l'observation. Chaque métrique permet d'évaluer un aspect spécifique de l'observation des cibles. Ainsi, la première métrique, nommée métrique A, calcule la moyenne des cibles observées par au moins un agent au cours de la mission. Cependant, évaluer une méthode uniquement à travers le prisme de la moyenne ne permet pas d'apprécier la performance des agents à observer une large diversité de cibles. Pour y pallier, d'autres métriques, telles que la métrique H ou la déviation standard, s'intéressent à la distribution de l'observation. Ces métriques permettent d'étudier si toutes les cibles sont observées de manière uniforme, ou si certaines sont plus souvent observées que d'autres. L'ensemble des métriques concernant la problématique de l'observation sont répertoriées au sein de cette section.

La définition des métriques repose sur la formulation de l'environnement suivante :

— A est un ensemble de m agents. Chaque agent appartenant à A est noté a_i, avec $i \in [1,m]$.

— O est un ensemble de n cibles. Chaque cible appartenant à O est notée o_j, avec $j \in [1,n]$.

— Les métriques s'opèrent sur un temps $t = [0,T]$. Le temps peut s'exprimer en pas de temps, ou de manière continue.

La métrique A

La métrique A [92] représente le nombre de cibles observées en moyenne par au moins un agent durant une période T. Elle peut aussi se nommer *average number of observed targets* (ANOT) [7], ou encore "la moyenne d'observation des cibles". Dans le cadre de la

problématique de l'observation, une stratégie efficace vise à maximiser la métrique A. Pour la définir, posons la fonction binaire suivante :

$$\alpha(j,t) = \begin{cases} 1 & \text{si la cible } o_j \text{ est observée au temps } t \\ 0 & \text{sinon} \end{cases}$$

Ainsi, avec n le nombre de cibles, et T la durée de l'évaluation, la métrique A est définie par :

$$A = \frac{1}{T} \sum_{t=0}^{T} \sum_{j=1}^{n} \alpha(j,t) \text{ avec } A \in [0;n] \tag{2.1}$$

Et sa formule normalisée s'exprime par :

$$A_n = A/n \text{ avec } A_n \in [0;1] \tag{2.2}$$

L'entropie H

Définie par DING et al. [38], l'entropie permet d'évaluer la diversité de l'observation parmi les cibles. Une forte entropie indique une observation bien répartie, les cibles étant équitablement vues. Au contraire, une entropie faible signifie qu'une partie des cibles est plus longtemps observée qu'une autre, soulignant ainsi une disparité au sein de l'observation. Par conséquent, une méthode efficace dans la répartition de l'observation tend à maximiser l'entropie H. Reposant sur la définition de l'entropie de Shannon [114], le calcul s'opère de la manière suivante :

$$H = H(p_1, p_2, ..., p_n) = -\sum_{j=1}^{n} H(p_j), \text{ avec } H \in [0; log_2(n)] \tag{2.3}$$

$$H(p_j) = \begin{cases} p_j \, log_2 \, p_j & \text{si } p_j > 0 \\ 0 & \text{sinon} \end{cases}$$

Où p_j représente la durée d'observation total de la cible j, divisée par le temps total d'observation de l'ensemble des n cibles. Ainsi, $p_j \in [0;1]$ est défini par :

$$p_j = \frac{\sum_{t=0}^{T} \alpha(j,t)}{\sum_{k=1}^{n} \sum_{t=0}^{T} \alpha(k,t)}$$

Néanmoins, si aucune cible n'a été observée pendant la période d'évaluation T, l'entropie H perd de son sens. En effet, la définition de p_j ne peut être utilisée, à cause de la division par zéro. Dans ce cas particulier, l'entropie H est considérée comme nulle.

La déviation standard σ_n

La déviation standard (également appelée écart-type) est un paramètre statistique utilisé pour mesurer la dispersion des valeurs autour de la moyenne. Elle est fréquemment employée, tout comme la métrique H, pour quantifier la diversité des observations des cibles. Une méthode cherchant à observer équitablement les cibles tend ainsi à minimiser cette métrique. Dans ce contexte, une déviation standard nulle signifie que les cibles ont été observées pendant des durées égales au cours de la mission, ce qui est considéré comme un critère de réussite. Notamment utilisée au sein des travaux de YAN, JIA et BAI [131] et BANFI et al. [14], la déviation standard se définit de la manière suivante :

$$\sigma_n = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (\overline{\alpha}(j) - A_n)} \tag{2.4}$$

Avec A_n la moyenne normalisée de l'observation des cibles définie précédemment, et n le nombre de cibles. De plus, la définition de $\overline{\alpha}(j)$ s'exprime par :

$$\overline{\alpha}(j) = \frac{1}{T} \sum_{t=0}^{T} \alpha(j,t)$$

Avec T la durée totale de la mission, ou la durée de l'évaluation, et t un instant temporel issu de la discrétisation du temps.

2.1.3 Le formalisme pionnier du problème de l'observation : CMOMMT

Le formalisme du *Multi-Robot Observation of Multiple Moving Targets* (CMOMMT), élaboré par PARKER [92], définit pour la première fois la problématique de l'observation. Les agents, qui peuvent être considérés comme des robots, évoluent dans un environnement fermé en deux dimensions. Ainsi, les agents ne peuvent ni entrer, ni sortir de l'environnement. Les agents, homogènes, peuvent observer les cibles autour d'eux sur un rayon d'observation. De plus, ils peuvent communiquer entre eux autour d'un rayon de communication, comme représenté au sein de la figure 2.2.

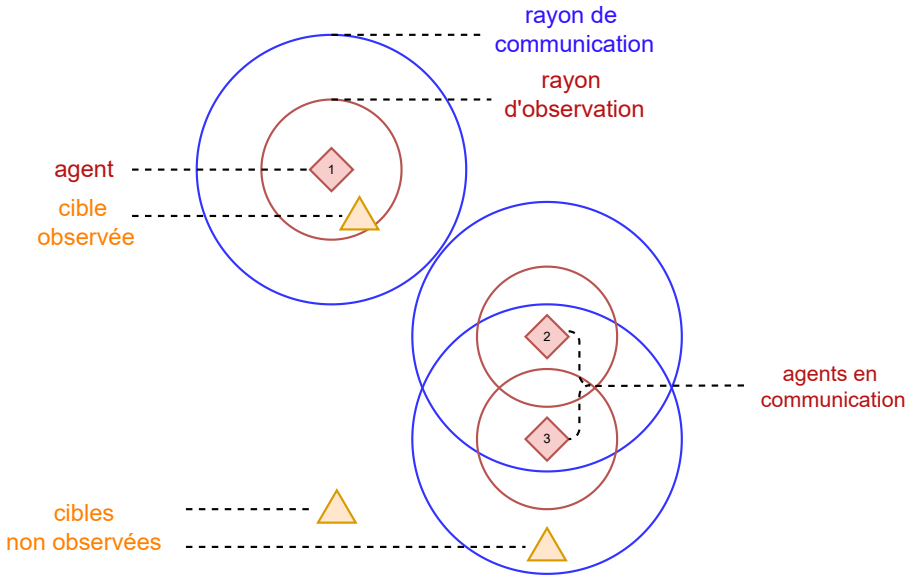


FIGURE 2.2 Illustration du formalisme CMOMMT.

Le **A-CMOMMT** [92] est une des premières approches cherchant à résoudre le problème du CMOMMT. Le comportement des agents est décrit grâce à l'utilisation d'un champ de force. Le comportement d'un agent i est décrit par une force d'attraction $f^t_{i,j}$ pour chaque cible j dans son rayon d'observation, mais aussi par une force de répulsion $f^r_{i,k}$ provenant de chacun des agents alliés k appartenant à son rayon de communication pour éviter tout risque de collision. Ainsi, les forces d'attraction sont orientées vers les cibles et les forces de répulsion dans le sens opposé des autres agents. Ainsi, le comportement de l'agent i est régi par la somme des forces qui se définit de la manière suivante :

$$F(a_i,t) = \sum_{j=1}^{n} \omega_{i,j} f^t_{i,j} + \sum_{k=1}^{m} f^r_{i,k} \tag{2.5}$$

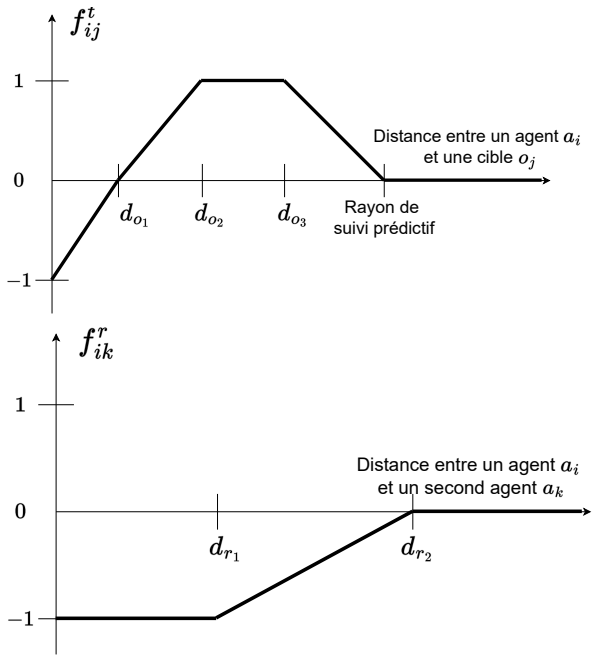


FIGURE 2.3 Magnitude des forces $f_{i,j}^t$ et $f_{i,k}^r$ en fonction de la distance entre un agent et une cible, ou entre deux agents.

Les magnitudes des forces $f_{i,j}^t$ et $f_{i,k}^r$ sont détaillées au sein de la figure 2.3 et s'expriment en fonction de la distance séparant l'agent de la cible ou de l'agent allié. À la différence d'un champ de force classique, appelé aussi *local force*, la méthode du A-CMOMMT attribue à chacune des forces d'attraction un coefficient $\omega_{i,j}$. Ainsi, ce coefficient est employé pour réduire l'attraction vers les cibles déjà observées par au moins un autre agent plus proche, tout en prenant en considération la densité des cibles dans son voisinage. Cependant, le coefficient ω n'est pas défini de manière formelle par PARKER [92] et est par conséquent libre d'interprétation de la part des expérimentateurs. Par ailleurs, une des limitations du A-CMOMMT, est le risque que les agents puissent perdre l'observation de plusieurs cibles dans le cas où elles se sépareraient de manière symétrique.

Pour faire face à ce phénomène de perte de cibles, les auteurs de [66] proposent une nouvelle méthode nommée le ***Behavior-CMOMMT*** **(B-CMOMMT)**. La méthode repose sur le principe du champ de force de son prédécesseur, le A-CMOMMT, en y ajoutant trois possibles comportements pour un agent : suivre, aider ou explorer. De plus, les auteurs prennent l'hypothèse que chaque cible est identifiable via un tag unique. En ne communiquant pas uniquement leurs positions, comme c'est le cas pour le A-CMOMMT, mais aussi une possible demande d'aide, les agents peuvent ainsi s'entraider et mieux se répartir les cibles à suivre. Cette demande d'aide s'opère lorsque les agents constatent que les cibles

suivies sortent progressivement de leurs zones d'observation. L'exploration est à un niveau d'abstraction supérieur et non détaillée.

Les travaux de DING et al. [38] mettent en évidence la relation entre le bénéfice individuel des agents à observer des cibles, c'est-à-dire le nombre de cibles observées par chaque agent, et le bénéfice collectif de l'observation, qui se définit comme le nombre total de cibles observées par au moins un agent. Les auteurs constatent que sans un comportement égoïste (bénéfice individuel) par les agents, les cibles ne seraient jamais observées. Cependant, le comportement uniquement égoïste peut nuire au bénéfice collectif si les cibles sont observées par plusieurs agents simultanément. Afin de trouver un compromis entre ces deux bénéfices, et d'encourager les agents à adopter en fonction du contexte soit un comportement égoïste, soit un comportement collaboratif altruiste, les auteurs ont conçu la méthode du ***Personality-CMOMMT*** **(P-CMOMMT)**. Cette approche repose sur le champ de force de la méthode du A-CMOMMT (cf. l'équation 2.5), tout en y apportant une redéfinition explicite au coefficient ω, renommé α. Ce coefficient est utilisé, initialement, pour que les agents aient moins d'attirance envers une cible déjà observée par un autre agent plus proche. Au sein du P-CMOMMT, les agents cherchent à équilibrer la répartition de l'observation des cibles, en se désintéressant de suivre une cible si cette dernière a été observée trop longtemps en comparaison des autres cibles de l'environnement. La distribution de l'observation parmi les agents est calculée selon la notion de l'entropie de Shannon (cf. équation 2.3). Les résultats expérimentaux montrent une amélioration de la distribution de l'observation des cibles en comparaison avec le A-CMOMMT.

Par la suite, DING et HE [37] se sont intéressés à développer des formations flexibles d'agents, rompant ainsi avec un système de champs de force virtuels. La méthode développée, appelée ***Flexible-CMOMMT*** **(F-CMOMMT)**, cherche à ce que les agents maintiennent une formation suivant une géométrie spécifique lors du suivi d'une cible, afin d'en améliorer l'observation globale. Pour ce faire, la formation cherche à limiter la superposition des surfaces d'observation des agents, résultant d'une meilleure observation des cibles en comparaison avec le A-CMOMMT et P-CMOMMT.

Dans la continuité du développement du A-CMOMMT, des approches proposent de développer des stratégies de suivi des cibles à travers de l'apprentissage. Une des premières approches par l'apprentissage [123] propose une méthode reposant sur du ***distributed lazy Q-learning*** afin de considérer les limites d'observabilité de chaque agent inhérent au problème du CMOMMT. Les agents perçoivent l'environnement à travers un cercle divisé en 16 portions. Deux vecteurs de 16 éléments chacun sont ainsi générés, permettant de renseigner la position des cibles et des autres agents environnants. Les composants des vecteurs contiennent

la distance relative de la cible ou de l'agent le plus proche. La distance avec une cible ou un agent est calculée uniquement si elle inférieure respectivement au rayon d'observation et de communication. Un même modèle est partagé au sein des agents, alimenté par les deux vecteurs de perception pour générer l'action à entreprendre. La coopération repose sur une récompense partagée parmi l'ensemble des agents en fonction du nombre de cibles observées. Même si l'efficacité d'observation moyenne du *distributed lazy Q-learning* est meilleure qu'une stratégie aléatoire, elle ne surpasse pas celle du A-CMOMMT.

L'implémentation d'une autre forme de Q-learning, le ***Evolutionary Nearest Neighbor Classifier -Q-learning*** **(ENNC-QL)**, est étudiée par FERNÁNDEZ, BORRAJO et PARKER [43] afin d'améliorer la collaboration parmi les agents. Ainsi, un modèle individualiste, considérant uniquement les coordonnées (x, y) de la cible la plus proche, est comparé avec un modèle collaboratif, incluant également les coordonnées de la cible observée la plus éloignée, et les coordonnées de l'agent allié le plus proche. Les auteurs ont montré que l'approche collaborative a une meilleure efficacité d'observation que l'approche individualiste, et dépasse l'approche du précédent *distributed lazy Q-learning*. Cependant, l'ENNC-QL collaboratif reste moins performant que le A-CMOMMT en termes d'observation moyenne des cibles.

Les performances de l'apprentissage par renforcement dépendent d'un ensemble de facteur, tels que le choix de l'algorithme d'apprentissage, des hyperparamètres, mais aussi de la manière de représenter les perceptions et les interactions avec l'environnement. Les observations et les actions des agents peuvent être ainsi représentées de manière continue ou discrète. En fonction du problème à résoudre, une représentation peut mener à un apprentissage plus performant comparée à une autre. Ce constat a justifié la création d'un formalisme reposant sur un environnement discret, explicité dans la section suivante.

2.1.4 Le formalisme discrétisé : CMUOMMT

Le formalisme du *Cooperative Multi-UAV Observation of Multiple Moving Targets* (CMUOMMT) [71] est une variante du CMOMMT. Au sein de ce formalisme, les agents et les cibles se déplacent au sein d'un environnement discrétisé, et non plus au sein d'un environnement continu, comme représenté au sein de la figure 2.4. Ainsi, les déplacements sont limités aux huit cellules voisines à chaque pas de temps. Cette discrétisation permet de représenter la perception de l'environnement par les agents, ici communément des drones, comme une image où chaque pixel est une cellule de l'environnement.

Pour résoudre le problème posé par le CMUOMMT, au sein de l'article [71], les auteurs y définissent la méthode du ***Profit-driven Adaptative Moving Targets Search*** **(PAMTS)**. Le

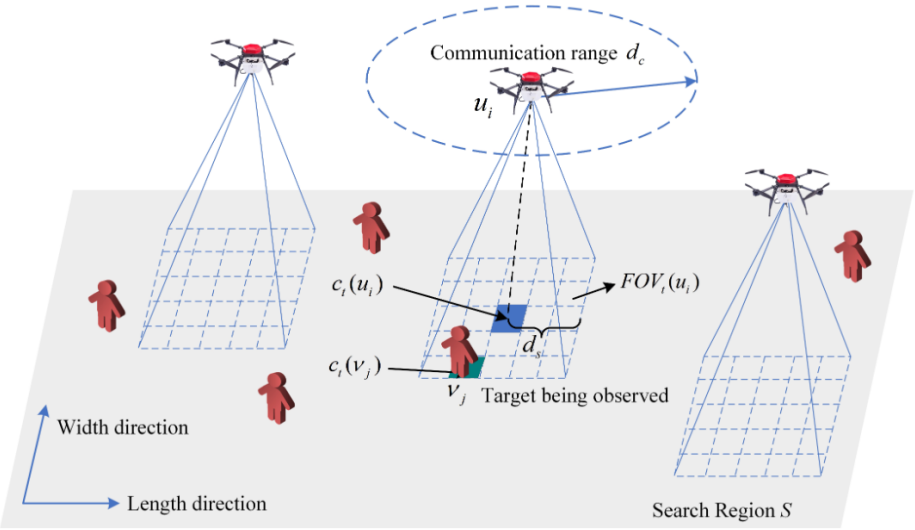


FIGURE 2.4 Représentation de l'environnement au sein du CMUOMMT. Schéma de [131].

problème de l'observation est approché par une dichotomie entre le suivi de cibles et l'exploration de l'environnement au sein d'une coordination distribuée. Les agents évaluent le gain du déplacement vers chaque cellule environnant vis-à-vis de l'observation des cibles (*profit of follow (PoF)*) et de l'exploration de l'environnement (*profit of exploration (PoE)*). Ces deux fonctions objectives sont pondérées par deux coefficients afin de trouver un compromis entre l'observation et la couverture. Les cellules ne contiennent pas de valeur d'oisiveté, mais possèdent deux états : visitée ou non. Une cellule observée redevient non visitée après un temps de rétablissement de χ pas de temps. Expérimentalement, la valeur optimale de $\chi = 5$ pas de temps. Au sein du formalisme du CMUOMMT, la méthode du PAMTS montre une meilleure efficacité d'observation moyenne que les méthodes du A-CMOMMT et du B-CMOMMT. Ces dernières ont été adaptées pour être fonctionnelles au sein de l'environnement discrétisé.

L'apprentissage par renforcement profond est employé par YAN, JIA et BAI [131] pour résoudre le CMUOMMT. La méthode n'ayant pas un nom spécifique, nous proposons de le nommer le ***Deep Reinforcement Learning - CMUOMMT*** **(DRL-CMUOMMT)**. La perception de chaque agent y est représentée sous forme de quatre images. La première image représente la localisation des cibles au sein de la surface d'observation. La deuxième positionne les agents appartenant à la surface de communication. La troisième met en évidence les limites de l'environnement environnant. Enfin, la dernière image représente le temps pendant lequel la zone n'a pas été visitée par l'agent en question. Le modèle entraîné repose sur une convolution des images suivie de deux couches cachées de 200 neurones chacune. Lors de l'entraînement, les agents obtiennent des récompenses à la fois individuelles et collectives, c'est-à-dire partagées parmi l'ensemble des agents. Les récompenses sont

obtenues en fonction de la couverture de l'environnement et de l'observation des cibles. Les punitions, c'est-à-dire les récompenses négatives, sont appliquées pour éviter toute collision entre agents ou sortie d'environnement. Ce même modèle est ensuite utilisé par l'ensemble des agents de manière distribuée. Les résultats expérimentaux montrent une moyenne d'observation des cibles relativement proche de la méthode du A-CMOMMT et légèrement plus faible que celle du PAMTS. Cependant, la distribution d'observation des cibles est améliorée par rapport aux deux précédentes méthodes évoquées, l'évaluation reposant sur la métrique de la déviation standard (cf. section 2.1.2).

2.1.5 Les formalismes intégrant des cibles coopératives : CTO et CMFMT

Les cibles dites coopératives partagent continuellement leurs positions à l'ensemble des agents au sein de l'environnement. L'objectif pour les agents reste inchangé, celui de maximiser l'observation des cibles. En effet, l'observation n'a pas pour unique but de connaître la position des cibles, ici déjà connue, mais peut aussi être utilisé pour avoir accès à l'état des cibles. Par exemple, les formalismes considérant des cibles coopératives peuvent être employés pour représenter des missions d'observation d'animaux sauvages [72] dotés de balise GPS, afin de surveiller et d'analyser le comportement de ces animaux à travers l'utilisation de drones fournissant un flux vidéo en temps réel.

Cooperative Targets Observation (CTO)

Le formalisme du *Cooperative Targets Observation* (CTO) [78] est une reformulation du formalisme du CMOMMT. La principale différence réside dans le changement d'attitude des cibles, considérée comme coopérative. Ainsi, les agents disposent à tout moment de la position de toutes les cibles au sein de l'environnement. En outre, les agents connaissent aussi la position des autres agents.

Les fondateurs du CTO ont initialement approché le problème à travers l'étude et la comparaison de deux méthodes de résolution centralisées. La première méthode, un **algorithme d'escalade** (ou *hill-climbing*), positionne tous les agents de manière à maximiser le nombre de cibles observées. La seconde méthode, un **algorithme de partitionnement en k-moyennes** (*k-means*) [76], identifie les regroupements (*cluster*) des cibles puis positionne les agents sur les centroïdes sans tenir compte des rayons d'observation des agents, comme illustrée par la figure 2.5. Ainsi, le nombre de regroupements est égal au nombre d'agents. Les résultats des expériences montrent que l'algorithme d'escalade est plus performant que l'algorithme de partitionnement dans le cas où la vitesse des cibles est relativement faible comparée à la vitesse des agents. Cependant, lorsque la vitesse des cibles augmente, ou que

le rayon d'observation est relativement grand, la méthode du *k-means* devient plus efficace pour maximiser l'observation des cibles. Cela s'explique par la nécessité pour l'algorithme d'optimisation de disposer de plus d'itération, et donc plus de temps de calcul, pour suivre efficacement les cibles rapides.

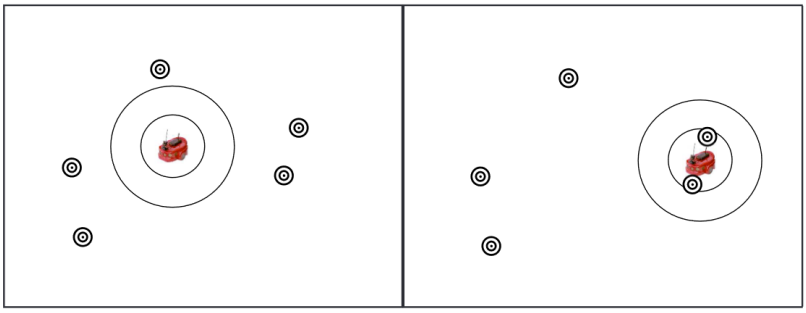


FIGURE 2.5 Représentation du comportement de l'agent pour la méthode par k-moyennes à gauche, et de l'algorithme d'escalade à droite. Illustration issue de [78].

Les auteurs LUKE et al. [78] proposent également une **méthode de combinaison** des deux précédentes approches. Ainsi, dans un premier temps, l'algorithme de partitionnement en k-moyennes est employé pour obtenir la position des centroïdes. Ensuite, ces positions sont utilisées comme référence pour initier l'algorithme d'escalade, qui optimisera la position des agents pour maximiser l'observation des cibles. La méthode de combinaison des deux algorithmes permet de trouver un compromis entre les avantages et les inconvénients des deux précédentes approches. Ainsi, cette dernière méthode est plus efficace que l'algorithme d'escalade dans la grande majorité des scénarios pour le suivi de cible, et rivalise bien avec l'approche du k-moyennes, à l'exception des cibles particulièrement rapides.

Afin de pallier les sous performances de la méthode du k-moyennes au sein des situations vues précédemment, ANDRADE et al. [7] conçoivent une méthode centralisée de partitionnement reposant sur le *Fuzzy C-means clustering* (FCM) [19], que nous nommons **FCM-CTO**. L'utilisation de l'algorithme FCM, à la place du k-moyennes, permet d'assigner une cible à plusieurs regroupements à la fois. L'objectif est de réduire le risque de positionner les agents à des centroïdes sans cibles à observer au sein du rayon d'observation. La méthode du FCM-CTO montre une amélioration de l'observation moyenne des cibles par rapport à la méthode du k-moyennes dans les cas où la vitesse des cibles est relativement lente (inférieure à la moitié de la vitesse des agents), et où le rayon d'observation est relativement faible (en deçà d'un rayon de 15 unités, pour un environnement de 150×150 unités). Cependant, cette nouvelle méthode reste moins performante que l'approche par k-moyennes lorsque, à l'inverse, les cibles sont rapides ou que le rayon d'observation est relativement grand.

L'observation des cibles ayant un comportement évasif est étudiée au sein des travaux de ASWANI, MUNNANGI et PARUCHURI [11]. Pour ce faire, les cibles sont équipées d'un rayon de détection, comparable au rayon d'observation des agents, afin de détecter si des agents sont à proximité et d'appliquer des stratégies d'évasions. Afin de maximiser l'observation des cibles évasives, la méthode développée du ***Cooperative Target Observation using Binary search for Randomization CTO*** **(BRLP-CTO)** prend inspiration de l'approche précédente par k-moyennes. Cependant, la méthode intègre également à l'algorithme de partitionnement une dimension de recherche de nouvelles cibles, à travers un dilemme exploration vs exploitation. Ainsi, l'objectif de cette méthode est d'identifier les regroupements des cibles, puis placer les agents sur les centroïdes tout en y ajoutant un déplacement aléatoire. La combinaison des deux objectifs, la recherche et le suivi de cibles, est pondérée par des coefficients. Afin d'évaluer la valeur des coefficients, trois configurations sont proposées : (1) Si aucune cible n'est à portée de l'agent, alors la priorité est à l'exploration. Sinon l'unique priorité est au suivi. (2) La priorité est à l'exploration si aucune cible n'est en vue. Si plus de deux cibles sont observées, alors la priorité est à l'observation. Sinon, la priorité entre l'exploration et l'exploitation est à 50/50. (3) La priorité à l'exploration est inversement proportionnelle au nombre de cibles observées. En conclusion, pour les différentes stratégies d'évasion des cibles évaluées, la première configuration des coefficients est la plus efficace. Malheureusement, les résultats obtenus ne sont pas comparés avec d'autres travaux de la littérature.

Les précédents travaux au sein du formalisme du CTO développent des méthodes d'observation en considérant uniquement des cibles ayant un comportement soit aléatoire (naïve), soit allant dans le sens opposé des agents (réactive). Afin de doter les cibles d'une stratégie d'évasion plus efficace, et ainsi de minimiser l'observation des cibles par les agents, plusieurs solutions sont proposées. Pour évaluer les stratégies d'évasion des cibles, la métrique *Average Number of Target Evasion* (ANTE) [31] est employée. L'ANTE mesure le succès d'évasion des cibles durant une mission et se définit par le nombre de cibles en moyenne qui ne sont pas observées par au moins un agent au cours de la mission. Le calcul correspond également au nombre de cibles total dans l'environnement moins la moyenne d'observation par les agents.

Afin de rendre les cibles rationnelles, et non pas uniquement réactives ou naïves, COSTA et al. [31] proposent que les cibles utilisent de l'apprentissage supervisé. Ainsi, les cibles améliorent la stratégie d'évasion en apprenant à prédire le comportement des agents appliquant la stratégie *k-means* [78]. Parmi les quatre types de réseaux de neurones expérimentés, le modèle ayant la meilleure performance maximisant l'ANTE est un réseau de neurones récurrent. En effet, la prédiction du comportement des agents est plus efficace avec un réseau

de neurones récurrent, car plus adapté à l'apprentissage d'une séquence temporelle qu'un perceptron multicouche. En parallèle de l'apprentissage, les cibles peuvent aussi améliorer le comportement évasif grâce à des algorithmes de *clustering* [32]. La mise en place d'un coordinateur, connaissant la position de l'ensemble des agents et des cibles, permet de positionner les cibles le plus éloigné possible des agents, et ce grâce aux méthodes *k-means*, FCM ou encore par une combinaison des deux méthodes. Après expérimentation, l'approche à travers l'utilisation d'un algorithme centralisé de *clustering* (*k-means*) est plus efficace que l'approche par l'apprentissage via un réseau de neurones vis-à-vis de la métrique ANTE, c'est-à-dire de la performance de fuite des cibles, minimisant ainsi leurs observations par des agents.

Cooperative Multirobot Fair Multitarget Tracking (CMFMT)

Le formalisme du *Cooperative Multirobot Fair Multitarget Tracking* (CMFMT) [14] considère l'hypothèse de cibles coopératives, fournissant à l'ensemble des agents leurs positions. Contrairement au CTO, les agents ne peuvent échanger d'information, et notamment leurs positions, qu'à travers un rayon de communication. De plus, la position future des cibles est supposée connue, mais de plus en plus imparfaite selon que l'horizon de prédiction s'accroît. Ces incertitudes sont représentées par les cercles violets de la figure 2.6. À l'instar du CMUOMMT, l'environnement est représenté de manière discrète. Ainsi, les agents et les cibles ne peuvent se déplacer que de cellule en cellule. Au sein du formalisme du CMFMT, les objectifs sont de maximiser à la fois l'observation moyenne des cibles, soit la métrique A (eq. 2.1), mais aussi que les cibles soient observées équitablement, de manière juste (décrit en anglais par le mot *"Fair"* dans le nom du formalisme), en minimisant la déviation standard σ_n (eq. 2.4).

Pour répondre aux objectifs du CMFMT, BANFI et al. [13] proposent la méthode de l'***Integer Linear Program*** **(ILP)**, composée de deux variantes, une centralisée et une autre distribuée. La méthode repose sur deux fonctions objectives. La première encourage les agents à se diriger vers des cellules ayant une forte probabilité d'apparition de cibles. Tandis que la seconde récompense les agents qui observent des cibles de manière à assurer une distribution équitable de l'observation de l'ensemble des cibles. Le compromis entre ces deux objectifs est arbitré par un coefficient α. Ainsi, les agents sont encouragés à suivre les cibles à proximité lorsque α tend vers 1, ou de privilégier l'exploration pour mieux uniformiser l'observation quand α tend vers 0. La prédiction de la position future des cibles est maintenue à l'aide d'un filtre Bayésien. La méthode ILP repose sur la planification d'actions sur un horizon fixe. Au sein de la version centralisée, un agent planificateur collecte l'ensemble

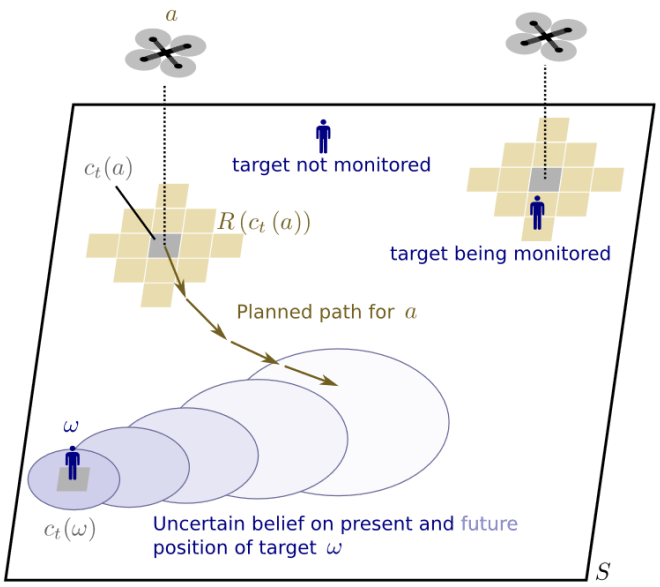


FIGURE 2.6 Représentation de l'environnement par le CMFMT, illustrée par BANFI et al. [13]. Un agent a se positionne sur une cellule $c_t(a)$ et observe un ensemble de cellules $R(c_t(a))$.

des observations des agents afin de calculer les chemins optimaux à effectuer par l'équipe d'agents. Cependant, ce calcul peut nécessiter un long temps d'exécution pour des grands scénarios. À la différence de la version distribuée, où chaque agent planifie indépendamment son parcours, à l'aide de ses observations et celles des agents en communication directe ou par relais.

Les approches ILP, A-CMOMMT et P-CMOMMT adaptées à un environnement discret ont été comparées. Les résultats montrent que la méthode ILP-distribuée affiche les meilleures performances en termes d'observation moyenne des cibles, suivie de près par les méthodes A-CMOMMT et ILP-centralisée. Le P-CMOMMT quant à lui, présente la plus faible moyenne d'observation. Toutefois, le P-CMOMMT distribue l'observation des cibles d'une manière plus équitable que le A-CMOMMT, bien que de manière moins efficace que l'approche ILP. Enfin, grâce à sa nature centralisée, l'ILP-centralisée offre une performance supérieure ou égale à celle de l'ILP-distribuée en termes de minimisation de la déviation standard de l'observation.

2.1.6 Les formalismes reposant sur une partition de Voronoï : SCAT et DMST

Les formalismes abordés au sein de cette section reposent sur le découpage de l'environnement via un diagramme de Voronoï. Appelée aussi décomposition de Voronoï, partition de Voronoï ou tessellation de Dirichlet, cette méthode permet de partitionner l'environnement en plusieurs régions. Soit un environnement en deux dimensions composé d'un ensemble discret de points, appelés "germes", comme matérialisé par la figure 2.7. Pour chaque germe, une région de Voronoï lui est associée. Cette région est caractérisée par une surface, dont les points ont une distance plus faible avec le germe associé qu'avec un autre germe de l'environnement. Dans notre cas d'étude, les germes représentent la position des agents.

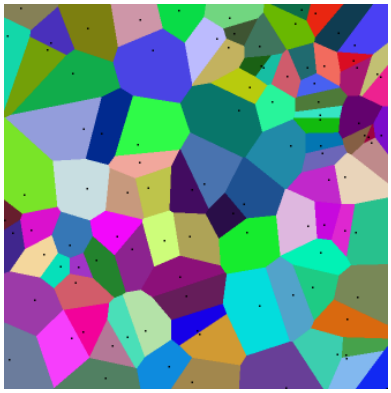


FIGURE 2.7 Représentation d'un partitionnement de l'environnement par un diagramme de Voronoï.

Dans le cadre d'un système multi-agents, le diagramme de Voronoï présente l'avantage de distribuer des régions en fonction de la position des agents dans l'environnement. Ainsi, les méthodes reposant sur ce type de diagramme tirent parti du fait que la zone assignée à un agent représente la totalité des emplacements où l'agent peut prendre des mesures plus rapidement que tout autre endroit.

Simultaneous Coverage and Tracking (SCAT)

Comme énoncé précédemment, le formalisme du *Simultaneous Coverage and Tracking* (SCAT)[94] utilise le diagramme de Voronoï. Les agents se déplacent au sein d'un environnement continu et peuvent observer l'ensemble de la région de Voronoï leur étant associé. La communication est possible entre deux agents si leurs régions ont une frontière commune. Les cibles sont non coopératives, par conséquent la position d'une cible n'est connue que de l'agent possédant la région où elle se situe. Le nombre de cibles peut évoluer au

cours du temps dans le cadre d'un environnement ouvert. Par ailleurs, les agents n'ont pas connaissance du lieu et du moment d'apparition des cibles.

Le formalisme du SCAT est abordé comme un problème d'attribution décentralisée de tâche [94]. La méthode proposée, que nous nommons ***Time Varying Density Function - SCAT*** **(TVDF-SCAT)**, a pour première tâche de répartir au mieux la couverture de l'environnement parmi les agents. Par la suite, une tâche de suivi de cibles est générée pour chaque cible présente au sein de l'environnement. L'assignation des tâches est modélisée par une fonction de densité évoluant au cours du temps (*time varying density function*), permettant d'attribuer la tâche la plus prioritaire à réaliser en fonction de la position de l'agent. L'objectif est, qu'au mieux, un agent observe au maximum une seule cible. Par ailleurs, la position et la vitesse des cibles est connue si ces dernières appartiennent à la région de Voronoï, ou si elles se situent à proximité. Durant les expérimentations, les agents se déplacent à l'aide d'une loi de commande reposant sur une fonction de Lyapunov. Ce contrôleur est fréquemment utilisé en robotique pour réduire de manière souple la distance entre la position désirée et la position actuelle de l'agent.

Le formalisme du SCAT a été critiqué sur le fait qu'une cible est par définition continuellement observée. En effet, la superposition des régions de Voronoï couvre l'ensemble de l'environnement. Et dans le cas extrême où il existe un seul agent au sein de l'environnement, alors sa région de Voronoï se confond avec l'environnement. Par conséquent, une cible appartient toujours à une région de Voronoï. La section suivante aborde un nouveau formalisme afin d'être plus proche de la problématique de l'observation.

Distributed Multi-Target Search and Tracking (DMST)

Le formalisme du *Distributed Multi-Target Search and Tracking* (DMST) [33] est similaire au formalisme du SCAT, à la différence que les cibles ne peuvent être observées qu'à travers un rayon d'observation. Ainsi, les cibles ne sont pas automatiquement observées lorsqu'elles appartiennent à une région. La communication entre deux agents ne peut s'effectuer que s'ils possèdent une frontière commune entre leurs régions de Voronoï. Dans ce cas, la communication est supposée parfaite, sans contrainte et instantanée. Par ailleurs, contrairement à l'ensemble des formalismes vu précédemment, le DMST intègre une limitation des capteurs au sein des agents à travers une probabilité d'observation, via une probabilité de faux positifs et de faux négatifs.

Une approche de résolution du DMST, que nous nommons **PHD-DMST**, propose d'obtenir le comportement des agents à l'aide d'un filtre *Probability Hypothesis Density* (PHD) [80].

Le filtre PHD permet de fusionner les données provenant de plusieurs capteurs afin d'estimer la position des cibles. Ainsi, lors de la détection d'une cible, une fonction de densité est générée autour d'elle, souvent représentée par une distribution gaussienne. En sommant ces fonctions de densité, la localisation des cibles est approximée aux emplacements correspondants aux pics de la fonction de densité du filtre PHD. Pour résoudre la problématique de l'observation dans le cadre du formalisme du DMST, chaque agent est considéré comme un capteur, et le filtre est utilisé par chaque agent de manière distribuée. Les agents communiquent aux autres agents des régions voisines leurs observations.

Les agents utilisent les pics de la fonction de densité générée par le filtre PHD pour suivre les cibles. Lorsque la localisation des cibles est inconnue, le filtre PHD est presque uniforme, obligeant les robots à couvrir l'ensemble de l'environnement. Si une zone ne contient pas de cibles, la probabilité d'y trouver une cible diminue et les agents sont moins susceptibles de s'y rendre. En revanche, lorsque qu'une cible est identifiée, la densité de cibles dans la région augmente, ce qui pousse un ou plusieurs agents à la suivre.

2.1.7 Le formalisme pour la capture des cibles : MPE

L'objectif du formalisme du *Multi-robot Pursuit Evasion* (MPE) [65] est de minimiser le temps de capture d'une ou plusieurs cibles évasives. Les cibles possèdent une connaissance parfaite de l'environnement et de la localisation des agents. De plus, les agents rentrent en compétition avec les cibles, pouvant faire appeler ainsi à la théorie des jeux. Ainsi, la stratégie des agents peut influer sur la stratégie des cibles. Enfin, au sein de certaines études, les cibles ont la capacité de se déplacer plus rapidement que les agents.

Dans la continuité des formalismes du SCAT et du DMST, une première piste de résolution du MPE est d'utiliser le diagramme de Voronoï dans le cas où les agents et les cibles connaissent respectivement leurs positions. Le diagramme de Voronoï est exploité de deux manières différentes. Sa première utilisation est d'attribuer à chaque cible une région de Voronoï, afin de connaître la répartition des agents pour la capture des cibles. Une cible ayant peu d'agent dans sa région est ainsi plus susceptible de ne pas être capturée. Afin d'améliorer la poursuite et la capture, un système de négociation est proposé [58] afin d'équilibrer l'attribution des agents à la capture des cibles. Sa seconde utilisation est de générer une région de Voronoï pour chacun des agents et chacune des cibles au sein de l'environnement. L'objectif est ainsi de développer une stratégie visant à réduire au mieux la surface des régions de Voronoï des cibles [54, 91], générant ainsi une stratégie d'encerclement par les agents en vue de capturer les cibles.

Cependant, la capture des cibles est une problématique qui diffère de l'observation des cibles. Afin de rapprocher le formalisme du MPE à un problème d'observation, c'est-à-dire maximiser l'observation des cibles et non pas uniquement les détecter, la première approche consiste à ce qu'un agent s'associe avec une cible capturée pour maintenir continuellement sa liaison [4]. La méthode s'approche d'autant plus à la problématique de l'observation grâce à l'intégration d'un rayon d'observation aux agents pour détecter les cibles. Ces dernières possèdent un rayon de détection des agents plus large que le rayon d'observation afin de les fuir. La seconde approche est d'ignorer de la mission chaque cible venant d'être capturée, permettant ainsi aux agents de rechercher continuellement de nouvelles cibles [127].

L'état de l'art concernant la problématique de l'observation se termine avec ce dernier formalisme du MPE. L'ensemble des formalismes parcourus sont répertoriés au sein de la figure2.1, en suivant la boîte "maximisation du nombre de cibles observées". Cette thèse s'intéresse également à une seconde problématique, celle de la patrouille, dont son état de l'art est explicité à la section suivante.

2.2 État de l'art sur la problématique de la patrouille

2.2.1 Contexte et définition

La problématique de la patrouille se définit comme la visite la plus régulière possible d'un ensemble de lieu au sein d'un environnement. Au sein de cette thèse, nous nous intéressons plus particulièrement à la patrouille multi-agents, c'est-à-dire lorsque que plusieurs agents se partagent la tâche de la patrouille, incluant par conséquent des défis de coordination afin d'assurer des visites aussi fréquente que possible.

La notion de patrouille se décline en deux versions aux objectifs bien distincts. La première version, **la patrouille adversariale**, se place dans le contexte de surveiller un environnement face à l'intrusion d'adversaire. L'objectif est de maximiser la capture d'intrus par les agents. Par conséquent, communément, les lieux à surveiller se situent aux frontières de l'environnement [85]. La patrouille adversariale s'intéresse notamment aux attaques coordonnées [115], aux attaques séquentielles [116], ou encore à appliquer des stratégies issues de la théorie des jeux [3]. La seconde version, sur laquelle nous nous focalisons au sein de cette thèse, se nomme **la patrouille temporelle**. Son objectif est de minimiser l'oisiveté [90] de l'ensemble des lieux de l'environnement. L'oisiveté représente la différence de temps entre deux visites d'un même lieu par au moins un agent.

Au sein de la littérature, un lieu est communément représenté par un nœud au sein d'une représentation de l'environnement sous forme de graphe. Ainsi, pour chaque nœud, une valeur d'oisiveté y est associée. Cette valeur s'incrémente à chaque pas de temps (pour un temps discret), ou chaque seconde (pour un temps continu). L'oisiveté d'un lieu est réinitialisé à zéro lors de la visite d'un agent. L'objectif de la patrouille temporelle est de couvrir l'environnement de manière continue, c'est-à-dire visiter aussi régulièrement que possible les lieux constituant l'environnement.

Nous définissons l'**oisiveté individuelle** comme l'oisiveté d'un ou plusieurs nœuds du point de vue de l'agent, qui résulte de la visite des lieux par l'agent ainsi que du partage d'information avec d'autres agents. Cependant, l'oisiveté individuelle est à différencier de l'**oisiveté réelle**, ou oisiveté globale. Cette dernière représente l'oisiveté du point de vue du nœud sur le terrain, s'incrémentant au cours du temps et revenant à zéro par la visite de n'importe quel agent. Les métriques d'efficacité permettent d'évaluer les méthodes à l'aide de l'oisiveté réelle. Un des enjeux de la problématique de la patrouille multi-agents est par conséquent de réduire l'oisiveté réelle à l'aide d'agents ayant une représentation individuelle des oisivetés de l'environnement.

La problématique de la patrouille permet d'aborder des applications concrètes aux thématiques larges, telles que l'utilisation de drone pour surveiller la propagation des feux de forêt [129, 36, 107], pour lutter contre le braconnage [49], ou encore pour détecter des intrusions ou des agissements suspects [64, 18].

2.2.2 Les métriques d'évaluation

Plusieurs métriques sont élaborées afin d'étudier et d'évaluer les stratégies de patrouille dans leurs manières de réduire l'oisiveté. Par exemple, certaines stratégies seront efficaces pour réduire l'oisiveté maximale enregistrée au cours d'une mission, ne laissant ainsi jamais longtemps une zone sans visite, tandis que d'autres seront performantes pour faire baisser l'oisiveté moyenne de l'ensemble de l'environnement.

La problématique de la patrouille a été initialement représentée par PAMPONET MACHADO et al. [90] comme un graphe $G = (N, E)$. Avec N un ensemble de nœuds représentant les lieux à visiter, et E un ensemble d'arêtes illustrant les déplacements possibles entre les nœuds. Par la suite, CHEVALEYRE [26] suggère d'améliorer la représentation en associant le poids des arêtes à la distance entre les deux nœuds reliés. À chaque nœud $n_k \in N$ est associé une valeur d'oisiveté instantanée $i_k(t)$. De cette manière, à chaque pas de temps Δt, l'oisiveté instantanée s'incrémente de la manière suivante : $i_k(t+\Delta t) = i_k(t) + \Delta t$. Si, à l'instant t, le nœud n_k est observé par au moins un agent, alors l'oisiveté instantanée du nœud est réinitialisée, c'est-à-dire $i_k(t) = 0$. Plusieurs métriques d'évaluation sont définies grâce à la notion d'oisiveté instantanée des nœuds.

L'oisiveté instantanée du graphe L'oisiveté instantanée du graphe $i_G(t)$ correspond à la moyenne des oisivetés des nœuds à un instant t :

$$i_G(t) = \frac{1}{|N|} \sum_{n_k \in N} i_k(t) \tag{2.6}$$

Avec $|N|$ le nombre de nœuds dans l'environnement.

L'oisiveté moyenne du graphe I_G^{ag} L'oisiveté moyenne du graphe, ou oisiveté moyenne, est l'évaluation de la moyenne des oisivetés instantanées du graphe durant la période $t \in [0, T]$:

$$I_G^{ag} = \frac{1}{|N| \times T} \sum_{t \geq 0}^{T} \sum_{n_k \in N} i_k(t) = \frac{1}{T} \sum_{t \geq 0}^{T} i_G(t) \tag{2.7}$$

La pire oisiveté du graphe I_G^{max} La pire oisiveté du graphe, ou oisiveté maximale, ou encore pire oisiveté, enregistre le record de la plus forte oisiveté qu'un nœud ait pu enregistrer durant la période $t \in [0,T]$:

$$I_G^{max} = \max_{t \in [0,T]} \max_{n_k \in N} i_k(t) \tag{2.8}$$

L'interférence Dans le contexte de la patrouille, l'interférence est une métrique utilisée pour compter le nombre de fois que des agents ont été assez proches pour risquer une collision. Dans ce cas de figure, une manœuvre d'évitement est généralement opérée par les agents. Cette métrique est particulièrement utile pour éprouver les méthodes face à la mise en échelle, c'est-à-dire d'étudier les résultats de la méthode dans le cas d'un grand nombre d'agents au sein de l'environnement. En effet, une méthode distribuant mal les tâches de patrouille pousse les agents à être au même moment sur la même zone, particulièrement lorsque la densité d'agents est importante, augmentant ainsi le risque de collision. L'interférence est représentée comme une fréquence, donc exprimée en hertz (Hz), comptant le nombre de risques de collision sur une minute [40] ou durant toute la mission [99].

L'intervalle moyen ou *Mean Interval* (MI) La notion d'oisiveté est une notion intuitive, cependant l'interprétation de l'oisiveté moyenne du graphe I_G^{ag} est plus complexe à saisir. Afin de développer des métriques plus facilement interprétables, SAMPAIO, RAMALHO et TEDESCO [108] se focalisent sur la notion d'intervalle entre les visites des nœuds, ce qui revient à s'intéresser à la succession des oisivetés instantanées des nœuds. Pour ce faire, nous utilisons les notations suivantes :

- Pour chaque nœud $n_k \in N$ est associé un ensemble d'intervalles de visite J_k.
- Un ensemble d'intervalle de visite J_k est composé de tous les intervalles $j_{k,i}$, qui se définissent par les temps successifs entre les visites du nœud n_k par n'importe quel agent a_i. Nous notons également $|J_k| = card(J_k)$ le nombre d'intervalles composant l'ensemble J_k.
- Soit $N_J = \sum_{n_k \in N} |J_k|$ le nombre total d'intervalles sur l'ensemble des nœuds.

L'intervalle moyen MI se définit comme le temps moyen entre deux visites des nœuds :

$$MI = \frac{1}{N_J} \sum_{n_k \in N} \sum_{j_{k,i} \in J_k} j_{k,i} \tag{2.9}$$

Pour illustrer la différence entre l'oisiveté moyenne et l'intervalle moyen, prenons l'exemple d'un seul nœud, visité par un agent au temps t_0 et t_5, comme représenté par la figure 2.8.

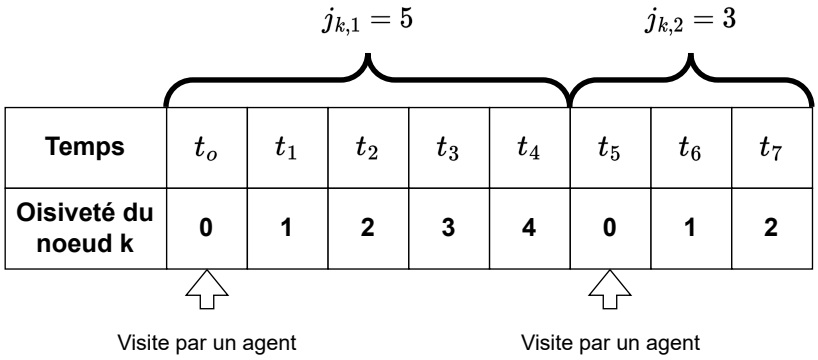


FIGURE 2.8 Exemple de scénario pour le calcul d'un intervalle moyen.

Dans ce cas de figure, nous avons $J_k = \{5,3\}$ pour un total d'intervalle $N_J = 2$. Avec $MI = (5+3)/2 = 4$ unité de temps, nous pouvons comprendre que le nœud est visité en moyenne tous les 4 temps. Par ailleurs, l'oisiveté moyenne est obtenue par le calcul suivant : $I_G^{ag} = (0+1+2+3+4+0+1+2)/8 = 1,625$ unité de temps. Par conséquent, cette métrique a une interprétation moins aisée, avec la conclusion que le nœud a une oisiveté en moyenne de $1,625$ unité de temps.

L'intervalle quadratique, ou *Mean Square Interval* (MSI) L'intervalle quadratique, noté également MSI [108], est une métrique permettant d'évaluer la distribution de la patrouille parmi les nœuds. Une méthode veillant à visiter de manière équitable les nœuds, c'est-à-dire d'avoir des intervalles de visite similaires entre les nœuds, tend par conséquent à minimiser cette métrique. L'intervalle quadratique se définit de la manière suivante :

$$MSI = \sqrt{\frac{1}{N_j} \sum_{n_k \in N} \sum_{j_{k,i} \in J_k} j_{k,i}^2} \tag{2.10}$$

2.2.3 Les méthodes pionnières de la patrouille

Cette section se focalise sur l'évolution des approches dites pionnières pour résoudre la problématique de la patrouille temporelle. Son but est d'examiner comment les méthodes ont été développées, en s'appuyant sur les travaux antérieurs, afin d'améliorer l'efficacité des agents à patrouiller l'environnement.

Afin de répondre à la problématique de la patrouille, les fondateurs du concept de l'oisiveté [90] ont proposé plusieurs méthodes multi-agents visant à minimiser l'oisiveté moyenne. Au sein de ces stratégies, les agents prennent leurs décisions en fonction de leurs oisivetés individuelles, et non pas de l'oisiveté réelle. La première méthode intuitive est nommée la stratégie réactive égoïste, ou encore ***Conscientious Reactive* (CR)**. Au sein de cette méthode, les agents choisissent de se diriger vers le nœud voisin ayant la plus grande oisiveté. Dans le cas d'un raisonnement au-delà uniquement des nœuds voisins, la méthode du ***Conscientious Cognitive* (CC)** guide les agents à se diriger vers le nœud au sein de l'ensemble du graphe ayant la plus grande oisiveté. L'efficacité des méthodes de patrouille est étudiée en comparaison de la méthode aléatoire, la ***Random Reactive* (RR)**, où les agents se déplacent au hasard parmi les nœuds environnants.

Au sein des précédentes approches évoquées, les agents prennent leurs décisions individuellement en s'appuyant sur la représentation personnelle des oisivetés du graphe. Ces croyances individuelles, reposant uniquement sur leurs chemins parcourus, peuvent s'éloigner des oisivetés réelles, ou autrement dit des oisivetés du terrain. Par exemple, un agent peut décider de visiter un nœud récemment visité par un autre agent, en ayant la croyance que ce même nœud possède une forte oisiveté. Ce comportement sous optimal peut être partiellement corrigé à l'aide d'une communication entre les agents. Une des solutions pour pallier ce phénomène est de centraliser les informations à l'aide d'un agent coordinateur. La première méthode évoquant ce concept s'appelle l'***Idleness Coordinator* (IC)**, ou encore le *Cognitive Coordinated* par ALMEIDA et al. [6]. L'agent coordinateur est dédié à la centralisation des informations concernant la visite des nœuds afin de diriger les agents vers les lieux ayant la plus forte oisiveté.

Plusieurs environnements de patrouille sont présentés au sein de la figure 2.9 [5]. Ces graphes servent de cadre de référence pour évaluer et comparer les différentes méthodes de patrouille. Chaque environnement possède sa propre spécificité, avec notamment un maillage carré pour la carte *Grid*, ou des îlots isolés pour la carte de l'*Island*.

Par ailleurs, ALMEIDA et al. [6] proposent d'incorporer la notion d'heuristique au sein des stratégies précédentes. Une méthode de patrouille dite heuristique intègre la distance entre les nœuds dans le choix des lieux à visiter par les agents. Ainsi, les agents cherchent à visiter les nœuds ayant la plus grande oisiveté, pondérés par la distance à parcourir. La distance entre deux nœuds repose sur le chemin le plus court, obtenue par exemple à l'aide de l'algorithme A* [48]. L'impact de la distance est représenté par un coefficient $r_h \in [0;1]$, sélectionné en fonction de la mission. Une valeur de r_h faible signifie qu'un agent priorisera la visite des nœuds proche de lui, et une valeur de r_h élevée poussera l'agent à aller vers

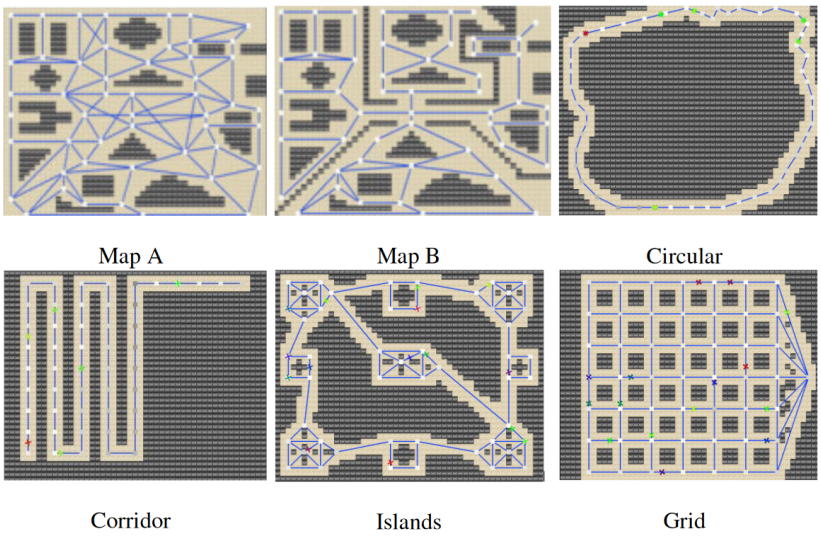


FIGURE 2.9 Environnements de références pour la patrouille [5].

des nœuds ayant une forte oisiveté, même si plus éloignés. La version heuristique de la méthode du CR se nomme ainsi l'***Heuristic Conscientious Reactive*** **(HCR)**, de même pour la méthode du CC avec l'***Heuristic Conscientious Cognitive*** **(HCC)**.

Lorsqu'un agent se dirige en direction d'un nœud désiré, il emprunte un chemin constitué d'un ensemble de nœud qu'il visitera par la même occasion. Une méthode dite navigatrice, ou *Pathfinder*, cherche à faire un compromis lors de la génération d'un chemin, qui doit être à la fois le plus court possible pour atteindre rapidement le nœud désiré, mais aussi qui tend à maximiser les oisivetés des nœuds visités sur le parcours. Ce compromis est régi par un coefficient $r_p \in [0;1]$ défini par l'expérimentateur. Si le coefficient r_p est proche de 1, alors la génération du chemin à emprunter priorisera la sélection de nœuds ayant une forte oisiveté, au détriment d'un chemin plus long. Inversement, un coefficient r_p proche de zéro cherchera à minimiser la distance à parcourir, en prenant peu en compte les oisivetés des nœuds parcourus.

La stratégie à coordinateur cognitif heuristique navigateur, ou encore l'***Heuristic Pathfinder Conscientious Cognitive*** **(HPCC)**, combine une approche à la fois navigatrice et heuristique. À l'instar de la méthode du IC, la répartition des tâches de la patrouille est assurée par un agent coordinateur. Cette méthode surpasse l'ensemble des précédentes méthodes vues précédemment pour la réduction de l'oisiveté moyenne de l'environnement. Sa version la plus performante, notée HPCC 0.4 0.2, a un coefficient heuristique $r_h = 0.4$ et un coefficient de navigation $r_p = 0.2$.

Cependant, une coordination centralisée souffre d'un manque de robustesse et suppose une communication sans faille. SAMPAIO, RAMALHO et TEDESCO [108] mettent en avant la **stratégie gravitationnelle SG**, une méthode distribuée exploitant des champs potentiels pour régir le mouvement des agents. Chaque nœud exerce une attraction sur les agents, proportionnelle à l'oisiveté du nœud et inversement proportionnelle avec la distance séparant le nœud de l'agent. De plus, au lieu de considérer chaque nœud indépendamment, le principe d'attraction permet d'être attiré par plusieurs nœuds à la fois lors du choix du prochain nœud à visiter.

Les approches précédentes sont dites *"online"*, c'est-à-dire que les agents entreprennent et adaptent leurs actions en fonction des observations réalisées. Les travaux de CHEVALEYRE, SEMPÉ et RAMALHO [27] diffèrent des approches précédentes en proposant une méthode dite *"offline"*, où la planification des chemins à suivre par les agents est effectuée en amont de la mission. Le parcours à suivre par les agents est obtenu grâce au cycle minimal du graphe, c'est-à-dire un cycle qui ne repasse pas deux fois par un même nœud (à l'exception de la fermeture) et dont sa longueur est la plus petite possible. Dans le cas où les agents suivent exactement ce même parcours avec un intervalle identique entre chaque agent, alors nous parlons de la méthode du cycle unique, ou ***Single Cycle* (SC)**. Dans le cas où les agents se repartissent ce parcours en plusieurs sous régions, alors nous parlons de la **stratégie régionalisée SR**. L'approche SC montre la meilleure performance, en comparaison avec les autres méthodes citées précédemment, pour minimiser la métrique de la pire oisiveté. La méthode du ***Cyclic Algorithm for Generic Graphs* (CGG)** est une extension du SC, permettant de généraliser la création de cycle. Cette approche est développée par PORTUGAL et ROCHA [98], en prenant inspiration des travaux sur les cycles de ELMALIACH, AGMON et KAMINKA [39]. Le CGG propose une solution dans le cas où le graphe ne présente aucun cycle unique. Ce phénomène peut se produire à cause notamment de la présence de nœud dans des impasses, c'est-à-dire lié qu'à un seul autre nœud. Dans ce cas, un "chemin long" (*longest path*) permet de générer un cycle passant plusieurs fois par un même nœud.

D'autres méthodes reposent sur des algorithmes de colonie de fourmis, ou en anglais *Ant Colony Optimization* (ACO), en prenant l'hypothèse que les agents puissent communiquer entre eux à travers l'environnement grâce au dépôt de phéromone. Les phéromones ont pour propriétés de pouvoir s'évaporer au cours du temps, mais également de se propager aux lieux environnants. Ainsi, les phéromones représentent une information qui peut être contenue au sein des noeuds d'un graphe [68, 69], mais aussi au sein de cellules dans le cadre d'un environnement discrétisé [30, 46]. Cependant, l'utilisation de phéromones implique une

interaction complexe entre les agents et leur environnement, ce qui rend sa mise en pratique difficile et en fait une hypothèse d'utilisation forte.

PORTUGAL, IOCCHI et FARINELLI [96] ont implémentés en open-source de nombreuses méthodes de patrouille telles que le RR, CR, HCR, HPCC, etc. au sein du simulateur Stage, sous le nom du package *"patrolling_sim"* [1]. Le simulateur repose sur le middleware *Robot Operating System* (ROS). Le package est compatible jusqu'à la version ROS Noetic, correspondant à la dernière et finale version de ROS1.

2.2.4 Les approches par partitionnement puis répartition de l'environnement en plusieurs régions

Les méthodes de patrouille nécessitent de coordonner les agents afin de visiter régulièrement les lieux en évitant toute collision. Pour ce faire, une des solutions est de segmenter l'environnement en plusieurs régions puis de les répartir parmi les agents pour éviter tout croisement. De plus, chaque conglomérat de nœud n'est associé et donc visité que par un seul agent, ainsi ce dernier a l'avantage d'avoir une oisiveté individuelle confondue avec l'oisiveté réelle pour ces mêmes nœuds. En effet, aucun autre agent n'est supposé patrouiller sur ce même territoire. Cette section cherche à expliciter l'évolution des approches de segmentation de l'environnement en région au sein de la littérature.

Afin de rendre l'attribution des nœuds parmi les agents dynamique, ALMEIDA et al. [5] proposent la mise en place d'un système de mise en enchère. Les nœuds sont attribués aléatoirement parmi les agents en début de mission. Durant la mission de patrouille, chaque agent cherche à améliorer la fréquence de visite des nœuds en sa possession en mettant en vente les nœuds éloignés du reste des nœuds possédés, et en cherchant à acquérir les nœuds proches. Ainsi, le système d'enchère permet de répartir les nœuds parmi les agents en rassemblant les nœuds d'une même propriété au sein d'un conglomérat, où la distance entre les nœuds tend à être minimisée. Deux architectures sont proposées pour réaliser le système d'enchère, le *Two-Shot-Bidder Agent* (TSBA) et le *Mediated Trader Bidder Agent* (MTBA). L'approche TSBA repose sur des agents égoïstes, ne cherchant qu'à augmenter l'utilité individuelle, via un système en deux tours, tandis que l'architecture MTBA repose sur un broker, faisant l'entremetteur entre les agents pour proposer les nœuds intéressants à échanger. Par la suite, les agents sélectionnent le nœud à visiter parmi leurs nœuds possédés selon une méthode propre à l'expérimentateur. Les méthodes les plus efficaces identifiées par

1. https://github.com/davidbsp/patrolling_sim

les auteurs sont à la fois heuristiques et navigatrices, avec l'***Heuristic Pathfinder Two-shots Bidder*** **(HPTB)** et l'***Heuristic Pathfinder Mediated Trader Bidder*** **(HPMB)**.

Le système d'enchère est enrichi par les travaux de MENEZES, TEDESCO et RAMALHO [84], à travers la méthode de l'Agent Enréchisseur Flexible, ou ***Flexible Bidder Agent*** **(FBA)**. Ce système gagne en efficacité en permettant aux agents d'échanger jusqu'à deux nœuds à la fois au sein d'une enchère en un seul tour. Les agents y sont égoïstes et suivent aussi une stratégie heuristique et navigatrice. La méthode FBA présente une efficacité comparable au HPCC, et légèrement inférieure au SC, dans la majorité des environnements dans le cadre d'un système fermé et d'échange synchrone. Au sein de l'environnement *Island*, la méthode FBA montre une performance bien supérieure que le HPCC, car la répartition des nœuds parmi les agents permet de pallier l'isolation des îlots. Cependant, les performances du FBA chutent dans le cas d'une communication asynchrone, où les échanges ne sont plus instantanés, mais simultanés, et où un même nœud ne peut être l'objet de plusieurs échanges simultanés. Lorsqu'un agent cherche à échanger un de ses nœuds, les autres agents ne peuvent proposer que les nœuds ne faisant pas l'objet d'un échange au même moment. Les propositions d'échange étant plus réduites qu'au sein d'une communication synchrone, la mise en enchère devient sous optimale. Cette mauvaise performance se constate par une inégalité de la répartition des nœuds entre les agents. Ainsi, certains agents possèdent une situation avantageuse avec des nœuds proches les uns des autres, et ne cherchent pas à aider les agents ayant des nœuds plus dispersés.

PORTUGAL et ROCHA [98] proposent la méthode du ***Multilevel Subgraph Patrolling*** **(MSP)** afin de segmenter l'environnement en s'inspirant de la répartition des zones parmi les agents de la méthode SR. Cependant, contrairement au SR où les agents se partagent un même parcours en plusieurs sections, la méthode du MSP divise dans un premier temps l'environnement en plusieurs régions de taille équivalente, afin de définir un parcours pour chacune d'entre elles. Ainsi, chaque agent suit un parcours unique, avec un nombre de régions identique au nombre d'agents. Si possible, le cycle du parcours suit un cycle minimal ou eulérien, c'est-à-dire qu'il visite tous les nœuds uniquement une fois en minimisant la longueur du chemin. Dans le cas où aucun cycle minimal n'existe, alors le cycle est si possible hamiltonien, validant le passage par tous les nœuds uniquement une fois sans considérer la distance à parcourir. Enfin, si le cycle ne peut pas être hamiltonien, alors il suit la logique du "chemin long" identique à l'approche du CGG.

Les méthodes du HPCC, CR, HCR, CGG et MSP sont comparées vis-à-vis de l'efficacité à réduire l'oisiveté moyenne, mais aussi l'interférence (cf. section 2.2.2) au sein des travaux de PORTUGAL et ROCHA [99]. Pour ce faire, l'étude repose sur trois environnements ayant

chacun une topologie spécifique. La figure 2.10 représente les méthodes les plus performantes pour réduire l'oisiveté moyenne, selon le nombre d'agents et la connectivité du graphe. La connectivité représente le ratio entre le nombre d'arêtes présentes au sein du graphe divisé par le nombre d'arêtes si le graphe était complet, correspondant à l'hypothèse où tous les nœuds seraient liés les uns aux autres. Dans le cas d'un nombre d'agents relativement faible (inférieur à cinq dans les expériences), la méthode HPCC est la plus performante pour une basse connectivité, car la planification de chemin fait la différence dans le cas où les connexions entre les nœuds sont réduites. Cependant, la navigation est plus aisée lorsque les nœuds sont fortement connectés entre eux, profitant ainsi aux méthodes de nature réactives (CR et HCR) qui sélectionnent les oisivetés les plus fortes dans leurs entourages. Enfin, les méthodes cycliques (CGG et MSP) sont davantage résilientes face à une forte densité d'agents en limitant fortement l'interférence entre les agents. En effet, que ce soit en suivant un unique cycle comme le CGG, ou en partitionnant l'environnement en région via des sous cycle distinct sans chevauchement pour le MSP, les agents sont moins susceptibles de se croiser et donc de réduire la performance de la patrouille. Ce constat se confirme grâce à la comparaison des interférences entre les méthodes, pour un nombre d'agents allant de 1 à 12. De gauche à droite sont présentées les méthodes ayant les pires résultats d'interférence, jusqu'aux plus performantes :

$$HCR > HPCC > CR > CGG > MSP$$

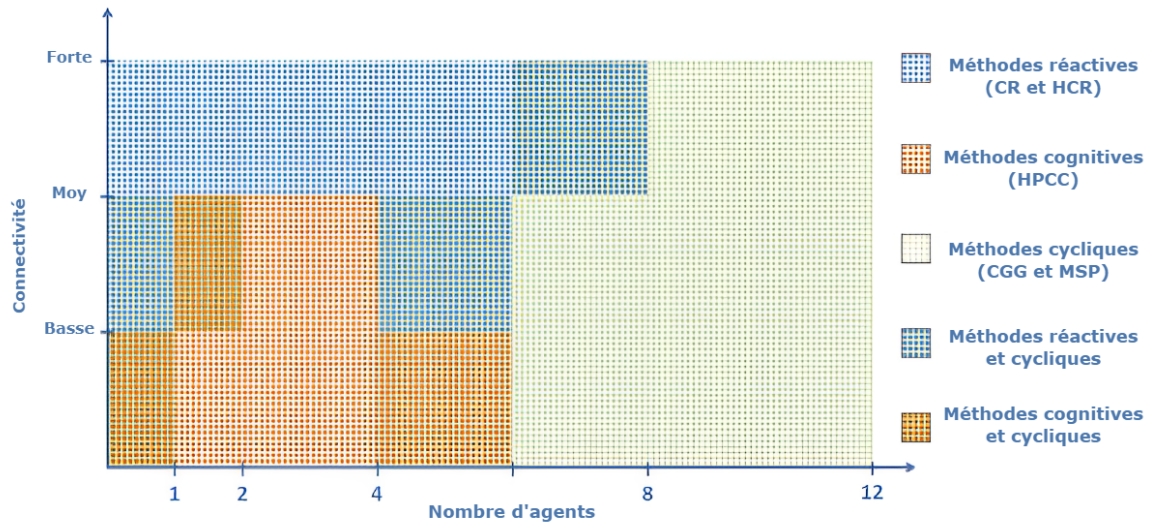


FIGURE 2.10 Identification par [99] des méthodes les plus performances pour réduire l'oisiveté moyenne du graphe selon le nombre d'agents et la connectivité du graphe.

2.2.5 L'adaptation des méthodes pionnières face à des nouveaux paradigmes environnementaux

Cette section s'intéresse aux études portant sur la robustesse et les limites des méthodes dites pionnières face à des environnements plus complexes. Les particularités étudiées au sein de la littérature sont regroupées en deux groupes : Un environnement ouvert, où les agents peuvent entrer ou sortir du système, et le cas d'un graphe dynamique, où les nœuds peuvent disparaître puis réapparaître. Nous nous intéressons ainsi aux limites des méthodes pionnières et aux solutions proposées pour les adapter face à ces deux nouveaux paradigmes.

Passage d'un système fermé à un système ouvert

Les méthodes pionnières sont élaborées avec l'hypothèse d'un système fermé, c'est-à-dire que les agents ne peuvent ni entrer ni sortir de l'environnement. POULET [101] s'est intéressé aux efficacités des méthodes RR, CR, SC, IC et HPCC dans le cadre d'un système ouvert. Afin que ces méthodes puissent être fonctionnelles dans le cadre de l'apparition ou de la disparition d'un agent, une adaptation a été mise en place. Ainsi, pour les méthodes IC et HPCC, les agents envoient un message au coordinateur pour l'informer qu'il rentre ou sort du système. De cette manière, le coordinateur connaît le nombre d'agents au sein du système et les nœuds qui n'ont pas pu être visités suite à la sortie d'un agent.

La méthode du SC repose sur le suivi par les agents du cycle minimal, dont le principe est conservé, car le graphe n'est pas modifié. Cependant, l'intervalle entre les agents au sein du cycle minimal, qui correspond à la distance que les agents doivent maintenir entre eux, doit être actualisée suite au changement du nombre d'agents. De plus, la vitesse est supposée constante et homogène parmi les agents en mouvement. Par conséquent, la modification de vitesse ne peut pas être exploitée pour repositionner les agents. Deux mécanismes sont proposés pour repositionner les agents à des intervalles réguliers :

— **SC lent** : La patrouille est stoppée lorsque l'agent coordinateur est informé d'un départ ou d'une arrivée d'un agent. L'intervalle optimal qui sépare les agents est recalculé, puis le coordinateur sélectionne un seul agent comme repère pour relancer la patrouille. Les autres agents patientent jusqu'à ce que la distance avec l'agent-repère corresponde à l'intervalle assigné, c'est-à-dire qu'ils soient positionnés correctement dans la file, pour se mouvoir de nouveau. Si le premier agent rencontré par l'agent-repère est celui qui attend le plus, alors nous parlons de la méthode du SC lent 1. À l'inverse, si c'est celui qui attend le moins longtemps, alors nous parlons de la méthode du SC lent 2.

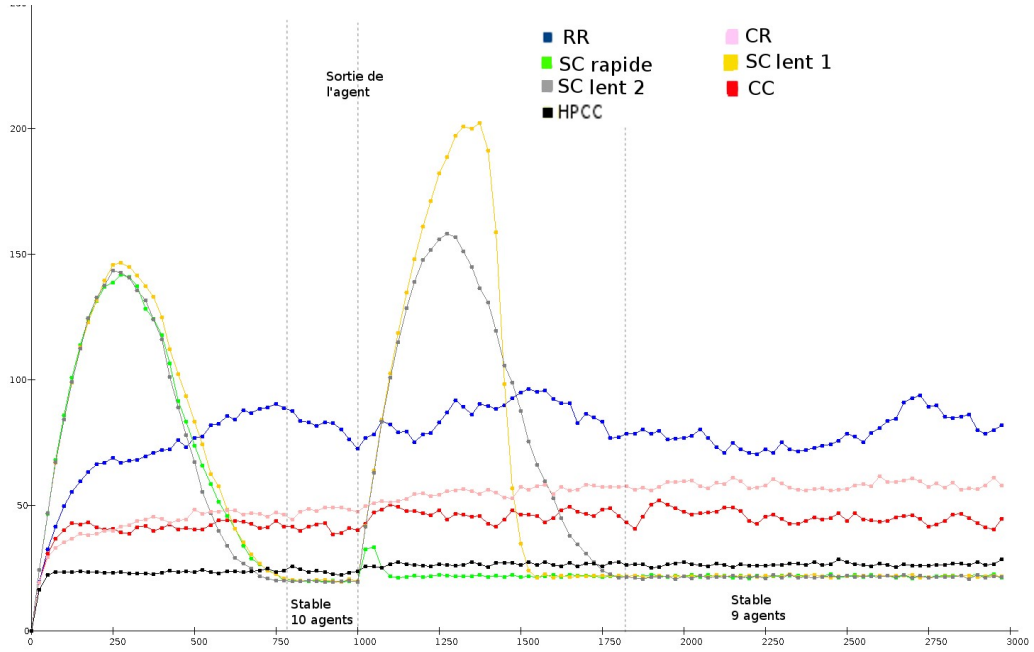


FIGURE 2.11 Évolution de l'oisiveté moyenne instantanée au cours du temps pour les méthodes RR, CR, SC, CC et HPCC au sein de l'environnement *Map A* au passage de 10 à 9 agents. Graphique provenant de [101].

— **SC rapide** : La patrouille est également stoppée par l'agent coordinateur à la réception d'un message d'entrée ou de sortie du système afin de recalculer l'intervalle optimal. Cependant, au lieu d'utiliser un agent-repère, chaque agent se déplace en fonction de l'agent devant lui dans le cycle. Ainsi, si la distance avec l'agent précédent est inférieure à l'intervalle optimal, alors l'agent attend. Sinon, si la distance est égale ou supérieure à l'intervalle optimal, alors l'agent continue de se déplacer.

Lors du départ ou de l'arrivée d'un agent, POULET [101] constate que pour toutes les méthodes étudiées, l'oisiveté moyenne instantanée évolue durant une phase de transition avant de s'équilibrer de nouveau durant une phase stable. La durée de cette phase de transition est appelée le temps de stabilisation. Par exemple, comme représenté au sein de la figure 2.11, lorsque le nombre d'agents passe de 10 à 9, la méthode SC lent au sein de l'environnement *Map A* voit son oisiveté moyenne instantanée s'accroître puis culminer autour 200*ut* avant de se stabiliser autour de 20*ut*. Cette évolution s'effectue avec un temps de stabilisation moyen de 600*ut*. L'unité *ut* correspond à une unité de temps, ou un pas de temps, dans le cadre d'un temps discrétisé.

Une étude comparative est menée sur l'amplitude de la variation de l'oisiveté moyenne instantanée pour l'ensemble des méthodes lors de la sortie d'un agent (avec initialement 5,

10, 15 et 25 agents). Ainsi, pour tous les environnements évoqués sur la figure 2.9, nous obtenons le classement suivant, avec à gauche la plus grande variation et à droite la plus faible :

$$SC_{lent} > SC_{rapide} > RR > CR \approx HPCC \approx IC$$

Ainsi que pour le temps de stabilisation, avec de gauche à droite les plus lentes aux plus rapides :

$$SC_{rapide} > CR > HPCC \approx IC > RR > SC_{lent}$$

L'objectif des travaux de POULET, CORRUBLE et EL FALLAH SEGHROUCHNI [102] est d'apporter une solution pour réduire l'impact du départ ou de l'arrivée d'un ou plusieurs agents durant la mission de patrouille. Ainsi, une méthode robuste à un système ouvert tend à minimiser à la fois la durée de la phase de transition et l'oisiveté moyenne instantanée durant la phase stable. Pour ce faire, les auteurs définissent la méthode du ***FBA Asynchronous* (FBAA)**. Son fonctionnement s'inspire directement du FBA, en l'adaptant au contexte d'un système ouvert et d'une communication asynchrone. Ainsi, les agents entrants cherchent à acquérir des nœuds, tandis que les agents sortants mettent en enchère leur possession. La perte de performance due à une communication asynchrone, résultant d'agents avantagés par la disposition des nœuds et ne cherchant pas à aider les agents désavantagés (cf. section 2.2.4), est palliée par une redistribution continue des régions grâce à l'entrée et la sortie des agents.

Deux types de règles de gestion sont étudiées et comparées pour l'attribution des nœuds du FBAA lors d'un départ ou d'une arrivée d'un agent : Les mécanismes égalitaristes veillent à ce que chaque agent donne le même nombre de nœuds selon plusieurs modèles, tandis que les mécanismes par proximité privilégient une attribution de nœud proche du lieu de départ ou d'arrivée de l'agent. L'auteur montre que les mécanismes de proximité montrent une stabilisation plus rapide ainsi que de meilleures performances dans la réduction de l'amplitude de variation que les mécanismes égalitaires.

L'introduction de fonctions d'utilité globales au sein du FBAA permet d'améliorer la qualité des enchères. Ainsi, dans le cadre d'agents coopératifs, une allocation de nœud est considérée comme bonne si elle maximise cette fonction d'utilité globale, même si elle est au détriment de l'utilité individuelle de l'agent. Au sein des travaux de POULET, CORRUBLE et EL FALLAH-SEGHROUCHNI [103], deux fonctions sont étudiées, directement issues de théorie du choix social. La première est une approche utilitaire, nommée ***minisum***, cherchant à minimiser la longueur moyenne des chemins des agents. Alors que la seconde suit une approche égalitaire, nommée ***minimax***, visant à minimiser le chemin le plus long du groupe d'agents. Les méthodes sont comparées selon plusieurs métriques. Un classement entre les

méthodes est réalisé sur le tableau 2.1. Pour chaque métrique, les méthodes sont classées de 1 à 5, avec 1 la meilleure performance et 5 la pire. À titre d'illustration, la méthode *minisum* montre la meilleure performance au regard de l'amplitude de variation, en réussissant à le minimiser plus efficacement que les autres méthodes comparées.

TABLE 2.1 Classement des performances de plusieurs méthodes de patrouille dans le contexte d'un environnement ouvert. Chaque méthode est classée de 1 à 5, avec 1 la meilleure performance et 5 la pire.

	SC	minisum	minimax	FBA	HPCC
Oisiveté moyenne	5	4	3	2	1
Pire oisiveté	X	2	1	3	4
Phase de transition	5	4	3	2	1
Amplitude de variation	5	1	4	2	3

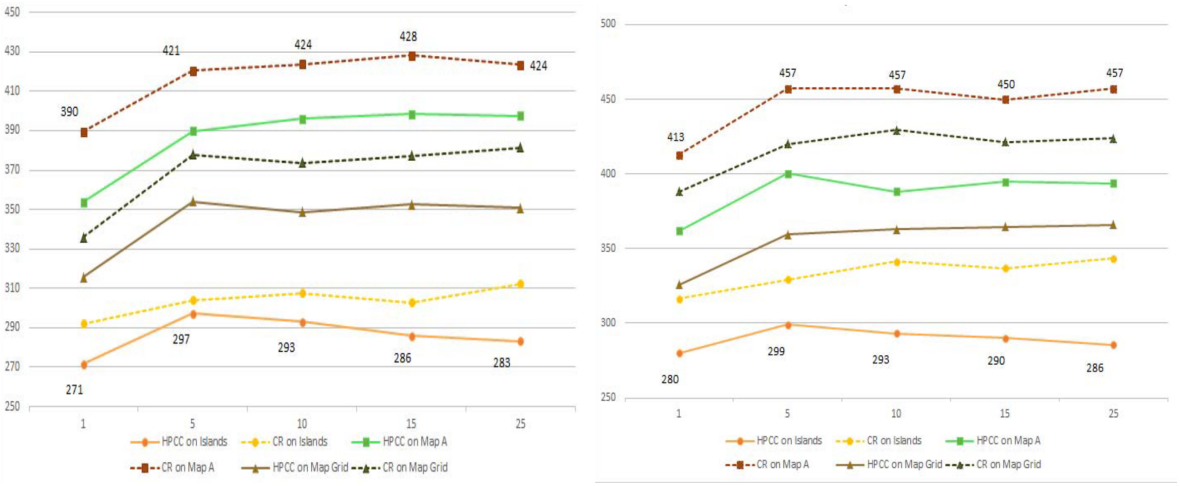
La méthode *minimax* obtient, en système ouvert, les meilleures performances au regard de l'oisiveté maximale et des performances comparables au FBA concernant l'oisiveté moyenne. Les performances du *minisum* sont relativement moins bonnes que la méthode du *minimax*, à l'exception de la réduction de l'amplitude de variation. D'après les auteurs, la méthode *minimax* est par conséquent la méthode privilégiée pour effectuer une patrouille multi-agents dans le cas d'un système ouvert. Cette dernière s'appuie sur une coordination décentralisée et une communication asynchrone.

Graphe dynamique

Un graphe dynamique se caractérise par l'apparition et la suppression au cours du temps de nœuds ou bien d'arêtes. Ce type de graphe est utile pour représenter l'évolution de l'environnement pour des scénarios avec des chemins rendus par exemple impraticables, ou dans le cas de lieux devenant dangereux à visiter. Ainsi, un graphe dynamique peut être utilisé lors de l'apparition de phénomène météorologique sur certaines zones, empêchant des drones de voler [44], ou encore de générer un chemin en cas d'incendie, dont certains accès peuvent être compromis au cours du temps [29].

Le comportement des méthodes HPCC et CR a été étudié dans le cadre de graphes statiques et dynamiques au sein des recherches de OTHMANI-GUIBOURG et al. [88]. Les environnements de référence utilisés, ici *Map A*, *Grid* et *Islands*, ont été modifiés pour inclure une probabilité de modification des arêtes, c'est-à-dire une probabilité d'apparition si l'arrête est absente, ou de disparition si celle-ci existe. Au sein des expériences, la probabilité de modification est entre 10 et 15% à chaque pas de temps. Par ailleurs, la méthode CR est

adaptée au graphe dynamique en considérant, dans le choix du nœud à visiter, uniquement les nœuds adjacents dont les arêtes permettent la liaison. Tandis que pour la méthode du HPCC, la planification prend l'hypothèse d'un graphe statique. Lorsque l'agent s'aperçoit qu'une des arêtes sur le chemin initial a été supprimé, alors le coordinateur central effectue une nouvelle planification.



(a) Scénario avec graphes statiques.

(b) Scénario avec graphes dynamiques.

FIGURE 2.12 Intervalle moyen MI en fonction du nombre d'agents, pour les méthodes HPCC et CR au sein de trois environnements. Illustration issue de [88].

La figure 2.12 représente les intervalles moyens des méthodes HPCC et CR pour les trois environnements de référence. Ces environnements sont représentés avec un graphe statique pour le graphique (a), et un graphe dynamique pour le graphique (b). Nous pouvons constater que le HPCC, de par nature centralisée et utilisant un coordinateur, est davantage résiliant face à la modification du graphe. Ce constat est également présent vis-à-vis de la métrique de la distribution de la patrouille entre les nœuds via l'écart-type des oisivetés. Cependant, HPCC nécessite une communication continue entre les agents, ce qui constitue une hypothèse forte. Tandis que la méthode CR, par sa nature décentralisée et ne résonnant que sur les nœuds alentours, perd grandement en efficacité face à un graphe dynamique.

2.2.6 Les méthodes décentralisées RLPM et RAMPAGER, visant à reproduire les performances et le comportement de la méthode centralisée HPCC

La méthode du HPCC est particulièrement efficace pour réaliser une mission de patrouille, car comme nous l'avons explicité au sein de la section 2.2.3, elle est issue de la combinaison

d'une approche à la fois heuristique, navigatrice, et centralisée. Cependant, l'utilisation d'un agent coordinateur, concentrant toutes les informations, nécessite une communication parfaite et continue de la part de l'ensemble des agents. De plus, une coordination centralisée nécessite que l'agent coordinateur dispose de ressources de calcul suffisant pour réaliser une mise à l'échelle, c'est-à-dire de garder sa performance malgré un grand nombre d'agents au sein de l'environnement. Ces contraintes rendent la méthode du HPCC vulnérable en cas de perte de la communication avec un agent, d'un délai important dans la transmission des informations, ou encore d'un temps de calcul trop lent pour la répartition des tâches par l'agent coordinateur. Dans l'objectif de pallier les contraintes issues d'une coordination centralisée, tout en conservant au mieux son efficacité, OTHMANI-GUIBOURG, EL FALLAH SEGHROUCHNI et FARGES [89] proposent deux méthodes décentralisées issues de deux approches distinctes.

La première méthode provient d'un préapprentissage empirique et se nomme la méthode du ***Random-Next-Neighbour-LSTM-Path-Maker*** **(RLPM)**. Son approche est de décentraliser la méthode du HPCC, et plus précisément sa version HPCC 0.2 0.2, en deux étapes. La première étape consiste à entraîner des agents, décentralisés et autonomes, à reproduire le comportement de la méthode du HPCC sans avoir recours à un agent coordinateur, à l'aide d'un apprentissage supervisé. Les agents apprennent individuellement à imiter la politique du HPCC en associant les observations et les actions entreprises par chaque agent issue de la méthode centralisée. Un seul modèle est entraîné, identique pour tous les agents, et repose sur un réseau *Long Short-Term Memory* (LSTM) [51]. L'avantage d'un réseau récurrent, tel que le réseau LSTM, est son efficacité à apprendre une séquence temporelle. Dans ce scénario précis, l'agent apprend à entreprendre une action en fonction de ses anciens nœuds visités. La sortie du réseau génère une distribution de probabilité, à l'aide d'une fonction softmax, pour visiter le prochain nœud voisin. Cependant, au lieu de choisir le nœud ayant la plus haute probabilité, les auteurs constatent une meilleure efficacité du modèle lorsque le nœud est sélectionné par un tirage aléatoire dans la distribution issue du LSTM. L'apprentissage a été effectué sur plusieurs structures du réseau LSTM, en faisant varier le nombre de neurones et le nombre de couches cachées. Les deux structures les plus performantes retenues possèdent 50 neurones, avec une et deux couches cachées.

L'efficacité de la méthode du RLPM est comparée avec celle du HPCC 0.2 0.2, méthode de référence pour l'entraînement, et du CR (cf. section 2.2.3). Les environnements de référence utilisés pour la comparaison sont : *Map A*, *Grid* et *Island*. Enfin, le nombre d'agents pour les expériences est compris entre 5 et 25, avec un pas de 5. Grâce à sa nature centralisée, la méthode du HPCC surpasse les deux autres en termes d'oisiveté moyenne et de la pire

oisiveté sur l'ensemble des environnements. La méthode du RLPM est plus efficace pour réduire l'oisiveté moyenne que la méthode du CR, sur les environnements *Map A* et *Grid* lorsque le nombre d'agents est inférieur à 20. Cependant, dans le contexte d'îlot isolé de l'environnement *Island*, la méthode du RLPM ne réduit pas mieux l'oisiveté moyenne que la méthode du CR. De plus, la méthode du CR est plus performante pour minimiser la pire oisiveté, sur tous les environnements et la variation du nombre d'agents étudiés.

La seconde méthode, nommée ***RAndom Multiagent PAtrollinG LSTM-Path-MakER* (RAMPAGER)**, provient d'un préapprentissage analytique. Ce qui signifie qu'à la différence de la méthode RLPM, le modèle LSTM est alimenté en entrée par un vecteur représentant l'oisiveté instantanée individuelle de chaque nœud composant le graphe de l'environnement. De plus, une étude approfondie des résultats du modèle HPCC montre que la version HPCC 0.5 0.5 est la plus efficace dans la réduction de l'oisiveté moyenne. C'est par conséquent cette version qui est utilisée comme référence lors de l'apprentissage. Au même titre que la méthode RLPM, plusieurs structures LSTM ont été étudiées sur l'environnement *Map A* au regard de l'oisiveté moyenne et de la pire oisiveté. La structure la plus efficace sur ces deux métriques possède deux couches cachées de 50 neurones chacune, nommée ainsi RAMPAGER 2-50.

L'efficacité de RAMPAGER 2-50 est comparée avec les méthodes HPCC 0.5 0.5, CR et RLPM. Les environnements et le nombre d'agents évalués sont les mêmes que pour l'évaluation du RLPM. La méthode centralisée HPCC montre la meilleure efficacité vis-à-vis des deux métriques d'évaluation, la pire oisiveté et de l'oisiveté moyenne, sur l'ensemble des environnements. Concernant l'oisiveté moyenne, RAMPAGER 2-50 est plus performante que les méthodes de CR et RLPM pour les trois environnements, peu importe le nombre d'agents. De plus, RAMPAGER 2-50 montre une efficacité comparable à la méthode CR pour réduire la pire oisiveté sur les cartes *Map A, Grid*, et légèrement supérieure sur la carte *Islands*.

2.2.7 Les approches pour réduire l'écart entre les oisivetés individuelles et l'oisiveté réelle

Une méthode de patrouille gagne en efficacité lorsque la différence entre les oisivetés individuelles des agents et les oisivetés globales de l'environnement est minimisée. En effet, les agents prennent ainsi la décision de visiter un lieu en se reposant sur des informations d'oisiveté proche des valeurs réelles du terrain. Les approches régionalisées (cf. section 2.2.4) permettent d'éliminer l'écart d'oisiveté, car chaque région est attribuée à un unique agent.

Cependant, cette solution souffre d'un manque de robustesse, tout particulièrement en cas de perte, par exemple, d'un agent. De même, une coordination centralisée, où toutes les informations sont concentrées par un agent coordinateur, élimine également les écarts d'oisiveté. Cependant, le problème se pose dans le cadre d'une coordination décentralisée ou distribuée. Dans ce cas, une manière de réduire l'écart d'oisiveté est la communication entre les agents. Au sein de cette section, nous nous intéressons aux travaux cherchent à pallier les écarts d'oisiveté, que ce soit entre les oisivetés individuelles des agents, ou entre les oisivetés individuelles d'un agent et les oisivetés globales de l'environnement.

Pour réduire l'écart d'oisiveté, les travaux de YAN et ZHANG [130] proposent que les agents communiquent entre eux leurs oisivetés individuelles, le lieu qu'ils ont l'intention de visiter, ainsi que le temps estimé pour l'atteindre. Au sein de la méthode distribuée ***Expected Reactive* (ER)**, les agents choisissent de visiter le nœud minimisant l'oisiveté estimée, parmi les nœuds voisins accessibles. L'oisiveté estimée se définit, du point de vue d'un agent, comme la différence entre le temps d'arrivée sur le lieu, prenant en compte le temps de trajet, et le temps de la dernière visite du lieu. La méthode prend également en considération le cas où un autre agent a exprimé l'intention de visiter un lieu, où la dernière visite du lieu est égale au temps d'arrivée de cet agent. La méthode ER montre globalement une meilleure efficacité, pour un nombre d'agents allant de 1 à 12, vis-à-vis de l'oisiveté moyenne et maximale que les méthodes CR, SEBS et une méthode de partitionnement centralisée CP (*Centralized Partition*) [59]. Les environnements utilisés reposent sur la forme "Grid", et une topographie représentant plusieurs salles de bureau nommé "Cumberland".

OTHMANI-GUIBOURG, EL FALLAH-SEGHROUCHNI et FARGES [87] proposent une méthode pour estimer l'oisiveté globale en fonction de l'oisiveté individuelle, sans passer par la communication entre les agents. Pour ce faire, chaque agent embarque un modèle de prédiction identique, prenant en entrée les oisivetés individuelles de chaque nœud, pour fournir en sortie l'estimation de leurs oisivetés globales respectives. Sur un même environnement, le modèle est entraîné par apprentissage supervisé. Les données proviennent d'un grand nombre de simulations de la méthode HCC-0.2, où à chaque pas de temps, les agents fournissent leurs oisivetés individuelles pour alimenter l'entrée du modèle, et l'environnement fournit l'oisiveté globale pour faire la correspondance avec la sortie du modèle.

Le premier type de modèle, nommé *Mean*, estime que l'oisiveté globale de chaque nœud correspond à la moyenne de toutes les oisivetés globales enregistrées au cours du temps et des simulations sur ce même nœud. Le second type de modèle sont des réseaux de neurones artificiels à entrainer, exploitant les fonctions d'activation *Linear* et *Relu*. Les méthodes reposant sur ces modèles prédictifs ont pour désignation ***Heuristic Pathfinder X Predicator* (HPXE)**,

avec X le nom du modèle, c'est-à-dire *Mean*, *Linear* ou *Relu*. Dans le cas où une stratégie est stochastique, et non déterministe, alors le nom de la méthode est précédé du terme *Random*. Par exemple, la méthode RHPME est la version stochastique exploitant le modèle *Mean*. L'évaluation des méthodes prédictives est réalisée sur les environnements *Map A*, *Islands* et *Grid*, avec 5, 10, 15 et 25 agents. La méthode RHPLE montre les meilleures performances vis-à-vis de l'intervalle moyen MI et de l'intervalle quadratique MSI, surpassant également la méthode CR sans atteindre l'efficacité de la méthode centralisée HCC-0.2.

2.2.8 Le formalisme avec une fréquence de visite des lieux hétérogène : CCPP

Définition du CCPP

Le formalisme du ***Continuous Cooperative Patrolling Problem*** **(CCPP)** [132] conceptualise un environnement avec des priorités hétérogènes parmi les nœuds, c'est-à-dire que certains lieux doivent être visités plus fréquemment que d'autres. La nécessité de concevoir différentes priorités de visite vient initialement des travaux de KATO et SUGAWARA [61], où l'objectif est de développer une méthode multi-agents pour effectuer des tâches de nettoyage, avec la particularité que certains endroits génèrent plus fréquemment des déchets que d'autres. Par la suite, le formalisme généralise le problème en considérant la création de déchet comme l'apparition d'un évènement. Le CCPP est représenté par un graphe, dont la pondération entre les nœuds correspond à la distance les séparant. L'environnement, initialement continue en deux dimensions, se décompose en un ensemble de nœud. Les agents ne peuvent se déplacer que de nœud en nœud à chaque pas de temps.

Adaptation des métriques de la patrouille

À la différence du formalisme de la patrouille temporelle originel, chaque nœud v ne possède pas une valeur d'oisiveté, mais une quantité d'évènement $L_t(v)$ au temps t :

$$L_t(v) = \begin{cases} L_{t-1}(v) + 1 & \text{si un évènement apparaît} \\ L_{t-1}(v) & \text{sinon} \end{cases}$$

La fréquence d'apparition d'un évènement est établie en amont au sein de l'environnement. Les figures 2.13 et 2.14 représentent les environnements de référence pour comparer les méthodes de patrouille au sein du formalisme du CCPP. Ainsi, les zones rouges ont une probabilité d'apparition d'évènement $P_v = 10^{-3}$, c'est-à-dire qu'à chaque pas de temps, un

évènement a une chance d'apparition de 0.1 % sur le nœud v. Les zones grises ont une probabilité de $P_v = 10^{-4}$ et les zones blanches de $P_v = 10^{-6}$.

Les métriques d'évaluation de la patrouille sont adaptées au formalisme du CCPP. L'équivalent de l'oisiveté moyenne est la moyenne cumulée des évènements non traités, notée D, durant la période d'évaluation $[t_s;t_e]$. La métrique D est définie par :

$$D_{t_s,t_e}(V) = \frac{1}{t_e - t_s} \sum_{v \in V} \sum_{t=t_s+1}^{t_e} L_t(v)$$

Avec V l'ensemble des nœuds de l'environnement. De la même manière, l'homologue de la pire oisiveté au cours d'une mission correspond au record U d'évènement cumulé sur un nœud durant la période d'évaluation :

$$U_{t_s,t_e} = \max_{v \in V, t_s \leq t \leq t_e} L_t(v)$$

Les travaux de WU, SUGIYAMA et SUGAWARA [128] s'intéressent également à minimiser la consommation d'énergie par les agents, et y définissent la consommation totale d'énergie C par :

$$C_{t_s,t_e} = \sum_{i \in A} \sum_{t=t_s+1}^{t_e} E_t(i)$$

Avec i un agent parmi l'ensemble des agents A. Et $E_t(i)$ la consommation d'énergie par l'agent i au temps t.

Les méthodes de résolution

Les quatre premiers environnements de référence pour la résolution du CCPP sont définis au sein des travaux de YONEDA et al. [133] (figure 2.13). Les auteurs y définissent plusieurs méthodes de résolution, qui sont composées de deux phases : le processus de sélection du nœud, puis le processus de navigation afin d'atteindre le nœud sélectionné.

La première approche de sélection proposée est le ***Random Selection*** **(RaS)**, qui consiste à choisir aléatoirement un nœud au sein de l'environnement. Cette approche est comparable à la méthode du RR. La méthode du ***Probabilistic Greedy Selection*** **(PGS)** repose sur la notion d'espérance d'évènement cumulée, s'exprimant pour un nœud par l'expression : $EL_t(v) = P_v(t - t^v_{visit})$, avec t^v_{visit} la dernière visite connue du nœud. L'agent liste les N nœuds

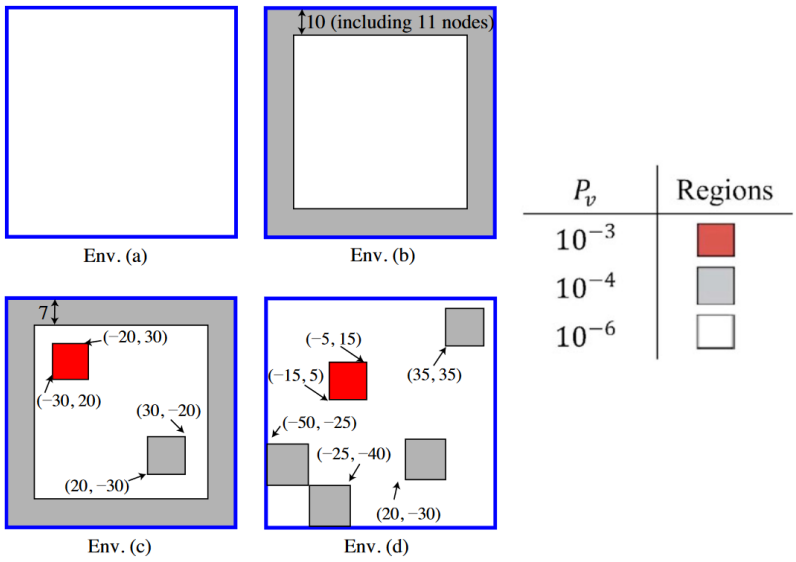


FIGURE 2.13 Quatre premiers environnements de référence pour le CCPP [133].

qui maximisent la valeur d'espérance EL_t puis décide de visiter aléatoirement un de ces nœuds. Le caractère aléatoire permet d'éviter que tous les agents aillent continuellement sur un même nœud. La méthode du ***Balanced neighbor-preferential selection*** **(BNPS)** inclut une notion heuristique, en privilégiant la visite des nœuds à proximité de l'agent ayant une espérance EL_t dépassant un certain seuil, avant de visiter les nœuds plus éloignés. Enfin, la méthode du ***Repulsive Selection*** **(RS)** consiste pour les agents de sélectionner aléatoirement N nœuds puis de se diriger vers celui étant le plus éloigné de tous les autres agents de l'environnement. Deux processus de navigation sont également proposés : le chemin le plus court et le ***Gradual Path Generation*** **(GPG)**. Ce dernier inclut une modification du chemin le plus court, pour visiter également les nœuds adjacents ayant une espérance d'évènement cumulée $EL_t(v)$ élevée, c'est-à-dire dépassant un seuil préétabli. L'approche du GPG est comparable aux méthodes navigatrices, cependant la différence majeure réside dans l'ordre des étapes : le GPG modifie le chemin une fois que le nœud à visiter est sélectionné, tandis que les autres approches navigatrices incorporent dans le choix du nœud à visiter l'oisiveté cumulée de chacun des chemins menant aux nœuds.

Les expériences montrent que la navigation par GPG est plus efficace pour réduire la métrique $D(V)$ que la navigation par le chemin le plus court, et ce quelle que soit la stratégie de sélection du nœud à visiter.

Les précédentes méthodes formelles, c'est-à-dire reposant sur des formules mathématiques explicites, sont comparées avec des approches issues de l'apprentissage. La première méthode d'apprentissage proposée est l'***Adaptative Meta Target Decision Strategy***

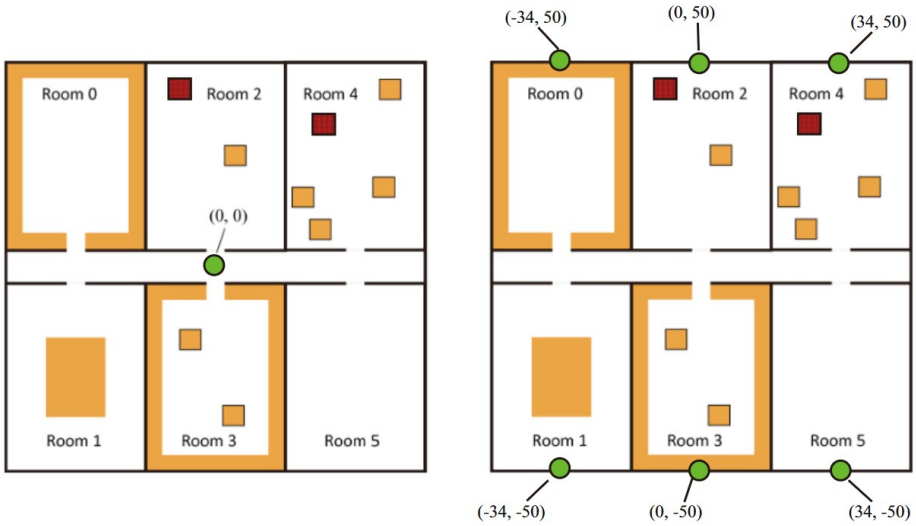


FIGURE 2.14 Deux scénarios pour l'environnement "Bureau", incluant des bornes de recharge pour les agents en vert [118].

(AMTDS), reposant sur de l'apprentissage par renforcement. L'algorithme d'apprentissage, ici le Q-learning, exploite une approche ε-glouton. La récompense est calculée comme le rapport entre la quantité d'évènement traitée L_t et la distance parcourue pour atteindre le nœud sélectionné, le long du chemin généré par la navigation GPG. Cependant, une situation problématique du AMTDS est la sur-sélection récurrente des mêmes nœuds au sein de sa stratégie, faisant décroître ainsi son efficacité. Pour pallier ce problème, l'approche ***AMTDS with relearning by self-monitoring*** **(AMTDS/RM)** propose de réinitialiser la *Q table* lorsque le phénomène de saturation se présente. Néanmoins, les précédentes approches prennent l'hypothèse forte de connaître les probabilités P_v en amont de la mission. Pour y remédier, la méthode du ***AMTDS/RM and learning of dirt accumulation probability*** **(AMTDS/RMLD)** propose d'inclure l'apprentissage également des probabilités d'apparition d'évènement au sein du modèle. Les auteurs SUGIYAMA, SEA et SUGAWARA [117] proposent d'incorporer un système de négociation au sein de la méthode AMTDS/RMLD pour gagner en efficacité dans la répartition des tâches parmi les agents. La méthode ***AMTDS with learning of event probabilities and enhancing divisional cooperation*** **(AMTDS/EDC)** propose que les agents puissent s'échanger des informations à travers un rayon de communication. Les communications sont supposées sans autre contrainte, c'est-à-dire instantanée, sans limitation de bande passante, etc. L'environnement "Bureau" est créé, avec plusieurs pièces séparées (cf. figure 2.14) afin de rendre la répartition des tâches d'autant plus nécessaire.

Parmi la multitude des variants du AMTDS, une comparaison exhaustive des efficacités de chacune d'elle est proposée au sein des travaux [133, 117]. Ainsi, au sein de l'environ-

nement (d), caractérisé par des zones isolées avec des fréquences disparates, le classement des méthodes vis-à-vis de la métrique D est le suivant (du moins efficace au plus performant) : $AMTDS > AMTDS/RM \approx AMTDS/RMLD$. Cependant, pour l'environnement (c), moins complexe avec peu d'îlot et des zones relativement proches, alors nous obtenons le classement : $AMTDS/RM > AMTDS/RMLD \approx AMTDS$. Enfin, dans l'environnement "Bureau", nous obtenons une meilleure efficacité moyenne de l'AMTDS/EDC de 26.7 % en comparaison avec la méthode AMTDS/RMLD.

Dans la continuité de l'approche AMTDS, WU, SUGIYAMA et SUGAWARA [128] proposent la méthode de l'***AMTDS for energy saving and cleanliness*** **(AMTDS/ESC)**. L'objectif est d'optimiser l'énergie consommée par les agents lors des déplacements, et donc de minimiser à la fois la métrique D et la métrique C. Pour ce faire, la récompense est modifiée en reprenant la formule du AMTDS originel, mais divisée par la consommation de l'agent pour atteindre le nœud désiré. Les agents montrent également un comportement faisant plus attention à l'autonomie de la batterie, en retournant se charger à la base plus régulièrement et en prenant également des pauses au sein du parcours. La méthode du AMTDS/ESC montre une efficacité comparable au AMTDS pour la métrique D, tout en préservant entre 20 et 50 % de l'énergie consommée par rapport aux méthodes RaS, PGS, RaS et BNPS.

Une comparaison entre les méthodes a permis d'obtenir leurs classements [133] au regard de la minimisation de la métrique D. Ce classement est obtenu grâce à la moyenne des performances sur les environnements de la figure 2.13, et ce, pour 10, 15 et 20 agents. Sur la gauche sont présentés les méthodes avec la plus grande quantité d'évènement non traité D en moyenne, et à droite les méthodes les plus efficaces minimisant cette même quantité :

$$RaS > RS \approx PGS > BNPS > AMTDS$$

Une comparaison équivalente est menée pour la métrique de la consommation d'énergie C. Nous nous focalisons sur l'environnement (d), qui est l'environnement le plus complexe à résoudre. De la même manière, les méthodes sur la gauche ont la plus grande consommation d'énergie, et sur la droite sont celles optimisant au mieux cette consommation :

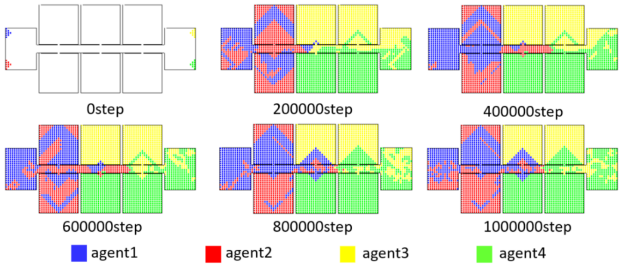
$$PGS > RaS > RS > BNPS > AMTDS/ESC$$

Il est intéressant de souligner que ces efficacités sont obtenues durant des durées de mission relativement longues (10^5 pas de temps). Tandis que les méthodes formelles ont un temps de stabilisation de leurs efficacités très faible, les approches par renforcement

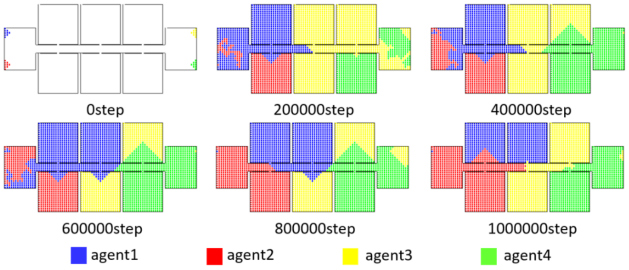
nécessitent naturellement un temps d'apprentissage bien plus long (en moyenne 10^5 pas de temps), rendant la méthode très peu efficace sur le court terme.

À l'instar des méthodes évoquées au sein de la section 2.2.4, une approche pour répondre au formalisme du CCPP est de régionaliser l'environnement et d'attribuer à chaque agent une région à patrouiller. La solution proposée par les travaux de SEA, SUGIYAMA et SUGAWARA [112] est d'attribuer à chaque nœud un poids correspondant à sa priorité de visite. Puis de rassembler les nœuds en région dont les poids cumulés doivent être équivalents. Pour réaliser ce regroupement, les auteurs ont adapté l'algorithme de k-moyennes (*k-means* [76]). Ainsi, au sein de la fonction objective redéfinie, chaque distance entre les centroïdes et les nœuds est multipliée par le poids du nœud en question. De plus, les auteurs ont remarqué que le regroupement est d'autant plus efficace en initialisant la position des centroïdes sur les nœuds ayant les priorités les plus fortes, contrairement à l'initialisation aléatoire. Chaque agent patrouille au sein de sa région attribuée, son chemin de patrouille est calculé en amont de la mission. L'algorithme *Simulated Annealing* (SA) permet d'obtenir le chemin le plus court où l'agent peut passer m_i fois par nœuds, avec m_i la priorité du nœud i. Malheureusement, l'approche n'a pas été évaluée sur les environnements de référence, mais sur un environnement personnalisé, ce qui ne permet pas de faire des comparaisons avec les autres méthodes de résolution.

Cependant, l'approche de regroupement par l'algorithme de *k-means* nécessite une coordination centralisée. Afin de proposer une solution plus robuste au CCPP, HATTORI et SUGAWARA [50] développent un système de négociation décentralisée pour la répartition de l'environnement. Au sein d'une coordination décentralisée, l'absence d'un agent coordinateur responsable de la bonne répartition des régions peut mener à la création de région enclavée, ce qui a pour impact de diminuer l'efficacité de la patrouille, comme cela a pu être constaté au sein des travaux de KATO et SUGAWARA [61] pour l'environnement "Bureau". Pour répondre à cette problématique d'enclavement, l'approche proposée est d'inclure un système de négociation entre les agents. Au sein de la négociation, les agents cherchent à s'échanger des nœuds attribués, mais éloignés de leurs régions, tout en cherchant à acquérir des nœuds proches. La figure 2.15 représente la répartition de l'environnement "Bureau" selon la méthode de référence [61] et la méthode de négociation [50]. Les deux méthodes ont une efficacité identique dans la minimisation de la métrique D. Le gain est essentiellement dans la réduction significative du nombre d'enclaves au cours du temps.



(a) Répartition selon la méthode de KATO et SUGAWARA [61].



(b) Répartition selon la méthode de HATTORI et SUGAWARA [50].

FIGURE 2.15 Différences de répartition de l'environnement entre les deux méthodes évaluées. Images issues de [50].

2.2.9 Améliorer la coordination grâce à l'apprentissage par renforcement

L'apprentissage par renforcement est une branche de l'intelligence artificielle qui se concentre sur l'apprentissage d'agents autonomes à prendre des décisions optimales dans des environnements complexes. Cela implique un processus itératif où l'agent interagit avec l'environnement, observe les états, effectue des actions et reçoit des récompenses ou des pénalités en fonction de ses actions. L'objectif de l'apprentissage par renforcement est d'apprendre à l'agent à maximiser les récompenses cumulées au fil du temps en ajustant ses actions en fonction des récompenses obtenues. Dans le cas où le contexte d'étude est un système multi-agents, alors le sous-domaine associé se nomme l'apprentissage par renforcement multi-agents, ou *Multi-Agent Reinforcement Learning* (MARL).

Les premières recherches sur l'utilisation du MARL dans le contexte de la patrouille ont été initiées par les travaux de SANTANA et al. [110], à travers la création de deux méthodes utilisant une représentation markovienne [16] : le ***Black-Box Learner Agent*** **(BBLA)** et ***Gray-Box Learner Agent*** **(GBLA)**. Ces méthodes reposent sur l'apprentissage *Q-Learning*, où les agents cumulent des récompenses en fonction de l'oisiveté réelle perçue durant la visite

d'un nœud. Pour déterminer la valeur d'oisiveté réelle, les agents placent, lors de la visite d'un nœud, un drapeau sur lequel est inscrit le dernier temps de visite. Au sein de la méthode GBLA, les agents communiquent également sur le drapeau l'intention de visite du prochain nœud voisin. Expérimentalement, la méthode du GBLA montre une réduction de l'oisiveté moyenne plus efficace que les méthodes BBLA, CR et HPCC. La méthode ***Extended-GBLA***, proposée par LAURI et KOUKAM [70], améliore les performances et la coordination entre les agents de la méthode du GBLA. Pour ce faire, la récompense est redéfinie sous la forme d'une punition, attribuée à tous les agents, selon la pire oisiveté enregistrée au cours d'un épisode. Cependant, l'utilisation de drapeau des méthodes BBLA, GBLA et *Extended-GBLA* est une contrainte forte et peu réaliste de la part d'agent réalisant une mission de patrouille.

Les travaux de JANA, VACHHANI et SINHA [56] s'inscrivent également dans le contexte d'une représentation markovienne. Cependant, contrairement aux approches précédemment citées, les agents n'obtiennent pas l'oisiveté des nœuds grâce à une interaction physique avec l'environnement. Plusieurs scénarios d'interactions entre les agents sont étudiés, allant d'agent individualiste ne partageant aucune information d'oisiveté, à des agents communicants uniquement leurs positions, en passant par des agents qui ont la liberté de communiquer avec tous les autres agents, échangeant ainsi des informations sur leurs oisivetés et positions respectives. L'apprentissage est indépendant à chaque agent et repose sur du *Deep Q-Learning*. Le modèle d'apprentissage utilisé consiste en un réseau de neurones à deux couches cachées, comprenant respectivement 128 et 84 neurones, avec une fonction d'activation de type *ReLu*. Le vecteur d'observation en entrée dépend du scénario envisagé, tandis qu'en sortie, les agents ont la possibilité de choisir entre aller au nœud en face, derrière, à gauche ou à droite d'eux. Afin d'assurer une collaboration entre les agents, après la visite d'un nœud, la récompense obtenue est égale au rapport entre l'oisiveté instantanée du nœud visité et l'oisiveté moyenne instantanée de tous les nœuds du graphe. Les résultats montrent que cette formulation de la récompense est plus efficace que celles utilisées précédemment par le GBLA [110] et l'*Extended-GBLA* [70].

Diverses méthodes s'appuient sur une représentation appelée semi-markovienne, ou Generalised Semi-Markov Decision Process (GSPDM) [81]. Dans cette représentation, tous les agents partagent les mêmes informations pour prendre leurs décisions. Ces informations comprennent la position de tous les agents ainsi que la valeur d'oisiveté de chaque nœud. La prise de décision repose également sur la notion de "fraîcheur", une valeur bornée entre -1 et 1, calculée par b^{i_k}, avec $0 < b < 1$ et i_k l'oisiveté du nœud n_k. Les auteurs proposent de résoudre ce problème via une approche algorithme en ligne, nommée *Anytime Error Minimization Search*, qui réalise une recherche heuristique parmi les nœuds voisins pour optimiser la

récompense liée à la fraîcheur des nœuds. Dans le cas où les agents communiquent entre eux, la méthode est appelée le ***Coordinated Anytime Error Minimisation Search*** **(C-AEMS)**. Elle est comparée avec une méthode de patrouille réactive, le ***Reactive Markov Decision Process*** **(RMDP)**. Les résultats montrent peu de différence entre ces deux méthodes, laissant les auteurs conclure que dans le cadre de la patrouille, une stratégie simple peut être aussi performante qu'un modèle d'une haute complexité computationnelle.

D'autres approches reposent sur un modèle bayésien, où la théorie des probabilités bayésiennes est appliquée au contexte de l'apprentissage par renforcement. Ce modèle considère les états de l'environnement, les actions de l'agent et les récompenses associées comme des variables aléatoires. En utilisant des probabilités conditionnelles, un modèle bayésien peut être utilisé pour prédire les conséquences des actions et aider l'agent à prendre des décisions en tenant compte de l'incertitude inhérente à l'environnement. Les premières approches utilisant un modèle bayésien sont développées par PORTUGAL et ROCHA [100], avec les méthodes ***Greedy Bayesian Strategy*** **(GBS)** et ***State Exchange Bayesian Strategy*** **(SEBS)**. Au sein de ces deux méthodes, les agents communiquent entre eux leurs oisivetés individuelles à chaque nouvelle visite d'un nœud. Ces méthodes reposent également sur deux variables aléatoires, le premier représentant l'action de se déplacer, ou non, sur un nœud voisin, tandis que le second représente le gain de se déplacer sur ce nœud en fonction de son oisiveté et de la distance à parcourir. Cependant, en supposant une priorité entre les nœuds uniformes, et un gain n'incorporant pas d'aléatoire, le GBS adopte un comportement similaire à la méthode du CR, où les agents sélectionnent systématiquement le nœud voisin ayant la plus grande oisiveté. Tandis que les agents de la méthode GBS adoptent une stratégie individualiste, la méthode SEBS a été définie pour limiter l'interférence (cf. section 2.2.2), c'est-à-dire le risque de présence de deux agents sur un même nœud. Pour ce faire, la probabilité d'un agent de sélectionner un nœud à visiter prend en considération les intentions des autres agents souhaitant visiter également ce même nœud. La méthode ***Concurrent Bayesian Learner Strategy*** **(CBLS)** [95, 97] s'inscrit dans la continuité des approches précédentes, mais introduit un système distribué où chaque agent ajuste son comportement en fonction de récompenses individuelles. En considérant qu'un nœud a plus de chance d'être visité lorsqu'il possède un grand nombre de voisins, la notion du nombre de visites normalisées est proposée comme le nombre de visites d'un nœud divisé par son nombre de connexions avec des nœuds voisins. Une récompense positive est obtenue si l'agent visite un nœud possédant un nombre de visites normalisées relativement faible par rapport à ses voisins et possédant une oisiveté forte. Sinon, une récompense négative est attribuée à l'agent, à l'exception de la visite des nœuds ayant un unique voisin, où la récompense est nulle.

L'évaluation des approches repose sur trois environnements et un nombre d'agents allant de 1 à 12. En termes de réduction moyenne de l'oisiveté, la méthode CBLS se distingue avec la meilleure performance en moyenne, suivie par SEBS, puis par CR. Les méthodes GBS et MSP sont à égalité en termes de performance, tandis qu'HPCC et HCR ferment la marche. Comme le mentionne au sein de sa thèse OTHMANI-GUIBOURG [86], il est inhabituel de constater une performance aussi élevée pour la méthode réactive CR, en contradiction avec les résultats communément rapportés dans la littérature, tandis que la méthode centralisée navigatrice et heuristique HPCC présente une performance inattendue et nettement inférieure à ce qui est généralement observé.

2.3 Discussion sur les approches intégrant la recherche de cibles à la problématique de l'observation

Cette section de discussion s'intéresse aux stratégies de recherche de cibles mises en place par les approches répondant à la problématique de l'observation. Nous nous intéressons à leurs fonctionnements, mais également leurs limites, justifiant la création d'un nouveau formalisme développé à la section suivante. Durant cette discussion, nous ne prenons pas en considération les méthodes appartenant à des formalismes supposant des cibles coopératives. En effet, ces cibles ont la spécificité de partager continuellement leurs positions avec les agents, rendant inutile la mise en place d'une stratégie de recherche.

La recherche de cibles avec un déplacement aléatoire L'emploi du déplacement aléatoire constitue une approche simple mais peu efficace dans la recherche de cibles, car elle ne tire pas parti des avantages découlant de la collaboration entre les agents pendant les opérations de recherche. Dans le contexte du formalisme du CMOMMT, toutes les approches formelles répertoriées (c'est-à-dire celles dont la stratégie est explicite et non dérivée de l'apprentissage) optent pour une stratégie de recherche aléatoire. En ce qui concerne les méthodes utilisant l'apprentissage par renforcement, à notre connaissance, aucune indication d'observation ou de récompense n'est fournie pour guider l'agent vers l'adoption d'une quelconque stratégie de recherche.

La recherche de cibles avec une fonction de probabilité de densité Au sein du formalisme du DMST, les agents sont attirés vers les zones ayant la plus forte densité d'après le filtre PHD. Dans le cas où aucune cible n'est repérée, le filtre retourne une densité identique sur l'ensemble de l'environnement, afin que les agents se dispersent uniformément pour

couvrir l'environnement. Cependant, le procédé de communication de ce filtre entre les agents repose sur une hypothèse forte et peu applicable sur de vrai robot. En effet, les agents entrent en communication dans le cas où ils partagent une frontière de Voronoï commune. À titre d'illustration, lorsque l'environnement est composé de deux agents, il est attendu que ces derniers soient en communication continue, quelle que soit la dimension de l'environnement.

La recherche de cibles en encodant des informations dans des cellules Le formalisme du CMUOMMT (cf. section 2.1.4) représente l'environnement sous forme discrétisée, à travers un déplacement des agents et des cibles sur des cellules. De cette manière, les méthodes peuvent encoder la perception des agents sur des matrices ou des images. Ainsi, les cellules peuvent être employées pour contenir des informations liées aux visites.

La méthode du PAMTS mets en place une stratégie de recherche de cibles en plaçant une information de visite booléenne à chaque cellule. Ainsi, une cellule peut être de deux natures : soit déjà visitée, soit à visiter. La transition du premier état vers le second s'effectue après χ pas de temps, avec χ un paramètre spécifié par l'expérimentateur. Le déplacement des agents pour la recherche de cibles consiste à couvrir un maximum de cellule à visiter.

La méthode du DRL-CMUOMMT propose, quant à elle, que chaque cellule enregistre le temps "estampillé" t_{stamp} à laquelle la dernière visite par un agent a eu lieu. L'objectif est de minimiser une métrique nommée le taux d'exploration, définie de la manière suivante :

$$\beta = \frac{1}{T}\frac{1}{C_L C_W}\sum_{k}^{C_L}\sum_{l}^{C_W} t_{stamp}(c_{kl})$$

Avec T le temps total de l'expérimentation, C_L et C_W le nombre de cellules en longueur et largeur de l'environnement, ce dernier étant supposé rectangulaire. Enfin, $t_{stamp}(c_{kl})$ correspond au temps de la dernière visite par un agent de la cellule positionné à l'emplacement (k,l). Ainsi formulé, le taux d'exploration β est une métrique à maximiser. Sa borne supérieure $\beta = 1$ signifie que toutes les cellules sont observées par au moins un agent au temps $t = T$. La notion de t_{stamp} est assez proche de la notion d'oisiveté. En effet, l'oisiveté correspond à la différence entre le temps actuel et le temps de la dernière visite d'un lieu. Soit $i(c_{kl},t)$ l'oisiveté de la cellule c_{kl} au temps t, nous obtenons le lien suivant : $i(c_{kl},t) = t - t_{stamp}$. Par conséquent, la métrique β pourrait s'apparenter à l'oisiveté moyenne de l'environnement.

Cependant, la métrique β ne reflète pas l'exploration de l'environnement tout au long de la durée de la mission. Cette métrique est calculée uniquement à la toute fin de la mission, en analysant le temps "estampillé" de chaque cellule au temps T. Par conséquent, une méthode

se focalisant sur la visite des lieux sur la fin de la mission aura un taux d'exploration plus intéressant qu'une méthode couvrant continuellement l'environnement sur l'ensemble de la durée de la mission.

Par ailleurs, une autre restriction du formalisme du CMUOMMT réside dans la contrainte imposée à la dynamique de déplacement des agents et des cibles, qui sont restreints à se déplacer uniquement d'une cellule à une autre.

La recherche de cible approchée par la problématique de la patrouille La problématique de la patrouille a pour objectif de couvrir continuellement l'environnement. Son emploi pour la recherche de cibles mobiles provient initialement des travaux de ROBIN [105]. Sa thèse se place dans le contexte d'un système multi-agents hétérogène, où les agents possèdent différentes compétences et spécificités. Par ailleurs, les cibles ont la capacité de se camoufler dans certaines zones de l'environnement. L'enjeu est de réaliser le suivi des cibles, identifier le meilleur moment pour appeler des renforts ayant potentiellement d'autres capacités d'observation ou de déplacement, mais aussi d'assurer la patrouille de l'environnement afin de ne pas se faire manipuler par une cible. Les agents et les cibles se déplacent de cellules en cellules, disposées en grille carrée ou hexagonale.

Le contexte d'étude de ROBIN [105] est essentiellement mono-cible. Ainsi, les agents ont pour objectif d'assurer une surveillance continue d'un environnement, avec une topographie incluant des obstacles et des propriétés de terrain particulier, tout en assurant le suivi d'une seule et même cible. Cependant, comme mentionné dans l'introduction de cette thèse, de nombreuses applications se placent dans un scénario avec plusieurs cibles mobiles. Par exemple, nous pouvons énumérer les missions de surveillance terrestre ou maritime, la protection de la biodiversité par le suivi d'animaux sauvages, ou encore des missions de secourisme.

Par conséquent, nous proposons au sein de cette thèse d'étendre la problématique à un contexte multi-cibles, n'ayant pas la capacité de se dissimuler, mais avec un déplacement des agents et des cibles libres et non contraint par des cellules. Le chapitre suivant porte sur la définition de ce nouveau formalisme, associant la problématique de l'observation avec celle de la patrouille, en présentant notamment une autre manière de représenter de l'environnement et la dynamique de déplacement des agents et des cibles.

Deuxième partie

Contributions

Chapitre 3

Contribution I : Formalisation du problème de l'observation appuyée par la patrouille (POP) et résolution par champs potentiels

Ce chapitre présente, dans un premier temps, la définition d'un nouveau formalisme appelé Problème de l'Observation appuyée par la Patrouille (POP), qui associe la problématique de l'observation à celle de la patrouille. Ce formalisme offre une représentation de l'environnement qui permet d'illustrer les oisivetés de l'environnement tout en assurant une liberté de déplacement pour les agents et les cibles. Par la suite, les métriques de la patrouille sont redéfinies afin d'être adaptées à ce formalisme. Dans un second temps, ce chapitre décrit la mise en place d'une méthode distribuée basée sur un champ potentiel. Cette méthode, nommée Idleness-CMOMMT (I-CMOMMT), a pour objectif de trouver un compromis entre les objectifs parfois divergents entre le problème de l'observation de cibles et celui de la patrouille de l'environnement.

3.1 Définition formelle du problème de l'observation appuyée par la patrouille

La problématique de l'observation a pour objectif la maximisation de l'observation des cibles, et ce à travers différentes métriques, telles que la moyenne d'observation ou la distribution de l'observation parmi les cibles (cf. section 2.1.2). Par conséquent, les méthodes cherchant à résoudre cette problématique se focalisent essentiellement sur la coordination

à mener au sein des agents pour améliorer la répartition et le suivi des cibles. Cependant, comme nous avons pu le constater au sein de l'état de l'art, et plus particulièrement discuté à la section 2.3, ces méthodes ne détaillent pas de stratégie de recherche de cibles, car considérée comme un niveau d'abstraction plus élevé. Dans le cas de cibles non coopératives, les stratégies de recherche sont au mieux assimilées à un déplacement aléatoire des agents [92, 66].

Au sein de cette thèse, nous proposons que la recherche active de cibles s'effectue par une couverture continue de l'environnement, c'est-à-dire que les agents cherchent à répondre également à la problématique de la patrouille. Nous nommons ce nouvel enjeu le Problème de l'Observation appuyée par la Patrouille (POP). Les agents sont ainsi face à un dilemme d'exploration, via la recherche des cibles, et d'exploitation, à travers la maximisation de l'observation des cibles. Ce dilemme permet notamment d'apporter une solution au risque de manipulation évoqué par MARKOV et CARPIN [82]. En effet, si la méthode employée par les agents consiste à uniquement suivre continuellement les cibles, alors des cibles intelligentes risquent d'exploiter ce comportement à leur avantage en influençant le déplacement des agents.

Cette problématique s'adresse également à des scénarios dans lesquels des agents sont déployés pour observer à la fois des cibles mobiles et identifier des évènements statiques au sein de l'environnement. Par exemple, nous pouvons envisager comme scénario l'emploi de drones pour l'identification et le suivi de requins en bord de plage tout en surveillant la densité de nageurs sur la surface à sécuriser [17]. Nous pouvons également envisager la couverture de l'environnement par drone en cas de feu de forêt afin de détecter les foyers et le front du feu [113], couplée avec l'observation d'animaux ou de riverains aux alentours à protéger. En dernier exemple, le déploiement de drones pour secourir des victimes d'avalanche peut être appréhendé comme un problème dans lequel les victimes sont considérées comme des cibles coopératives, statique ou bien mobile, dont il faut maximiser l'observation [12].

Au sein de cette section, nous présentons dans un premier temps le formalisme, incluant la représentation de l'environnement, sur lequel repose le problème de l'observation appuyée par la patrouille. Par la suite, une reformulation des critères d'évaluation est présentée pour être adaptée à cette nouvelle représentation de l'environnement. Enfin, dans la dernière section, nous listons les objectifs à atteindre pour répondre au POP.

3.1.1 Formalisme

Dans l'objectif de clarifier les concepts, les notations et le cadre d'étude, nous proposons que le problème de l'observation appuyée par la patrouille soit présenté à travers le formalisme suivant :

— Soit S un espace en deux dimensions fermé, dont les agents et les cibles ne peuvent pas entrer ou sortir. $S \subset R^2$.
— Soit A est un ensemble de m agents.
— Soit O est un ensemble de n cibles.

Chaque agent $a_i \in A$, avec $i \in [1,m]$ est caractérisé par trois capacités :

— Une capacité de déplacement de sa position cartésienne $(x_{a_i}, y_{a_i}) \in S$, caractérisée par une vélocité v_{a_i}. La vitesse est bornée par une vitesse maximale, c'est-à-dire $v_{a_i} \leq v_{a_{max}}$.
— Une capacité d'observation des cibles, déterminée par une surface d'observation $s_o \in S$. Cette capacité prend en considération les propriétés d'identification, telles que le pourcentage de faux positifs, faux négatifs, ou le temps de calcul.
— Une capacité de communication avec les autres agents, décrit par une portée s_c, ainsi que les contraintes de communication comme le délai de transmission ou la capacité de la bande passante.

Chaque cible $o_j \in O$, avec $j \in [1,n]$, se définit par deux principales capacités :

— Une capacité de déplacement de sa position cartésienne $(x_{o_j}, y_{o_j}) \in S$, caractérisée par sa vélocité v_{o_j}, où $v_{o_j} < v_{o_{max}}$.
— Si la cible a un comportement évasif, elle possède également une capacité de détection des agents environnants afin de les éviter. Cette capacité se caractérise par une surface de détection $s_d \in S$.

Nous supposons que l'ensemble des agents utilisent un système de coordonnées global et partagé par tous. Ils disposent également d'une horloge interne synchronisée avec le groupe.

3.1.2 Représentation de l'environnement

Les objectifs

La manière de représenter l'environnement est une étape importante pour la résolution de problème complexe, notamment parce que la représentation conditionne la façon d'approcher et de proposer une solution à un problème. Comme le souligne le philosophe Alfred Korzybski [67] : "La carte n'est pas le territoire". Autrement dit, la représentation de notre environnement (la carte) n'est qu'un prisme, parmi tant d'autres, pour refléter la réalité du monde (le territoire). De cette manière, la carte simplifie et se focalise sur un ou plusieurs

aspects précis du monde. Par conséquent, le choix d'une représentation plutôt qu'une autre d'un problème peut être déterminant et significatif dans le développement de ses solutions.

Comme énoncé au sein de la section précédente, le formalisme de l'observation appuyée par la patrouille cherche à résoudre deux problématiques bien distinctes, possédant des objectifs, des métriques d'évaluation, des méthodes de résolution, mais aussi une représentation de l'environnement qui leur sont propres. D'une part, la problématique de l'observation repose sur une représentation continue ou discrète, où les agents et les cibles se déplacent sans discontinuité, ou de cellule en cellule. D'autre part, la problématique de la patrouille est représentée majoritairement au sein de la littérature sous forme d'un graphe, où chaque lieu d'intérêt est symbolisé par un nœud. De cette manière, chaque nœud possède une valeur d'oisiveté qui lui est propre.

L'objectif de cette section est de trouver une représentation de l'environnement permettant d'illustrer les informations essentielles aux deux problématiques, qui sont de positionner les agents et les cibles, ainsi que de stocker les valeurs d'oisiveté. Plus particulièrement, au sein du POP, nous souhaitons que la dynamique des agents et des cibles puisse être au plus proche d'une application réelle avec un déplacement continue, tout en s'assurant de pouvoir modéliser l'oisiveté au sein de l'environnement.

Les représentations étudiées et leurs limites

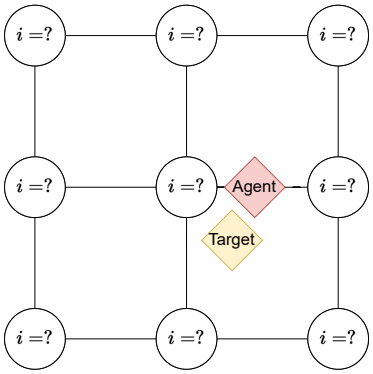


FIGURE 3.1 Scénario avec déplacement des agents continu.

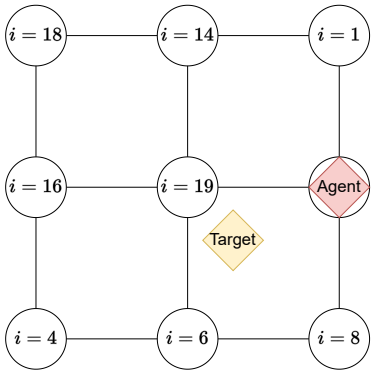


FIGURE 3.2 Scénario avec déplacement des agents sur un graphe.

Afin d'atteindre l'objectif fixé, plusieurs scénarios de représentation ont été étudiés. Le premier de ces scénarios est illustré par la figure 3.1. Les agents se déplacent de manière continue au sein de l'environnement. Cependant, au sein de ce scénario, les agents ne se déplacent plus de nœud en nœud, rendant inexploitables leurs valeurs d'oisiveté vis-à-vis

des métriques de la patrouille. Dans le second scénario, nous avons restreint les agents à se mouvoir uniquement sur les nœuds. Or, comme nous pouvons le constater sur la figure 3.2, le déplacement des agents devient sous efficace pour assurer l'observation des cibles. Par ailleurs, ce contexte ne respecte pas notre objectif fixé en amont d'avoir une mobilité continue des agents au sein de l'environnement.

Ces deux scénarios montrent les limites de l'exploitation d'un graphe pour représenter l'oisiveté dans le cadre du POP. Certains travaux étudiés au sein de l'état de l'art mettent en évidence d'autres manières d'illustrer la notion d'oisiveté. Ainsi, le formalisme du CCPP (cf. section 2.2.8), mais aussi les travaux de CHU et al. [30], proposent un environnement discrétisé en un maillage de nœud, assimilable à un ensemble de cellules sur deux dimensions. Chaque cellule possède une valeur d'oisiveté propre. Cependant, les agents sont contraints de se déplacer uniquement vers une cellule voisine. De plus, chaque agent ne peut visiter que la cellule sur lequel il se trouve, omettant ainsi la capacité d'un agent à observer les cellules aux alentours. Nous nous inspirons de l'emploi de cellule pour représenter l'oisiveté sur un environnement en deux dimensions. Or, nous souhaitons dans le cadre du POP que le déplacement des agents et des cibles ne soit pas restreint aux cellules, et qu'un agent puisse observer les cellules aux alentours en fonction de sa capacité d'observation.

Une représentation de l'oisiveté sous forme matricielle

Afin d'associer un déplacement continu des agents et des cibles avec une représentation de l'oisiveté sous forme de cellule, nous proposons de conserver les informations d'oisiveté au sein d'une matrice nommée **la carte d'oisiveté** M. Cette carte est composée d'un ensemble de cellules issues de la discrétisation de l'environnement. Chaque cellule possède ainsi une valeur d'oisiveté, symbolisant l'oisiveté de la surface représentée. La discrétisation s'effectue grâce à un coefficient de discrétisation d_f, s'exprimant en nombre de cellules par mètre carré. Plus ce coefficient est élevé, plus les informations sur un lieu seront précises, au prix d'un stockage des informations plus important. Une illustration de cette représentation est présentée au sein de la figure 3.3. La représentation matricielle de l'oisiveté permet notamment de stocker ces informations au sein de la mémoire des agents.

Comme nous pouvons le voir au sein de la figure 3.3, une ambiguïté demeure pour définir si une cellule est observée lorsqu'elle n'appartient pas intégralement à la surface d'observation d'un agent. Afin de clarifier cette ambiguïté, les cas envisagés pour l'observation d'une cellule sont les suivantes :

— Une cellule est considérée comme observée si la surface qu'elle représente est entièrement couverte par la surface d'observation d'un agent.

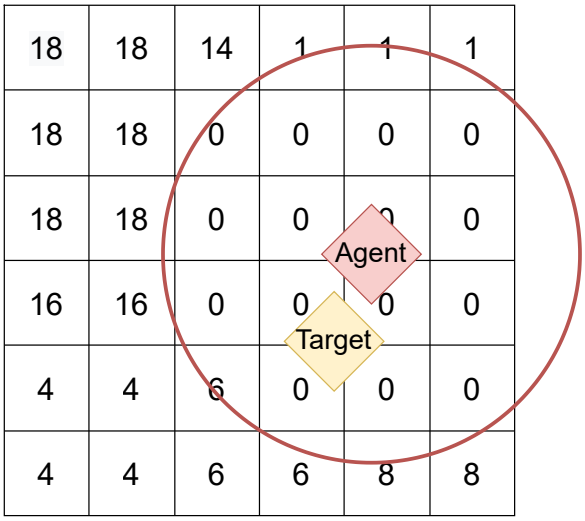


FIGURE 3.3 Scénario du POP, avec le déplacement continu des agents et la carte d'oisiveté en arrière plan.

— Une cellule est considérée comme observée si la surface qu'elle représente est couverte au moins à hauteur de 50% par la surface d'observation d'un agent.
— Une cellule est considérée comme observée si la surface qu'elle représente possède au minimum un point de contact avec la surface d'observation d'un agent.

Nous considérons que ce choix est libre à l'expérimentateur. Au sein de nos travaux, dans un objectif de clarté et de simplicité, nous considérons qu'une cellule est observée selon la définition du premier cas.

Formulation de la fonction d'oisiveté

L'évolution des oisivetés au sein de la carte M est régie selon une fonction d'oisiveté $i(a,b,t)$, retournant l'oisiveté pour chaque cellule positionnée aux coordonnées (a,b) au temps t :

$$\mathbb{N}^2 \times \mathbb{R} \to \mathbb{R}^+$$
$$a,b,t \mapsto i(a,b,t)$$

En début de mission, l'ensemble des oisivetés sont initialisés à une valeur α propre à l'expérimentateur :

$$i(a,b,t=0) = \alpha$$

Par la suite, à chaque pas de temps Δt, l'oisiveté de toutes les cellules s'incrémente par le même intervalle temporel :

$$i(a,b,t+\Delta t) = i(a,b,t) + \Delta t$$

Enfin, lorsqu'un agent a_i observe une surface s_o au temps t, alors l'oisiveté des cellules présentes sur cette même surface est réinitialisée à 0 :

$$\forall (a,b) \in s_o : i(a,b,t) = 0$$

Afin de simplifier l'écriture, nous notons par la suite qu'une cellule c_k positionnée aux coordonnées (a,b) possède une oisiveté au temps t notée $i_k(t)$.

3.1.3 Redéfinition des critères d'évaluation

Au sein du formalisme du POP, l'oisiveté n'est plus représentée sous forme de graphe, mais sous forme matricielle à travers la carte d'oisiveté. Cette transformation, explicitée au sein de la section précédente, nécessite par conséquent de redéfinir les critères d'évaluation liés à la notion d'oisiveté. Par conséquent, afin de comparer l'efficacité des méthodes au sein du formalisme du POP, nous proposons de redéfinir les critères d'évaluation suivants :

L'oisiveté moyenne I^{av}

L'oisiveté moyenne permet de connaître, au cours d'une période d'évaluation T, la moyenne des oisivetés de l'ensemble des cellules. Ce critère d'évaluation lié à la patrouille est par conséquent à minimiser. Sa définition mathématique s'exprime par :

$$I^{av} = \frac{1}{|C| \times T} \sum_{t \geq 0}^{T} \sum_{c_k \in C} i_k(t) \tag{3.1}$$

Avec $|C|$ le nombre de cellules c_k, C l'ensemble des cellules c_k et $i_k(t)$ la valeur d'oisiveté de la cellule c_k à l'instant t.

L'oisiveté maximale, ou pire oisiveté I^{max}

L'oisiveté maximale permet d'obtenir le record d'oisiveté qu'une cellule a pu obtenir au cours d'une période d'évaluation T. Par conséquent, cette métrique permet d'analyser si une

ou plusieurs zones ont été laissés longtemps sans patrouille. Sa définition est la suivante :

$$I^{max} = \max_{t \in [0,T]} i^m(t) \tag{3.2}$$

Avec $i^m(t)$ la pire oisiveté à l'instant t :

$$i^m(t) = \max_{c_k \in C} i_k(t)$$

L'oisiveté maximale régionale I_r^{max}

Une cellule correspond à la plus petite surface de l'environnement suite à sa discrétisation. Par conséquent, une cellule peut aisément ne pas être observée durant toute la durée T de la mission. Et ce plus particulièrement dans les cas où la carte d'oisiveté est obtenue avec un grand coefficient de discrétisation d_f, c'est-à-dire qu'un grand nombre de cellules est employé par mètre carré, ou sinon que le rayon d'observation des agents est relativement faible. Dans ce cas de figure, la métrique d'oisiveté maximale risque de plafonner à la valeur maximale $I^{max} = T$.

Pour obtenir une métrique plus représentative de la distribution de la visite des lieux, nous proposons d'analyser non pas l'oisiveté maximale d'une seule cellule, mais d'un ensemble de cellule que nous nommons **région**. Pour ce faire, nous n'utilisons pas la carte d'oisiveté M, mais une carte d'oisiveté régionale notée M_r. La carte d'oisiveté régionale est obtenue à l'aide d'une convolution de la carte d'oisiveté avec un filtre moyen représenté par son masque (ou noyau) ω : $M_r = \omega * M$. Nous proposons que la taille du filtre corresponde au rayon d'observation des agents. Ainsi l'oisiveté maximale régionale s'exprime de la manière suivante :

$$I_r^{max} = \max_{t \in [0,T]} \max_{c_k \in C} ir_k(t) \tag{3.3}$$

Avec $ir_k(t)$ l'oisiveté régionale en position k à l'instant t.

Pour illustrer le fonctionnement de l'oisiveté maximale régionale, considérons deux cas d'étude. La figure 3.4 présente un exemple de cas où des oisivetés sont distribuées de manière sporadique, c'est-à-dire que les cellules ayant de fortes oisivetés sont isolées des autres. Tandis que la figure 3.5 expose un cas où les cellules ayant de fortes oisivetés sont regroupées. Dans les deux cas, la valeur maximale d'oisiveté est identique, avec $I^{max} = 100$. Les cartes régionales M_r sont obtenues en prenant l'hypothèse que le rayon d'observation

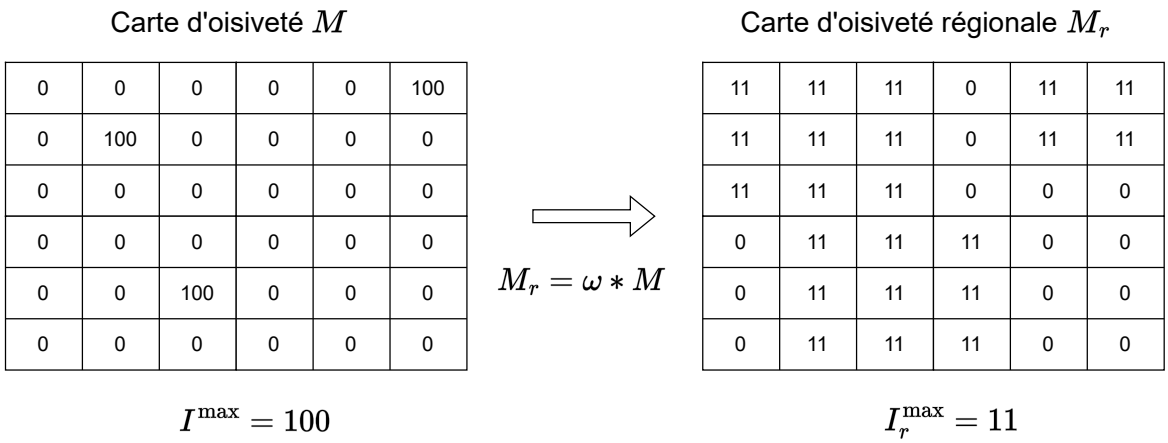


FIGURE 3.4 Calcul des oisivetés maximales avec une distribution sporadiques.

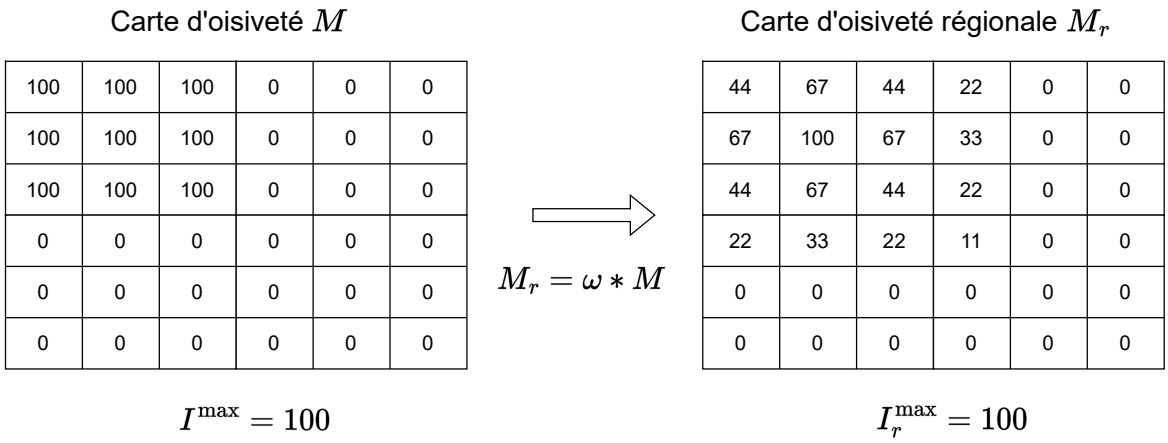


FIGURE 3.5 Calcul des oisivetés maximales avec une distribution regroupée.

permet d'observer une cellule aux alentours, c'est-à-dire assimilable au masque suivant :

$$\omega = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Dans le contexte de la distribution sporadique de la figure 3.4, nous obtenons une oisiveté régionale maximale $I_r^{max} = 11$, donc bien plus faible que dans le cas d'une distribution regroupée, avec $I_r^{max} = 100$. Cet exemple permet d'illustrer que l'oisiveté régionale maximale est une métrique utile pour identifier la durée record pour laquelle une région, soit un ensemble de cellules voisines, n'a pas été visitée.

L'intervalle moyen MI L'intervalle moyen est redéfini pour considérer les successions des visites, non plus des noeuds, mais des cellules. Les notations sont donc conservées, avec J_k l'ensemble d'intervalles de visite de la cellule c_k. De cette manière, l'intervalle moyen MI est adapté de la manière suivante :

$$MI = \frac{1}{N_J} \sum_{c_k \in C} \sum_{j_{k,i} \in J_k} j_{k,i} \tag{3.4}$$

L'intervalle quadratique MSI L'intervalle quadratique est adapté de la même manière que l'intervalle moyen, où les noeuds $n_k \in N$ sont remplacés par les cellules $c_k \in C$. La redéfinition de la métrique MSI s'exprime ainsi par :

$$MSI = \sqrt{\frac{1}{N_j} \sum_{c_k \in C} \sum_{j_{k,i} \in J_k} j_{k,i}^2} \tag{3.5}$$

3.1.4 Les objectifs à atteindre

Les méthodes multi-agents cherchant à répondre au POP ont plusieurs objectifs à atteindre. Ainsi, le POP peut être considéré comme une problématique multi-objectif, avec quatre principaux à objectifs à atteindre :

- Évitement de collision entre les agents : Les agents ne doivent pas entrer en collision les uns avec les autres, et maintenir au mieux une distance de sécurité.
- Aucune sortie de l'environnement : L'environnement est considéré comme un système fermé. Par conséquent, les agents ainsi que les cibles ne doivent pas sortir de l'environnement.
- Maximisation de l'observation : Les méthodes déploient les agents de manière à observer le plus de cibles possible, via par exemple la métrique A, mais également à équilibrer au mieux l'observation entre les cibles, à travers notamment la métrique H ou la déviation standard σ_n.
- Minimisation de l'oisiveté : Les méthodes cherchent également à minimiser l'oisiveté de l'environnement, que ce soit à travers les métriques de l'oisiveté moyenne, de la pire oisiveté, ou de l'oisiveté maximale régionale.

3.2 Une résolution multicritère par des champs potentiels : La méthode du I-CMOMMT

Comme décrit au sein de la section précédente, le problème de l'observation appuyée par la patrouille cherche à répondre à un double enjeu : maximiser l'observation des cibles, et minimiser l'oisiveté de l'environnement. L'objectif est également de réduire la disparité d'observation entre les cibles. Le formalisme du POP représente la dynamique des agents et des cibles de la même manière que le formalisme du CMOMMT (cf. section 2.1.3), c'est-à-dire avec des déplacements continus et un environnement fermé. Un grand nombre de méthodes s'appuyant sur le formalisme du CMOMMT proposent une résolution à travers des champs potentiels, tels que le A-CMOMMT [92], le B-CMOMMT [66] ou encore le P-CMOMMT [38]. Les champs potentiels, ou champs de force, ont l'avantage de décrire le comportement et la dynamique des agents de manière simple. Ainsi, chaque objectif est assimilé à une force, dont sa magnitude caractérise son niveau de priorité ou d'importance.

Au sein de cette section, nous proposons une première résolution du POP à travers l'exploitation d'un champ de force. Dans un premier temps, nous explicitons le fonctionnement des forces employées au sein de la méthode. Dans un second temps, nous détaillons les informations communiquées entre les agents, et les conséquences que cela peut impliquer dans leurs comportements. Par la suite, nous proposons des solutions pour améliorer la coordination entre les agents. Puis, nous mettons en évidence les expériences réalisées et les résultats obtenus en comparant notre approche avec des méthodes se focalisant sur l'observation ou la patrouille. Enfin, nous présentons notre conclusion et les perspectives d'amélioration de la méthode.

3.2.1 Description du champ potentiel

Nous proposons de résoudre le POP à travers une méthode distribuée que nous nommons ***Idleness-CMOMMT*** **(I-CMOMMT)**. Son champ de force est comparable à la méthode du A-CMOMMT (cf. équation 2.5), cette dernière ayant pour objectif de maximiser l'observation des cibles et d'éviter la collision avec les autres agents. Cependant, au sein du I-CMOMMT, nous ajoutons un troisième objectif, celui de patrouiller également l'environnement. Ainsi, chaque objectif est représenté par une force. La somme des forces, dictant le comportement de l'agent a_i au temps t, se définit de la manière suivante :

$$F(a_i,t) = (1-\lambda(t)) \sum_{j=1}^{n} \omega_{i,j} f_{i,j}^{t} + \sum_{k=1}^{m} f_{i,k}^{r} + \lambda(t) f_i^{p} \tag{3.6}$$

Chaque agent a_i, au temps t, subit une force d'attraction $f^t_{i,j}$ pour chaque cible o_j dans sa surface d'observation s_o, une force de répulsion $f^r_{i,k}$ pour chaque autre agent a_k dans sa surface de communication s_c, et enfin une dernière force d'attraction f^p_i vers une région sélectionnée ayant une forte oisiveté. Le choix de la cellule à visiter est explicité au sein de la section 3.2.3.

L'expression des forces d'attraction $f^t_{i,j}$ et des forces de répulsion $f^r_{i,k}$ est identique à l'approche du A-CMOMMT (cf. section 2.1.3), s'exprimant ainsi en fonction de la distance avec une cible j ou avec un autre agent k. Les magnitudes des forces sont présentées en figure 2.3. La force de patrouille f^p_i a une magnitude constante de 1.

Les objectifs de suivre les cibles, mais aussi de chercher de nouvelles cibles via la patrouille, constituent un dilemme à résoudre. Nous proposons d'arbitrer la priorité entre ces deux objectifs à travers un coefficient évoluant en fonction de l'oisiveté de l'environnement. De cette manière, la priorité est donnée à l'observation lorsque l'oisiveté est faible, et est rendue à la patrouille lorsque l'oisiveté devient importante. La notion de forte oisiveté est subjective et dépend essentiellement du scénario étudié. Par conséquent, nous permettons à l'expérimentateur d'exprimer à partir de quelle valeur une oisiveté est considérée comme forte à travers la configuration d'un paramètre σ, nommé "seuil d'oisiveté", exprimé en seconde ou en pas de temps.

Nous définissons ainsi la fonction $\lambda(t) \in [0;1]$ comme la priorité de la patrouille sur l'observation, grâce à la pondération de la force de patrouille f^p_i et des forces d'observation f^t_{ij}. Naturellement, si un agent a_i n'a aucune cible dans son champ d'observation, alors la patrouille est son unique priorité, avec $\lambda(t) = 1$. Sinon, la fonction $\lambda(t)$ évolue selon la valeur de l'oisiveté régionale maximale présente au sein de la carte individuelle de l'agent. Nous proposons ainsi de définir cette fonction par :

$$\lambda(t) = tanh\left(\frac{\max_{c_k \in C}\, ir_k(t)}{\sigma}\right)$$

Avec $ir_k(t)$ l'oisiveté régionale au sein de la cellule k, C l'ensemble des cellules de la carte d'oisiveté individuelle de l'agent a_i et σ le seuil d'oisiveté.

La figure 3.6 illustre un exemple d'application des forces au sein de la méthode du I-CMOMMT. Ainsi l'agent 1 possède un rayon d'observation, où il peut observer les cibles 1 et 2, mais ne peut pas voir la cible 3. Chaque cible au sein du rayon d'observation génère une force d'attraction, respectivement $f^t_{1,1}$ et $f^t_{1,2}$. De plus, l'agent 1 dispose d'un rayon de communication lui permettant d'être en contact avec l'agent 2, mais pas l'agent 3. L'agent 2

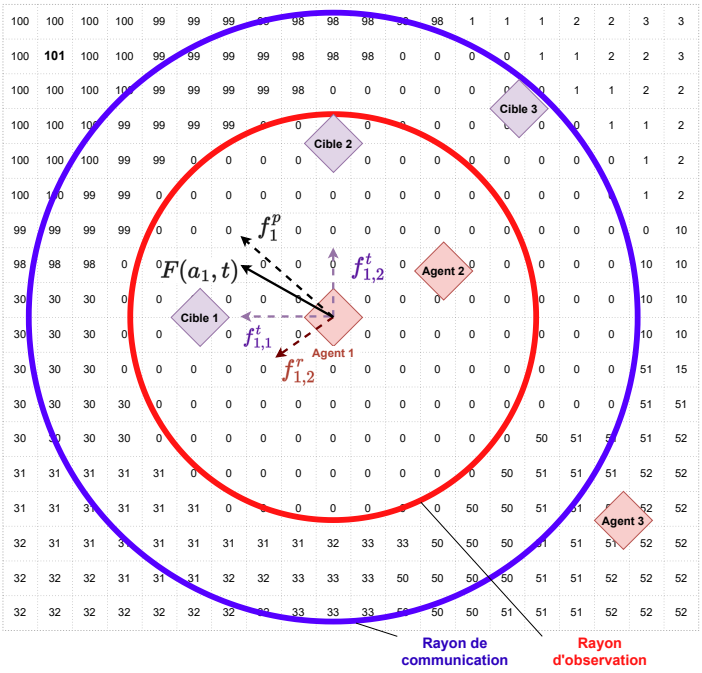


FIGURE 3.6 Illustration des forces subies par un agent au sein du I-CMOMMT.

étant relativement proche de l'agent principal, il exerce une force de répulsion $f_{1,2}^r$. Enfin, la carte d'oisiveté est représentée par l'ensemble des cellules en arrière-plan de la figure, contenant chacune une valeur d'oisiveté. L'agent est attiré par l'oisiveté la plus forte, ici la valeur 101 en haut à gauche, générant ainsi une force d'attraction f_1^p. La somme des forces est représentée par le vecteur $F(a_1,t)$, l'agent se déplaçant dans cet exemple vers le haut à gauche.

3.2.2 Communication entre les agents

La méthode du I-CMOMMT repose sur une coordination distribuée, impliquant que la communication entre les agents n'est pas forcément continuellement maintenue. Ainsi, deux agents ne peuvent communiquer que lorsque ces derniers rentrent respectivement au sein de leur rayon de communication. Durant leurs communications, plusieurs informations sont échangées afin d'améliorer la coordination.

Premièrement, les agents partagent respectivement leurs positions, afin de calculer la force de répulsion à appliquer pour éviter tout risque de collision. Deuxièmement, les agents se partagent également leurs cartes d'oisiveté. L'objectif pour un agent est d'actualiser les valeurs d'oisiveté de sa propre carte d'oisiveté avec celle de l'agent en communication, en ne conservant que les informations les plus récentes. En conséquence, pour chaque cellule,

la valeur d'oisiveté conservée est celle possédant la plus faible valeur entre les deux cartes, c'est-à-dire celle mise à jour le plus récemment. Ce procédé peut s'exprimer par la formule suivante, entre un agent a_1 réceptionnant une carte d'un agent a_2 :

$$\forall c_k \in C_{a_1}, i_k(t)_{a_1} = min(i_k(t)_{a_1}, i_k(t)_{a_2})$$

Avec C_{a_1} l'ensemble des cellules de l'agent a_1, et $i_k(t)_{a_i}$ la valeur d'oisiveté de la cellule c_k au temps t de l'agent a_i.

Ce partage des cartes d'oisiveté permet aux agents de réduire l'écart entre les oisivetés locales, propre à chaque agent, et l'oisiveté globale de l'environnement. Être au plus proche de l'oisiveté réelle du terrain permet aux agents de prendre de meilleures décisions dans les lieux à visiter. Cependant, en homogénéisant les cartes d'oisiveté parmi les agents en communication, ces derniers risquent de désirer de visiter les mêmes lieux. Pour pallier ce problème, la section suivante met en évidence des systèmes de coordination présents au sein de la méthode du I-CMOMMT.

3.2.3 Coordination dans le choix des lieux à visiter

Une communication régulière des cartes d'oisiveté entre les agents a pour conséquence de les homogénéiser. En possédant les mêmes cartes, les agents identifient les fortes oisivetés aux mêmes endroits, avec comme risque de désirer visiter les mêmes lieux. Ce phénomène a pour conséquence d'augmenter le risque de collision, mais aussi d'avoir un impact négatif sur l'efficacité de la couverture de l'environnement. Pour y remédier, un système de coordination dans la distribution des tâches de la patrouille est mise en place.

La coordination au sein du I-CMOMMT repose sur deux mécanismes, illustrés à travers une architecture fonctionnelle à la figure 3.7. Le premier mécanisme repose sur un système de sélection stochastique, en s'inspirant de la coordination mise en place au sein de la méthode du *Probabilistic Greedy Selection* (PGS) (cf. section 2.2.8). Pour le I-CMOMMT, les agents répertorient N cellules ayant la plus forte oisiveté régionale au sein de leur carte. Les cellules les plus proches de l'agent sont privilégiées en cas d'égalité d'oisiveté, afin d'ajouter une dimension heuristique à la couverture de l'environnement. Ensuite, l'agent sélectionne aléatoirement, parmi ces N cellules, une seule à visiter. Le paramétrage de la valeur N est propre à l'expérimentateur. Une forte valeur est propice à réduire le risque que les agents sélectionnent une même cellule à visiter. Cependant, cela a pour conséquence de permettre aux agents de choisir de visiter des lieux possédant des oisivetés moins importantes.

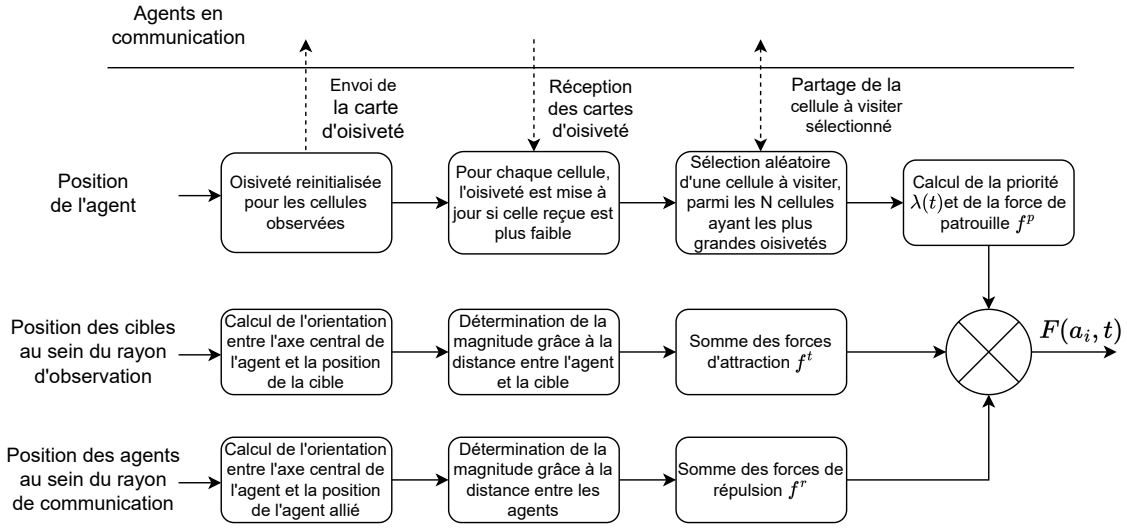


FIGURE 3.7 Architecture fonctionnelle du I-CMOMMT.

Le second mécanisme de coordination repose sur un système de sélection par priorité. Chaque agent se voit attribuer un identifiant qui lui est propre, avec une valeur allant de 1 à m. Lorsque des agents rentrent en communication, ces derniers échangent le lieu qu'ils désirent visiter ainsi que leurs identifiants. Dans le cas où la sélection du lieu est identique, alors l'agent ayant l'identifiant le plus faible décide de choisir un autre lieu à visiter. Ce système de priorité permet d'éviter que deux agents visitent deux fois un même lieu. Cependant, l'agent ayant l'identifiant le plus faible, et donc laissant la priorité de visite, considère l'hypothèse que ce lieu sera effectivement visité prochainement. Cette hypothèse peut ne pas être confirmée dans le cas où l'agent prioritaire croise une ou plusieurs cibles et décide de privilégier l'observation à la patrouille.

3.2.4 Expériences et résultats

Dans cette section expérimentale, nous présenterons les détails de notre méthodologie ainsi que les procédures mises en œuvre pour mener à bien l'étude de l'efficacité du I-CMOMMT. Dans un premier temps, nous nous intéressons à l'impact du paramètre σ pour répondre au problème de l'observation appuyée par la patrouille. Par la suite de cette étude, nous fixons une valeur de σ trouvant le meilleur compromis parmi les objectifs du POP, et nous comparons le I-CMOMMT avec le A-CMOMMT, mais aussi deux méthodes de patrouille et enfin une méthode de déplacement aléatoire.

L'environnement de simulation repose sur un environnement Rviz/Gazebo/ROS2 détaillé au sein du chapitre 6. Les cibles suivent un comportement aléatoire, en choisissant arbi-

trairement une position appartenant à l'environnement S. Une fois atteint, avec une vitesse constante, une nouvelle position est sélectionnée. Nous ne supposons aucune communication entre les cibles.

Impact du seuil d'oisiveté σ sur l'efficacité du I-CMOMMT

Le paramètre σ permet à l'expérimentateur de spécifier à partir de quelle valeur une oisiveté est considérée comme importante. Comme explicité au sein de la section 3.2.1, ce paramètre est utilisé par les agents pour prioriser l'exploration de l'environnement ou le suivi de cibles. En effet, lorsqu'une région dans l'environnement possède une oisiveté proche du seuil d'oisiveté σ, alors la priorité est donnée principalement à la patrouille. À l'inverse, lorsque les régions ont une oisiveté relativement faible, alors les agents se concentrent sur l'observation des cibles. Afin de gagner en clarté d'écriture, nous exprimons le paramètre de σ en fonction de la durée de la mission T. Pour ce faire, nous définissons le coefficient σ_c de la manière suivante :

$$\sigma_c = \frac{\sigma}{T}$$

De cette manière, le seuil d'oisiveté est inclus dans la notation de la méthode via l'écriture suivante : I-CMOMMT-x, avec x la valeur du rapport σ_c.

L'impact du coefficient σ_c sur les efficacités de la méthode I-CMOMMT est étudié à travers une série d'expérimentations, où la majorité des paramètres caractérisant la mission sont fixes, à l'exception du nombre d'agents, du nombre de cibles et du rapport σ_c qui seront variables. Le cadre environnemental consiste en une mission d'une heure, avec un environnement carré de $100m$ de côté. La carte d'oisiveté est discrétisée selon un facteur $d_f = 1$ cellule par mètre carré. Les agents possèdent un rayon d'observation de $5m$ et un rayon de communication de $7m$. Les cibles ont une vitesse maximale de $1m/s$, tandis que les agents peuvent se déplacer au mieux à $2m/s$. Nous expérimentons avec un environnement de 2, 3 et 5 agents, traquant 3, 5 ou 7 cibles. Enfin, nous comparons les résultats du I-CMOMMT prenant des valeurs de $\sigma_c = \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

Dans un premier temps, nous nous intéressons à l'évolution des métriques au cours du temps sur un scénario, puis dans un second temps, nous étudions les moyennes et écart-type de ces mêmes métriques pour l'ensemble des scénarios expérimentés. Afin d'évaluer l'observation des cibles, nous retenons la moyenne d'observation (métrique A), ainsi que la déviation standard de l'observation. Pour évaluer la patrouille, nous nous concentrons sur la moyenne d'oisiveté, et l'oisiveté maximale régionale.

Le scénario étudié est composé de 2 agents et 3 cibles. La figure 3.8 représente l'évolution de la moyenne d'observation, tandis que la figure 3.9 représente la distribution de l'observation des cibles via la métrique de la déviation standard. Comme nous pouvons le constater, plus le seuil d'oisiveté σ est grand, à travers un coefficient σ_c élevé, et plus la moyenne d'observation des cibles augmente. Cependant, l'observation des cibles est aussi plus hétérogène, avec certaines cibles plus observées que d'autres. Cette tendance s'explique par une priorisation du suivi de cibles sur la recherche de nouvelles cibles lorsque le seuil d'oisiveté est plus grand.

La patrouille de l'environnement est également impactée par le coefficient σ_c. En effet, les agents désirent visiter les lieux dont la valeur d'oisiveté est supérieure au seuil d'oisiveté σ. Par conséquent, plus la valeur de σ_c est faible, plus l'oisiveté moyenne présentée au sein de la figure 3.10 est minimisée. Cette tendance à prioriser la patrouille pour des seuils d'oisiveté faibles est d'autant plus significative au regard de l'oisiveté maximale régionale. Au sein de la figure 3.11, nous constatons que les courbes sont linéaires, comparable à la fonction $y = x$, jusqu'à ce que l'oisiveté maximale régionale soit trop forte comparée à l'oisiveté seuil, motivant les agents à visiter ces régions.

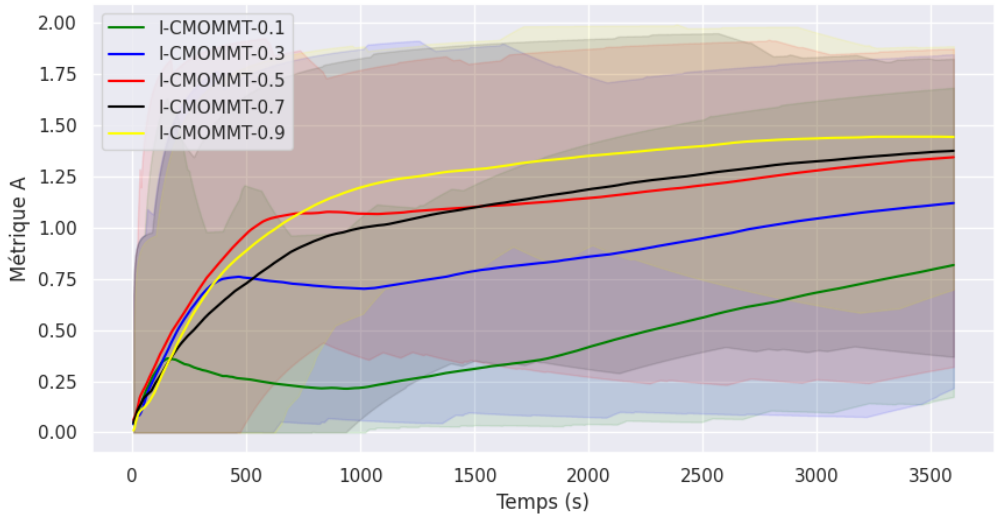


FIGURE 3.8 Évolution de la moyenne d'observation au cours du temps pour plusieurs paramétrages de σ_c

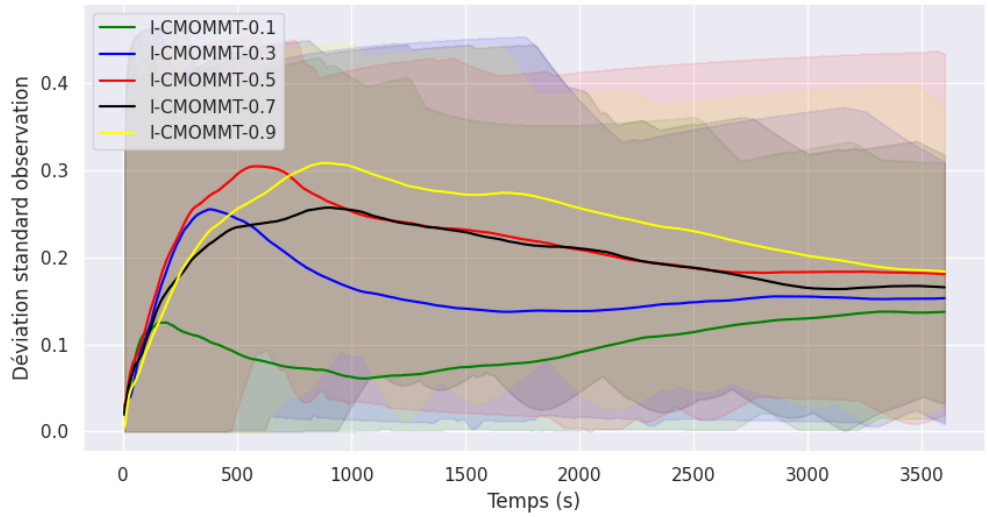


FIGURE 3.9 Évolution de la déviation standard de l'observation des cibles au cours du temps pour plusieurs paramétrages de σ_c

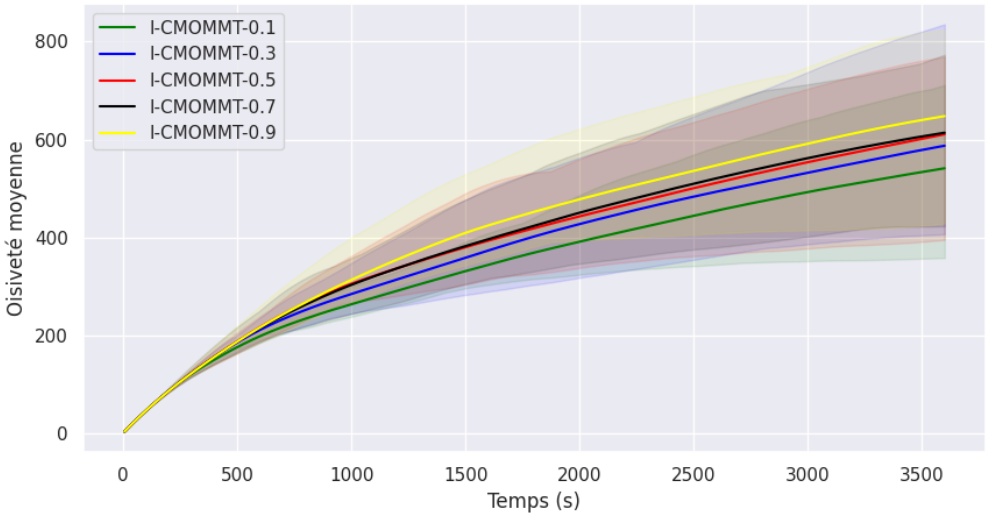


FIGURE 3.10 Évolution l'oisiveté moyenne au cours du temps pour plusieurs paramétrages de σ_c

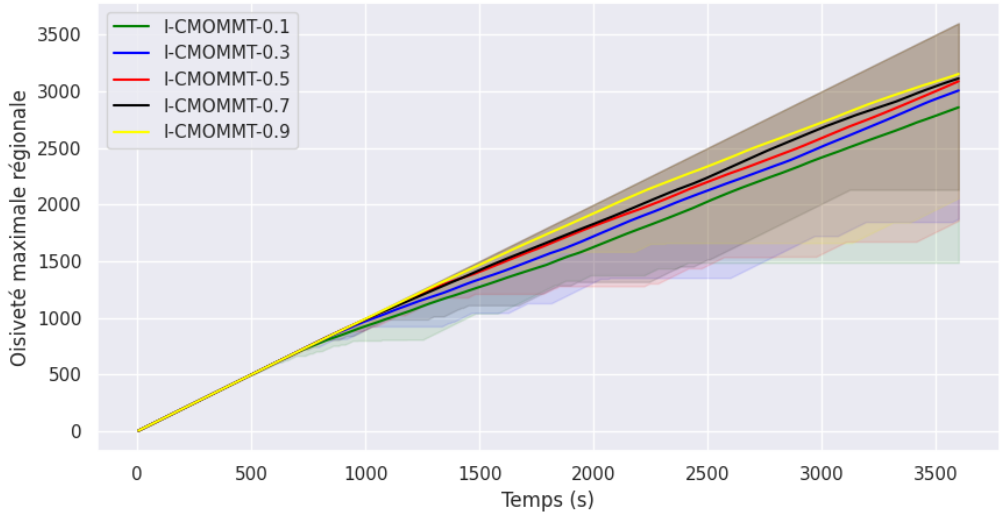


FIGURE 3.11 Évolution de l'oisiveté maximale régionale au cours du temps pour plusieurs paramétrages de σ_c

Comparaison avec d'autres méthodes de patrouille et d'observation

L'efficacité de la méthode du I-CMOMMT à répondre au POP est comparée sur une série de simulations, dont les caractéristiques sont présentées au sein du tableau 3.1. Nous explorons deux scénarios en faisant varier le rayon d'observation et le rayon de communication. Nous proposons d'étudier le I-CMOMMT avec le paramètre $\sigma_c = 0.5$, car le choix de ce seuil d'oisiveté a montré précédemment une ambivalence à répondre au suivi de cibles et à la distribution de l'observation de ces dernières. Les résultats du I-CMOMMT sont comparés avec la méthode d'observation du A-CMOMMT [92], ainsi que trois autres méthodes de patrouilles développées par PAMPONET MACHADO et al. [90]. Cependant, comme explicité au sein de la section 2.2.3, ces méthodes ont été conçues avec une représentation de l'environnement sous forme de graphe. Par conséquent, le comportement des agents a été adapté au formalisme du POP de la manière suivante :

***Random Reactive* (RR)** Les agents se dirigent vers une position sélectionnée aléatoirement au sein de l'environnement. Une fois atteint, les agents font route vers une nouvelle position aléatoire. Au sein de [90], les agents choisissent aléatoirement des noeuds à visiter.

***Closest Idleness* (CI)** Inspirée de la méthode du *Conscientious Reactive* (CR), les agents visitent, parmi les cellules environnantes proches, celle possédant la plus grande valeur d'oisiveté. Les agents peuvent partager aux autres agents au sein du rayon de communication leur propre carte d'oisiveté.

***Highest Idleness* (HI)** Inspirée de la méthode du *Conscientious Cognitive* (CC), les agents se dirigent, parmi toutes les cellules de la carte d'oisiveté, vers celle détenant le record maximal d'oisiveté. Les agents peuvent partager aux autres agents au sein du rayon de communication leur propre carte d'oisiveté.

Paramètre	**Valeur scénario 1**	**Valeur scénario 2**
Taille de l'environnement	$100m \times 100m$	
Durée mission T	3 600 s	
Vitesse maximale des cibles	0.5 m/s	
Vitesse maximale des agents	1 m/s	
Coef. de discrétisation df	1 cellule/m²	
σ	$0.5 \times T$	
N	10	
(dr_1, dr_2)	(1m, 2m)	
Rayon d'observation	4m	3m
Rayon de communication	5m	7m
$(do_1, do_2, do_3$, rayon de suivi prédictif)	(1m, 2m, 4m, 5m)	(1m, 2m, 3m, 5m)

TABLE 3.1 Paramètres de simulation pour deux scénarios, incluant les configurations I-CMOMMT et A-CMOMMT.

Le cadre expérimental consiste à évaluer l'efficacité de toutes les méthodes pour un environnement ayant successivement 2, 4 puis 6 agents, cherchant à observer 3, 5 puis 7 cibles. Chaque configuration agents/cibles est évaluée 20 fois. La position initiale des agents et des cibles est aléatoire. L'évaluation de la performance liée à l'observation des cibles repose sur la moyenne d'observation (métrique A) et la déviation standard de l'observation, présentées respectivement au sein des figures 3.12 et 3.14 pour le premier scénario, et des figures 3.13 et 3.15 pour le second scénario. Tandis que les objectifs de la patrouille sont évalués à travers la moyenne d'oisiveté ainsi que l'oisiveté maximale régionale, mises en évidence respectivement par les figures 3.16 et 3.18 pour le premier scénario, et les figures 3.17 et 3.19 pour le second scénario.

FIGURE 3.12 Moyenne d'observation en fonction du nombre d'agents et nombre de cibles pour le scénario 1.

FIGURE 3.13 Moyenne d'observation en fonction du nombre d'agents et nombre de cibles pour le scénario 2.

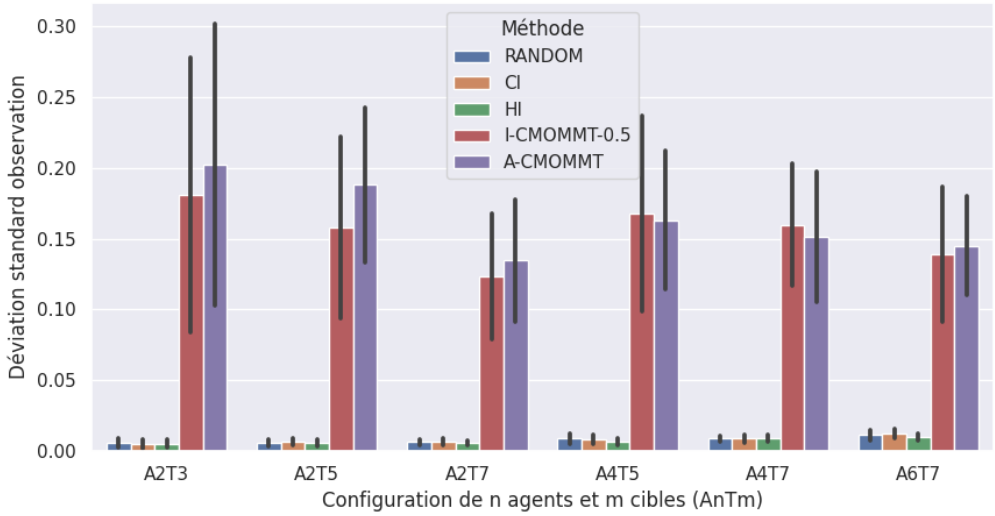


FIGURE 3.14 Déviation standard de l'observation des cibles en fonction du nombre d'agents et du nombre de cibles pour le scénario 1.

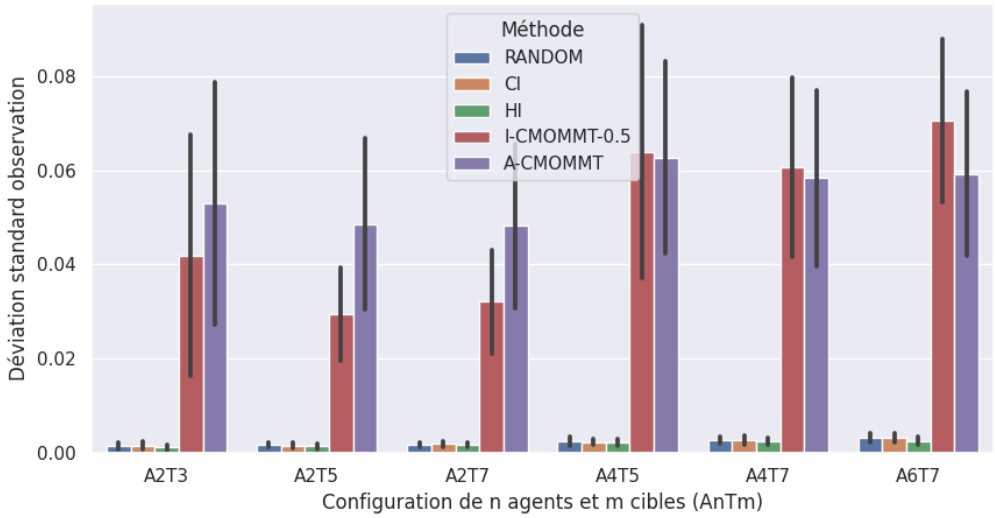


FIGURE 3.15 Déviation standard de l'observation des cibles en fonction du nombre d'agents et du nombre de cibles pour le scénario 2.

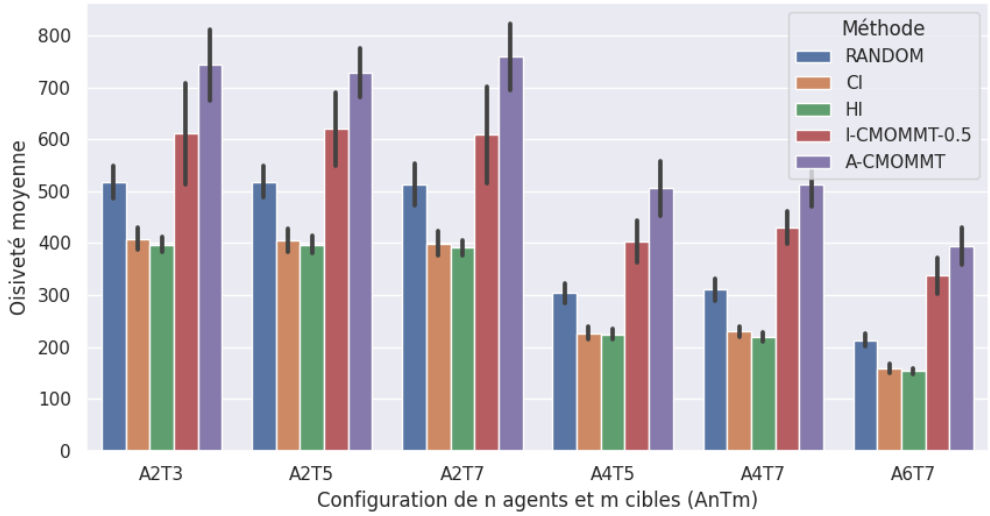


FIGURE 3.16 Oisiveté moyenne en fonction du nombre d'agents et du nombre de cibles pour le scénario 1.

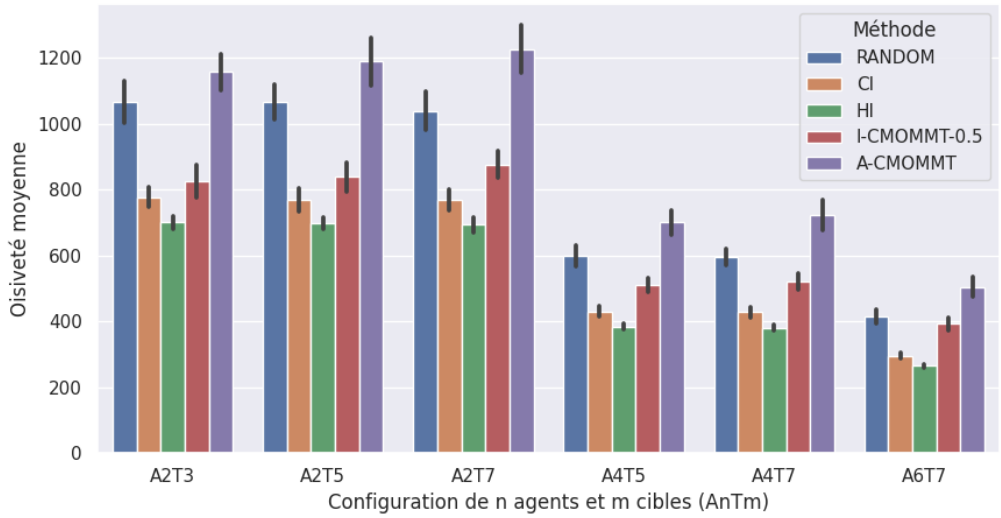


FIGURE 3.17 Oisiveté moyenne en fonction du nombre d'agents et du nombre de cibles pour le scénario 2.

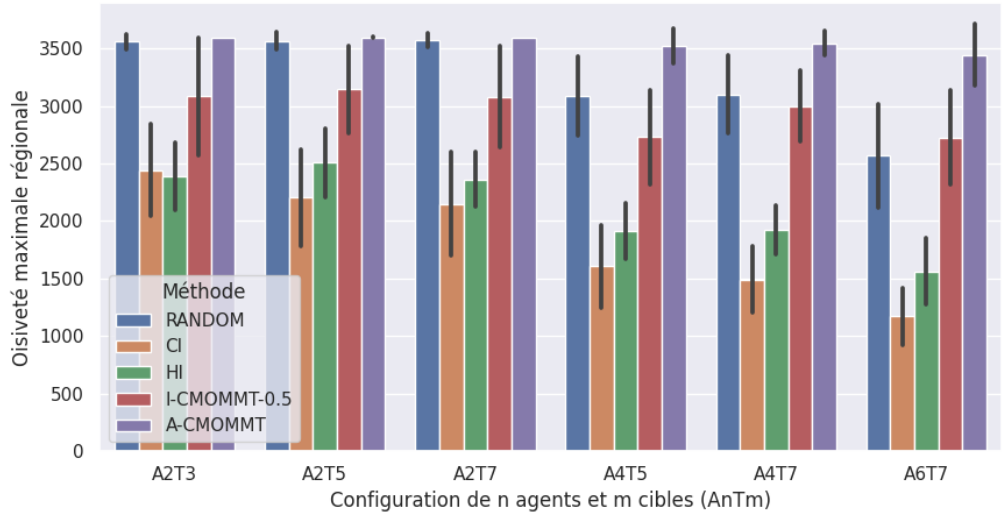


FIGURE 3.18 Oisiveté maximale régionale en fonction du nombre d'agents et du nombre de cibles pour le scénario 1.

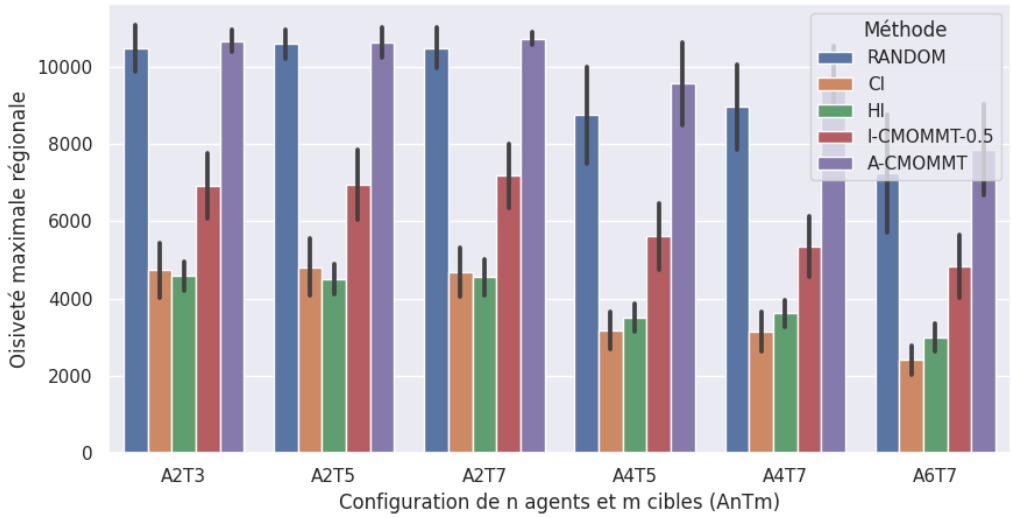


FIGURE 3.19 Oisiveté maximale régionale en fonction du nombre d'agents et du nombre de cibles pour le scénario 2.

On peut constater, en comparant le scénario 1 et le scénario 2, que la diminution du rayon d'observation a un effet significatif sur les performances des méthodes. Dans l'ensemble, lorsque le rayon d'observation est faible, les oisivetés moyennes et maximales régionales ont tendance à augmenter, tandis que les performances d'observation des cibles sont réduites.

Par ailleurs, il est manifeste que les méthodes de patrouille, telles que le CI et le HI, présentent des bonnes performances pour réduire les oisivetés de l'environnement, tout en ayant des efficacités quasi nulles pour le suivi de cibles. Au sein des méthodes mentionnées, de même que la méthode aléatoire, les cibles sont rencontrées et donc observées de manière aléatoire sans effort de suivi, ce qui se traduit dans les figures 3.14 et 3.15 par une faible disparité d'observation entre les cibles. En revanche, la méthode A-CMOMMT se concentre spécifiquement sur le suivi des cibles, ce qui lui permet de présenter de bonnes performances pour maximiser leurs observations. Cependant, cette méthode ne se préoccupe pas de la patrouille de l'environnement et ne réduit l'oisiveté que pendant les déplacements effectués pour le suivi des cibles.

Dans le contexte du scénario 1, on peut constater que l'emploi de la méthode I-CMOMMT permet d'observer les cibles avec une efficacité atteignant 90% de celle du A-CMOMMT, tout en réduisant l'oisiveté régionale moyenne et maximale d'environ 20%. Cette tendance est particulièrement accentuée dans le scénario 2, où le rayon d'observation est diminué et le rayon de communication étendu. La méthode I-CMOMMT permet d'observer les cibles de manière quasi équivalente, voire supérieure, à la méthode A-CMOMMT, tout en réduisant l'oisiveté moyenne de 30% et l'oisiveté moyenne régionale de 40%. Cependant, pour la configuration étudiée et pour $\sigma_c = 0.5$, la méthode I-CMOMMT présente des déviations standards pour l'observation des cibles qui sont comparables à celles du A-CMOMMT, ce qui suggère que la méthode n'apporte pas d'amélioration significative dans la répartition de l'observation entre les cibles.

3.2.5 Conclusion

Pour résoudre la problématique de l'observation appuyée par la patrouille, la méthode I-CMOMMT adopte une approche basée sur un champ de force. Ce champ de force comprend une attraction envers les cibles, une répulsion des agents voisins pour éviter les collisions, ainsi qu'une attraction vers une zone de forte oisiveté. Afin de parvenir à un compromis entre l'objectif de suivre les cibles et la couverture de l'environnement, l'expérimentateur spécifie un seuil d'oisiveté. Ce seuil indique à partir de quel niveau d'oisiveté atteint dans l'environnement, un agent doit privilégier la patrouille plutôt que l'observation des cibles.

La coordination entre les agents est enrichie, en plus du champ potentiel, par deux mécanismes permettant de limiter le risque que les agents ne décident de visiter le même endroit. Le premier mécanisme consiste à ce que les agents sélectionnent de manière stochastique le lieu à visiter, parmi les N cellules ayant la plus forte oisiveté régionale. Le second mécanisme repose sur un système de priorité, où chaque agent partage le lieu qu'il souhaite visiter. En cas de conflit, l'agent ayant l'identifiant le plus fort prend la priorité de la visite.

La méthode I-CMOMMT est comparée à deux autres méthodes de patrouille, le CR et le CC, adaptées au formalisme du POP. Elle est également comparée à une méthode axée sur l'observation des cibles, le A-CMOMMT, ainsi qu'à une méthode de déplacement aléatoire. Expérimentalement, les résultats montrent que la méthode I-CMOMMT offre un compromis entre l'observation moyenne des cibles, et l'oisiveté moyenne et maximale régionale de l'environnement.

La méthode du I-CMOMMT considère qu'une cellule est intéressante à visiter uniquement si cette dernière possède une valeur d'oisiveté supérieure aux autres cellules. Cependant, des améliorations pourraient être apportées à cette méthode en redéfinissant l'intérêt accordé aux cellules. Par exemple, en prenant en compte non seulement l'oisiveté de la cellule elle-même, mais aussi la valeur d'oisiveté des cellules environnantes ou la distance entre la cellule et la position de l'agent. Une réflexion approfondie sur ces aspects est entreprise dans la prochaine section.

3.3 Identifier les zones à patrouiller : La génération de points d'intérêt dynamiques

Le formalisme du problème de l'observation appuyée par la patrouille permet d'associer l'objectif du suivi des cibles avec celle de la patrouille de l'environnement. Pour réaliser les tâches de patrouille avec un déplacement continu des agents, ces derniers disposent d'une carte d'oisiveté représentant le niveau d'oisiveté de chaque lieu de l'environnement (cf. section 3.1.1).

Dans le cadre d'un système multi-agents, distribuer la visite des lieux entre les agents permet de répondre plus efficacement à la problématique de la patrouille. Cependant, cette distribution nécessite d'identifier en amont les lieux ayant un intérêt à être visités, que nous nommons **points d'intérêt (PI)**. Cet intérêt peut être suscité, par exemple, par une oisiveté forte ou une proximité du lieu avec l'agent. Dans le cas d'une représentation sous forme de graphe, chaque lieu à visiter est représenté par un nœud. Ainsi, la position des points d'intérêt est dite prédéfinie, car restreinte par l'emplacement de ces nœuds. Nous nommons ces points en anglais *Predefined Interest Points* (PIP). Cependant, dans le cadre du POP, les lieux à visiter sont représentés par des cellules. Or, lorsqu'un agent se positionne sur une cellule, ce dernier observe également toutes les cellules environnantes contenues au sein de sa surface d'observation. Nous nommons ainsi les **points d'intérêt dynamiques**, ou en anglais *Dynamic Interest Points* (DIP), les lieux d'intérêt dont la localisation géographique s'adapte aux valeurs de la carte d'oisiveté.

L'objectif de cette section est de fournir aux méthodes cherchant à résoudre le POP, un outil permettant d'identifier les lieux intéressants à visiter au sein de l'environnement. Ces lieux sont obtenus grâce à des algorithmes de **génération de points d'intérêt dynamiques**. L'utilisation de l'outil est d'autant plus utile dans le cadre d'un nombre important de cellules à traiter. En effet, la carte d'oisiveté est issue d'une discrétisation de l'environnement. Dans le cas d'une forte discrétisation, ou d'un environnement relativement grand, le nombre de cellules constituant la carte d'oisiveté peut être important.

Dans un premier temps, nous élaborons un lexique afin d'expliciter les termes et les concepts utilisés au sein des algorithmes de génération de points d'intérêt. Dans un second, nous définissons les objectifs que les points d'intérêt dynamiques doivent atteindre et nous proposons trois algorithmes pour leur génération. Par la suite, l'efficacité des algorithmes est éprouvée selon plusieurs critères d'évaluation, tels que la durée d'exécution. Enfin,

le positionnement des points d'intérêt dynamiques est comparé face aux points d'intérêt prédéfinis sur un environnement de référence.

3.3.1 Définition des concepts pour les algorithmes de génération de points d'intérêt

Dans l'objectif de détailler les algorithmes de génération des points d'intérêt dynamiques, nous proposons de définir certains concepts via un lexique spécifique. De cette manière, la compréhension du fonctionnement et des enjeux de la génération des points d'intérêt sera plus accessible.

La génération des points d'intérêt s'inscrit au sein du formalisme du POP, reprenant ainsi les concepts de carte d'oisiveté, constitué de cellules notées $c_{x,y}$, possédant une oisiveté notée $i_{x,y}$. Nous décrivons le **voisinage** $\nu(c_{a,b})$ d'une cellule $c_{a,b}$, comme l'ensemble des cellules qui seraient contenues au sein du rayon d'observation d'un agent s'il était placé en cellule $c_{a,b}$. Le concept de voisinage est illustré au sein de la figure 3.20. Dans le contexte d'une surface rectangulaire, nous obtenons la formulation suivante :

$$\begin{aligned}
&\forall x \in [max(0; a - r_d); min(D_m; a + r_d)], \\
&\forall y \in [max(0; b - r_d); min(D_n; a + r_d)] : \\
&c_{x,y} \in \nu(c_{a,b}))
\end{aligned}$$

Par ailleurs, nous nommons **oisiveté surfacique** $\hat{i}(c)$ d'une cellule la moyenne des oisivetés des cellules au sein de son voisinage. Soit N le nombre de cellules au sein d'un voisinage :

$$\hat{i}(c) = \frac{1}{N} \sum_{c_{a,b} \in \nu(c)} i_{a,b}$$

Par la suite, nous spécifions qu'une cellule $c_{a,b}$ est dite de **domination forte** si et seulement s'il n'existe aucune cellule dans son voisinage $\nu(c_{a,b})$ avec une oisiveté plus grande :

$$\nexists c_{x,y} \in \nu(c_{a,b}) \text{ tel que } i_{x,y} > i_{a,b}$$

Sinon, une cellule $c_{a,b}$ est dite de **domination faible** si et seulement s'il n'existe aucune cellule dans son voisinage $\nu(c_{a,b})$ avec une oisiveté plus grande, à l'exception des cellules déjà dominées au sein de leur voisinage respectif. Ainsi, nous définissons qu'une cellule de domination faible ou forte est un **point d'intérêt**.

Les algorithmes de génération de point d'intérêt dynamiques reposent également sur le concept de cellule non dominée par un point d'intérêt, que nous nommons **cellule libre**. Ainsi, nous définissons un **voisinage libre** $v'(c_{a,b})$ comme l'ensemble des cellules libres présent au sein d'un voisinage $v(c_{a,b})$. Une carte d'oisiveté M est dite complètement **couverte** si chaque cellule est soit une cellule dominée, soit est un point d'intérêt. Par conséquent, il n'existe plus aucune cellule libre au sein de l'environnement discrétisé. Enfin, la **redondance** se définit comme étant le pourcentage de cellule ayant plus d'un point d'intérêt dans son voisinage.

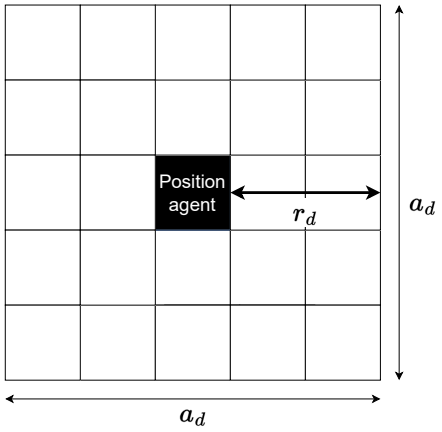


FIGURE 3.20 Illustration du voisinage d'une cellule, dans le cas où la surface d'observation des agents est carrée.

3.3.2 Génération des points d'intérêt

Au sein de cette section, plusieurs algorithmes sont proposés afin de générer des points d'intérêts dynamiques, aussi appelé en anglais *Dynamic Interest Points* (DIP). Ces algorithmes ont plusieurs objectifs à atteindre pour positionner les points d'intérêt dynamiques :

— Les points d'intérêt doivent permettre d'assurer la couverture de l'ensemble de l'environnement. Autrement dit, aucune cellule ne doit être libre.
— Les points d'intérêt doivent être positionnés de manière à maximiser les oisivetés couvertes.
— Les algorithmes tendent à minimiser le temps d'exécution pour la génération des points d'intérêt.

L'ensemble des algorithmes proposés possède la même structure, saisissant en entrée la carte discrétisée M, et retournant en sortie la liste des DIP. La fonction $cellules\_libres(M)$ retourne l'ensemble des cellules libres de la carte M. Tandis que la fonction $max(M)$ retourne

une liste de toutes les cellules possédant la même valeur d'oisiveté maximale au sein de la carte M.

Algorithme 1 : Itération parmi les cellules libres - *Iteration among free cells* (IFC)

L'algorithme IFC (cf. Algorithme 1) parcourt l'ensemble des cellules libres, les unes après les autres, afin d'identifier si une cellule est dominante dans son voisinage. Si c'est le cas, alors elle est considérée comme un point d'intérêt. Après l'évaluation de toutes les cellules de la carte M (appelée itération), si des cellules sont toujours libres au sein de la carte M, alors une nouvelle itération est effectuée parmi ces mêmes cellules libres afin d'identifier des dominations faibles.

Algorithme 1 Iteration among free cells (IFC)

Entrée: M
Sortie: PI

1: **répéter**
2: **pour chaque** $c_{x,y}$ libre $\in M$ **faire**
3: **si** $c_{x,y}$ domine $v'(c_{x,y})$ **alors**
4: $PI.ajouter(c_{x,y})$
5: **fin si**
6: **fin pour**
7: **tant que** Aucune cellule libre

Algorithme 2 : Unique itération parmi les oisivetés maximales - *Single iteration among highest idleness* (SIHI)

L'algorithme SIHI (cf. Algorithme 2) identifie, parmi l'ensemble des cellules libres de la carte M, celles ayant l'oisiveté maximale. L'algorithme considère qu'une oisiveté maximale sur la carte M est par définition une oisiveté maximale locale, dominant ainsi son voisinage. Si plusieurs cellules possèdent la même oisiveté maximale, alors uniquement la première de la liste est sélectionnée. Cette cellule est automatiquement considérée comme un point d'intérêt. Le processus est répété jusqu'à ce qu'il n'y ait plus de cellule libre.

Algorithme 2 Single iteration among highest idleness (SIHI)

Entrée: M

Sortie: PI

1: **répéter**
2: $M' \leftarrow cellules\_libres(M)$
3: $liste_{max} \leftarrow max(M')$
4: $c_{x,y} \leftarrow liste_{max}(1)$ ▷ Sélection du premier élément de la liste
5: $PI.ajouter(c_{x,y})$
6: **tant que** Aucune cellule libre

Algorithme 3 : Double itération parmi les oisivetés maximales - *Double iterations among highest idleness* (DIHI)

L'algorithme DIHI (cf. Algorithme 3) est très proche du fonctionnement du précédent algorithme SIHI. Cependant, dans le cas où plusieurs cellules possèdent la même oisiveté maximale, alors l'algorithme boucle parmi cette liste et vérifie que chaque cellule n'est pas dominée avant de la considérer comme un point d'intérêt.

Algorithme 3 Double iterations among highest idleness (DIHI)

Entrée: M

Sortie: PI

1: **répéter**
2: $M' \leftarrow cellules\_libres(M)$
3: $liste_{max} \leftarrow max(M')$
4: **pour chaque** $c_{x,y}$ libre $\in liste_{max}$ **faire**
5: $PI.ajouter(c_{x,y})$
6: **fin pour**
7: **tant que** Aucune cellule libre

Preuve de couverture

L'ensemble des algorithmes bouclent tant que des cellules sont libres, c'est-à-dire qu'elles ne sont pas encore dominées par un point d'intérêt au sein de leur voisinage. Cependant, il existe nécessairement au moins une cellule avec une domination, forte ou faible, dans le voisinage d'une cellule libre. Par conséquent, les algorithmes garantissent une couverture de l'ensemble de l'environnement par les points d'intérêts. Chaque cellule est soit dominée, soit

est un point d'intérêt. Les expériences menées au sein de la section suivante confirment cette logique.

Exemple de fonctionnement de l'algorithme IFC

La figure 3.21 illustre le fonctionnement de l'algorithme IFC, en explicitant les différentes étapes. La carte d'oisiveté est d'une taille 7×7 cellules. Les valeurs d'oisivetés sont aléatoirement choisies entre 0 et 100, comme représentées par la figure 3.21a. La surface d'observation a une dimension 5×5 cellules, ce qui est l'équivalent d'un rayon d'observation de 2 cellules.

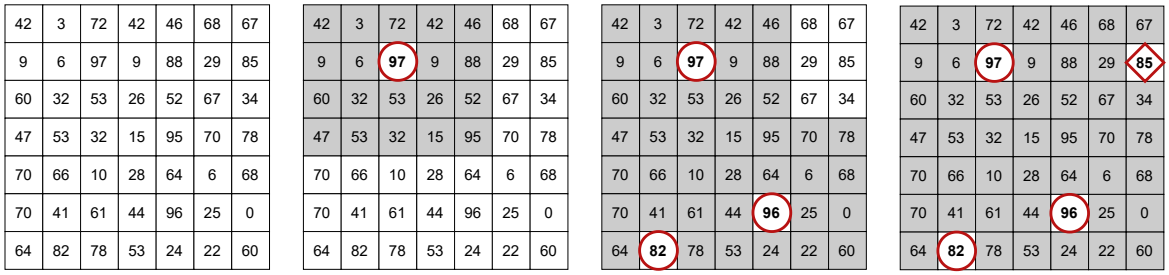
Dans l'objectif d'identifier les points d'intérêt, l'algorithme effectue une évaluation séquentielle de toutes les cellules, en commençant par celle en haut à gauche et en progressant vers celle en bas à droite. Par exemple, la première cellule évaluée possède une valeur d'oisiveté de 42, mais elle est dominée par une cellule voisine ayant une oisiveté de 97. En conséquence, la première cellule est considérée comme étant "dominée".

Au fur et à mesure que l'algorithme parcourt les cellules, la première cellule identifiée comme étant dominante possède une valeur d'oisiveté de 97, comme illustré à l'étape 3.21b. Par conséquent, elle est enregistrée en tant que point d'intérêt, et l'ensemble de son voisinage est considéré comme étant "dominé".

À l'étape 3.21c, une première évaluation de toutes les cellules est effectuée, permettant d'identifier l'ensemble des cellules fortement dominantes. Cependant, certaines cellules restent encore non attribuées, nécessitant ainsi une nouvelle itération spécifique pour les cellules libres. Par exemple, la cellule possédant une oisiveté de 85 n'est pas considérée comme une domination forte, car elle est voisine d'une cellule avec une oisiveté de 95. Cependant, cette dernière est déjà dominée et n'est donc pas un point d'intérêt. Toutefois, lors de la seconde itération, l'algorithme ne considère que les cellules libres, ce qui permet d'identifier la cellule d'oisiveté 85 comme une domination faible et donc un point d'intérêt. Une fois l'algorithme IFC terminé, toutes les cellules ont soit un point d'intérêt dans leur voisinage, soit elles sont elles-mêmes un point d'intérêt. Cela signifie que chaque partie de l'environnement est couverte, et ce constat est observable à l'étape finale 3.21d.

3.3.3 Expérimentations

Cette section expérimentale permet d'éprouver les trois algorithmes précédemment proposés au regard des objectifs de la génération des points intérêt, c'est-à-dire la rapidité



(a) Carte avec des oisivetés aléatoires comprise entre 0 et 100 (7×7 cellules).

(b) Première domination trouvée, considérée comme un PI.

(c) Toutes les dominations fortes sont identifiées, mais certaines cellules restent libres.

(d) L'environnement est complètement couvert par les PI, en identifiant une domination faible.

FIGURE 3.21 Illustration des étapes de l'algorithme IFC. Sont représentées en blanc les cellules libres, en gris les cellules dominées, par un losange les cellules à domination faible, et par un cercle les cellules à domination forte.

d'exécution des algorithmes, la vérification de la couverture de l'environnement et la maximisation des oisivetés couvert par les points d'intérêt dynamiques (cf. section précédente 3.3.2).

Dans un premier temps, au sein de la partie 3.3.3, les algorithmes sont comparés au regard de leur temps d'exécution. Dans un second temps, une solution est étudiée afin de maximiser l'ensemble des oisivetés couvertes par un point d'intérêt (appelé aussi oisiveté surfacique), en opposition à la maximisation de l'oisiveté de la cellule (appelé aussi oisiveté cellulaire) où est placé le point d'intérêt. Par la suite, les points d'intérêt dynamiques sont comparés aux points d'intérêt statiques sur un environnement de référence. Les métriques de comparaison sont la couverture de l'environnement et l'oisiveté surfacique couverte par les points d'intérêt. Enfin, les points d'intérêt dynamiques générés sont étudiés au regard de la redondance de l'observation. Les algorithmes et les expériences sont disponibles sous le langage MATLAB en accès libre sur GitHub [1].

Temps d'exécution des algorithmes

Dans un premier temps, une comparaison des trois algorithmes proposés au sein de la section 3.3.2 est réalisée en ce qui concerne le temps requis pour générer les points d'intérêt dynamiques. Il est intéressant de souligner que les trois algorithmes positionnent les points d'intérêt exactement aux mêmes endroits, et ce peu importe les paramètres du scénario tels que la taille de l'environnement, sa topographie ou le rayon d'observation des agents.

1. https://github.com/JamyChahal/dynamic_interest_point_generation

La durée moyenne pour la génération des points d'intérêts, exprimée en seconde, est présentée par le tableau 3.2. Plusieurs tailles d'environnement, sous forme carré, ainsi que plusieurs rayons d'observation sont évalués. Pour chaque configuration, la moyenne est issue de 100 expériences, où les oisivetés des cellules sont réinitialisées aléatoirement entre 0 et 100. Les expériences sont menées au sein d'un ordinateur doté d'un processeur i5 9e génération d'une fréquence de 2,40 GHz. Les algorithmes sont développés sous MATLAB 2022a.

TABLE 3.2 Durée moyenne, en seconde, pour la génération des points d'intérêt dynamiques pour chaque algorithme. *uc* désigne l'unité d'une cellule.

Obs. range (*uc*)	Taille env. (*uc*)	IFC (*s*)	SIHI (*s*)	DIHI (*s*)
3	30	1,6e-02	8,2e-04	**7,9e-04**
5	30	1,7e-02	2,8e-04	**2,7e-04**
10	30	1,6e-02	1,3e-04	**1,2e-04**
3	50	7,5e-02	2,6e-03	**2,5e-03**
5	50	8,0e-02	9,9e-04	**9,7e-04**
10	50	9,2e-02	3,4e-04	**3,4e-04**
3	100	1,3e+00	2,8e-02	**2,3e-02**
5	100	1,4e+00	1,0e-02	**9,9e-03**
10	100	1,4e+00	2,7e-03	**2,6e-03**

Mis en évidence en gras, l'algorithme DIHI (algorithme 3) montre la meilleure performance pour générer le plus rapidement les points d'intérêt dynamiques. L'algorithme IFC (algorithme 1) est le plus lent, car il évalue de manière séquentielle les cellules les unes après les autres. Par conséquent, l'étude systématique du voisinage est chronophage. Tandis que les algorithmes SIHI (algorithme 2) et DIHI (algorithme 3) reposent sur le principe qu'un maximum d'oisiveté global est par définition également un maximum d'oisiveté local. Ainsi, ces algorithmes n'ont pas le besoin d'évaluer systématiquement le voisinage de chaque cellule. Par ailleurs, le temps de calcul est d'autant plus réduit grâce à la fonction *max* de Matlab bénéficiant d'une parallélisation multithreading. Enfin, l'algorithme DIHI est légèrement plus efficace que l'algorithme SIHI car le DIHI évalue les cellules ayant la même forte oisiveté une seule fois au sein d'une boucle.

De l'oisiveté d'une cellule à une oisiveté surfacique

Lorsqu'un agent se positionne sur un point d'intérêt, il n'observe pas uniquement la cellule sur laquelle il est positionné (appelé oisiveté cellulaire), mais aussi l'ensemble des

cellules au sein de son voisinage (appelé oisiveté surfacique). En fonction du scénario envisagé, il peut être intéressant de positionner les DIP en vue de maximiser l'oisiveté surfacique. Cependant, les algorithmes de génération de DIP ne prennent en entrée qu'une carte d'oisiveté M afin de positionner les DIP en fonction de l'oisiveté cellulaire.

Notre proposition afin de maximiser l'oisiveté surfacique est de filtrer la carte originale M en entrée par une convolution. Cette convolution est réalisée par un filtre moyen ω, ayant la même taille que la surface d'observation d'un agent. Dans le cas où la surface d'observation est un carré de côté a_d, alors le noyau est défini de la manière suivante :

$$\omega = \frac{1}{a_d^2} A(a_d, a_d) \tag{3.7}$$

Avec A une matrice carrée de longueur et largeur a_d, composée uniquement de 1.

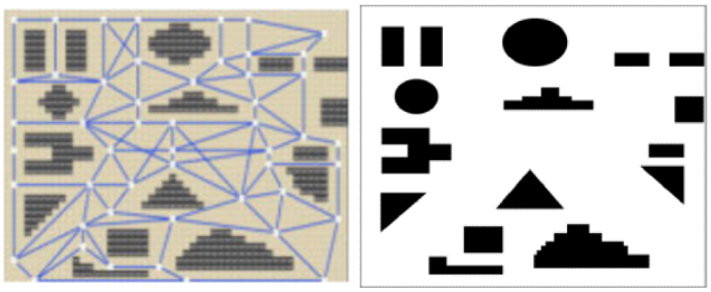
TABLE 3.3 Évaluation de la moyenne d'oisiveté $i(c)$ et de la surface d'oisiveté $\hat{i}(c)$, avec et sans filtre de la carte d'oisiveté, nommés respectivement M_f et M. *uc* désigne l'unité d'une cellule.

Rayon d'obs. (uc)	Taille d'env. (uc)	Moy. $(i(c))$		Moy. $(\hat{i}(c))$	
		M	M_f	M	M_f
3	30	**84,8**	41,7	49,1	**52,9**
5	30	**91,6**	43,2	49,6	**51,6**
10	30	**96,5**	44,9	49,9	**50,5**
3	50	**85,0**	41,5	49,0	**53,1**
5	50	**91,8**	43,2	49,6	**51,9**
10	50	**96,7**	45,5	49,9	**50,8**
3	100	**85,2**	41,5	49,0	**53,3**
5	100	**92,0**	43,2	49,6	**52,2**
10	100	**96,9**	45,5	49,9	**51,1**

Le tableau 3.3 présente la moyenne d'oisiveté cellulaire $i(c)$ et la moyenne de l'oisiveté surfacique $\hat{i}(c)$ avec une carte originale M et une carte filtrée M_f. Plusieurs configurations sont étudiées, avec différentes tailles d'observation et différentes tailles d'environnement carré. Les résultats sont obtenus à partir de 1000 évaluations pour chaque configuration, où les cellules prennent des valeurs d'oisiveté aléatoires entre 0 et 100. Comme nous pouvons le constater, l'utilisation d'une carte originale M tend naturellement à maximiser les oisivetés cellulaires, cependant l'utilisation d'une carte filtrée M_p en entrée d'algorithme tend à maximiser les oisivetés surfaciques. Les prochaines expérimentations ne considéreront que les oisivetés surfaciques comme métrique d'évaluation.

Comparaison entre les points d'intérêt prédéfinis et les points d'intérêt dynamiques

Plusieurs environnements de référence sous forme de graphe ont été présentés par les travaux de ALMEIDA et al. [5] afin d'évaluer les méthodes de patrouille (cf. section 2.2.3). Ces environnements présentent chacune une topologie particulière, que ce soit sous forme circulaire, en grille ou en étoile.



(a) Environnement *map A*. (b) Environnement discretisé

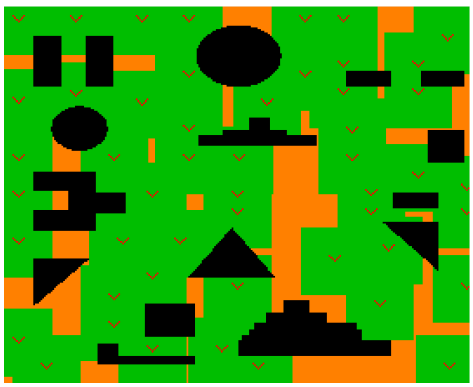
FIGURE 3.22 Génération d'un environnement discrétisé, inspiré de l'environnement *map A* [34].

Pour illustrer la différence de performance entre les points d'intérêt dynamiques et les points d'intérêt prédéfinis, nous sélectionnons le premier environnement de référence, c'est-à-dire la *map A* représentée par la figure 3.22a. La figure 3.22b illustre la discrétisation de la *map A* en 802×651 cellules, où les obstacles sont représentés par des cellules noires. Cependant, l'environnement *map A* ne possède aucune dimension concrète. Par conséquent, une cellule y représente une unité de surface arbitraire.

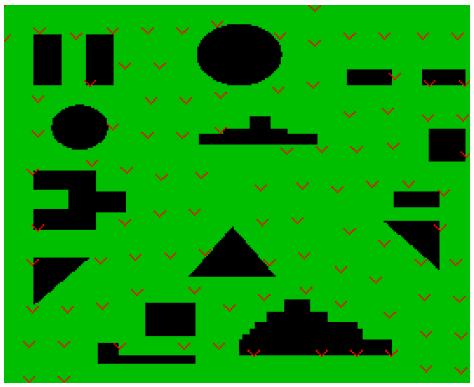
La figure 3.23 représente le positionnement des points d'intérêt dans un cas prédéfini (figure 3.23a) et dynamique (figure 3.23b et 3.23c), pour un rayon d'observation fixe de 20 cellules. Nous faisons l'hypothèse que les agents ont besoin de s'arrêter sur un lieu, et ne peuvent pas être en mouvement, pour observer et analyser l'environnement à l'aide de leurs capteurs. Pour la génération des DIP, deux scénarios sont étudiés :

— Soit les points d'intérêt doivent être placés en dehors des obstacles, car ces zones sont inatteignables, comme représentés par la figure 3.23b
— Soit les points d'intérêt peuvent être placés par-dessus les obstacles, comme présenté par la figure 3.23c. Ce scénario correspond, par exemple, à l'utilisation de drones comme agents, pouvant survoler des obstacles au sol.

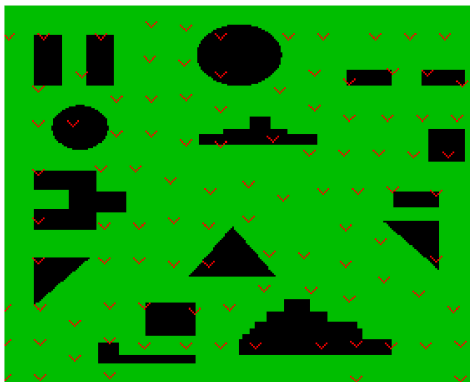
Au sein de l'exemple de la figure 3.23, les points d'intérêt prédéfinis (PIP) ne couvrent pas l'ensemble de l'environnement, comme matérialisé par les cellules oranges de la figure 3.23a.



(a) Zone couverte par les points d'intérêt prédéfinis.



(b) Zone couverte par les points d'intérêt dynamiques.



(c) Zone couverte par les points d'intérêt dynamiques au-dessus des obstacles.

FIGURE 3.23 Illustration des zones couvertes (en vert) par un point d'intérêt (V rouge) avec un rayon d'observation de 20 cellules. Les zones non couvertes sont représentées en orange.

Par conséquent, dans le cas du problème du POP, des cibles intelligentes pourraient se cacher indéfiniment au sein de ces zones oranges sans la crainte d'être observée. Tandis que la génération des points d'intérêt dynamiques permet d'assurer la couverture de l'environnement, où l'ensemble des cellules sont en vert.

Ce constat de couverture complète se confirme par le tableau 3.4, représentant la comparaison entre les PIP et les DIP vis-à-vis de la couverture de l'environnement *map A*, ainsi que le nombre de points d'intérêt générés, pour différents rayons d'observation. Le tableau présente la moyenne de 1000 expériences par configuration, avec des valeurs d'oisiveté attribuées aléatoirement au sein de la carte.

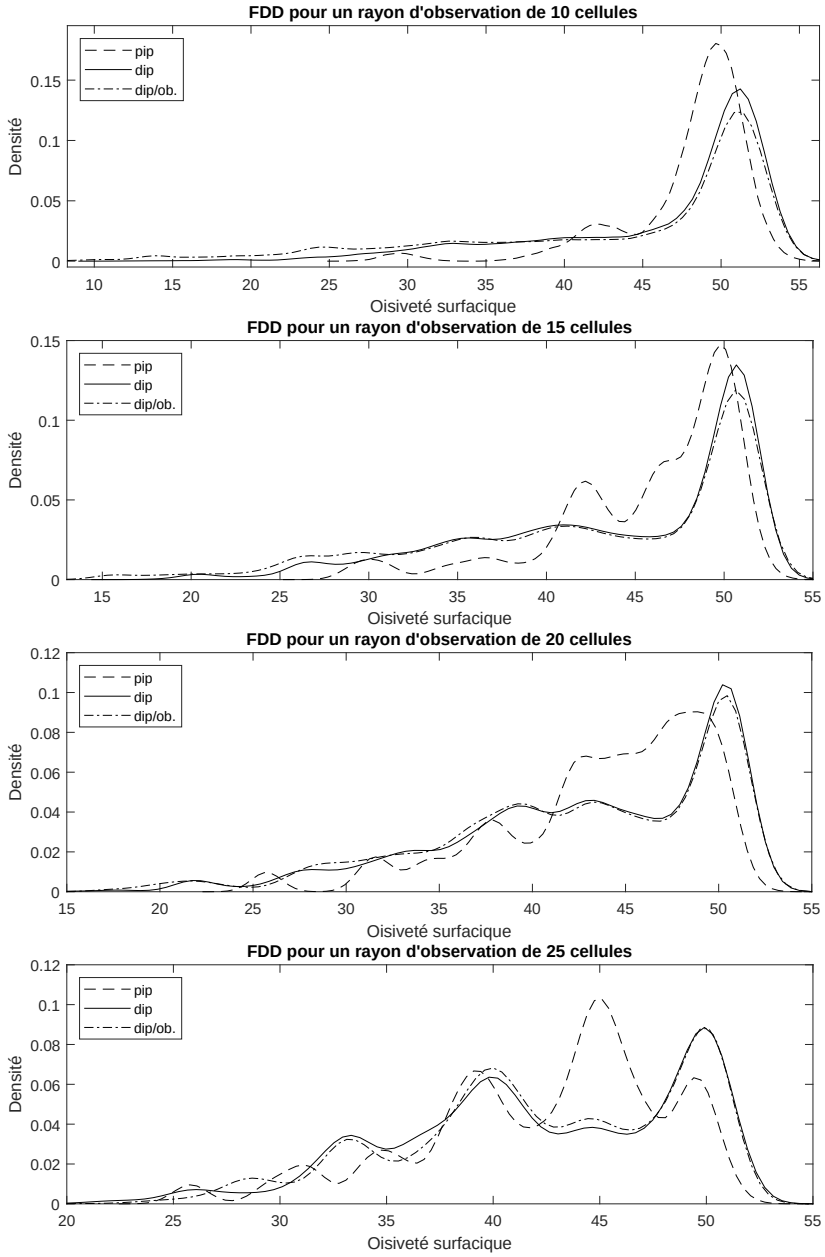
La figure 3.24 représente la distribution des oisivetés surfaciques pour les trois types de points d'intérêt : les points d'intérêt prédéfinis (PIP), les points d'intérêt dynamiques (DIP), et les points d'intérêts dynamiques pouvant se positionner sur les obstacles (DIP/obs).

TABLE 3.4 Couverture de l'environnement par des points d'intérêt prédéfinis (PIP) et des points d'intérêt dynamiques (DIP) pour plusieurs rayons d'observation. *uc* désigne l'unité d'une cellule.

Rayon d'obs. (*uc*)	PIP		DIP en dehors des obstacles		DIP pouvant se positionner sur les obstacles	
	Cellules non couvertes	Nombre de PIP	Cell. non couvertes	Nombre de DIP (moy.)	Cell. non couvertes	Nombre de DIP (moy.)
10	68,13%	50	0%	354,68	0%	399,87
15	39,30%	50	0%	169,70	0%	182,30
20	18,46%	50	0%	98,30	0%	102,32
25	7,43%	50	0%	65,88	0%	66,20
30	2,30%	50	0%	46,65	0%	46,75
35	0,89%	50	0%	33,14	0%	33,47

Pour chaque rayon d'observation étudiée, 1000 expérimentations ont permis de générer de manière empirique la fonction de densité (FDD). Ainsi, l'axe des abscisses illustre l'oisiveté surfacique, à maximiser, et l'axe des ordonnées représente la densité de point d'intérêt possédant une même valeur d'oisiveté surfacique. Au sein des graphiques, les PIP sont en pointillés, les DIP en ligne continue et les DIP/obs en point tiret.

Nous examinons ces graphiques en comparant la disposition des pics, qui présentent une similitude avec des courbes de Gauss, entre les diverses approches. Ainsi, nous pouvons constater que la distribution des oisivetés surfaciques est plus concentrée vers des valeurs élevées pour les DIP comparé aux PIP. Cela signifie que, empiriquement, les DIP sont positionnés de manière à couvrir des oisivetés plus importantes que les PIP.


FDD pour un rayon d'observation de 10 cellules
pip
dip
dip/ob.
Densité
Oisiveté surfacique
FDD pour un rayon d'observation de 15 cellules
pip
dip
dip/ob.
Densité
Oisiveté surfacique
FDD pour un rayon d'observation de 20 cellules
pip
dip
dip/ob.
Densité
Oisiveté surfacique
FDD pour un rayon d'observation de 25 cellules
pip
dip
dip/ob.
Densité
Oisiveté surfacique

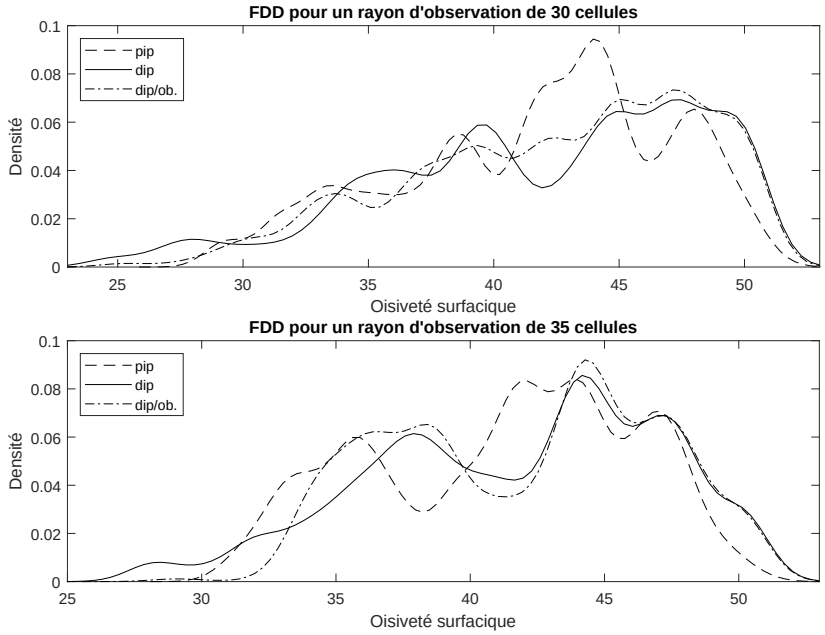


FIGURE 3.24 Fonction de densité (FDD) de l'oisiveté surfacique des points d'intérêt pour plusieurs rayons d'observation. La notation dip/ob. symbolise les points d'intérêt dynamiques pouvant être placés sur les obstacles.

Nous pouvons par conséquent en déduire que sur l'environnement *map A*, la méthode des points d'intérêt dynamiques permet de maximiser les oisivetés surfaciques, en comparaison avec les points d'intérêt prédéfinis. Cependant, lorsque la surface d'observation devient grande, avec 30 ou 35 cellules de rayon, alors la comparaison devient plus difficile. Ceci s'explique par des surfaces d'observation des points d'intérêt recouvrant plusieurs fois une même zone. Lorsqu'une cellule est recouverte par plus d'un point d'intérêt, nous parlons alors de redondance. Cette notion est étudiée au sein de la section suivante.

Étude de la redondance

La redondance se définit comme le pourcentage de cellules ayant plus d'un point d'intérêt au sein de son voisinage. Les algorithmes de génération de points d'intérêt se focalisent uniquement sur la couverture entière de l'environnement et la maximisation des oisivetés, et non de la redondance. Cependant, il est intéressant d'étudier cette métrique, car une redondance forte signifie que le positionnement des points d'intérêt peut être amélioré, car les zones couvertes s'entrecroisent.

La redondance moyenne pour plusieurs tailles d'environnement et rayons d'observation est présentée au sein du tableau 3.5. Chaque configuration est exécutée 1000 fois. Pour chaque expérience, les cellules prennent une valeur aléatoire entre 0 et 100. Comme nous pouvons

le voir, pour les paramètres considérés, en employant la méthode des DIP, la redondance ne dépasse jamais 1 %, c'est-à-dire qu'au maximum un pourcent des cellules ont plus d'un point d'intérêt dans leurs voisinages.

TABLE 3.5 Moyenne de la redondance pour plusieurs configurations avec des DIP. *uc* désigne l'unité d'une cellule.

Taille env. (*uc*) / Rayon obs. (*uc*)	100	250	500	700
3	0,72%	0,73%	0,73%	0,73%
5	0,79%	0,79%	0,79%	0,79%
10	**1,00%**	**1,00%**	**1,00%**	**1,00%**
30	0,89%	0,86%	0,86%	0,85%
50	0,74%	0,87%	0,87%	0,86%

3.3.4 Conclusion

Au sein de ce chapitre, nous avons étudié la mise en place d'algorithmes permettant d'identifier des zones intéressantes à patrouiller au sein d'une carte d'oisiveté, que l'on nomme les points d'intérêt dynamiques. En opposition à la représentation sous forme de graphe, aussi nommés les points prédéfinis, ces points d'intérêt dynamiques ont pour premier objectif d'être placés afin de maximiser les valeurs d'oisiveté présentes dans leur voisinage. Le second objectif est d'assurer la couverture de l'environnement, autrement dit, de s'assurer que toutes les cellules appartiennent au voisinage d'au moins un point d'intérêt. De plus, comme démontré au sein de la section expérimentation, ces points d'intérêt dynamiques s'adaptent aux différents rayons d'observation et à la topographie de l'environnement.

Grâce à cette solution, les agents disposent ainsi d'un outil permettant d'analyser plus simplement et efficacement une carte d'oisiveté. Lorsque plusieurs agents sont en communication, partageant ainsi la même carte d'oisiveté, cet outil peut également servir de support pour coordonner la répartition des lieux à visiter.

Trois algorithmes ont été proposés et évalués selon le critère de la rapidité de génération des points d'intérêt, sachant que ces algorithmes génèrent le même résultat vis-à-vis du positionnement des points d'intérêt. Par la suite, nous avons démontré que les points d'intérêt dynamiques permettent d'identifier plus efficacement les zones d'oisivetés fortes que les points d'intérêt prédéfinis qui sont fixes dans l'environnement, et permettent aussi

d'assurer la couverture entière de l'environnement. Enfin, nous avons étudié la redondance des points d'intérêt dynamiques, c'est-à-dire la tendance à générer des zones de couverture se superposant. La redondance est un aspect qui tend à être minimisé, et est donc un critère qui fera l'objet de recherche future.

Chapitre 4

Contribution II : Des méthodes d'apprentissage pour résoudre le POP

Ce chapitre explore l'étude et la mise en œuvre d'algorithmes d'apprentissage multi-agents pour faciliter la coordination distribuée entre les agents. Dans un premier temps, la méthode du Force Field Reinforcement Learning (FFRL) est développée par le biais de l'apprentissage par renforcement, visant spécifiquement à résoudre le problème de l'observation, en prenant en considération des cibles aléatoires et difficiles à suivre. Ensuite, la méthode Force Field MultiAgent Reinforcement Learning (F2MARL), dans la continuité du FFRL, intègre également la problématique de la patrouille pour répondre au formalisme du POP. Enfin, le chapitre présente la méthode du MultiAgent Learning using Optimized Strategy (MALOS), qui répond au formalisme du POP à travers la mise en place d'un apprentissage supervisé.

4.1 Force Field Reinforcement Learning (FFRL) : S'adapter au comportement des cibles

Dans cette section, nous examinons en détail le fonctionnement de la méthode intitulée *Force Field Reinforcement Learning* (FFRL). Cette approche s'inspire du concept de déplacement à l'aide de champs de force, en mettant l'accent sur la maximisation de la détection des cibles grâce à l'apprentissage par renforcement. En outre, nous étudions deux comportements différents dans le déplacement des cibles. Ainsi, les cibles peuvent soit suivre un mouvement aléatoire, soit adopter une stratégie évasive vis-à-vis des agents. L'entraînement de la méthode F2MARL est disponible en open-source sur Github [1].

1. https://github.com/JamyChahal/FFRL_F2MARL

4.1.1 L'approche

La méthode du *Force Field Reinforcement Learning* (FFRL) s'inspire du champ de force développé au sein du A-CMOMMT [92] pour répondre au problème de l'observation. Chaque agent a_i, au temps t, subit une force provenant de sa politique F^{π} et une force de protection F^{P} de la manière suivante :

$$F(a_i,t) = F^{\pi}(a_i,t) + F^{P}(a_i,t) \tag{4.1}$$

La force de protection est employée pour prévenir l'agent de rentrer en collision avec les autres agents, mais aussi de sortir de l'environnement. Cette force est composée d'une force de répulsion f^r de la part des autres agents k au sein de son rayon de sécurité, et d'une force de répulsion f^b lorsque l'agent est trop proche des limites de l'environnement :

$$F^{P}(a_i,t) = f^b + \sum_{k=1}^{m} f^r_{i,k} \tag{4.2}$$

Avec m le nombre d'agents. Lorsque la distance entre un agent et les autres agents, ou les limites de l'environnement, est inférieure à une distance de sécurité (DS), les forces de répulsion entrent en jeu. À partir de ce moment, les forces f^b et $f^r_{i,k}$ augmentent de manière linéaire jusqu'à atteindre une magnitude maximale de 1 lorsque la distance devient inférieure ou égale à une distance dangereuse (DD). De cette manière, en plus du comportement obtenu lors de l'apprentissage, les forces de protection incitent les agents à éviter des actions critiques qui pourraient mettre en péril leur intégrité ou celle des autres agents. Par conséquent, la méthode FFRL permet de réaliser des expérimentations plus sûres dans le cadre d'un système multi-robots et multi-drones.

4.1.2 Définition de l'environnement

La résolution du problème de l'observation grâce à l'apprentissage par renforcement nécessite de développer un environnement dédié, qui inclut la définition de la perception des agents, les actions possibles et la définition des récompenses. Cet environnement, nommé *pop_env*, est développé grâce à la librairie dédiée à l'apprentissage multi-agents PettingZoo [120], qui s'inspire de la structure de la librairie Gym [21]. La figure 4.1 illustre une représentation visuelle de cet environnement, avec la présence des agents, des cibles, des rayons d'observation et de communication. Son développement est inspiré par l'environne-

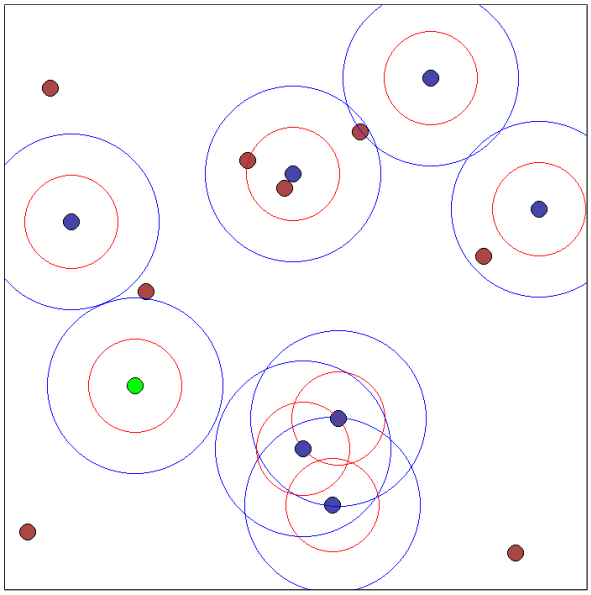


FIGURE 4.1 Représentation de l'environnement *pop_env* au sein de PettingZoo. Les agents sont illustrés en bleu (à l'exception d'un agent en vert), les cibles en rouge, les rayons d'observation par un cercle rouge et les rayons de communication par un cercle bleu.

ment *simple_adversary*, créé par LOWE et al. [77] au sein de la représentation *Multi-agent Particle-World Environment* (MPE) [2].

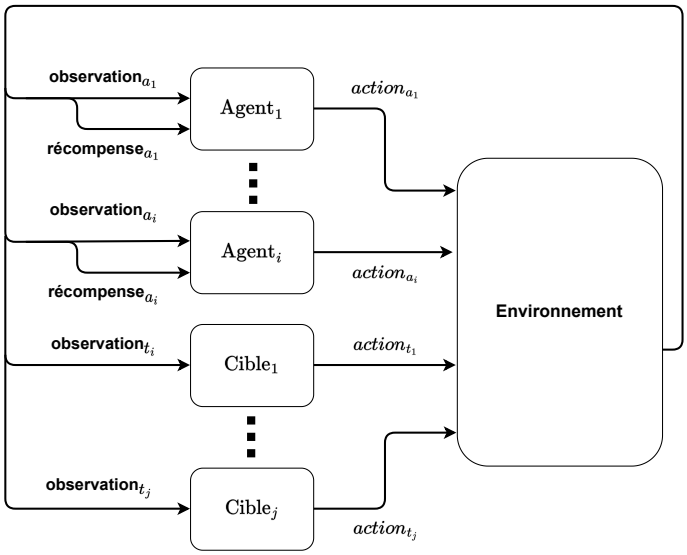


FIGURE 4.2 Diagramme d'interaction entre l'environnement et les agents ainsi que les cibles.

2. https://pettingzoo.farama.org/environments/mpe/

La figure 4.2 illustre les interactions entre les agents et les cibles avec l'environnement dans le cadre de l'apprentissage par renforcement multi-agents. Ces interactions se manifestent sous différentes formes :

— L'espace d'observation, également appelé espace d'état, fait référence à l'ensemble des informations perçues par l'agent sur l'environnement dans lequel il évolue. Ces observations peuvent inclure des données sur l'état de l'agent lui-même, telles que sa position, sa vitesse, son orientation, etc. Elles peuvent également englober les informations perçues par les capteurs de l'agent, comme les flux vidéo, les données de capteurs de proximité, les mesures de température, etc.

— L'espace d'action fait référence à l'ensemble des actions possibles qu'un agent peut choisir à chaque étape de son interaction avec un environnement. L'espace d'action peut prendre deux formes différentes. Il peut être discret, où toutes les actions possibles sont énumérées individuellement, par exemple aller en haut, à gauche, en bas ou à droite. Alternativement, l'espace d'action peut être continu, ce qui signifie qu'il est défini à l'intérieur d'un intervalle opérationnel, tel que la plage de vitesse allant de 0 à 1.

— La récompense, dans le contexte de l'apprentissage par renforcement, est une mesure de l'utilité ou de la valeur qu'un agent obtient en effectuant une action dans son environnement. Sa valeur peut être positive ou négative, si son action mérite une récompense ou une punition.

Espace d'observation

Au sein de l'environnement *pop_env*, nous structurons l'observation comme un vecteur de 18 éléments, composé de :

— La position euclidienne (x_{self}, y_{self}) de l'agent dans les coordonnées de l'environnement.

— La position relative $(\delta x_{a_i}, \delta y_{a_i})$ des quatre plus proches agents au sein du rayon de communication. Les emplacements en trop sont remplacés par des zéros.

— La position relative $(\delta x_{o_j}, \delta y_{o_j})$ des quatre plus proches cibles au sein du rayon d'observation. Les emplacements en trop sont remplacés par des zéros.

Dans le but de faciliter l'apprentissage, il est courant de normaliser les espaces d'observations. Cette normalisation est réalisée en divisant chaque élément du vecteur d'observation par ses valeurs extrêmes (maximum ou minimum). Cela permet de réduire les effets potentiels des caractéristiques spécifiques de l'environnement sur le comportement de l'agent. Ce processus est réalisé de la manière suivante :

— La position euclidienne (x_{self}, y_{self}) est divisée par la taille de l'environnement, ainsi $(x_{self}, y_{self})_n \in [-1,1]^2$.
— Les positions relatives des agents $(\delta x_{a_i}, \delta y_{a_i})$ sont divisées par le rayon de communication, de manière que $(\delta x_{a_i}, \delta y_{a_i})_n \in [-1,1]^2$.
— Les positions relatives des cibles $(\delta x_{o_j}, \delta y_{o_j})$ sont divisées par le rayon de communication, ainsi $(\delta x_{o_j}, \delta y_{o_j})_n \in [-1,1]^2$.

Espace d'action

L'action d'un agent consiste à se déplacer dans l'environnement à chaque pas de temps. Ce déplacement se matérialise par deux composantes de la force F^{π} : (1) La force désirée sur l'axe x et (2) la force désirée sur l'axe y. Ces actions sont continues et comprises dans l'intervalle $[-1;1]$. Par ailleurs, les vitesses des agents sont plafonnées par une vélocité maximale.

Récompense

La fonction récompense est conçue pour maximiser l'observation des cibles, tout en évitant les collisions entre les agents. Elle s'exprime de la manière suivante :

$$R(a_i,t) = R_{obs}(a_i,t) + R_{collision}(a_i,t)$$

La fonction de récompense $R_{collision}(a_i,t)$ incite les agents à éviter toute collision entre eux, afin d'adopter un comportement plus sûr, en plus de la présence de la force de protection F^P. Cette fonction est décrite par la figure 4.3, avec DS la distance de sécurité et DD la distance dangereuse. Ces deux distances sont paramétrées par l'expérimentateur. La fonction de récompense est continue afin de considérer le problème du *credit assignment*, et potentiellement ainsi améliorer l'apprentissage.

La récompense liée à l'observation des cibles est examinée selon deux perspectives, une individuelle et une autre collective :

Récompense individuelle : Chaque agent obtient une récompense égale au nombre de cibles au sein de son rayon d'observation.

Récompense partagée : Chaque agent obtient une récompense pour toutes les cibles au sein de son rayon d'observation, mais aussi pour les cibles observées par les autres agents dans son rayon de communication.

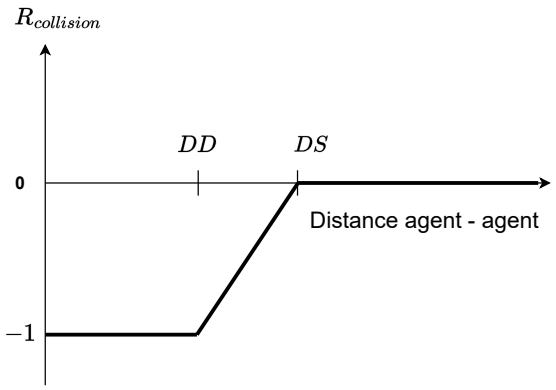


FIGURE 4.3 Fonction de récompense associée à la collision entre agents.

La figure 4.4 illustre la différence des deux fonctions de récompense, selon l'action de l'agent, dans un scénario spécifique. Nous prenons l'hypothèse que les deux agents sont en communication. Dans le cas d'une récompense individuelle, l'agent reçoit une récompense de +1 en observant soit la cible de gauche, soit la cible de droite. Or, dans le cas d'une récompense partagée, l'agent est en communication avec le second agent. Ainsi, en se dirigeant vers la cible à droite, collectivement les agents n'observeraient qu'une seule cible, avec une récompense de +1, tandis que le déplacement vers la gauche permet d'observer collectivement deux cibles, ce qui vaut une récompense +2.

Comportement des cibles

L'environnement considère deux types de comportement pour les cibles :

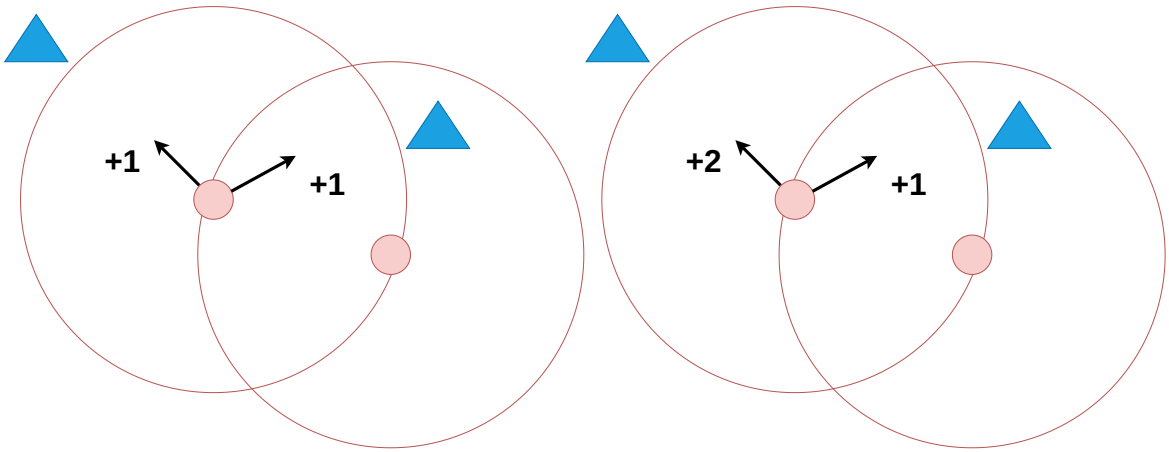
— Les cibles naïves ou aléatoires : La cible sélectionne aléatoirement une position désirée, et va droit dans sa direction, jusqu'à sélectionner une nouvelle position une fois atteinte.

— Les cibles réactives ou évasives : La cible utilise une force répulsive pour aller en direction opposée des agents environnants, appartenant à un rayon de détection. Si aucun agent n'est aperçu, la cible adopte un comportement aléatoire.

4.1.3 Entraînement et résultats

Entraînement du modèle

L'entraînement repose sur le partage de paramètres, également appelé *parameter sharing*. Un seul modèle est entraîné pour apprendre une politique commune à tous les agents homogènes, ce qui permet d'obtenir une politique applicable à différentes échelles. Comme expliqué par TERRY et al. [121], la technique du partage de paramètres est en contraste



(a) Récompense individuelle. (b) Récompense partagée.

FIGURE 4.4 Illustration des deux fonctions de récompense en fonction du déplacement d'un agent, pour un scénario avec deux cibles et deux agents en communication. Les déplacements envisagés d'un agent sont représentés par des flèches noires, les cibles par des triangles bleus, les agents par des ronds rouges et leurs rayons d'observation par des cercles rouges.

avec un apprentissage entièrement indépendant, où chaque agent individuel apprend sa propre politique. Pendant la session d'entraînement, la position initiale de tous les agents et cibles est définie de manière aléatoire afin d'éviter un apprentissage trop spécifique. Cependant, la configuration de l'environnement est fixe. Ainsi, la taille de l'environnement est de $100m \times 100m$, la portée de l'observation est de $5m$ et la portée de la communication est de $10m$.

L'algorithme d'entraînement choisi est le *Proximal Policy Optimization* (PPO) [111], disponible dans la bibliothèque Rllib [74] avec l'implémentation du modèle Acteur-Critique. Le choix du PPO comme algorithme d'apprentissage réside principalement dans sa compatibilité avec des espaces d'actions discrets ou continus, et dans sa capacité à limiter les mises à jour de la politique pour éviter des changements radicaux. De plus, les recherches menées par ANDRYCHOWICZ et al. [8] démontrent que l'algorithme PPO obtient des résultats d'apprentissage prometteurs dans la résolution de divers problèmes complexes, surpassant ainsi d'autres méthodes d'apprentissage. Dans le cadre d'un apprentissage multi-agents, l'*Independent Proximal Policy Optimization* (IPPO) WITT et al. [125] propose une extension naturelle du PPO, afin d'entraîner des agents dans un schéma coopératif, collaboratif ou encore compétitif. La figure 4.5 représente son fonctionnement, où chaque agent est entraîné selon l'observation perçue, alimentant le modèle acteur et critique, afin d'obtenir une action pour interagir avec l'environnement. Nous nous plaçons dans un cadre particulier de l'IPPO, où un seul modèle acteur et un seul modèle critique est utilisé, commun à tous les agents.

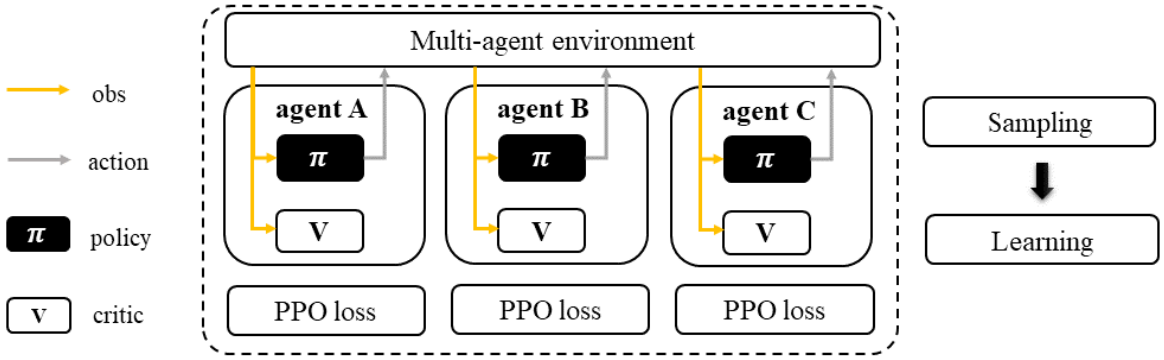


FIGURE 4.5 Illustration des mécanismes de l'architecture IPPO, réalisée par HU et al. [53].

L'algorithme d'apprentissage par renforcement gère le processus d'apprentissage d'une politique en utilisant un ensemble d'hyperparamètres. Le choix d'un ensemble d'hyperparamètres approprié dépend de divers facteurs tels que la conception de l'environnement et la nature du problème à résoudre. Effectuer plusieurs configurations et ajuster manuellement les hyperparamètres peut être chronophage et nécessiter des ressources importantes. Pour sélectionner rapidement un bon ensemble d'hyperparamètres, nous utilisons l'algorithme *Population Based Training algorithm* (PBT) développé par JADERBERG et al. [55]. En définissant une plage de valeurs acceptables pour les hyperparamètres, le PBT entraîne simultanément plusieurs modèles basés sur des configurations aléatoires. Les modèles sont autorisés à exploiter les résultats partiels des autres modèles jugés prometteurs à un moment donné. Enfin, nous sélectionnons le modèle présentant les meilleures performances. La perturbation de la configuration est effectuée tous les 10 épisodes et 6 entraînements sont exécutés simultanément pour chaque type de récompense. Les plages d'hyperparamètres de PPO, ainsi que les valeurs d'hyperparamètres sélectionnées, sont décrites dans le tableau 4.1.

L'architecture du modèle utilisé est un réseau de neurones, avec trois couches cachées contenant 128 neurones. La taille de la couche d'entrée est déterminée par le nombre d'éléments du vecteur d'observation, et la couche de sortie comprend deux neurones pour représenter les composantes x et y de la force. Les figures 4.6 et 4.7 représentent respectivement le résultat des entraînements pour des récompenses individuelles et des récompenses collectives. Les courbes illustrent la récompense moyenne, tandis que les surfaces colorées symbolisent les valeurs maximales et minimales des récompenses. La courbe associée correspond au meilleur entraînement obtenu parmi les différentes populations de l'algorithme PBT. Le modèle final retenu d'un entraînement ne correspond pas à la dernière étape, mais à celle où la récompense moyenne est maximale.

TABLE 4.1 Hyperparamètres pour chaque type de récompense. SGD signifie *Stochastic gradient descent*.

Hyperparamètres	**PBT Plage de valeur**	**Valeur finale Réc. individuelle**	**Valeur finale Réc. partagée**
Horizon temporel pour chaque épisode	Fixe	1 000	1 000
Taille du mini-batch	[128 ;16e3]	2653	6830
Clipping ratio	[0.01 ; 0.5]	0.0344	0.0228
Gamma	[0.9 ; 1.0]	0.9	0.9
Taux d'apprentissage (Learning rate)	[5e-05 ; 1.0]	1e-05	5e-05
Coefficient d'entropie	[0 ;0.1]	0.0342	0.0499
Couches cachées	Fixe	128 x 128 x 128	128 x 128 x 128
Lambda	[0.7 ;1.0]	1	0.9601
Nombre d'itérations SGD	[1 ;30]	1	1
Taille du mini-batch SGD	[128 ; 512]	708	470

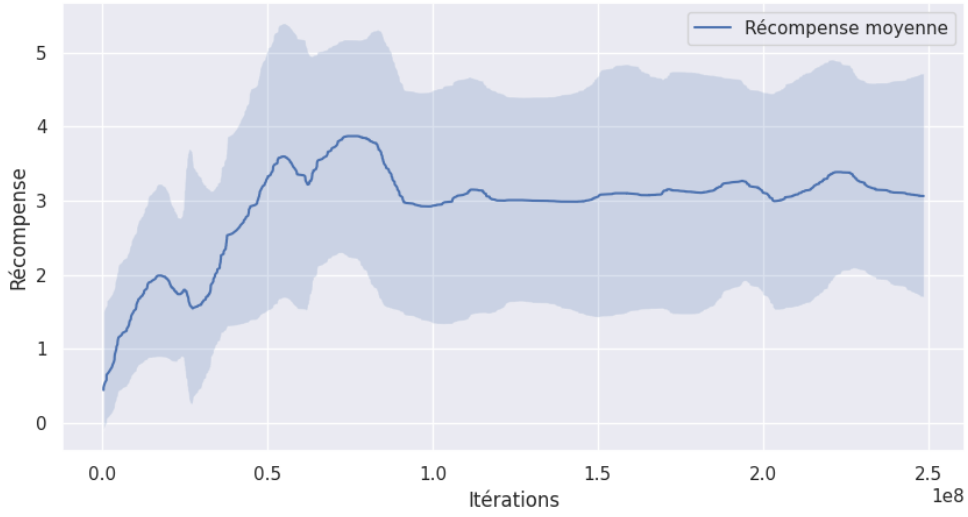


FIGURE 4.6 Résultat du meilleur entraînement avec une récompense invididuelle.

Les détails concernant l'environnement utilisé lors de l'entraînement sont présentés dans la prochaine section expérimentale. Dans cet environnement, huit agents sont déployés et entraînés avec un modèle partagé. De plus, afin de diversifier la nature des cibles, l'environnement est composé de quatre cibles avec un déplacement aléatoire et quatre autres cibles adoptant un comportement évasif.

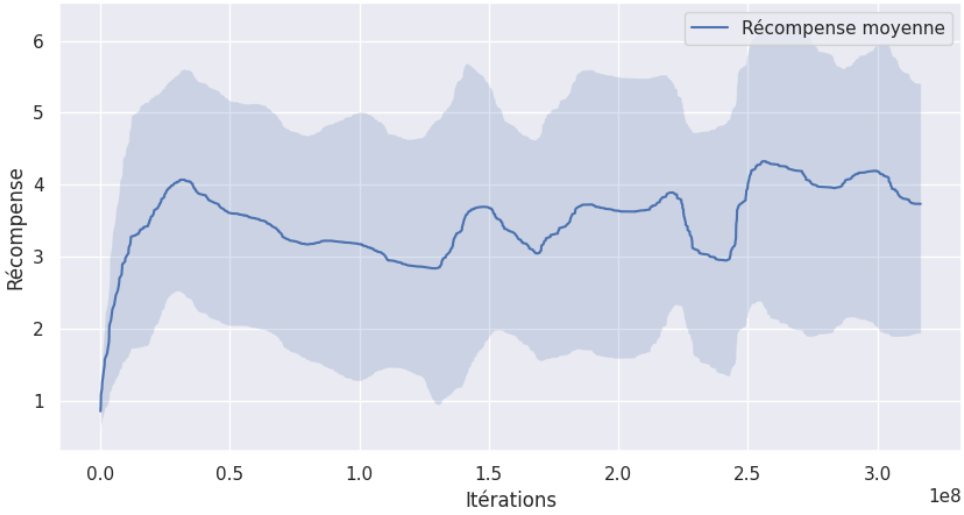


FIGURE 4.7 Résultat du meilleur entraînement avec une récompense partagée.

Résultats

La méthode FFRL est focalisée principalement dans le suivi des cibles, par conséquent, nous la comparons avec d'autres méthodes d'observation : le A-CMOMMT, le I-CMOMMT-0.5 et une méthode aléatoire. Le cadre expérimental proposé consiste à évaluer l'efficacité de toutes les méthodes pour un environnement ayant successivement 2, 4 puis 6 agents, cherchant à observer 3, 5 puis 7 cibles. Les cibles adoptent soit un comportement aléatoire, soit un comportement évasif. Chaque configuration agents/cibles est évaluée 20 fois. La position initiale des agents et des cibles est aléatoire. L'évaluation de performance repose sur la moyenne d'observation (métrique A) et la déviation standard de l'observation, présentées respectivement au sein des figures 4.8 et 4.10 pour des cibles aléatoires, et par les figures 4.9 et 4.10 pour des cibles évasives.

Les caractéristiques de l'environnement expérimental sont similaires à l'environnement d'entraînement. Il est représenté sous forme d'un carré de $100m \times 100m$, où les agents ont un rayon d'observation de $5m$ et un rayon de communication de $10m$. Par ailleurs, les cibles évasives possèdent un rayon de détection de $7m$. La mission a une durée d'une heure, soit $3600s$, et la vitesse des agents est de $2m/s$, tandis que les cibles ont une vitesse de $1m/s$. Enfin, les méthodes A-CMOMMT et I-CMOMMT ont pour paramètres $do_1 = 1m$, $do_2 = 2m$, $do_3 = 5m$ et un rayon de suivi prédictif de $6m$.

Dans un premier temps, nous pouvons constater qu'entre les deux systèmes de récompense pour entraîner la méthode FFRL, les récompenses partagées permettent une meilleure

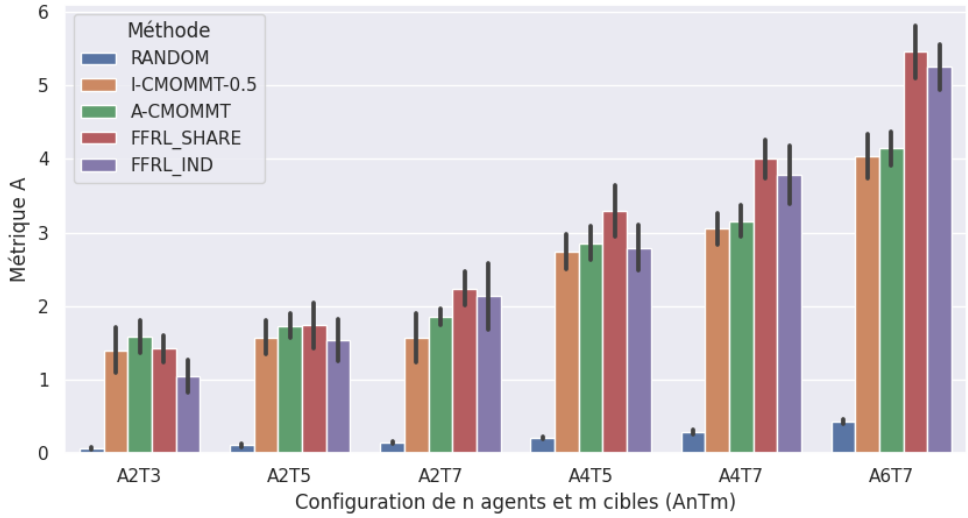


FIGURE 4.8 Performance des différentes méthodes pour l'observation moyenne des cibles aléatoires.

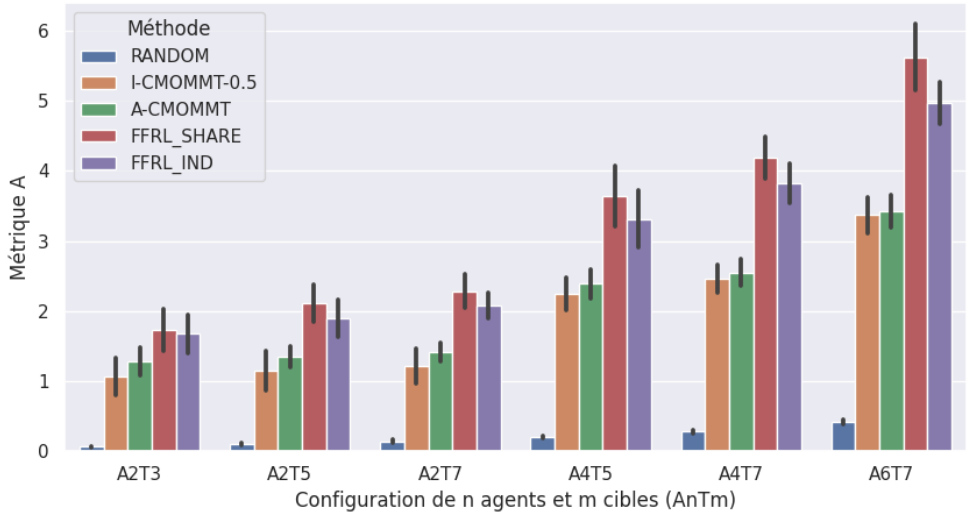


FIGURE 4.9 Performance des différentes méthodes pour l'observation moyenne des cibles évasives.

coordination entre les agents que les récompenses individuelles. Cela se traduit par de meilleures performances en termes d'observation moyenne des cibles (métrique A), mais également une meilleure répartition de l'observation entre les cibles (métrique de déviation standard).

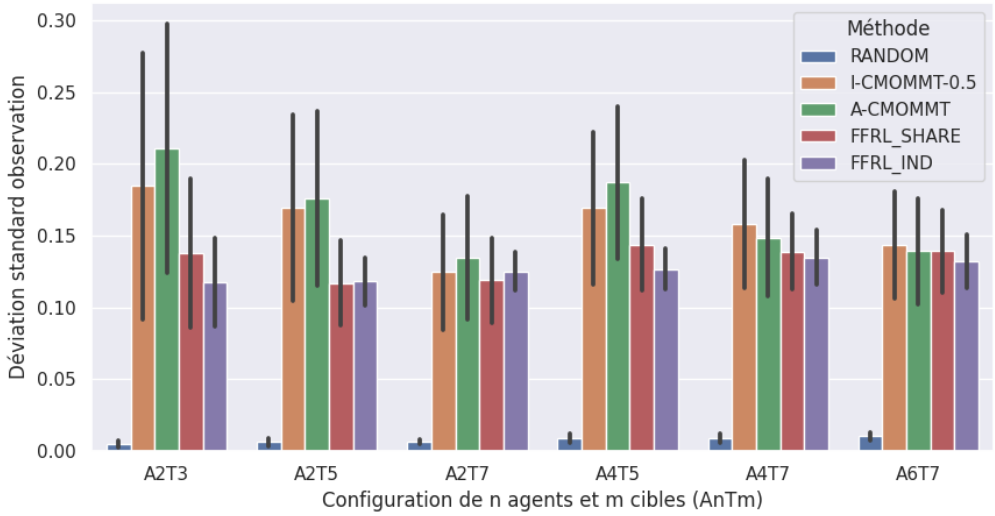


FIGURE 4.10 Déviation standard σ_n de l'observation des cibles au comportement aléatoire.

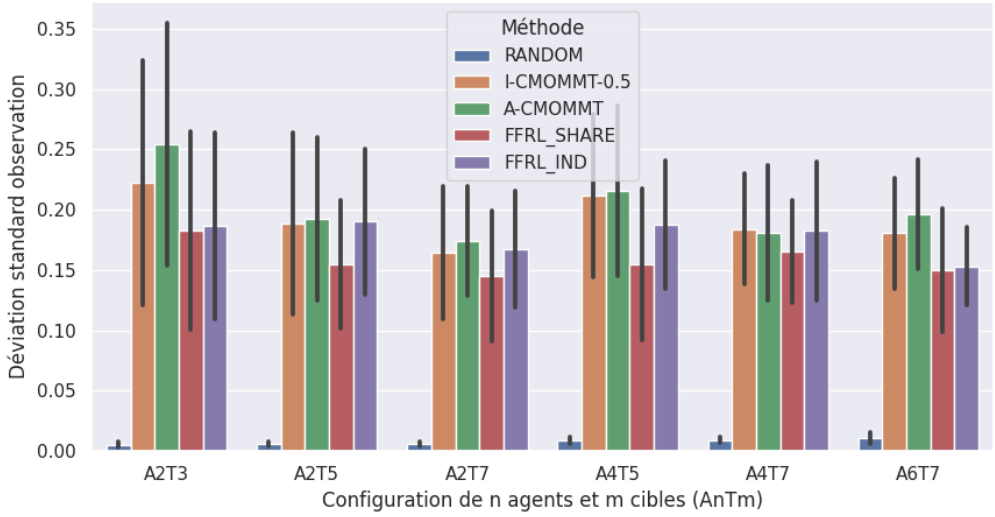


FIGURE 4.11 Déviation standard σ_n de l'observation des cibles au comportement évasif.

En second lieu, il convient de remarquer que les performances du FFRL sont principalement influencées par la densité des agents et des cibles. Lorsque l'environnement contient un nombre réduit d'agents et de cibles, le FFRL affiche des performances comparables, voire inférieures, aux méthodes A-CMOMMT et I-CMOMMT-0.5. Cependant, dans le cas où la densité devient importante, le FFRL se démarque par des performances significatives. Pour expliquer ce phénomène, il est pertinent de mettre en évidence que lorsque la densité

des agents et des cibles est élevée, la répartition des cibles entre les agents devient un défi majeur, en plus du suivi des cibles lui-même. La méthode FFRL a été entraînée avec une forte densité d'agents et de cibles, permettant ainsi au modèle d'être confronté aux problèmes de répartition. Enfin, comme nous pouvons le constater, les méthodes A-CMOMMT et I-CMOMMT présentent logiquement des performances inférieures lorsque les cibles adoptent un comportement évasif. Cela crée un écart plus marqué avec la méthode du FFRL, qui se révèle être plus performante dans ces conditions.

4.1.4 Conclusion

Le développement du *Force Field Reinforcement Learning* (FFRL) a pour objectif de répondre à la problématique de l'observation à travers une coordination distribuée. Et ce plus particulièrement dans le cas où les cibles adoptent un comportement évasif, c'est-à-dire que les cibles se dirigent dans le sens opposé des agents présents au sein d'un rayon de détection. Pour ce faire, les agents partagent un même modèle entraîné par de l'apprentissage par renforcement. L'entraînement utilise l'algorithme *Proximal Policy Optimization* (PPO) [111], dont les hyperparamètres sont obtenus par l'algorithme *Population Based Training algorithm* (PBT) JADERBERG et al. [55]. Nous comparons deux formulations de la récompense : La récompense individuelle incite uniquement l'agent à maximiser le nombre de cibles observées, tandis que la récompense partagée gratifie l'agent du nombre de cibles observées, que ce soit par lui ou par un agent en communication.

Les résultats expérimentaux indiquent que la collaboration entre les agents pour distribuer les cibles se révèle plus performante lorsqu'une récompense partagée est utilisée plutôt qu'une récompense individuelle. De plus, les performances d'observation des cibles obtenues grâce à la méthode FFRL sont évaluées en comparaison avec celles obtenues par les méthodes A-CMOMMT, I-CMOMMT-0.5 ainsi qu'avec une approche aléatoire. La méthode FFRL montre significativement une meilleure efficacité dans la moyenne d'observation des cibles, ainsi que dans la distribution de leurs observations, dans le cas où la densité d'agents et de cibles est importante. Ce phénomène est d'autant plus marqué dans le cas où les cibles adoptent un comportement évasif. Ceci s'explique par l'importance de la répartition des cibles entre les agents lorsque la densité devient importante.

4.2 Force Field MultiAgent Reinforcement Learning (F2MARL) : Un apprentissage centralisé pour une exécution distribuée

Cette section présente la méthode du *Force Field MultiAgent Reinforcement Learning* (F2MARL), qui est une extension de la méthode FFRL. Contrairement au FFRL, qui se concentre uniquement sur l'observation, le F2MARL se focalise sur la résolution du problème de l'observation appuyée par la patrouille. Pour atteindre cet objectif, le modèle F2MARL intègre la carte d'oisiveté de l'agent et la traite à l'aide d'un réseau neuronal convolutif. L'entraînement de la méthode F2MARL est disponible en open-source sur Github [3].

4.2.1 La méthode d'entraînement multi-agents

L'architecture

L'apprentissage par renforcement dans le cadre d'un système multi-agents peut reposer sur plusieurs architectures. Les auteurs WONG et al. [126] identifient les suivantes :

— Apprenants indépendants, ou *Independent learners* (IL) : Chaque agent apprend sa propre politique à partir de ses propres observations et récompenses. Cependant, ces schémas d'apprentissage ignorent le problème de non-stationnarité, défini ci-dessous.

— Contrôleur centralisé, ou *Centralized controller* (CC) : Cette architecture réduit le problème à une approche mono-agent. Toutes les observations et toutes les récompenses des agents individuels sont centralisées vers un seul agent, qui prend toutes les décisions. Cette architecture n'est pas adaptée à une méthode distribuée, telle que le F2MARL.

— Entraînement centralisé et exécution décentralisée, ou *Centralized training and decentralized execution* (CTDE) : Les agents sont autorisés à partager des informations pendant l'entraînement, telles que leurs observations ou leurs actions. Pendant l'exécution, les agents ont une coordination décentralisée ou distribuée, où l'observation et la communication sont locales.

Pour le F2MARL, nous avons choisi l'architecture CTDE pour adopter une coordination distribuée enrichie d'un apprentissage centralisé. Un modèle d'agent unique est entrainé pour apprendre une politique partagée parmi tous les agents homogènes. De cette manière, la politique peut être appliquée à différentes échelles. De plus, l'architecture CTDE prend en compte le problème d'un environnement non stationnaire. En effet, dans un environnement

3. https://github.com/JamyChahal/FFRL_F2MARL

multi-agents, chaque agent adapte sa propre politique en fonction de la stratégie des autres agents. Par conséquent, les transitions d'état et les récompenses peuvent changer pendant l'entraînement, rendant l'environnement non stationnaire. Cependant, plusieurs algorithmes d'apprentisage par renforcement reposent sur l'hypothèse de Markov d'un environnement stationnaire pour la convergence de la politique. Grâce à un entraînement centralisé, les agents sont capables de partager des informations, ce qui réduit potentiellement l'impact du problème de non-stationnarité.

L'algorithme d'apprentissage

L'algorithme du *Proximal Policy Optimization* (PPO) repose sur un modèle Acteur-Critique, permettant d'appliquer l'architecture CTDE de plusieurs manières pour développer des méthodes multi-agents. Au sein de l'approche *Multi Agent Proximal Policy Optimization* (MAPPO), développée par YU et al. [134], les agents se partagent entre eux les observations, ainsi que les actions effectuées, afin d'alimenter le modèle critique durant la phase d'apprentissage. Cette capacité de partage, comme illustré à la figure 4.12, permet de distinguer le MAPPO de l'approche IPPO employée par la méthode FFRL. Les auteurs ont démontré l'efficacité d'entraînement du MAPPO sur trois bancs d'essai multi-agents populaires, incluant le *Multi-agent Particle-World Environment* (MPE) dont est tiré l'environnement du F2MARL.

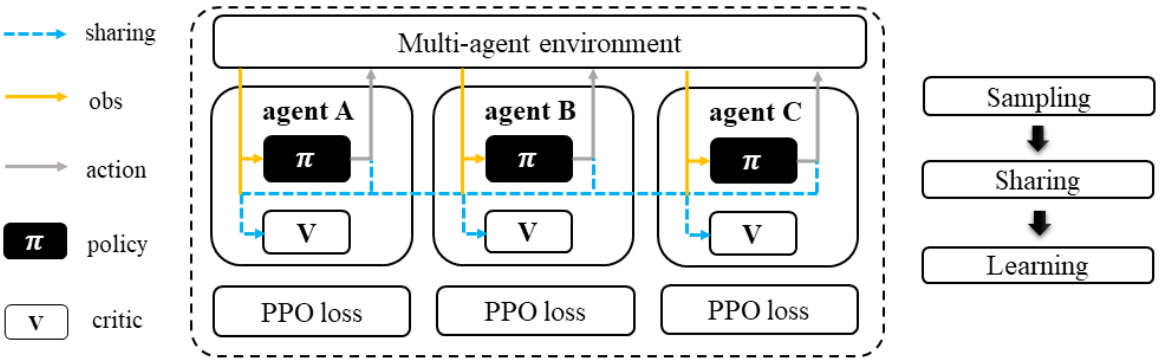


FIGURE 4.12 Illustration des mécanismes de l'architecture MAPPO, réalisée par HU et al. [53]

La figure 4.13 représente l'architecture de partage de paramètres du F2MARL dans le cas spécifique d'un modèle Acteur-Critique. Un seul modèle acteur, et un seul modèle critique, est employé et partagé entre tous les agents $i \in [1, m]$. Les cibles j adoptent un comportement issu d'une stratégie formelle, ne nécessitant pas d'apprentissage. Les agents i et les cibles j

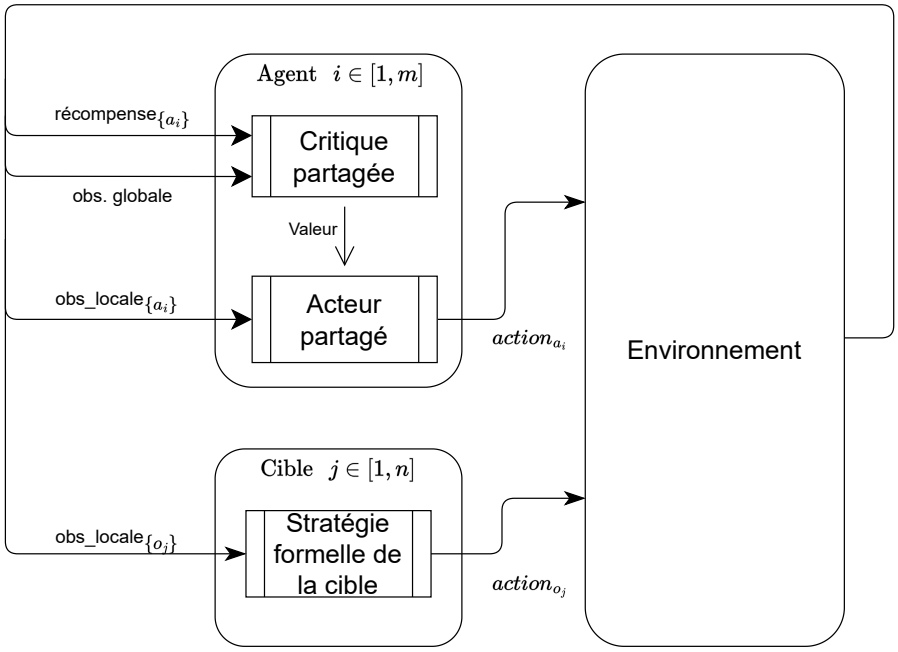


FIGURE 4.13 Diagramme de l'architecture MAPPO appliquée au POP.

agissent dans l'environnement par leurs actions, définies respectivement par le modèle de l'acteur et le comportement formel. En conséquence de cette action, l'environnement fournit la récompense appropriée ($reward_{a_i}$) décrite dans la section précédente, ainsi que l'observation locale ($local_{obs}$) et globale ($global_{obs}$) pour l'entraînement des agents. Ainsi, lors de l'évaluation, les agents n'obtiennent que l'observation locale pour agir sur l'environnement.

Définition de l'environnement

L'environnement de simulation pour l'entraînement est identique à celui employé pour le FFRL (cf. section 4.1.2), en incluant des modifications au regard de l'espace d'observation et des récompenses.

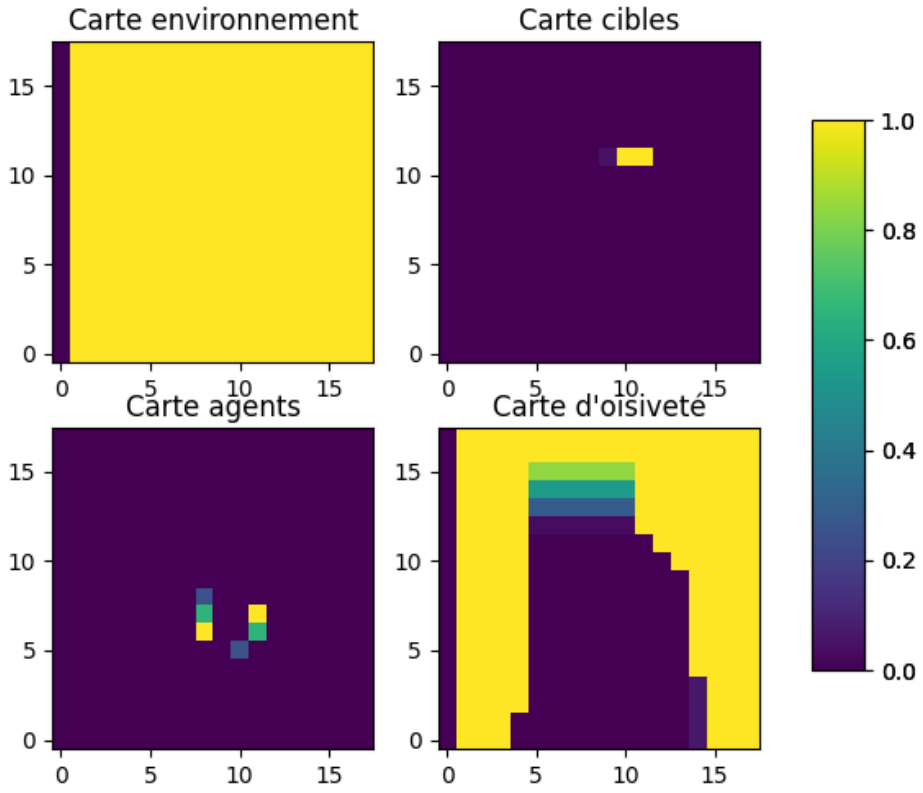
L'espace d'observation Dans l'environnement, pour chaque agent, nous proposons de structurer les observations sous la forme d'un ensemble d'images. L'avantage d'utiliser une observation sous forme d'image est d'avoir une représentation en 2D des environs de l'agent. Ce choix de conception s'oppose à l'utilisation d'un vecteur d'observation de taille fixe, comme le FFRL, qui n'est pas flexible avec une variation du nombre d'agents et de cibles environnantes. L'environnement est discrétisé par un facteur d_f. Deux types de centrage de l'image sont étudiés :

— L'image centrée sur l'agent : Les images sont centrées sur la position de l'agent. Ainsi, la localisation des autres agents et des cibles est représentée de manière relative, comme illustré par la figure 4.14
— L'image centrée sur l'environnement : Les images représentent l'ensemble de l'environnement, faisant ainsi coïncider les bordures de l'image avec celle de l'environnement carré. Un exemple d'illustration est proposé à la figure 4.15.

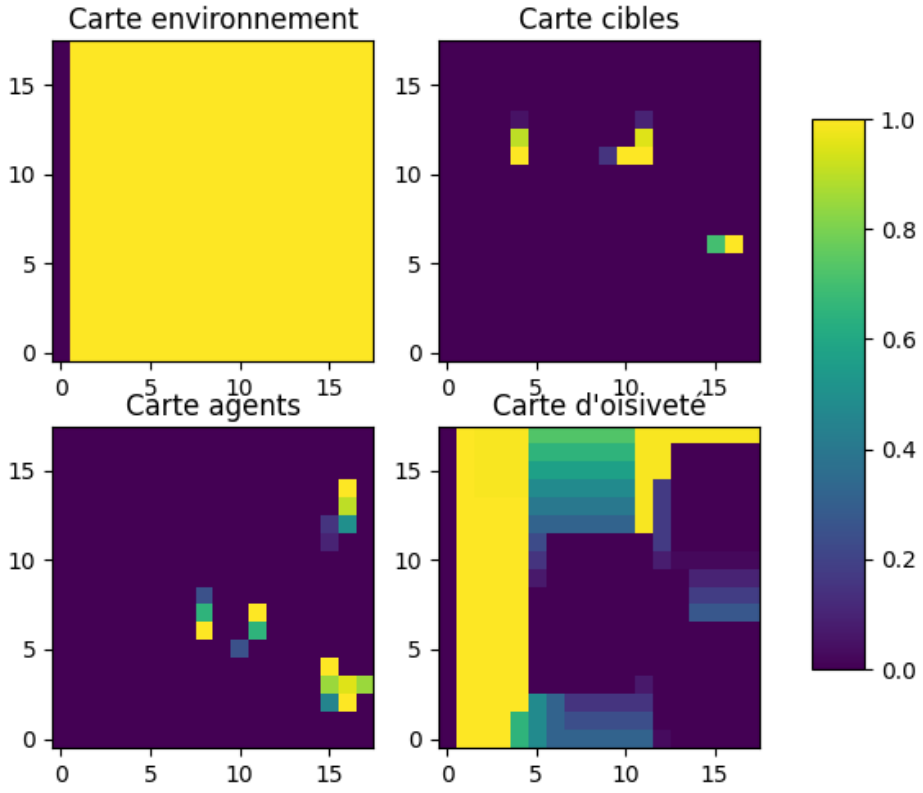
L'observation est composée de quatre couches d'image, chacune permettant de représenter géographiquement une information :

— Carte de l'environnement : Dans le cas où l'image est centrée sur l'agent, l'image représente la topologie et les limites de l'environnement, où chaque cellule a une valeur de 1 si la surface est disponible et 0 pour les obstacles et les limites de l'environnement. Dans le contexte de l'image centrée sur l'environnement, cette carte inscrit une valeur de 1 pour indiquer la position de l'agent dans l'environnement.
— Carte des cibles : L'image représente la position actuelle des cibles présentes dans un rayon d'observation. Chaque cible est représentée par une valeur de 1. À chaque pas de temps, la valeur de la cellule est réduite par une évaporation γ_t.
— Carte des agents : L'image fournit la position actuelle des agents dans le rayon de communication. Chaque agent est représenté par une valeur de 1. À chaque pas de temps, la valeur de la cellule est réduite par une évaporation γ_a.
— Carte d'oisiveté : L'image est un sous-ensemble de la carte d'oisiveté locale de l'agent M. L'image est normalisée entre 0 et 1, où la valeur 1 représente l'oisiveté la plus élevée de la carte locale M.

Le modèle critique, comme son nom l'indique, permet de critiquer les performances du modèle acteur. Il est uniquement utilisé durant la phase d'entraînement, tandis que le modèle acteur correspond au modèle utilisé par les agents en mission. Au sein de l'architecture MAPPO, le modèle critique peut bénéficier d'informations supplémentaires au modèle acteur, afin de mieux guider son apprentissage. Nous proposons que le modèle critique puisse accéder, grâce à son observation, à des informations de l'environnement sans les contraintes du rayon d'observation ou du rayon de communication propres au modèle acteur de l'agent. Autrement dit, que l'observation alimentant le modèle critique soit une observation globale de la réalité du terrain. Comme nous pouvons le voir aux figures 4.14 et 4.15, que ce soit pour des images centrées sur l'agent, ou centrées sur l'environnement, le modèle critique a accès à l'ensemble des positions des agents, des cibles et aux valeurs réelles des oisivetés. Alors que le modèle acteur est contraint d'observer les cibles uniquement présentes dans son

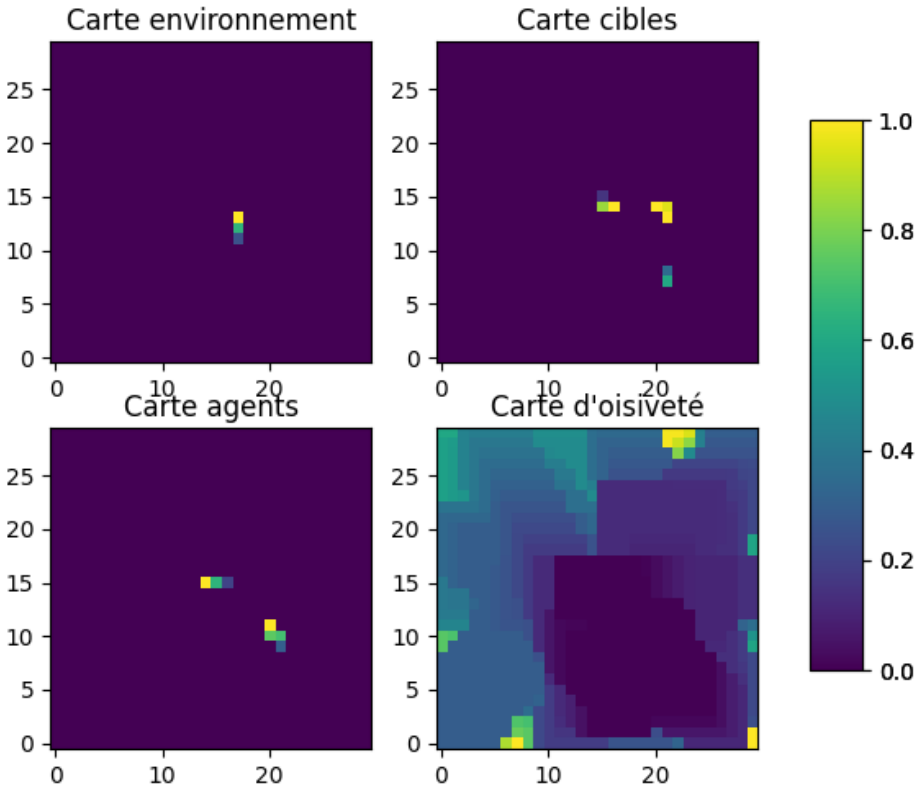


(a) Observation pour le modèle acteur.

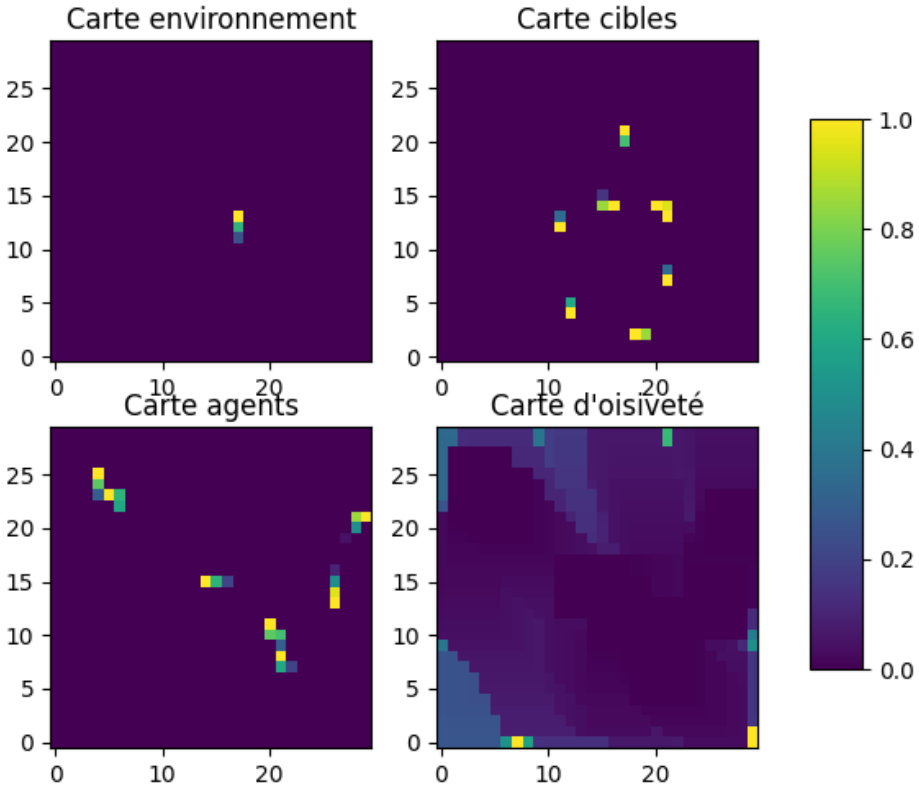


(b) Observation pour le modèle critique.

FIGURE 4.14 Exemple des cartes de perception du F2MARL centrées sur l'agent.



(a) Observation pour le modèle acteur.



(b) Observation pour le modèle critique.

FIGURE 4.15 Exemple des cartes de perception du F2MARL centrées sur l'environnement.

rayon d'observation, ainsi que positionner et communiquer avec les agents dans son rayon de communication.

Les récompenses La récompense d'un agent a_i à un instant t est composée de quatre composantes pondérées, chacune liée à une carte de perception : une récompense d'observation R_o, une récompense de patrouille R_p, une pénalité de collision R_c et une pénalité de sortie R_s :

$$R(a_i,t) = \omega_o R_o(a_i,t) + \omega_p R_p(a_i,t) + \omega_c R_c(a_i,t) + \omega_s R_s(a_i,t)$$

— La récompense d'observation : Partagée, chaque agent obtient une récompense pour chaque cible observée par au moins un agent dans l'environnement. Ainsi $R_o \in [0,m]$, avec m le nombre de cibles.
— La récompense de patrouille : Individuelle, cette récompense est égale à la somme des oisivetés des cellules observées après le déplacement de l'agent. Afin de la normaliser, cette somme est divisée par le nombre de cellules appartenant à la surface d'observation et par l'oisiveté maximale connue de l'agent. Ainsi $R_p \in [0,1]$.
— La pénalité de collision : Individuelle, elle est calculée de manière identique à la méthode du FFRL. $R_c \in [-1,0]$.
— La pénalité de sortie : Individuelle, cette pénalité est égale à -1 lorsque l'agent sort de l'environnement. Ainsi $R_s \in [-1,0]$.

Nous considérons quatre coefficients : ω_o, ω_p ω_c et ω_s. Ces poids sont fixés par l'expérimentateur en tenant compte de la priorité de certains incitatifs par rapport à d'autres. Par exemple, si la distance entre les agents est moins importante que la stratégie d'observation, alors ω_c devrait être diminué par rapport à ω_o.

4.2.2 Entraînement et résultats

Entraînement du modèle

Durant la phase d'entraînement, nous fixons les coefficients liés aux récompenses de la manière suivante : Les récompenses positives, liées à l'observation et à la patrouille, ont un coefficient $\omega_o = \omega_p = 1$. Alors que les récompenses négatives, liées au risque de collision et de sortie de l'environnement, ont un coefficient bien plus important, avec $\omega_c = \omega_s = 10$.

La figure 4.16 représente l'architecture du modèle acteur/critique employé pour la méthode du F2MARL. En entrée, les quatre images issues de l'observation sont traitées par un réseau neuronal convolutif (CNN), avec une taille de convolution de (3×3), 4 cartes d'attributs et d'une dimension surfacique identique à la couche d'entrée, c'est-à-dire 30×30

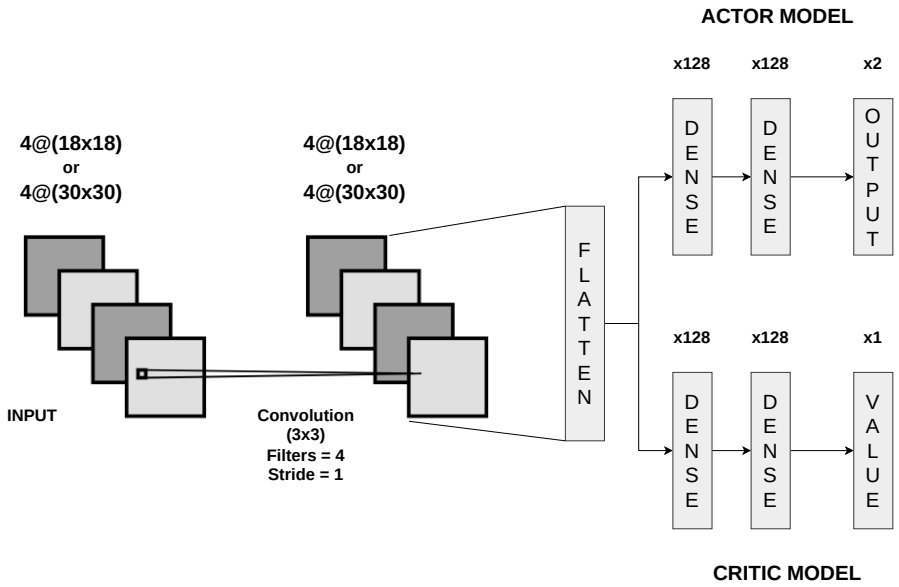


FIGURE 4.16 Architecture des modèles acteur et critique, avec un partage du réseau de convolution entre les deux modèles.

cellules pour une représentation centrée sur l'environnement et 18×18 cellules pour une représentation centrée sur l'agent. Pour l'observation centrée sur l'environnement, les dimensions d'images correspondent à une taille d'environnement de 100×100 cellules avec une discrétisation de $d_f = 0.3$. Alors que pour l'observation centrée sur l'agent, nous proposons que la taille de l'image soit deux fois plus grande que le rayon de communication, avec ici un rayon de 15 cellules, donc un diamètre de 30 cellules. Avec une discrétisation $d_f = 0.3$, nous obtenons des images de taille 18×18 cellules. Les deux modèles partagent le même réseau de convolution. Ce choix de structure provient de l'intuition que les deux modèles cherchent à extraire les mêmes caractéristiques des images en entrée, impliquant également une réduction du nombre de paramètres du modèle à entraîner. La sortie de la convolution est par la suite vectorisée, afin de s'intégrer au modèle agent et au modèle critique, composée chacun de deux couches cachées de 128 neurones. En sortie, le modèle agent retourne un vecteur de dimension 2, correspondant aux mouvements à réaliser sur les axes x et y. Le modèle critique, quant à lui, retourne une seule valeur utilisée pour l'entraînement du modèle acteur. Dans le cas où les dimensions de l'observation ne sont pas identiques entre le modèle acteur et le modèle critique, avec donc des représentations hétérogènes centrées sur l'agent et sur l'environnement, alors les réseaux de convolution sont séparés entre les modèles.

À l'instar du FFRL, l'entraînement du modèle du F2MARL repose sur un modèle commun à tous les agents. Par ailleurs, les hyperparamètres du MAPPO, et donc par extension du PPO, sont obtenus grâce à l'algorithme PBT, avec une taille de population de 7 entraînements

TABLE 4.2 Hyperparamètres du F2MARL. SGD signifie *Stochastic gradient descent*.

Hyperparamètres	PBT Plage de valeur	Valeur finale (agent ;agent)	Valeur finale (agent ;env)	Valeur finale (env ;env)
Horizon temporel pour chaque épisode	Fixe	1000	1000	1000
Taille du mini-batch	[128 ;16e3]	1952	2464	1420
Clipping ratio	[0.01 ; 0.5]	0.017	0.3204	0.1553
Gamma	[0.9 ;1.0]	0.9	0.9	0.9
Taux d'apprentissage (Learning rate)	[5e-05 ; 1.0]	0.01	1e-04	1e-05
Coefficient d'entropie	[0 ;0.1]	0.0123	0.0045	0.0119
Lambda	[0.7 ;1.0]	0.7254	0.7389	0.9961
Nombre d'itérations SGD	[1 ;30]	21	1	30
Taille du mini-batch SGD	[128 ;512]	52	383	139

simultanés. La courbe d'apprentissage des meilleurs modèles est présentée par la figure 4.17. Lors de l'apprentissage, trois configurations sont étudiées :

— Acteur et critique centrés sur l'agent, notée *(agent ; agent)* : Comme présenté à la figure 4.14, les observations à la fois locales et globales sont centrées sur l'agent.

— Acteur et critique centrés sur l'environnement, notée *(env ; env)* : Comme présenté à la figure 4.15, les observations à la fois locales et globales considèrent l'ensemble de l'environnement.

— Acteur centré sur l'agent et critique centré sur l'environnement, notée *(agent ; env)* : Dans cette configuration, le modèle acteur est alimenté par une représentation centrée sur l'agent, tandis que le modèle critique est alimenté par une représentation centrée sur l'environnement.

En observant la figure 4.17, nous pouvons constater que les performances d'apprentissage varient selon les configurations utilisées. Dans un premier temps, la pire performance est obtenue par la configuration *(env ; env)*, représentée en bleu. Le modèle ne semble pas réussir à évoluer vers une meilleure politique. Les causes potentielles sont une architecture de modèle pas assez complexe, un nombre d'itérations insuffisant ou encore une représentation de l'environnement inadaptée pour le modèle acteur. Dans un second temps, la configuration *(agent ; agent)* en orange, donc centrée sur l'agent, converge vers une politique plus performante que la configuration précédente *(env ; env)*. Nous supposons que lorsque les éléments de l'environnement sont disposés de manière relative à la position de l'agent dans

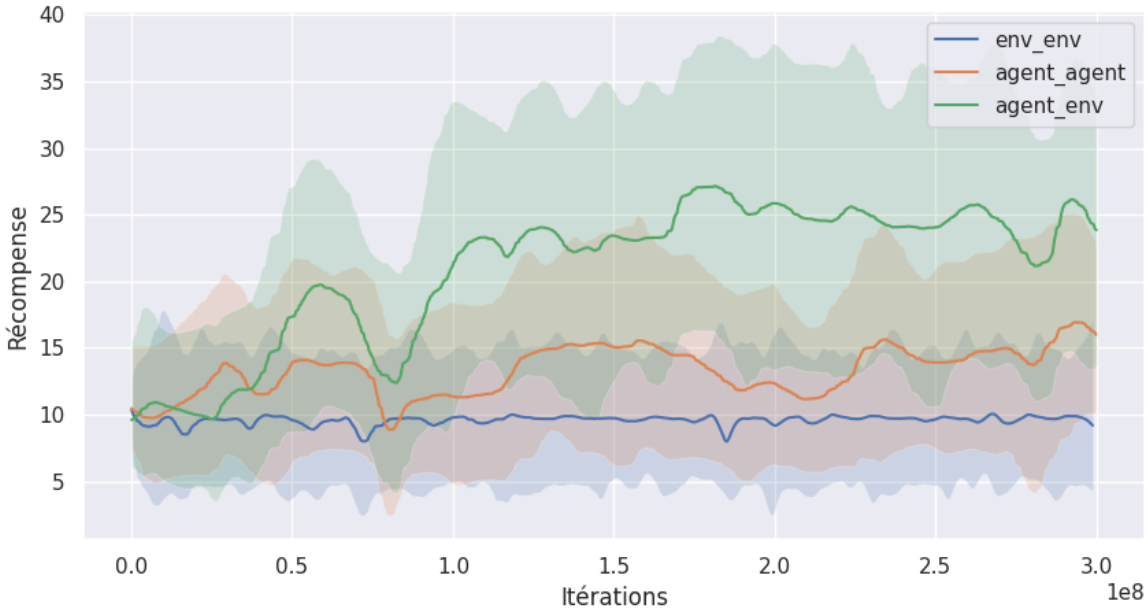


FIGURE 4.17 Courbe d'apprentissage du F2MARL, selon plusieurs représentations des observations. La moyenne des récompenses est tracée en ligne continue, tandis que l'amplitude entre les valeurs maximales et minimales est représentée par les surfaces colorées.

l'observation, il est plus facile pour l'agent d'établir une corrélation entre ses actions et les récompenses obtenues. Enfin, la meilleure performance est obtenue avec la configuration *(agent;env)*, représentée en vert, où l'observation du modèle acteur est centrée sur l'agent et l'observation du modèle critique est centrée sur l'environnement. Nous faisons l'hypothèse que le modèle acteur est plus performant avec une observation centrée sur l'agent, comme vu précédemment, tandis que le modèle critique oriente plus facilement l'apprentissage en observant l'entièreté de l'environnement. Lors des évaluations expérimentales de la prochaine section, nous conservons donc ce modèle comme référence de la méthode F2MARL.

Résultats

La méthode F2MARL aborde la double problématique présentée par le formalisme du POP. Dans cette optique, nous évaluons cette méthode en la comparant à d'autres techniques d'observation, telles que le A-CMOMMT et le I-CMOMMT-0.5, ainsi qu'à des approches de patrouille comme le HI et le CI, en plus d'une méthode de déplacement aléatoire. L'évaluation des performances en ce qui concerne l'observation des cibles repose sur deux métriques : la moyenne d'observation (métrique A) et l'écart-type de l'observation. En ce qui concerne la patrouille, nous comparons les méthodes en utilisant également deux critères : la moyenne de l'oisiveté et l'oisiveté maximale régionale. Ces évaluations sont obtenues dans un environnement carré de $100m \times 100m$, composé de 2, 4 puis 6 agents, ainsi que 3, 5 puis 7 cibles adoptant un comportement aléatoire. Chaque scenario a une durée de $3600s$ et est évalué 20 fois. Les autres caractéristiques de la mission, comme le rayon d'observation ou la vitesse des agents, sont similaires aux expériences menées pour l'évaluation de la méthode du FFRL.

Les figures 4.18 et 4.19 montrent respectivement les performances des méthodes pour observer les cibles et la répartition de l'observation entre les cibles. Comme prévu, les méthodes qui ne sont pas axées sur l'observation des cibles, telles que la méthode aléatoire, CI ou HI, présentent des performances nettement inférieures en termes d'observation moyenne des cibles. De plus, en l'absence de suivi des cibles, celles-ci sont observées de manière uniforme, ce qui se traduit par une très faible dispersion des observations et donc une déviation standard de l'observation particulièrement faible. Alors que le FFRL a montré une amélioration significative de l'observation des cibles dans des situations impliquant une grande variété d'agents et de cibles, le F2MARL démontre une amélioration de l'observation qui reste constante, indépendamment de la densité, par rapport aux méthodes évaluées. De plus, selon les configurations examinées, le F2MARL parvient à équilibrer l'observation entre les cibles de manière comparable, voire supérieure, à celle du I-CMOMMT.

Par ailleurs, contrairement à la méthode du FFRL, la méthode du F2MARL cherche également à réaliser une patrouille efficace. Les figures 4.20 et 4.21 représentent respectivement les oisivetés moyennes et les oisivetés maximales régionales obtenues par toutes les méthodes évaluées. Les méthodes se focalisant uniquement sur la patrouille, comme le CI ou le HI, présentent les meilleures performances pour minimiser les deux métriques liées à l'oisiveté. La méthode F2MARL utilise une stratégie de patrouille avec un niveau d'efficacité intermédiaire, situé entre celui de I-CMOMMT et celui de A-CMOMMT pour les deux métriques d'évaluation.

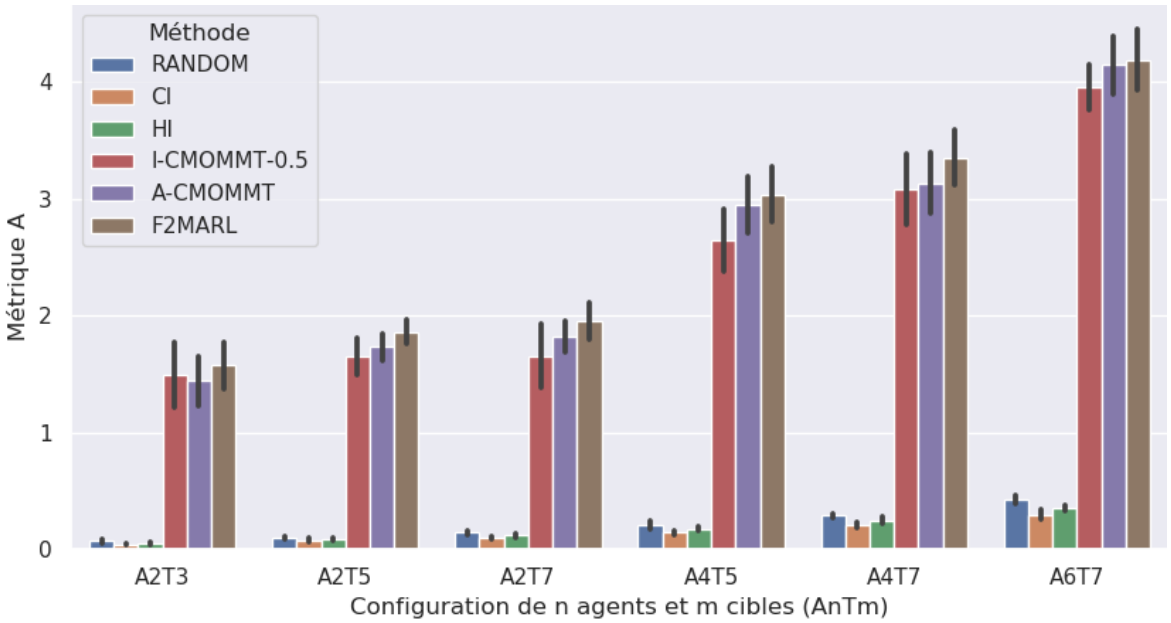


FIGURE 4.18 Efficacité des diverses approches pour l'observation moyenne des cibles (métrique A).

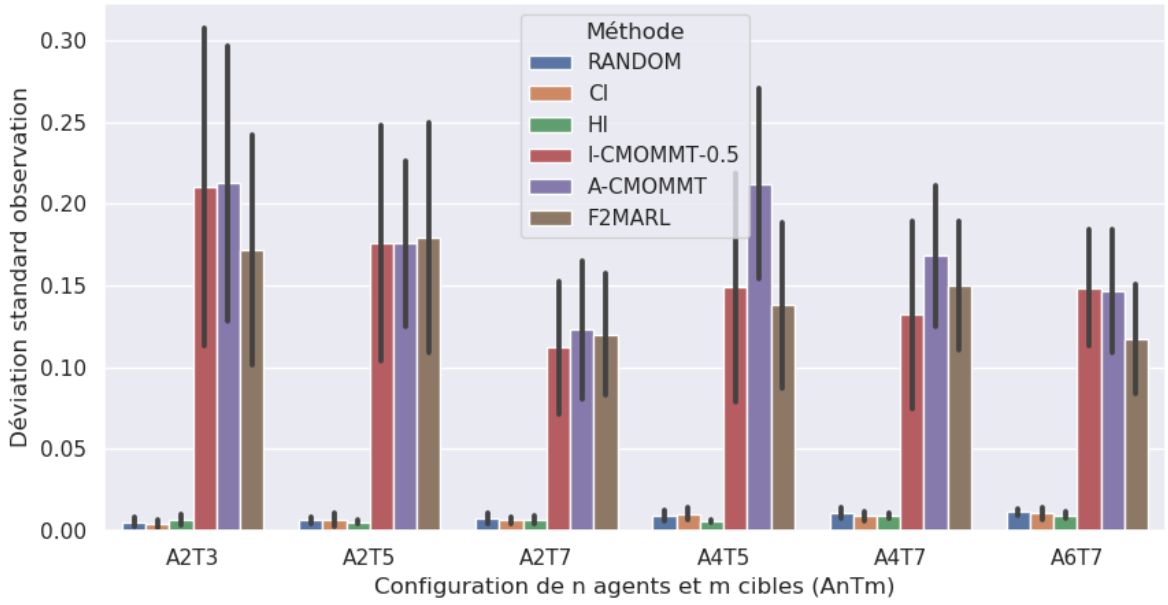


FIGURE 4.19 Efficacité des diverses approches au regard de la déviation standard σ_n de l'observation des cibles.

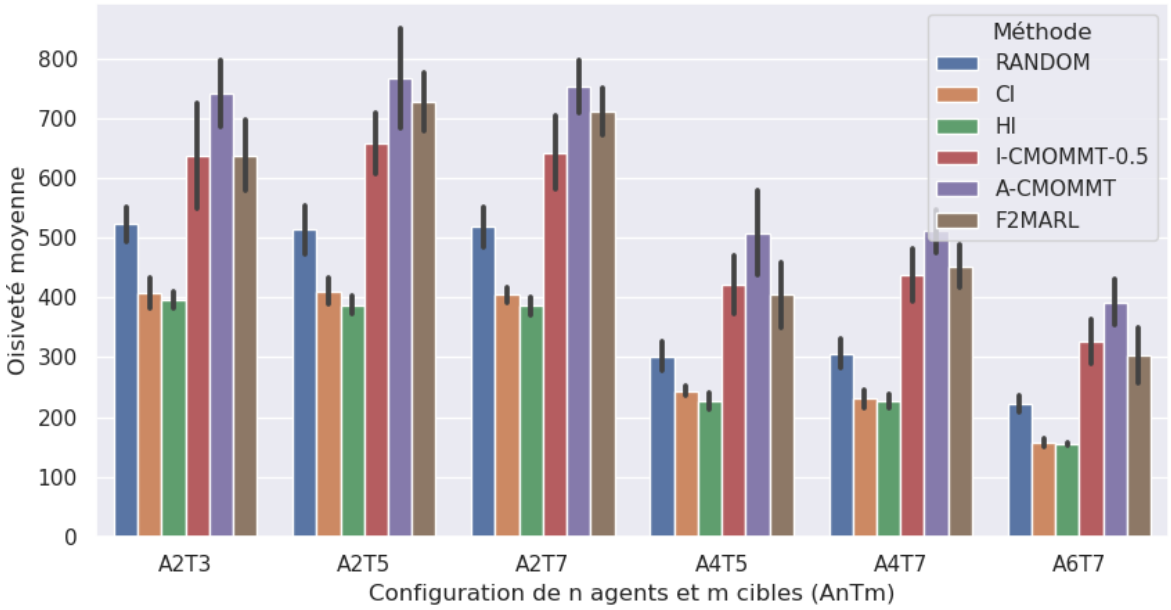


FIGURE 4.20 Efficacité des diverses approches pour minimiser l'oisiveté moyenne de l'environnement.

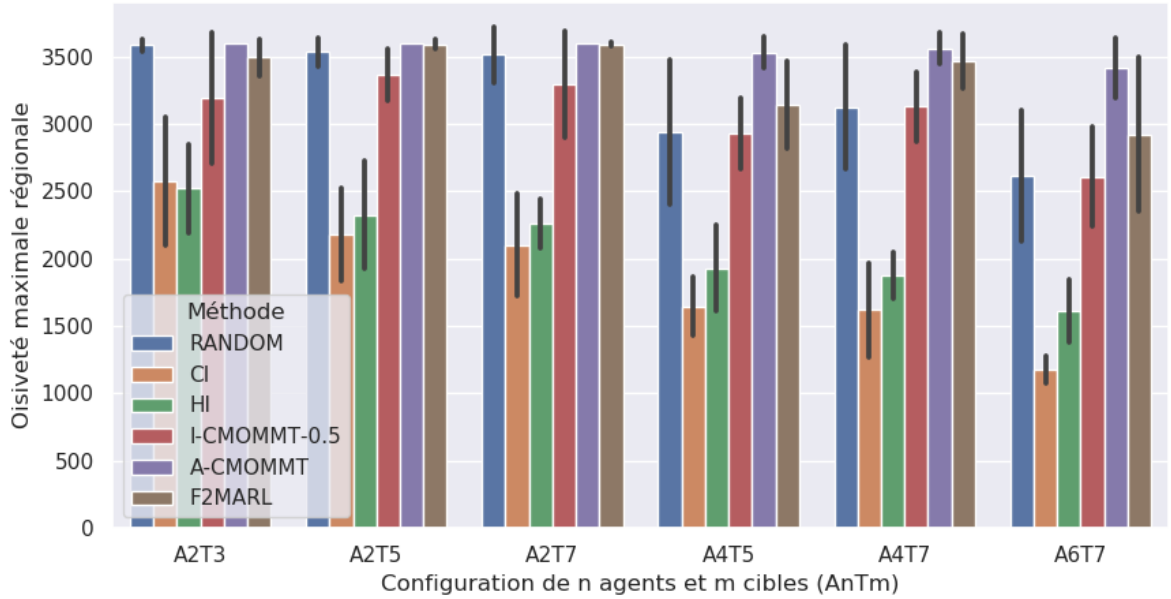


FIGURE 4.21 Efficacité des diverses approches pour minimiser l'oisiveté maximale régionale de l'environnement.

4.2.3 Conclusion

La méthode F2MARL vise à résoudre le problème formulé dans le cadre du POP en utilisant l'apprentissage par renforcement. Pour ce faire, elle associe à chaque objectif du POP (voir section 3.1.4) une perception de l'environnement ainsi qu'une récompense. Une caractéristique distinctive du F2MARL est la perception par les agents de l'environnement à travers des images. Cette représentation bidimensionnelle permet de localiser les agents et les cibles sans aucune limitation en termes de nombre. De plus, cette représentation permet d'intégrer les informations d'oisiveté sous forme matricielle. Ces images sont traitées à l'aide de réseaux de convolution neuronaux. L'entraînement des modèles repose sur l'utilisation de l'architecture MAPPO [134], qui permet d'incorporer des perceptions plus étendues dans le modèle critique que celles limitées par les capacités de l'agent destinées au modèle acteur. Pour ce faire, trois structures de représentation de l'environnement sont examinées pour nourrir les modèles acteur et critique : soit les images sont centrées sur l'agent, soit sur l'environnement, soit la perception du modèle critique est centrée sur l'environnement tandis que celle du modèle acteur est centrée sur l'agent. De manière expérimentale, il a été observé que cette dernière configuration produit l'apprentissage le plus efficace.

La méthode F2MARL a été comparée avec les méthodes du A-CMOMMT, le I-CMOMMT, le CI, le HI et enfin une méthode de déplacement aléatoire. Les résultats montrent que la méthode du F2MARL permet une amélioration de l'observation des cibles en comparaison des méthodes précédemment citées, et une efficacité pour réduire l'oisiveté comprise entre celle du I-CMOMMT et du A-CMOMMT. En perceptive future, nous pensons que les performances de patrouille, qui sont honorables, pourraient être nettement améliorées en redéfinissant la récompense liée. La récompense de patrouille est comprise entre $[0;1]$, et tend expérimentalement autour d'une valeur maximale de 0,2 pour un agent après un déplacement. Alors que les récompenses d'observation sont partagées entre les agents et sont comprises entre $[0;m]$, avec m le nombre de cibles. Par conséquent, les récompenses liées à la patrouille sont inférieures à celles obtenues pour l'observation des cibles, ce qui accorde une nette priorité à cette dernière. Les récentes recherches menées par JANA, VACHHANI et SINHA [56] ont mis en lumière une comparaison entre diverses définitions de la récompense de patrouille. La première récompense est définie comme étant égale à l'oisiveté du nœud visité, tandis que la seconde consiste en une récompense négative proportionnelle à la pire oisiveté de l'environnement. Enfin, la troisième définition, la plus performante, est basée sur le rapport entre l'oisiveté instantanée du nœud visité et l'oisiveté moyenne instantanée de tous les nœuds du graphe. Ces différentes formulations de la récompense pourraient être adaptées au cadre du POP afin d'améliorer la stratégie de patrouille.

4.3 MultiAgent Learning using Optimized Strategy (MALOS) : Une optimisation multicritère centralisée et un apprentissage distribué

Les méthodes précédemment développées, c'est-à-dire le FFRL et le F2MARL, répondent au POP grâce à de l'apprentissage par renforcement. Cependant, d'autres techniques d'apprentissage permettent également aux agents de développer des stratégies. Par exemple, comme nous l'avons évoqué au sein de l'état de l'art (cf. section 2.2.6), OTHMANI-GUIBOURG, EL FALLAH SEGHROUCHNI et FARGES [89] propose d'employer un apprentissage supervisé, afin que les agents tendent à imiter le comportement d'une méthode centralisée, tout en le déployant sur une coordination décentralisée. Cette section présente la méthode *MultiAgent Learning using Optimized Strategy* (MALOS), s'inspirant de la démarche d'apprentissage citée précédemment. L'objectif est de mener une stratégie combinant à la fois l'efficacité d'une méthode centralisée avec les avantages d'une coordination distribuée. L'implémentation de MALOS ainsi que toutes les expériences sont disponibles sur Github [4].

4.3.1 La méthode

La méthode MALOS repose sur une succession de trois étapes clés, permettant respectivement d'observer le comportement des agents d'une méthode centralisée, puis d'apprendre à les imiter, pour enfin exécuter la stratégie apprise de manière distribuée. Les étapes fonctionnent de la manière suivante :

- **Étape 1 :** Dans un premier temps, un algorithme d'optimisation est employé comme stratégie centralisée. Cet algorithme examine les diverses options de mouvement des agents, puis les déplace simultanément en fonction de la configuration qui satisfait au mieux les objectifs du POP.
- **Étape 2 :** Par la suite, les agents apprennent à imiter le déplacement des agents de la stratégie centralisée, en comparant la perception de chaque agent avec son action. Le modèle est ainsi entraîné selon une approche d'apprentissage supervisé.
- **Étape 3 :** Enfin, durant l'exécution, le modèle appris est appliqué par chaque agent, individuellement, de manière distribuée.

La figure 4.22 présente l'architecture des trois grandes étapes. Initialement, à chaque pas de temps, l'algorithme d'optimisation cherche un ensemble de déplacement x à faire réaliser aux agents permettant d'optimiser les fonctions objectives. Pour ce faire, l'algorithme étudie les performances $g(x)$ potentielles pour plusieurs ensembles d'actions x. Ensuite, une

4. https://github.com/JamyChahal/MALOS

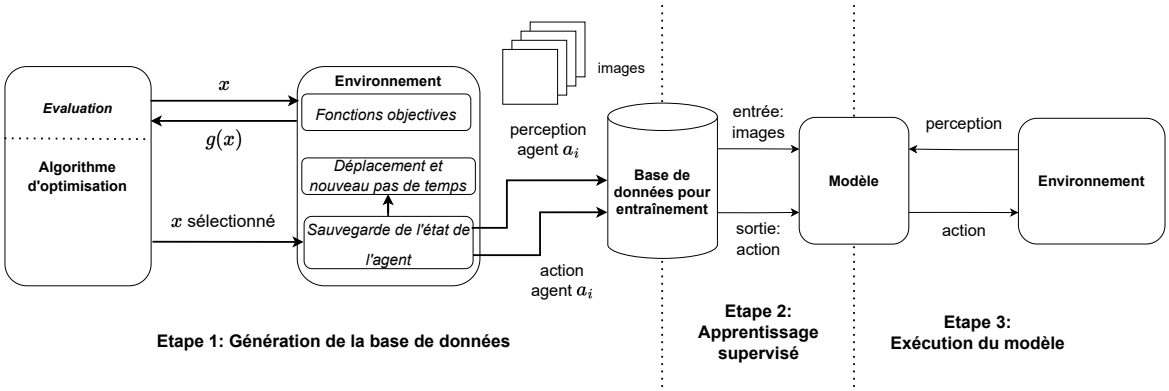


FIGURE 4.22 Architecture du processus d'entraînement de la méthode MALOS.

fois qu'une stratégie présentant la meilleure performance est sélectionnée, la perception et l'action de chaque agent sont enregistrées dans une base de données d'entraînement avant de passer au pas de temps suivant.

Une fois la mission terminée, la seconde étape consiste à réaliser un apprentissage supervisé hors ligne pour entraîner le modèle à partir de la base de données d'entraînement. Ainsi, le modèle cherche à apprendre la corrélation entre la perception et l'action entreprise du point de vue d'un agent. La dernière étape permet à ce que chaque agent, de manière individuelle, puisse appliquer le modèle entraîné pendant la phase d'exécution de la stratégie distribuée.

Étape 1 : L'algorithme d'optimisation et la génération de la base de données.

L'environnement La simulation de l'environnement reprend le contexte du *opp_env* développés au sein de la section 4.1.2. Les agents d'un côté et les cibles de l'autre sont considérés comme homogènes, en partageant les mêmes caractéristiques telles que le rayon d'observation, le rayon de communication ou la vitesse maximale.

Les perceptions Les agents perçoivent l'environnement avec la même structure d'image que l'observation locale centrée sur l'agent adoptée par la méthode du F2MARL (cf. section 4.2.1), incluant également les coefficients d'évaporation γ_t et γ_a.

Les actions À chaque étape, chaque agent est capable de se déplacer dans l'environnement. Nous définissons ce mouvement de la manière suivante : (1) la force désirée le long de l'axe des abscisses (x) et (2) la force désirée le long de l'axe des ordonnées (y). Ces actions sont définies comme une valeur continue entre -1 et 1, représentant la vitesse maximale normalisée de l'agent.

Variable x à optimiser La variable x est conçue pour représenter toutes les actions, c'est-à-dire les mouvements le long des deux axes, de tous les agents simultanément. Par conséquent, la variable x est conçue de la manière suivante :

$$x = [\delta_{x_1}, \delta_{y_1}, ..., \delta_{x_i}, \delta_{y_i}, ..., \delta_{x_n}, \delta_{y_n}]$$

Nous nous plaçons dans le contexte de l'optimisation de fonctions continues, où les éléments de la variable x sont des valeurs numériques réelles. Ainsi, l'espace de définition de x est :

$$\Omega = [-1;1]^m$$

La fonction objective Le but de la fonction objective $g(x)$ est d'estimer dans quelle mesure les prochains mouvements évalués des agents, notés x, permettront d'atteindre les objectifs prédéfinis.

$$\text{maximiser } \; g(x) = \sum_{i=1}^{M} \lambda_i f_i(x) \; \text{ with } \; x \in \Omega$$

La résolution du POPrepose sur $M = 4$ objectifs, tous identifiés au sein de la section 3.1.4. Ces objectifs, dont leurs priorités relatives sont définies par les coefficients λ_i, sont décrits par les fonctions suivantes :

1. La fonction d'observation :

$$f_1(x) = \sum_{j=1}^{n} f_{obs}(o_j)$$

$$f_{obs}(o_j) = \begin{cases} 1 & \text{si la cible } j \text{ est observée par au moins un agent} \\ 0 & \text{sinon} \end{cases}$$

La fonction d'observation $f_1(x)$ tend à maximiser le nombre de cibles observées par au moins un agent à l'instant t. Ainsi, $f_1(x) \in [0,n]$. Grâce à la fonction $f_1(x)$, les agents sont encouragés à se partager les cibles entre eux et à éviter d'avoir plus d'un agent par cible.

2. La fonction de patrouille : Cette fonction, notée $f_2(x)$, tend à minimiser l'oisiveté de l'environnement aux alentours de l'agent. La fonction repose sur la carte d'oisiveté locale de l'agent. Une première intuition a été de considérer sa valeur égale à la

somme des oisivetés des cellules observées. Cependant, le mouvement d'un agent $(\delta_{x_i}, \delta_{y_i})$ peut être relativement petit, et l'agent ne considérerait qu'une très petite partie de l'environnement environnant. Ainsi, nous considérons le vecteur (u_i, v_i), représentant la direction suivie par l'agent a_i en réalisant son déplacement $(\delta_{x_i}, \delta_{y_i})$. La valeur de la fonction de patrouille correspond à la somme des oisivetés des cellules observées et potentiellement observables au cours des prochains h pas si la direction de l'agent est maintenue. h est appelé l'horizon temporel.

3. La fonction de collision :

$$f_3(x) = \sum_{i=1}^{m-1} \sum_{k=i+1}^{m} f_c(a_i, a_k)$$

Les agents doivent adopter un comportement sûr en tenant compte des agents environnants. La fonction $f_c(x)$ est égale à 0 si la distance entre les agents est supérieure à une distance de sécurité, notée DS. Si la distance est inférieure à DS, la valeur diminue jusqu'à atteindre -1 en atteignant une distance dangereuse DD. En dessous de DD, la fonction est saturée à -1. Par conséquent, la fonction $f_3(x)$ attribue une valeur négative si les agents sont trop proches les uns des autres.

4. La fonction de sortie :

$$f_4(x) = \begin{cases} -1 & \text{si un agent sort de l'environnement} \\ 0 & \text{sinon} \end{cases}$$

Les agents ne sont pas censés quitter l'environnement, même pour suivre les cibles. Par conséquent, lorsqu'un agent a_i sort des limites de l'environnement, la fonction de sortie renvoie une valeur négative.

Étape 2 : L'apprentissage supervisé

La deuxième étape de l'entraînement MALOS consiste à appliquer la stratégie centralisée issue de l'algorithme d'optimisation à une coordination distribuée. Dans ce but, afin de comprendre la corrélation entre la perception des agents et leurs actions, plusieurs simulations sont réalisées. L'entraînement est effectué une fois que toutes les simulations souhaitées sont terminées.

Par conséquent, les simulations doivent représenter un large éventail de configurations possibles, telles qu'une large variété n d'agents dans l'environnement, plusieurs nombres m

de cibles, diverses vitesses maximales des agents de suivi ou différentes tailles d'environnement. L'entraînement est réalisé du point de vue de chaque agent. Comme illustré dans la Figure 4.22, à chaque pas de temps de la simulation, l'environnement enregistre dans une base de données la perception de chaque agent en tant qu'entrée, ainsi que l'action effectuée en tant que sortie.

L'architecture du modèle d'entraînement proposé repose sur l'utilisation de quatre images en tant qu'entrée. Pour cela, nous proposons d'adopter un réseau neuronal convolutif (CNN), suivi de deux couches cachées comprenant chacune 32 neurones. Les fonctions d'activation utilisées sont l'Unité Linéaire Rectifiée (ReLU) pour les convolutions et la tangente hyperbolique (Tanh) pour les couches cachées et de sortie. Le noyau utilisé pour effectuer la convolution a une taille de (3×3).

Étape 3 : Exécution du modèle

Une fois le modèle entraîné, la méthode MALOS peut être déployée en mission. Afin d'accomplir cela, chaque agent exécute indépendamment ce même modèle, dans le cadre d'une coordination distribuée. En entrée, le modèle est alimenté par les perceptions de l'agent, représentées sous forme d'images. Et en sortie, le modèle fournit l'action à entreprendre. Ces interactions avec le modèle sont ainsi formatées de la même manière que durant l'étape 2, avec l'apprentissage supervisé.

4.3.2 Expérimentations et résultats

Les expériences en simulation reposent sur l'environnement ROS2/Gazebo décrit au sein du chapitre 6. L'implémentation de MALOS utilise la librairie Pymoo [20], ainsi que les outils d'apprentissage de la bibliothèque Tensorflow [1]. Pour la phase d'optimisation de la première étape, nous utilisons l'algorithme d'optimisation mono-objective *Covariance matrix adaptation evolution strategy* (CMA-ES) développée par HANSEN et AUGER [47]. Deux scénarios sont mis en place pour évaluer l'efficacité de MALOS avec le A-CMOMMT [92], du I-CMOMMT (cf. section 3.2.1) et d'une stratégie aléatoire. Les résultats obtenus sont issus de 20 itérations de la simulation pour chaque configuration étudiée.

MALOS a été entraîné sur *2 600k* échantillons. Chaque échantillon représente un ensemble de perceptions et d'actions d'un agent à un pas de temps spécifique. Étant donné que les actions sont continues, l'entraînement est considéré comme un problème de régression. Nous utilisons l'erreur quadratique moyenne comme fonction de perte, la somme des erreurs sur les axes x et y des actions est illustrée dans la figure 4.23. Le jeu de données de validation

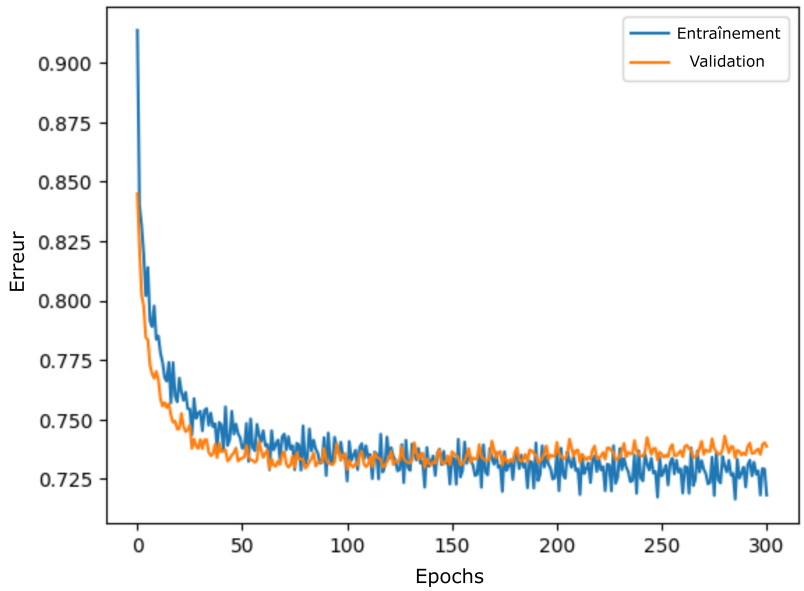


FIGURE 4.23 Courbe d'apprentissage, et de validation, du modèle pour MALOS.

représente 20% de l'ensemble des données. Les simulations ont été paramétrées pour couvrir plusieurs configurations, décrites dans le tableau 4.3.

Premier scénario : Variation du nombre d'agents

Le premier scénario consiste à comparer les méthodes en ce qui concerne le nombre d'agents dans l'environnement. Les configurations utilisées lors de l'évaluation sont décrites dans le tableau 4.4. Chaque configuration est évaluée 50 fois. Nous avons comparé les trois méthodes formelles (A-CMOMMT, I-CMOMMT et méthode aléatoire) avec MALOS en utilisant les métriques A et H. Les figures 4.24 et 4.25 présentent respectivement les performances de ces métriques. Les maximum et minimum sont représentés par la surface colorée, tandis que la moyenne est illustrée par une ligne continue. Le scénario considère $m = 5$ cibles et $n = \{2,4,6,8,10\}$ agents. Une méthode efficace pour l'observation et une distribution équitable de l'observation tend à maximiser à la fois la métrique A et la métrique H (cf. section 2.1.2).

La méthode aléatoire, illustrée en bleu, n'a pas pour but de suivre les cibles, ce qui explique pourquoi elle obtient les pires observations des cibles parmi les autres méthodes évaluées. Cependant, du fait qu'elle traverse les cibles de manière aléatoire, la distribution entre celles-ci est équilibrée, ce qui se traduit par une valeur élevée de la métrique H. Le champ de force utilisé dans le A-CMOMMT [92] permet aux agents d'être attirés par les

TABLE 4.3 Paramètres de simulation et d'optimisation pour générer la base de données d'entraînement. *ut* corresponds à l'unité de temps ou pas de temps.

Paramètre	**Valeur**
n agents	$\{3,5,8\}$
m cibles	$\{3,5,8\}$
Taille de l'environnement	$60m \times 60m$
Coefficient $\lambda_1, \lambda_2, \lambda_3, \lambda_4$	$(1,1,10,10)$
Évaporation γ_t, γ_a	$(0.1, 0.1)$
Horizon h	5 *ut*
DD, DS	($1m$, $2m$)
Durée de l'expérience T	1800 *ut*
Rayon d'observation	5 *m*
Rayon de communication	10 *m*
Vitesse max. des cibles	1 *m*/*ut*
Vitesse max. des agents	2 *m*/*ut*

TABLE 4.4 Paramètres de simulation.

Catégorie	**Paramètre**	**Valeur**
Environnement	Taille de l'environnement	$180m \times 180m$
	Durée de l'expérience T	900 *s*
	Rayon d'observation	5 *m*
	Rayon de communication	8 *m*
	Vitesse maximale des cibles	0.5 *m*/*s*
	Vitesse maximale des agents	1 *m*/*s*
I-CMOMMT	σ	$0.8 \times T$
A-CMOMMT & I-CMOMMT	$(do_1, do_2, do_3,$ portée de suivi prédictive)	(1m, 2m, 4.5m, 5m)
	(dr_1, dr_2)	($1m$, $2m$)

cibles et repoussés par les autres agents environnants. En l'absence de cibles dans le rayon d'observation, les agents adoptent un comportement aléatoire. La méthode A-CMOMMT, présentée en orange, montre une meilleure observation des cibles que le comportement aléatoire et le I-CMOMMT. Cependant, les agents se concentrent uniquement sur le suivi des cibles, sans coordination pour répartir l'observation des cibles entre les agents ou pour chercher de nouvelles cibles, ce qui conduit à une mauvaise répartition de l'observation.

Le I-CMOMMT-0.5 (cf. section 3.2.1) est représenté en rouge. Cette méthode parvient à suivre les cibles et à en rechercher de nouvelles en ajoutant une force de patrouille au champ de force utilisé dans l'A-CMOMMT. Le résultat de ce compromis entre l'exploration et l'exploitation se traduit par une moyenne inférieure des cibles observées par rapport à

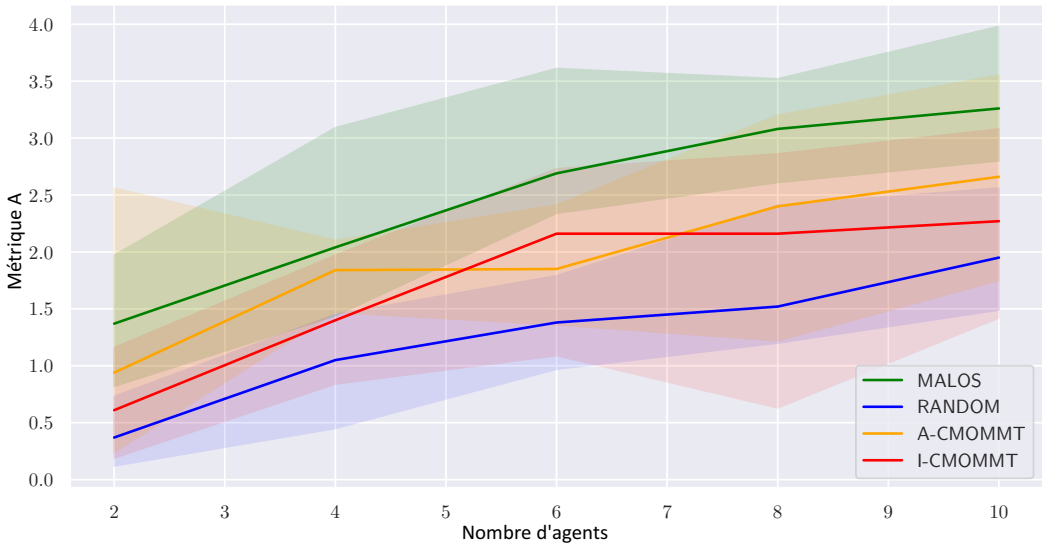


FIGURE 4.24 Évolution de la métrique A pour différents nombres d'agents.

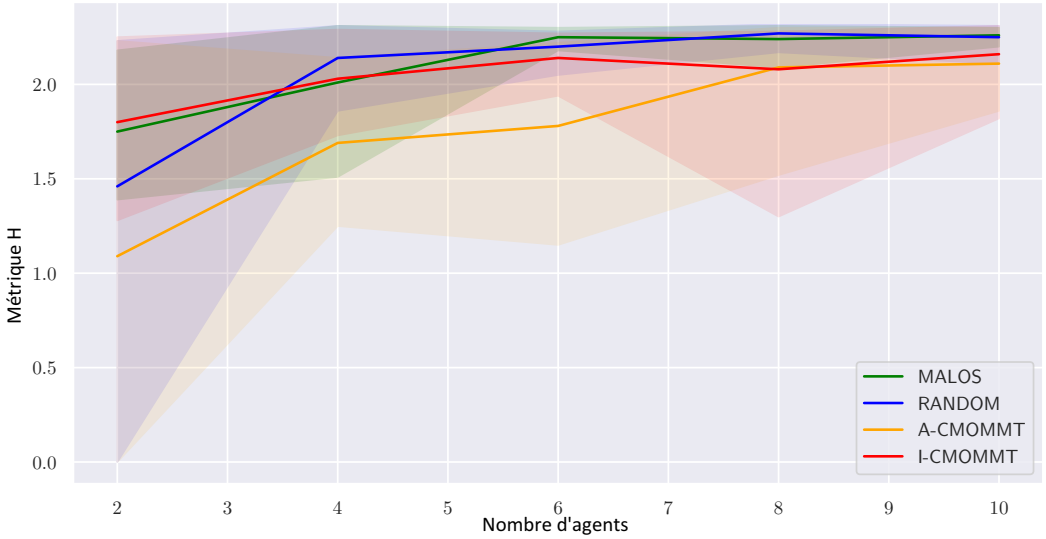


FIGURE 4.25 Évolution de la métrique H pour différents nombres d'agents.

l'A-CMOMMT, mais une amélioration significative de la répartition des cibles. La méthode MALOS, en vert, parvient à obtenir la meilleure moyenne des cibles observées par rapport aux autres méthodes évaluées. Dans le même temps, la répartition des observations est comparable à celle de la méthode aléatoire, ce qui indique un bon équilibre de l'observation entre les cibles.

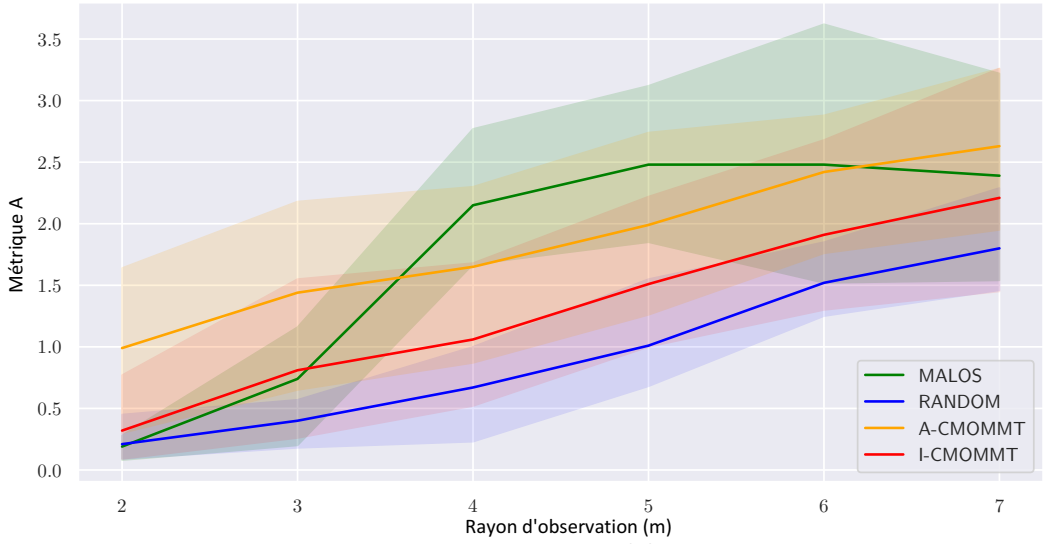


FIGURE 4.26 Évolution de la métrique A pour différents rayons d'observation.

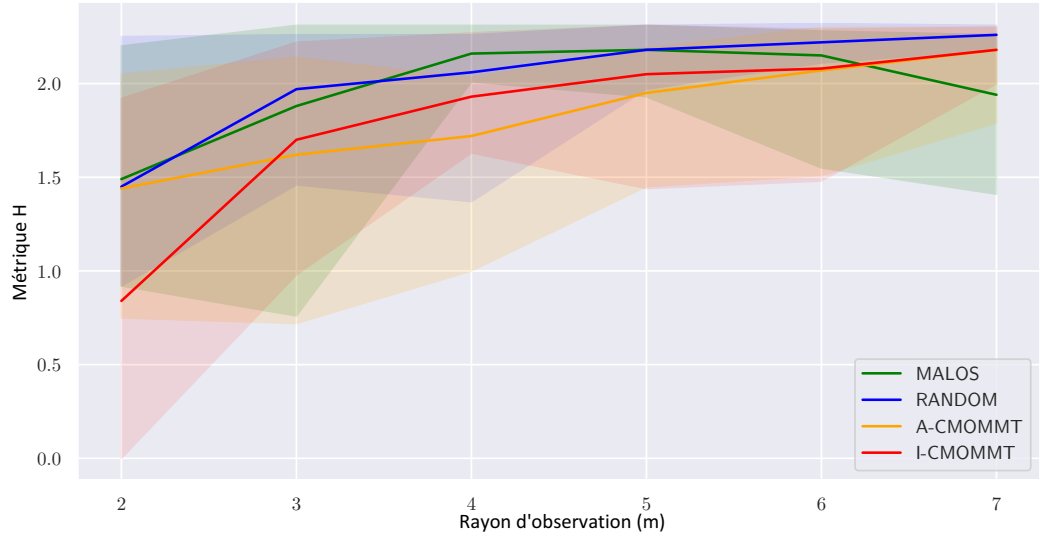


FIGURE 4.27 Évolution de la métrique H pour différents rayons d'observation.

Second scénario : Variation du rayon d'observation

Au sein de ce scénario, l'efficacité de MALOS est évaluée avec plusieurs rayons d'observation. Dans ce cas, le nombre d'agents et le nombre de cibles sont fixés et sont tous deux égaux à $m = n = 5$. Les figures 4.26 et 4.27 illustrent les résultats obtenus avec toutes les méthodes comparées respectivement à la métrique A et à la métrique H.

Lorsque le rayon d'observation se situe entre 4 et 6 mètres, la méthode MALOS est plus efficace pour les deux métriques. Cependant, son efficacité diminue par rapport aux

autres méthodes lorsque le rayon d'observation est en dehors de cet intervalle. Ces résultats peuvent être expliqués par le fait que MALOS a été entraîné uniquement avec une valeur d'observation fixe de 5 mètres. Contrairement à l'expérience précédente, où MALOS a été entraîné sur différentes configurations d'agents, le modèle a été formé avec un rayon d'observation constant. Par conséquent, MALOS se distingue des autres méthodes lorsque la configuration de la mission se rapproche de celle sur laquelle il a été entraîné.

4.3.3 Conclusion

La méthode MALOS répond au problème de l'observation appuyée par la patrouille en cherchant à associer l'efficacité d'une approche centralisée avec la robustesse d'une coordination distribuée. Afin d'accomplir cela, un modèle est entraîné de manière supervisée pour reproduire le comportement de chaque agent suivant une stratégie centralisée. Par la suite, ce même modèle est employé de manière distribuée par chaque agent.

Une architecture fonctionnelle est établie pour détailler le fonctionnement des trois principales étapes : (1) La génération d'une base de données issue d'une méthode centralisée (2) L'apprentissage supervisé pour l'imitation des comportements des agents (3) Le déploiement et l'exécution du modèle de manière distribuée.

Les performances de MALOS sont évaluées en comparaison de la méthode A-CMOMMT, I-CMOMMT, et la méthode aléatoire. Dans un premier temps, en faisant évoluer le nombre d'agents, et dans un second temps, en réalisant une variation du rayon d'observation. Nous constatons que MALOS montre une meilleure performance, que ce soit dans l'observation moyenne des cibles, que dans la distribution de l'observation de ces dernières, pour des paramètres expérimentaux sur lesquelles la méthode a été entraînée. Toutefois, si un paramètre tel que le rayon d'observation prend une valeur qui n'a pas été prise en compte pendant la phase d'entraînement, MALOS perd en efficacité.

En ouverture, nous considérons qu'il serait intéressant d'incorporer l'évolution des paramètres du coefficient de priorité ω_n pendant la phase d'entraînement, tout en alimentant également le modèle avec ces valeurs en entrée. L'objectif est ainsi de permettre à l'expérimentateur de déployer MALOS en spécifiant les priorités entre les différents objectifs.

Chapitre 5

Contribution III : Définition d'une architecture d'aide à la décision dans le cadre du POP

L'efficacité d'une méthode à résoudre le POP peut être étudiée à travers de nombreuses métriques, comme énoncé au chapitre 3. Ces performances dépendent notamment des paramètres caractérisant l'environnement, tels que le nombre de cibles ou encore le rayon de communication entre les agents, mais aussi des paramètres intrinsèques employés par la méthode elle-même.

Il est intéressant pour un expérimentateur de connaître l'impact des paramètres intrinsèques et extrinsèques de la méthode employée sur son efficacité. Dans cette perspective, les objectifs tant techniques qu'économiques consistent à garantir les performances d'une mission à un niveau souhaité, tout en réduisant les coûts associés ou en optimisant les capacités techniques des agents en conséquence. Néanmoins, compte tenu de la diversité des paramètres à prendre en considération, il n'est pas évident de trouver un ensemble de paramètres pouvant assurer les contraintes d'efficacité souhaitées.

Pour illustrer la problématique que nous avons soulevée, voici quelques exemples de questions auxquelles nous cherchons à répondre, avec chacune un scénario spécifique :

— Combien a-t-on besoin d'agents, au minimum, pour s'assurer qu'en moyenne, les trois quarts des cibles soient continuellement observées avec la méthode du I-CMOMMT ?

— À quelle vitesse optimale les drones doivent-ils se déplacer afin de garantir que l'oisiveté maximale régionale ne dépasse pas un seuil critique avec la méthode du F2MARL ?

— À partir de quel nombre de cibles et à quelle vitesse de ces cibles, l'observation moyenne (métrique A) chute en dessous de 50 % avec la méthode du A-CMOMMT ?

Au sein de ce chapitre, nous proposons de développer un outil d'aide à la décision permettant de fournir à l'utilisateur un ensemble de paramètres optimisés qui satisferont les contraintes d'efficacité préalablement spécifiées. Ainsi, en précisant la méthode à évaluer, les paramètres fixes et les paramètres variables à optimiser, l'outil retourne l'ensemble des configurations optimales possibles validant les contraintes utilisateurs. Le chapitre est organisé comme suit : Tout d'abord, nous présentons les approches et techniques trouvées dans la littérature qui se rapprochent de notre problématique. En utilisant ces recherches, nous proposons ensuite une architecture fonctionnelle pour notre outil. Enfin, nous évaluons cette architecture à travers l'expérimentation de plusieurs scénarios.

5.1 Les approches précédentes

Un outil d'aide à la décision nécessite une connaissance précise du comportement du système afin de conseiller sur le meilleur choix à faire. Le comportement du système peut être compris à travers un modèle généré par l'apprentissage automatique qui se base sur des conditions réelles [109] ou des simulations [122], pour identifier les causalités et les impacts de chaque paramètre sur les résultats.

La simulation d'un système complexe ou sophistiqué, comme le comportement de plusieurs agents et plusieurs cibles, peut être chronophage. Par conséquent, si l'outil d'aide à la décision cherche une solution optimale en expérimentant directement sur la simulation, l'obtention d'un résultat peut prendre beaucoup de temps. Pour résoudre ce problème, HONG et JIANG [52] propose d'entraîner un modèle prédictif à partir de plusieurs simulations, afin d'apprendre à reproduire leurs résultats. L'objectif est ainsi de fournir à l'outil des résultats plus rapidement lors de l'évaluation des solutions étudiées.

Les auteurs RUEDEN et al. [106] ont défini un cadre conceptuel combinant l'apprentissage automatique (ML) et la simulation. Ils ont identifié trois types de combinaisons : (1) ML assisté par simulation : Les données d'apprentissage d'un modèle sont issues exclusivement de la simulation. (2) Simulation assistée par ML : La simulation emploie un modèle d'apprentissage. (3) Hybride : L'apprentissage automatique et la simulation sont interdépendants, avec une relation symbiotique. Dans notre cas, nous nous plaçons dans la combinaison ML assistée par simulation, où l'apprentissage automatique entraîne son modèle directement à partir des données de la simulation. Une architecture d'outil de prise de décision est obtenue en ajoutant une fonctionnalité d'optimisation à la combinaison ML assistée par simulation.

Cette architecture a été appliquée précédemment dans le contexte de l'amélioration des problèmes de décision dans les terminaux de conteneurs. Ainsi, l'outil conçu par KASTNER et al. [60] fournit la meilleure quantité d'équipement associée aux politiques opérationnelles pour traiter un grand nombre de conteneurs dans un laps de temps court. Pour une application de planification de construction, les auteurs FENG, CHEN et LU [41] ont élaboré une architecture similaire, appelée *Machine Learning based simulation and optimization* (MSO), qui intègre une optimisation par essaim de particulaires (ou *particle swarm optimization* en anglais) pour trouver des plans de construction optimaux. TESTOLINA et al. [122] emploie un outil d'aide à la décision pour optimiser quatre paramètres d'une antenne, dans le but d'améliorer plusieurs métriques et contraintes au niveau du réseau en un temps réduit.

5.2 Architecture de l'outil

En nous inspirant des recherches précédentes, nous proposons une approche en trois étapes pour le fonctionnement de notre outil d'aide à la décision :

— Étape 1 : L'utilisateur spécifie le paramétrage de la mission, avec les valeurs fixes et variables, ainsi que la méthode à optimiser. De ce paramétrage, une base de données est générée à l'aide de multiples simulations, afin de lier les paramètres d'entrée avec les efficacités obtenues.

— Étape 2 : Un modèle est entrainé, par apprentissage supervisé, afin d'approcher la corrélation entre les paramètres d'entrée et les performances obtenues en sortie.

— Étape 3 : Obtention de la configuration optimale, grâce à un algorithme d'optimisation exploitant le modèle entraîné précédemment.

Ces trois grandes étapes s'intègrent dans l'architecture de l'outil d'aide à la décision présentée au sein de la figure 5.1. Les sections suivantes décrivent, point par point, chaque aspect de cette architecture.

5.2.1 Étape 1 : Paramétrage et génération d'une base de données

Paramètres d'entrée Les paramètres d'entrée, détaillés au chapitre 3, correspondent à l'ensemble des paramètres définissant le formalisme du POP. L'utilisateur y précise la valeur des paramètres fixes ainsi que les intervalles des paramètres variables à optimiser.

Les critères d'évaluation L'utilisateur choisit un ou plusieurs critères d'évaluation et indique la performance souhaitée pour le scénario donné. Ces critères peuvent être liés à la problématique de l'observation, de la patrouille ou du POP, présentés respectivement au sein

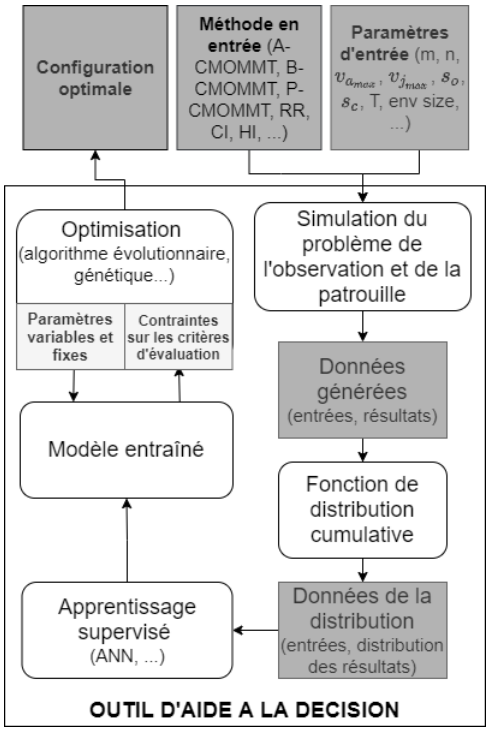


FIGURE 5.1 Architecture de l'outil d'aide à la décision par apprentissage

des sections 2.1.2, 2.2.2, 3.1.3. Par exemple, l'oisiveté moyenne de l'environnement doit être inférieure à 800s. Ce sont ces mêmes performances qui seront stockées au sein des bases de données.

La méthode à évaluer L'utilisateur est libre de choisir la méthode pour la résolution du POPqu'il souhaite évaluer. Ces méthodes sont répertoriées dans le chapitre 2 sur l'état de l'art, ainsi que dans les chapitres 3 et 4 qui traitent des méthodes développées dans cette thèse.

Simulation du POP Le bloc de simulation permet de générer une base de données (nommée « données générées »), liant les paramètres d'entrée avec les performances obtenues. Pour accomplir cela, les paramètres d'entrée définis comme variables par l'utilisateur sont explorés avec toutes les valeurs possibles, en utilisant un pas d'incrémentation préalablement défini. Cependant, pour une même configuration en entrée et une même méthode, la résolution du POP peut retourner des efficacités différentes. Ce phénomène provient notamment du positionnement initial aléatoire des agents et des cibles, mais aussi du comportement des cibles adoptant une stratégie aléatoire. Afin d'obtenir une représentation de la diversité possible des résultats, plusieurs simulations sont exécutées pour chaque configuration. L'uti-

lisateur est libre du logiciel de simulation à utiliser, ainsi que le nombre de simulations à exécuter par configuration et enfin du pas d'incrémentation pour les paramètres variables.

Données et fonction de distribution cumulative Un jeu de données est généré par les successives simulations, afin d'enregistrer la relation obtenue entre les configurations en entrée et les efficacités en sortie. Cependant, comme vu précédemment, un même ensemble de paramètre en entrée peut générer une variété de résultats possibles. Nous ne faisons aucune hypothèse sur la nature de la distribution et supposons qu'elle est non paramétrée.

Pour analyser la répartition des résultats, nous suggérons de les représenter sous la forme d'une fonction de distribution cumulative (FDC) empirique. De cette manière, pour chaque configuration d'entrée évaluée à plusieurs reprises, la distribution des résultats est représentée par une FDC. La fonction de distribution cumulative, ou encore nommée fonction de répartition, notée $F_X(x)$, a la caractéristique de fournir la probabilité P pour une variable aléatoire X de prendre une valeur inférieure ou égale à x. Cette propriété est décrite par l'équation suivante :

$$F_X(x) = P(X \leq x)$$

De la même manière, nous pouvons obtenir la probabilité pour la variable aléatoire X d'être supérieure ou égale à x :

$$P(X > x) = 1 - F_X(x)$$

Dans notre cas d'application, la fonction de distribution cumulative est utilisée pour obtenir la probabilité estimée P qu'une configuration donnée puisse générer des résultats supérieurs ou égaux à une efficacité spécifique x.

La FDC est construite empiriquement à partir des données brutes du bloc de simulation. De plus, étant donné que le jeu de données est fini, nous pouvons améliorer sa précision en utilisant la technique de l'estimation par noyau (ou en anglais *Kernel Density Estimation* (KDE)) pour approximer au mieux la distribution réelle. Au sein du chapitre sur l'expérimentation (cf. 5.3), nous comparons et évaluons les estimations avec et sans affinement par noyau. Grâce à la FDC, un nouveau jeu de données est généré. Pour chaque ensemble d'entrée, la distribution de chaque efficacité à évaluer est discrétisée en 11 éléments. Ainsi, nous représentons la FDC pour une probabilité $p_0 = 0$ jusqu'à $p_{11} = 1$ avec un pas $\Delta p = 0.1$.

5.2.2 Étape 2 : Modèle d'apprentissage

Un modèle est entraîné par un apprentissage supervisé afin d'identifier les relations entre les paramètres d'entrée et les valeurs discrétisées de la fonction de distribution cumulative. Chaque méthode évaluée en entrée correspond à l'entraînement d'un modèle. Pour une première expérience, nous proposons d'utiliser un réseau de neurones artificiel, où chaque paramètre en entrée correspond à un neurone sur la première couche, et chaque efficacité à évaluer par 11 neurones sur la couche de sortie, correspondant aux points discrétisés de la FDC. La structure et le nombre de couches cachées, ainsi que les fonctions d'activation, dépendent du scénario évalué et de l'expérience de l'utilisation. Ainsi, l'objectif du modèle entraîné est de pronostiquer la distribution des efficacités pour une configuration donnée.

5.2.3 Étape 3 : Optimisation et configuration optimale

Le bloc de l'optimisation représente la dernière partie de l'architecture. Pour obtenir la configuration optimale, l'utilisation précise en amont :

— Les paramètres variables à maximiser ou minimiser
— Les contraintes sur critères d'évaluation x à respecter
— Les probabilités désirées p_d pour lesquelles la configuration optimale retournée doit respecter les contraintes de performance x. Par exemple, $p_d = 0.5$ correspond à un minimum d'une expérience sur deux validant la contrainte spécifiée.

Nous nous situons dans le cadre d'une fonction d'optimisation continue, où les variables d'entrée sont des nombres réels. La littérature propose plusieurs algorithmes d'optimisation pour ce contexte. Par exemple, les algorithmes évolutionnaires s'inspirent de l'évolution biologique, impliquant des mutations et des sélections afin de réaliser une forme d'optimisation. Les auteurs Li et al. [73] ont réalisé une étude globale de l'état de l'art concernant les algorithmes d'optimisation, ainsi que les considérations pratiques à prendre en compte. L'utilisateur a la liberté de sélectionner l'algorithme d'optimisation qui correspond le mieux à son scénario spécifique. Cela dépend du nombre de fonctions objectif, des paramètres variables et de son propre niveau d'expérience. Cependant, il est important de noter que selon l'algorithme choisi, la solution obtenue peut être un optimum local plutôt qu'un optimum global.

5.3 Expérimentations et résultats

L'outil d'aide à la décision proposé est polyvalent et peut être utilisé dans diverses applications et scénarios. Nous avons réalisé deux scénarios distincts. L'implémentation

des algorithmes d'optimisation s'appuie sur la librairie Python Pymoo [20], tandis que l'apprentissage est effectué à l'aide de la bibliothèque Tensorflow [1].

5.3.1 Scénario 1 : Un paramètre à optimiser

Nous considérons le scénario suivant : l'utilisateur possède 10 drones et souhaite minimiser le nombre d'agents à déployer sur une mission, tout en assurant une couverture de l'environnement permettant de maintenir une oisiveté moyenne inférieure à 800 secondes avec la méthode du *Closest Idleness* (cf. section 3.2.4).

Dans ce contexte, le paramètre d'entrée variable est le nombre d'agents, que nous faisons varier de 1 à 10, avec un pas de 1. La contrainte utilisateur est la métrique de l'oisiveté moyenne sous les 800 secondes. Nous proposons d'exécuter la simulation 50 fois par configuration, et étudions les résultats pour une probabilité $p_1 = 0.5$, $p_2 = 0.9$ et $p_3 = 1.0$. Le tableau 5.1 contient l'ensemble des variables fixes et variables définissant ce scénario.

TABLE 5.1 Scénario 1 : Paramètres de simulation.

Paramètre	Type	Valeur
Taille de l'environnement	Fixe	$50\ m \times 50\ m$
Durée d'une mission T	Fixe	$1\,800\ s$
Rayon d'observation	Fixe	$4\ m$
Rayon de communication	Fixe	$5\ m$
Vitesse maximale des agents $v_{a_{max}}$	Fixe	$2\ m/s$
Nombre d'agents m	Variable	De 1 à 10

Nous proposons un modèle d'apprentissage basé sur un réseau de neurones artificiels, qui comporte une seule entrée correspondant au nombre d'agents, et onze sorties correspondant à la prédiction de la diversité de l'oisiveté moyenne. De plus, nous considérons trois couches cachées, constituées de 256 neurones. Les couches utilisent une fonction d'activation en *tanh*, tandis que la couche de sortie utilise une fonction d'activation linéaire. La figure 5.2 illustre les prédictions apprises par le modèle pour seulement trois probabilités, dans le but de faciliter la lecture et la compréhension.

Pour la dernière phase d'optimisation, nous avons sélectionné l'algorithme mono-objectif *Covariance matrix adaptation evolution strategy* (CMA-ES), tel que proposé par Hansen [47]. Les configurations optimales, obtenues en utilisant les distributions cumulatives empiriques (FDCE) et lissées par l'estimation par noyau (KDE), sont présentées dans le tableau 5.2. Les résultats sont arrondis, car le nombre d'agents est un nombre entier.

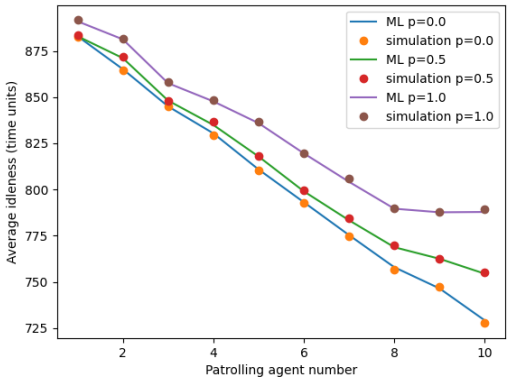


FIGURE 5.2 Scénario 1 : Prédiction par apprentissage de la distribution des résultats.

TABLE 5.2 Scénario 1 : Nombre optimal d'agents obtenu.

Distribution \ Probabilité	$p_1 = 0.5$	$p_2 = 0.9$	$p_3 = 1$
KDE	$6.17(=7)$	$6.54(=7)$	$8.58(=9)$
FDCE	6	$6.55(=7)$	$7.24(=8)$

Afin de vérifier si la configuration optimale fournie par l'outil satisfait les critères de performance spécifiés par l'utilisateur tout en minimisant le nombre d'agents, nous avons effectué 200 simulations avec ladite configuration. Les résultats de ces simulations sont présentés dans le tableau 5.3.

TABLE 5.3 Scenario 1 : Performances issues des paramètres optimaux.

Nombre d'agents	Pourcentage où $I^{av} < 800s$	I^{av} moyenne (s)
9	100%	763
8	97%	782
7	94%	787
6	30%	804

À la lecture des deux tableaux 5.2 et 5.3, l'outil d'aide à la décision a réussi à fournir les bonnes configurations pour les différentes probabilités souhaitées en utilisant la distribution KDE. Cependant, avec la distribution FDCE, les 8 agents recommandés n'ont pas été suffisants, ce qui se traduit par un taux de réussite de 97% au lieu de 100%, avec un maximum de moyenne d'observation de 811 secondes.

Afin d'obtenir la configuration optimale, l'algorithme d'optimisation a nécessité l'évaluation de 3 810 configurations, tandis que l'entraînement du modèle n'a nécessité que

500 simulations. Ainsi, nous pouvons en déduire que l'outil d'aide à la décision a permis d'économiser 3 310 simulations.

5.3.2 Scénario 2 : Deux paramètres à optimiser

Au sein du second scénario envisagé, l'utilisateur dispose de 3 agents, et cherche à savoir à partir de combien de cibles, et avec quelles vitesses de ces cibles, l'observation moyenne (métrique A) chute en dessous de 50%, avec la méthode du A-CMOMMT ? La probabilité désirée est $p = 0.9$. Le tableau 5.4 décrit en détail les paramètres de ce scénario.

TABLE 5.4 Scénario 2 : Paramètres de simulations.

Paramètre	Type	Valeur
Taille de l'environnement	Fixe	100 $m\times$ 100 m
Durée d'une mission T	Fixe	3 600 s
Rayon d'observation	Fixe	5 m
Rayon de communication	Fixe	3 m
Vitesse maximale des agents $v_{a_{max}}$	Fixe	2 m/s
Nombre d'agents m	Fixe	3
Nombre de cibles n	Variable	De 1 à 5
Vitesse maximale des cibles $v_{j_{max}}$	Variable	De 1 à 5 m/s
A-CMOMMT (d_{o1}, d_{o2}, d_{o3})	Fixe	(1 m, 1.5 m, 5 m)
A-CMOMMT (d_{r1}, r_{r2})	Fixe	(0.5 m, 1 m)

Le réseau de neurone possède deux entrées, correspondant au nombre de cibles et leurs vitesses, et de onze sorties, représentant les différentes probabilités de la métrique A. En dehors de ces changements, la structure du réseau est identique au scénario précédent. Avec deux fonctions objectives à maximiser, le nombre de cibles et la vitesse, avec une seule contrainte de performance, nous avons utilisé l'algorithme d'optimisation NSGA-II[35] (Non-dominated Sorting Genetic Algorithm-II). La population a une taille de 100, avec des chromosomes d'une taille de 2, correspondant aux paramètres à optimiser.

Les configurations optimales obtenues, en considérant la métrique sur la moyenne d'observation normalisée A, sont les suivants :

— Pour le FDCE : 1 cible avec une vitesse de 5 m/s, ou 2 cibles avec une vitesse de 1.5 m/s.

— Pour le KDE : 1 cible avec une vitesse de 5 m/s, ou 2 cibles avec une vitesse de 1.3 m/s.

Afin de valider ces paramètres optimaux, 200 expériences ont été menées pour chacune des configurations ci-dessus. Les résultats sont présentés au sein du tableau 5.5.

TABLE 5.5 Scénario 2 : Performances issues des paramètres optimaux., avec A_n l'observation moyenne normalisée.

Nombre de cibles	Vitesse des cibles (u/tu)	Pourcentage p où $A_n > 0.5$	A_n moyen
1	5	84.5%	0.796
2	1.3	85%	0.776
2	1.5	77%	0.738

Après analyse des résultats du tableau 5.5, le nombre d'expériences respectant les contraintes (84.5% et 85% pour les deux configurations respectivement) sont proches de la probabilité désirée de 90% mais ne satisfont pas intégralement le besoin de l'utilisateur. Ceci est en particulier dû à la haute variation de la métrique A, pour la méthode du A-CMOMMT, entre deux expériences avec la même configuration d'entrée. Pour améliorer les résultats de l'outil d'aide à la décision, il semble nécessaire d'affiner la représentation de la distribution des efficacités dans le modèle entraîné en augmentant le nombre de simulations.

5.4 Conclusion, limites et améliorations possibles

L'objectif de l'outil d'aide à la décision est de permettre à l'utilisateur d'obtenir une optimisation des paramètres d'une mission, tout en prenant en compte les objectifs d'efficacité à atteindre. Pour ce faire, une architecture fonctionnelle a été développée, reposant sur l'utilisation de simulations et de l'apprentissage supervisé d'un modèle, afin d'apprendre la corrélation entre les paramètres en entrée, et les efficacités en sortie. Enfin, l'outil utilise un algorithme d'optimisation pour fournir les paramètres optimaux en se basant sur le modèle précédemment appris.

Cependant, comme nous l'avons constaté, l'outil d'aide à la décision peut présenter des imprécisions lorsqu'une métrique présente une forte variation. Il serait intéressant que l'outil puisse détecter les écarts types des métriques afin d'ajuster le nombre de simulations nécessaires pour obtenir une meilleure représentation de leurs distributions. Pour étayer cette idée, dans le scénario 2, nous avons observé que l'apprentissage s'est basé sur 50 simulations, un nombre qui s'est avéré insuffisant pour garantir des paramètres optimaux respectant les contraintes utilisateur sur 200 évaluations. De plus, l'architecture du réseau de neurone peut mener à un sur-apprentissage (*overfitting*) ou un sous-apprentissage (*underfitting*) de la distribution à apprendre. Il serait tout aussi intéressant d'étudier ce comportement.

Chapitre 6

Expérimentation sur drones en volière

Ce chapitre est dédié à expliciter la mise en place d'expérimentation sur drone des approches développées, en comparaison avec certaines méthodes de la littérature. Pour atteindre cet objectif, nous présentons dans un premier temps le framework, appelé ROS, permettant les interactions entre les agents, les cibles et avec l'environnement. Dans un second temps, nous mettons en place le simulateur Gazebo afin de valider le comportement des agents et des cibles, en reproduisant leur dynamique de manière réaliste. Par la suite, nous présentons l'environnement de travail, afin de faire voler les drones au sein d'une volière équipée d'un système de positionnement d'intérieur. Pour conclure, nous expliquons les résultats obtenus lors des missions de vol.

Dans un souci de garantir la reproductibilité des expériences et des résultats, les codes développés sont rendus open-source et sont accessibles sur la plateforme GitHub [1].

6.1 Simulation - ROS et Gazebo

6.1.1 Robot Operating System (ROS)

Robot Operating System (ROS) [104] est un framework de développement open-source largement utilisé dans le domaine de la robotique, offrant une plateforme flexible pour la création de systèmes robotiques en intégrant des fonctionnalités telles que la gestion des capteurs, le contrôle des actionneurs et la communication entre les différents composants.

Nous utilisons ROS pour faciliter et harmoniser les interactions entre les agents et l'environnement, mais aussi assurer la communication entre les agents. Au sein de ROS,

1. https://github.com/JamyChahal/uav_obs_pat

chaque entité, que ce soit un agent, ou un programme informatique, est appelé un nœud. Les nœuds interagissent entre eux en structurant les échanges par des messages. Un **message** est une structure de données standardisée utilisée pour l'échange d'informations entre les nœuds d'un système. Les messages sont utilisés pour assurer une normalisation des informations échangées. Au sein de ROS, il existe plusieurs types de communication permettant de répondre à un fonctionnement spécifique :

- **Les *topics*** : Un *topic* est un canal de communication unidirectionnel permettant l'échange de données entre les différents nœuds d'un système. Un *topic* permet d'échanger un seul type de message. Ainsi, un nœud est dit *publisher* lorsqu'il publie des messages sur un *topic*, et *subscriber* lorsque le nœud reçoit les messages d'un *topic*. Les *topics* sont particulièrement prisés pour envoyer un flux d'information continue, comme le flux d'un capteur.
- **Les services** : Contrairement aux topics, où les messages sont échangés de manière asynchrone, les services permettent des échanges de données synchrones où un nœud client envoie une requête (*request*) à un nœud serveur, qui traite la requête et renvoie une réponse (*response*) au nœud client. Ces interactions sont utiles pour les échanges nécessitant un retour, comme l'exécution d'une commande de décollage et attendre le retour de la réussite ou de l'échec.
- **Les actions** : Les actions sont comparables aux services, avec une interaction client/serveur, à la différence que les communications sont asynchrones. Le client envoie une demande d'action (goal) au serveur, qui exécute la tâche en arrière-plan. Pendant l'exécution, le serveur peut envoyer des mises à jour d'état (*feedback*) au client pour l'informer de l'avancement de la tâche. Une fois la tâche terminée, le serveur envoie une réponse (*result*) au client pour indiquer le succès ou l'échec de l'action. Ces interactions permettent par exemple de s'assurer que la mission d'un drone se déroule comme prévu, en obtenant des retours réguliers de sa part.

La première structure de ROS, appelé ROS1, repose sur une architecture centralisée. Les nœuds peuvent interagir entre eux à la condition d'être connecté avec un nœud central, appelé le nœud *master*. Cette structure pose un problème dans le scénario envisagé, où les drones ne sont pas supposés être continuellement lié avec un nœud commun, qu'il soit embarqué au sein d'un drone ou d'un ordinateur central, appelé également *Ground Control Station* (GCS). Pour remédier à cette problématique, nous adoptons la deuxième version de ROS, connue sous le nom de ROS2 [83]. Grâce à son architecture distribuée, les nœuds peuvent interagir directement les uns avec les autres, sans avoir besoin d'être connectés à un nœud central. Au sein de cette thèse, nous utilisons la version ROS2 Galactic développée par MACENSKI et al. [79], reposant sur Ubuntu 20.04 LTS.

6.1.2 Gazebo

Gazebo est une plateforme de simulation open-source utilisée pour la modélisation et la simulation de robots. Ce simulateur offre la possibilité de créer des modèles de robots et d'environnements personnalisés, ainsi que de simuler les interactions physiques entre les robots, les objets et l'environnement. De plus, Gazebo permet le développement de *plugin*, permettant de connecter l'environnement et les robots avec ROS, afin par exemple de publier sur un *topic* les données issues d'un capteur.

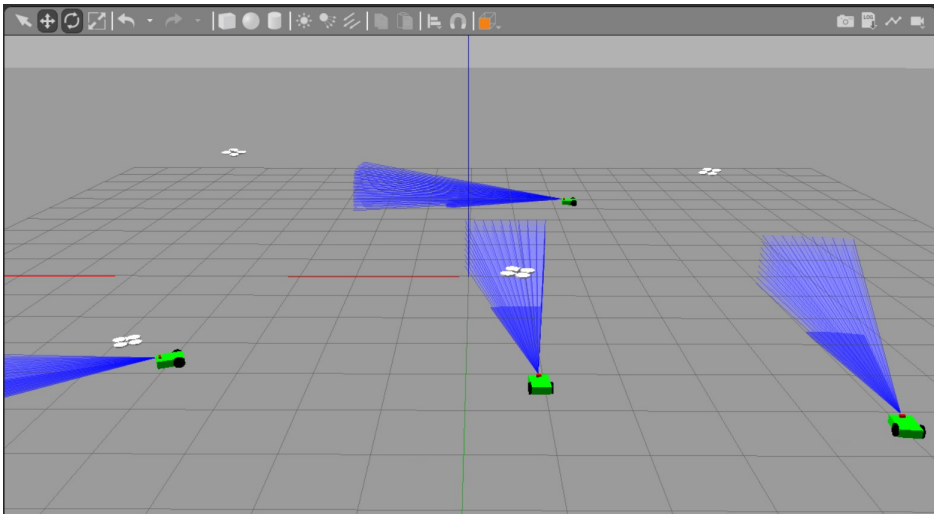


FIGURE 6.1 Représentation de l'environnement Gazebo, avec les drones et les cibles mobiles au sol. Les capteurs de distance des cibles sont illustrés par des faisceaux bleus.

Architecture d'interaction avec le drone via un plugin

La modélisation des drones, ainsi que leurs dynamiques, sont obtenus grâce au *plugin* open-source "tello_ros" [2] développé par Clyde McQueen. En modifiant et adaptant ce *plugin* à nos besoins, il devient possible d'interagir avec un drone Tello, que ce soit en simulation ou en situation réelle.

Afin qu'un drone puisse accomplir une mission avec une méthode spécifique, il est essentiel de mettre en place une architecture reliant les fonctionnalités requises, chacune étant représentée par un nœud. Les interactions entre ces nœuds sont illustrées par la figure 6.2 sous forme de graphe. Les nœuds inclus dans ce contexte fonctionnent de la manière suivante :

– **/droneX_strategy** : Ce nœud a pour fonctionnalité d'exécuter la méthode (par exemple I-CMOMMT) à appliquer au drone. Pour ce faire, une fois que le nœud a

2. https://github.com/clydemcqueen/tello_ros

obtenu les informations nécessaires au sujet de son environnement, comme détaillé à la section suivante, la stratégie est mise en œuvre en publiant une commande en vitesse sur le *topic* correspondant ***/droneX/cmd_vel***

- **/droneX/TelloPlugin** : Le *plugin* permet l'interaction avec le drone. Pour ce faire, il écoute la commande en vitesse sur le *topic* associé, puis le transmet au drone lors d'une expérimentation en vol réel, ou simule la dynamique du drone sous Gazebo. Par ailleurs, ce nœud permet de publier l'état du drone sur ***/droneX/flight_data***, et dans le cas de la simulation, également sa position sur ***/droneX/pose***.
- **/evaluator** : Ce nœud écoute la position de tous les drones et toutes les cibles afin de calculer et d'enregistrer les métriques d'évaluation de la méthode expérimentée.

Par ailleurs, le nœud **/drone0_strategy** utilise le service ***/droneX_action*** mise en place par le *plugin* afin de faire décoller ou atterrir le drone. En retour, **/droneX/TelloPlugin** informe si l'action s'est bien passée, si la manœuvre n'a pu être réalisée, ou s'il y a eu une erreur de communication avec le drone. La manœuvre n'est pas traitée si, par exemple, une demande de décollage a été émise tandis que le drone est déjà en vol.

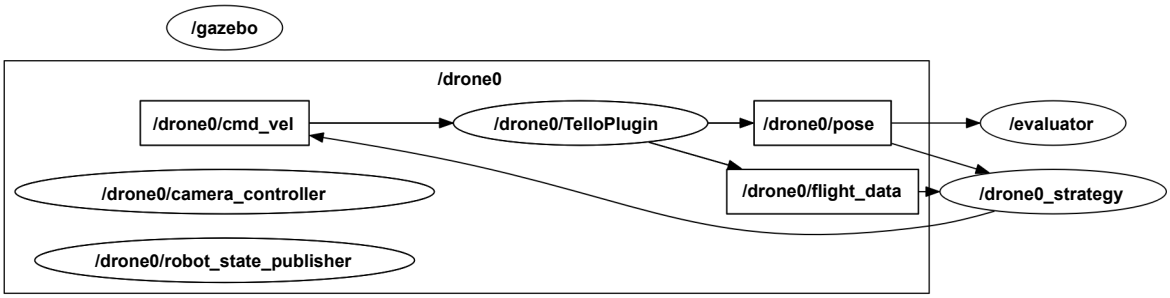


FIGURE 6.2 Représentation des interactions entre les nœuds au sein d'un drone. Les nœuds sont représentés par des ellipses, tandis que les *topics* sont représentés par des rectangles.

Architecture d'interaction entre les agents et les cibles

Pour mener à bien sa mission, un agent, ou un drone, a la possibilité de communiquer avec ses compères positionnés dans son rayon de communication, et d'observer la position des cibles appartenant à son rayon d'observation. Ces interactions sont structurées selon des *topics* et des services prédéfinis. La figure 6.3 permet d'illustrer l'interaction d'un drone0 avec son environnement. Nous nous plaçons ici dans la situation où le drone peut communiquer avec un second drone /drone1, et observe deux cibles, /target0 et /target1. La majorité des *topics* ont une fréquence d'actualisation de 10Hz, et sont utilisés de la manière suivante :

- ***/droneX/map_idleness*** : Message std_msgs/Int32Array : Chaque drone publie sur ce *topic* sa carte d'oisiveté, structurée sous forme d'une matrice. Lorsqu'un drone est

en communication avec un second, il écoute également ce *topic* afin d'actualiser sa propre carte d'oisiveté.

- ***/droneX/pose*** : Message geometry_msgs/PoseStamped : Chaque drone écoute sa propre position, ainsi que celles des drones en communication afin d'appliquer sa méthode. La position est exprimée en mètre, dans le repère Gazebo ou celui de l'environnement.
- ***/droneX/desired_patrol_position*** : Message geometry_msgs/PoseStamped : Ce *topic* est utilisé exclusivement par la méthode I-CMOMMT pour partager la position que le drone désire patrouiller, comme détaillé au sein de la section 3.2.3.
- ***/droneX/cmd_vel*** : Message geometry_msgs/Twist : Permet de publier la commande en vitesse, exprimé en m/s ou rad/s, dans le repère propre du drone. Ainsi, les vitesses en translations sur l'axe x et y du drone sont dirigés respectivement vers l'avant et à la gauche du drone. La vitesse angulaire sur z permet de contrôler l'angle du lacet.
- ***/targetX/odom*** : Message nav_msgs/Odometry : La position des cibles est obtenue par le drone lorsque ces dernières appartiennent au rayon d'observation. Cette position est exprimée en mètre dans le repère Gazebo ou celui de l'environnement.
- ***/droneX/flight_data*** : Message tello_msgs/FlightData : L'état du drone est récupéré puis publié grâce au *Plugin* Tello. Ces informations contiennent, entre autres, le niveau de batterie, les angles issus du gyroscope et de l'accéléromètre (tangage, roulis, lacet), le niveau de température interne, l'altitude estimée par les capteurs internes ainsi que les données du baromètre et de l'accéléromètre.

6.2 Mise en place de l'expérimentation en volière

Dans cette section, nous abordons la transition de l'environnement de simulation à l'environnement expérimental en volière. La simulation joue un rôle essentiel en validant le comportement des drones, incluant leur stabilité, leur capacité à éviter les obstacles, à rester dans les limites de l'environnement, ainsi que l'application correcte des méthodes théoriques. Une fois ces aspects validés, l'expérimentation sur drone permet de progresser graduellement vers des niveaux de complexité plus élevés. En effet, la simulation suppose un environnement idéal où les communications sont sans contraintes, les mouvements des drones ne sont pas influencés par des forces externes comme le vent et le positionnement dans l'espace est précis.

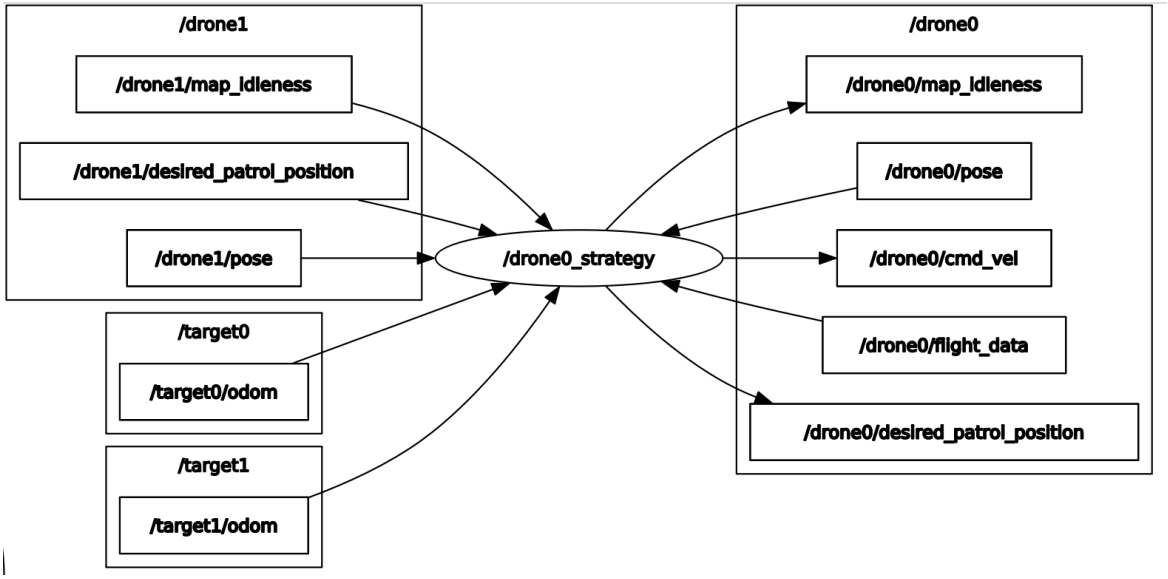


FIGURE 6.3 Représentation des interactions centrées sur un drone /drone0, avec un second drone en communication /drone1, et deux cibles observées /target0 et /target1. Les nœuds sont représentés par des ellipses, tandis que les *topics* sont représentés par des rectangles.

6.2.1 Système de positionnement

Le positionnement des drones et des cibles présente un défi technique. Pour assurer la sécurité de nos vols expérimentaux, nous avons opté pour la réalisation de nos expériences à l'intérieur d'une volière. Alors que les positions étaient aisément obtenues dans la simulation Gazebo grâce à l'utilisation d'un *plugin* dédié, les drones utilisent généralement un système GPS pour se localiser en extérieur. Cependant, le signal GPS n'est pas suffisamment fiable en vol intérieur, et sa précision ne convient pas aux contraintes d'une volière où l'espace est restreint.

Afin de localiser en temps réel les robots dans la volière, nous utilisons le système Vicon [3]. Ce système de positionnement utilise des caméras infrarouges pour suivre de manière précise les marqueurs placés sur les robots, permettant ainsi de déterminer leur position et leur orientation en temps réel. Le système discerne une cible d'un drone grâce à la disposition spécifique des marqueurs sur les robots, comme nous pouvons le voir sur les photos de la figure 6.7. L'ensemble des caméras est relié à un ordinateur dédié au système Vicon, calculant ainsi l'estimation de position et d'orientation.

L'illustration de la transmission des informations, du système Vicon jusqu'à la publication des positions des drones et des cibles sur les *topics* dédiés, est présentée dans la figure 6.4.

3. Les caméras Vicon sont du modèle Vero : https://www.vicon.com/hardware/cameras/vero/

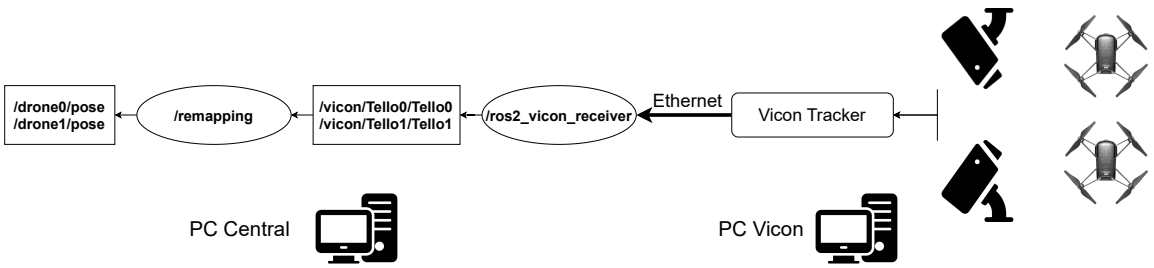


FIGURE 6.4 Flux d'information provenant du système Vicon.

Initialement, le logiciel Vicon Tracker est exécuté sur un ordinateur spécifique, permettant d'estimer la position des robots dans leur environnement à partir des flux d'images capturées par les caméras. Afin de garantir une transmission rapide des positions, cet ordinateur est connecté au même réseau physique que l'ordinateur central via une connexion Ethernet filaire. Dans un second temps, le nœud open-source **/ros2_vicon_receiver** [4] est responsable de récupérer les positions à l'aide de la bibliothèque DataStreamSDK_10.1, puis les publie directement, sans effectuer de traitement, sur les *topics* appropriés ***/vicon/TelloX/TelloX***.

Dans un troisième temps, le nœud **/remapping** assure que la publication des positions soit cohérente avec la structure employée en simulation. Ainsi, les positions en millimètres sont exprimées en mètre, structurées dans un message geometry_msgs/PoseStamped, et les *topics* possèdent les mêmes noms qu'en simulation. Enfin, si le système de positionnement Vicon n'arrive plus à localiser un robot, alors les *topics* vicon émettent une position aux coordonnées nulles. Le nœud **/remapping** ne republie pas ces informations afin que les agents et les cibles ne les considèrent pas comme de réelles positions. Par contre, lors de la perte de positionnement d'un drone ou d'une cible par le système, un message d'alerte est envoyé à l'expérimentateur pour l'avertir de la situation.

6.2.2 Environnement réseau

Pour communiquer avec un drone, il est nécessaire de se connecter à son réseau Wi-Fi dédié. Toutefois, lorsque plusieurs drones sont impliqués dans une mission, il est essentiel d'établir une architecture réseau spécifique pour assurer la communication avec l'ensemble des drones. Un ordinateur central est utilisé pour calculer la stratégie à mettre en œuvre pour chaque drone, cependant il ne peut pas se connecter aux réseaux Wi-Fi de chaque drone individuellement. Pour remédier à ce problème, la figure 6.5 représente l'architecture réseau proposée, où la communication avec chaque drone est assurée par un microcontrôleur

4. https://github.com/OPT4SMART/ros2-vicon-receiver

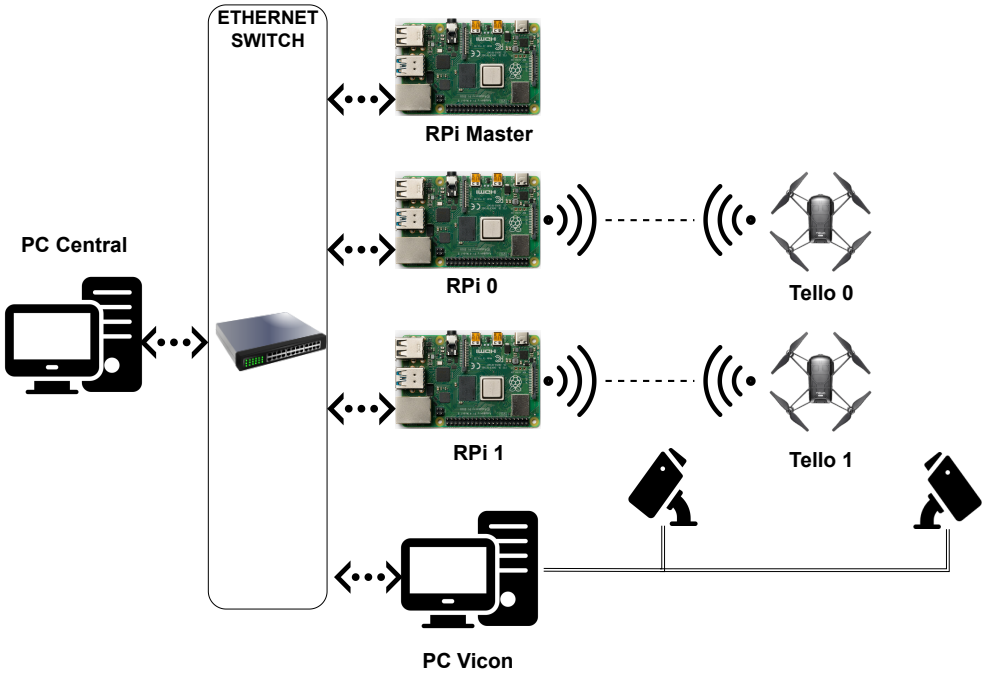


FIGURE 6.5 Architecture réseau entre les drones, le système de positionnement et l'ordinateur central.

Raspberry Pi. Ce dernier est relié à un réseau physique, via un commutateur réseau, ou *switch*, afin d'établir un pont entre l'envoi des commandes de la part du PC central et la réception des informations émanant du drone. La Raspberry Pi dite "master" permet de fixer les IP des autres Raspberry, du PC Central et du PC Vicon.

TABLE 6.1 Numéro de port associé à chaque drone.

	Par défaut	Drone 0	Drone 1
Envoi de commande	8889	3010	3020
Accusé de réception commande	38065	3011	3021
Flux état du drone	8890	3012	3022
Flux vidéo	11111	3013	3023

Dans le réseau interne de chaque drone, il existe des ports spécifiques pour chaque type d'interaction, tels que la réception de commandes, l'accusé de réception des commandes, le flux vidéo et le flux d'état du drone. Ces numéros de port UDP sont les mêmes pour tous les drones. Les Raspberry Pi ont pour rôle de rediriger le flux d'information vers des ports uniques et spécifiques à chaque drone sur le réseau physique, défini au sein du tableau 6.1. L'objectif est d'éviter tout conflit d'utilisation des ports.

6.3 Résultats expérimentaux

Cette section présente les résultats expérimentaux obtenus grâce au cadre expérimental défini à la section précédente. Les expérimentations ont eu lieu au sein de la volière de l'université d'Evry. L'utilité d'un vol en intérieur en volière est d'assurer un environnement sécurisé, en cloisonnant l'espace de vol des drones, comme montré sur la photo 6.6. Par ailleurs, les drones ne disposant pas de GPS, ils bénéficient du système de positionnement Vicon. L'emplacement des marqueurs sur les drones et les cibles est indiqué sur la photo 6.7.

FIGURE 6.6 Photo de la volière

Durant les expériences, les cibles sont représentées par des robots terrestres. Ces robots sont munis de trois roues, donc deux manœuvrables à l'arrière, pour assurer son déplacement et sa rotation. Les robots effectuent des déplacements aléatoires grâce à deux capteurs : un capteur ultrason, à l'avant, pour éviter les bordures de l'environnement et une centrale à inertie pour contrôler son orientation. La stratégie mise en œuvre consiste à choisir une orientation de manière aléatoire, puis à la maintenir. Pendant son déplacement, le robot a une probabilité de 5% de sélectionner une nouvelle orientation chaque seconde. Sinon, un nouveau cap est également choisi lorsque le robot rencontre un obstacle aux limites de l'environnement.

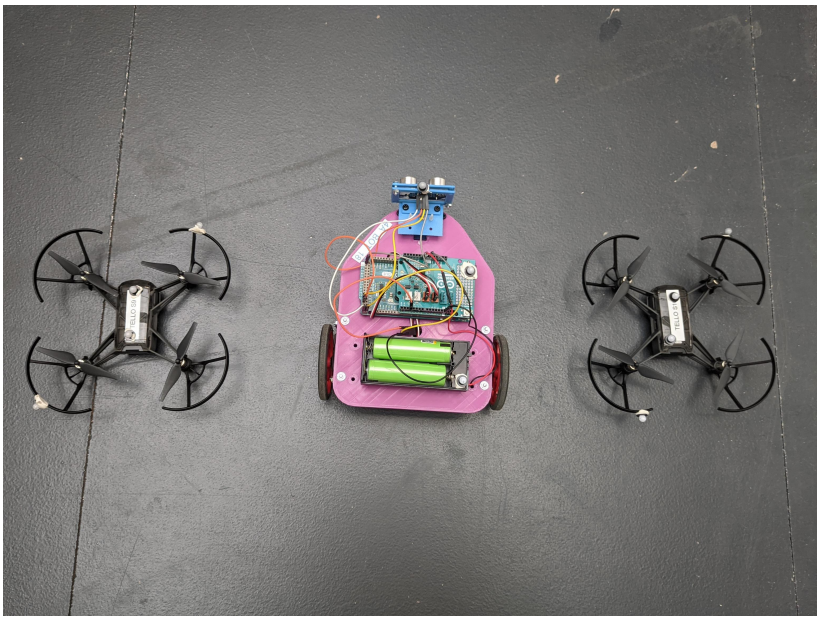


FIGURE 6.7 Photo de deux drones et d'une cible avec des marqueurs réflectifs (boules grises).

6.3.1 Déplacement aléatoire

La première étape de l'expérimentation a pour objectif de sécuriser les décollages, les atterrissages, y compris les atterrissages d'urgence du drone. Cela permet de confirmer le bon comportement du drone et de s'assurer d'une communication efficace entre le drone et l'ordinateur central. Par la suite, l'objectif est de réaliser des déplacements aléatoires à l'intérieur de l'enceinte de vol, dont la trajectoire d'une des expériences est présenté à la figure 6.8. Ces vols permettent de valider expérimentalement l'exactitude du positionnement du drone grâce au système de localisation, mais également de l'architecture réseau entre les différents modules présentés à la section précédente.

Enfin, ces déplacements valident le bon fonctionnement des boucles d'asservissement de position en boucle fermée, ce qui a pour conséquence de générer des translations fluides et stables du drone. Ce mécanisme d'asservissement repose à la fois sur la fréquence de mise à jour de la position, fixée à 100 Hz dans ce cas, ainsi que sur une précision de localisation satisfaisante, de l'ordre du centimètre pendant les expérimentations, et enfin sur la conception soignée du régulateur, un contrôleur PID en l'occurrence.

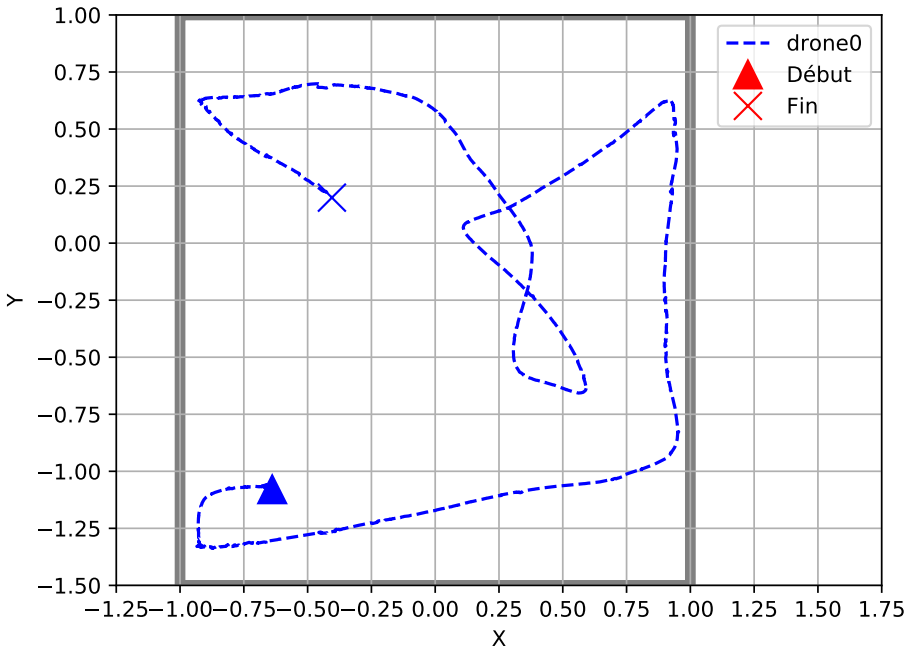


FIGURE 6.8 Trajectoire pour un seul drone se déplaçant aléatoirement.

6.3.2 Évitement de collision entre drones

Le passage d'une configuration mono-drone à un système multi-drones nécessite de réaliser puis valider une stratégie d'évitement d'obstacle efficace. L'enjeu est d'autant plus important que les drones sont des robots aériens rapides et majoritairement peu résistants au choc. Ainsi, que ce soit à cause de l'impact d'une hélice, ou de la chute du drone, les conséquences d'une collision sont particulièrement préjudiciables pour l'intégrité du matériel à bord.

Pour éviter tout risque de collision, la stratégie d'évitement consiste à employer un champ de force répulsif entre les agents, à l'instar des méthodes du A-CMOMMT et du I-CMOMMT (cf. respectivement les sections 2.1.3 et 3.2.1). La démarche expérimentale pour valider l'évitement de collision est de faire voler deux drones, et de demander un déplacement sur la même localisation. Deux paramètres caractérisent le champ de force :

— Le premier paramètre est appelé la distance de sécurité, notée DS, mais aussi dr_1 pour les méthodes CMOMMT. Elle correspond à la distance optimale à laquelle les agents doivent être les uns des autres. La force de répulsion s'active dès lors que la distance entre deux agents est inférieure à DS.

— Le second paramètre est appelé la distance dangereuse, notée DD, mais aussi dr_2 pour les méthodes CMOMMT. Cette distance, inférieure à DS, spécifie la longueur

minimale à laquelle deux agents peuvent se rapprocher. La magnitude des forces de répulsion entre les agents croît linéairement jusqu'à atteindre son maximum lorsque la distance est inférieure ou égale au seuil de dangerosité *DD*.

Les figures 6.9 et 6.10 illustrent les trajectoires des deux drones, dans le cas où la distance d'évitement est relativement importante, ici $DS = 1m$ et $DD = 0.5m$. Le point désiré se situe aux coordonnées $(x = 0m; y = 0.25m)$. À la lecture des trajectoires, le drone n°1, dont le tracé est en noir, effectue un déplacement vers la position avant de reculer pour maintenir la distance de sécurité. Le drone n°0, quant à lui, a effectué son décollage puis est resté stationnaire. Les deux drones désirant atteindre la même destination, mais repoussés par le champ de répulsion, restent ainsi statiques pour éviter toute collision.

L'espace de vol étant relativement restreint, nous avons expérimenté une distance de dangerosité plus faible, afin que les drones puissent se déplacer avec moins de contrainte. Ainsi, les figures 6.11 et 6.12 représentent les trajectoires de deux drones pour une valeur $DD = 0.25m$. Comme prévu, les drones s'approchent mutuellement afin de se placer à l'emplacement souhaité, jusqu'à ce qu'ils atteignent la distance dangereuse. À partir de ce point, les forces de répulsion deviennent plus intenses, induisant un recul symétrique des drones. Par la suite, les drones conservent une distance de sécurité en adoptant une position statique, ayant pour cause un équilibre entre les forces d'attraction et de répulsion.

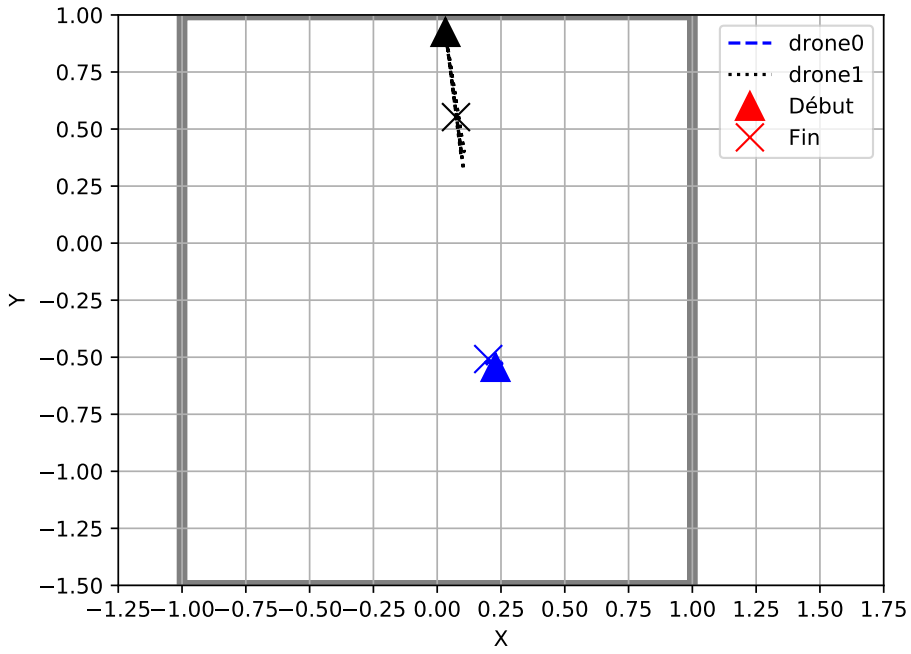


FIGURE 6.9 Trajectoire de deux drones avec une distance d'évitement élevée.

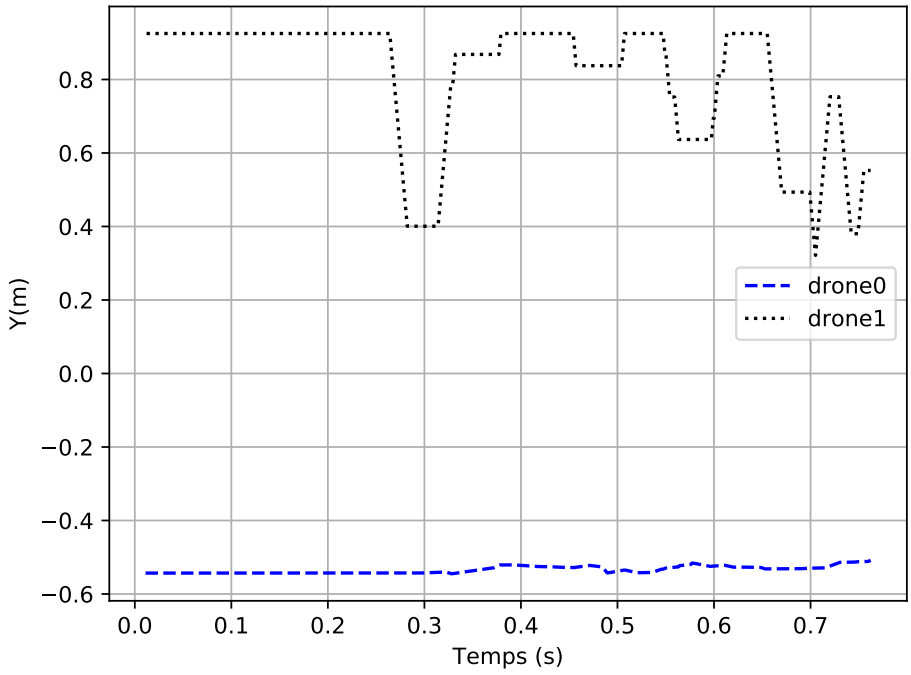


FIGURE 6.10 Déplacement des drones sur l'axe y, avec une distance d'évitement élevée.

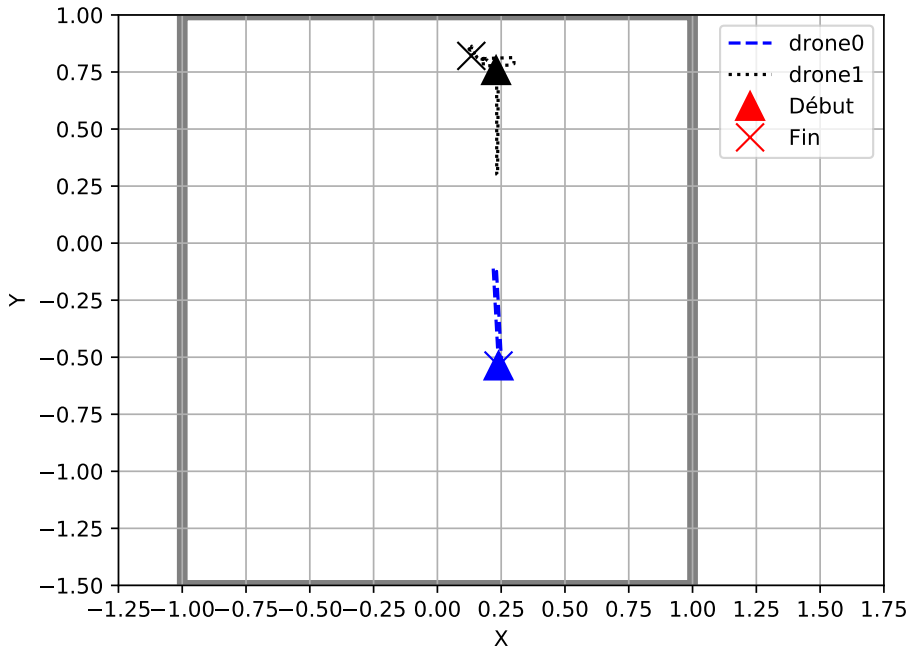


FIGURE 6.11 Trajectoire de deux drones, avec une distance d'évitement rapprochée.

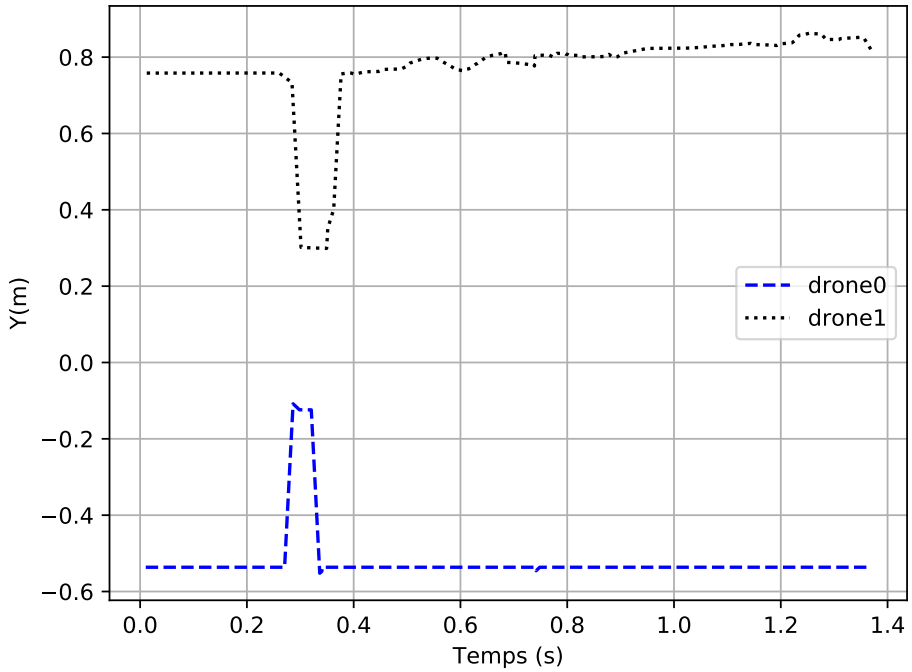


FIGURE 6.12 Déplacement des drones sur l'axe y, avec une distance d'évitement rapprochée.

6.3.3 Patrouille et suivi de cible du I-CMOMMT

La dernière expérience réalisée consiste à employer la méthode du I-CMOMMT, avec deux drones et une cible. La figure 6.13 montre les trajectoires de chaque entité. Le rayon d'observation est fixé à 0.25m, tandis que le rayon de communication est de 0.5m. L'environnement est défini sur x dans l'intervalle $[-1;1]$, et sur y dans l'intervalle $[-1;2]$. Au départ de la mission, un drone décolle à proximité de la cible, tandis qu'un second est placé au hasard dans l'environnement, afin d'étudier le comportement de suivi et de patrouille de la méthode. La cible adopte un déplacement aléatoire dans l'environnement.

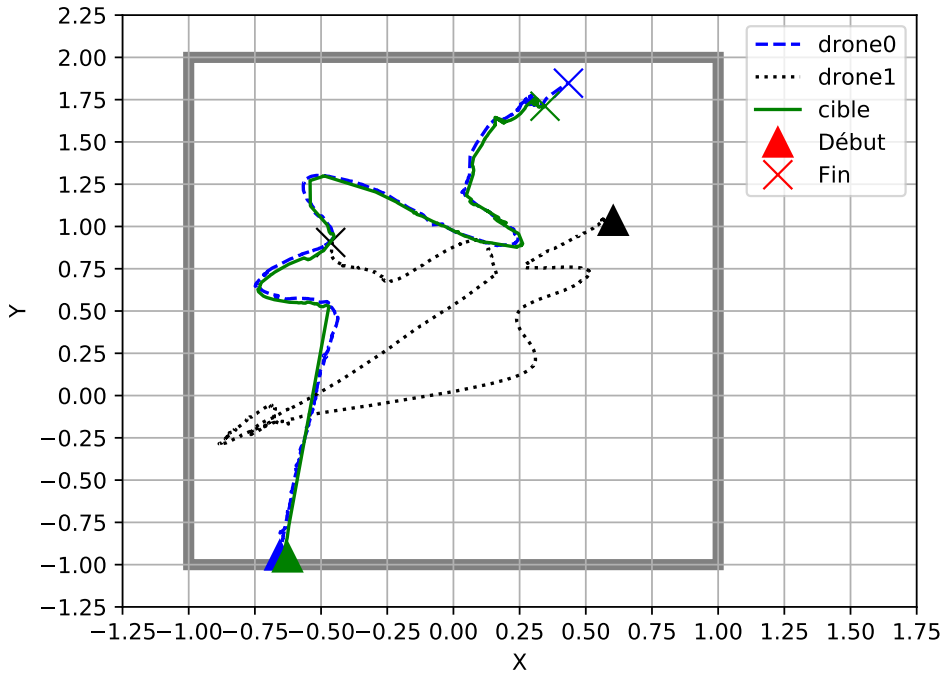


FIGURE 6.13 Trajectoire de deux drones et d'une cible, avec la méthode I-CMOMMT.

À la lecture des trajectoires, le drone n°0 dont le tracé est en bleu, effectue correctement le suivi de la cible dont le tracé est représenté en vert. Tandis que le drone n°1, n'ayant pas dans son champ d'observation la cible, effectue une patrouille de l'environnement. Il est par exemple intéressant de noter que ce drone marque un arrêt aux coordonnées $(0.2;0.75)$, puis change de cap, réalisant que la zone face à lui a déjà été patrouillé récemment. Cependant, le tracé du drone n°1 est parfois perturbé par la précision du système de positionnement aux abords de l'environnement. En outre, en raison du rayon de communication relativement restreint, les deux drones n'ont jamais eu l'opportunité d'échanger des informations, ce qui explique la patrouille dans des zones déjà visitées par le drone n°0.

6.3.4 Les difficultés rencontrées

La mise en œuvre des expériences s'est heurtée à de nombreuses difficultés techniques, que ce soit une mauvaise estimation de l'angle du lacet des drones, ou encore une mauvaise détection des obstacles par les robots terrestres. Tous ces défis techniques ont été résolus par des solutions d'ingénierie, à l'exception de deux problèmes majeurs :

— Interférences Wi-Fi : La communication avec les drones est réalisée grâce au Wi-Fi 2.4 GHz. Malheureusement, ces communications ont souvent été perturbées par les autres réseaux Wi-Fi présents dans le laboratoire, entraînant des délais significatifs dans les échanges entre les drones et l'ordinateur central. Le PC central agit ici comme une boucle de rétroaction, donnant des instructions aux drones en fonction de leurs positions, des positions des cibles et des autres agents. Par conséquent, un délai important impacte directement la réactivité des drones.

— Luminosité du lieu : En fonction de la météo, et de l'heure de la journée, la lumière issue du Soleil inonde les locaux du laboratoire. Cette forte luminosité, tant dans le spectre visible que dans l'infrarouge, entraîne des difficultés de détection des marqueurs réfléchissants embarqués sur les drones et les robots.

En raison de l'impact significatif de ces deux problèmes majeurs sur la qualité des expérimentations, nous n'avons pas pu mener suffisamment d'expériences pour obtenir des résultats probants concernant la comparaison des approches à évaluer.

Chapitre 7

Conclusion

Ce chapitre synthétise les principales contributions qui constituent le cœur de la thèse, tout en exposant nos aspirations pour les évolutions à venir dans ce domaine.

7.1 Contributions de la thèse

Les avancées récentes en matière d'intelligence artificielle distribuée s'efforcent d'améliorer la coordination entre diverses entités, en ayant recours à des méthodes d'apprentissage ou en employant des mécanismes formels, dans le but de résoudre des problèmes complexes. Une illustration concrète de cette sous-discipline de l'intelligence artificielle se manifeste dans les systèmes multi-agents, avec une application particulièrement notable dans le domaine de la robotique. C'est dans ce contexte que cette thèse aborde les défis inhérents aux problématiques multi-agents de l'observation, mais aussi de la patrouille, à travers le développement de méthode de résolution distribuée, embarqué par la suite sur drone.

Pour ce faire, une étude approfondie de la littérature, conduite dans le chapitre 2, a mis en évidence les formalismes qui encadrent chacune de ces problématiques : l'observation et la patrouille, ainsi que les méthodes de résolution qui leur sont associées. Cette étude a ensuite ouvert la voie à une réflexion sur la pertinence d'intégrer le concept de patrouille dans le cadre de la problématique de l'observation, dans le but d'adopter une stratégie de recherche de cibles reposant sur une couverture continue de l'environnement.

L'association de ces deux problèmes a été concrétisée, au sein du chapitre 3, par la formalisation du Problème de l'Observation appuyée par la Patrouille (POP). Cette approche repose sur un formalisme représentant l'environnement selon deux perspectives superposées. La première repose sur une représentation continue, évitant ainsi d'imposer des contraintes sur

la dynamique des agents et des cibles. La seconde opte pour une représentation matricielle qui permet de conserver les informations d'oisiveté, essentielles pour aborder la problématique de la patrouille. Une adaptation des métriques de patrouille a été réalisée, ces dernières étant initialement définies dans le contexte d'une représentation sous forme de graphe.

Le POP met en évidence un dilemme entre l'exploration (patrouille) et l'exploitation (observation) auquel les agents sont confrontés. Pour y répondre, nous avons développé dans le chapitre 3 des méthodes distribuées reposant sur des champs de potentiel (I-CMOMMT). D'autres solutions, exposées au sein du chapitre 4, reposent sur des techniques d'apprentissage par renforcement (FFRL et F2MARL) et d'apprentissage supervisé (MALOS). Ces approches ont démontré leur efficacité par rapport aux méthodes existantes. Les expérimentations menées dans un environnement de simulation, ainsi que sur de véritables drones en volière, ont validé la pertinence et la praticabilité de nos approches dans des contextes variés.

Toutes les méthodes élaborées possèdent des particularités propres, accompagnées de leurs avantages et de leurs inconvénients respectifs. Le I-CMOMMT ne demande aucun apprentissage préalable, mais exige une configuration pour déterminer le compromis souhaité entre l'observation et la patrouille. Cependant, il est important de noter que ce compromis ne permet pas d'améliorer l'observation moyenne des cibles par rapport aux méthodes déjà existantes dans la littérature. La méthode du FFRL affiche des performances supérieures en ce qui concerne l'observation des cibles, notamment dans des environnements caractérisés par une diversité élevée d'agents et de cibles, en particulier lorsqu'il s'agit de cibles évasives. Ces résultats découlent d'une répartition plus efficace des agents parmi les cibles par rapport aux méthodes évaluées. En revanche, cette approche ne considère pas la dimension de la patrouille dans ses choix de déplacement. En contraste, le F2MARL, qui effectue une patrouille de l'environnement, démontre une capacité d'observation des cibles moyenne légèrement supérieure à celle de l'A-CMOMMT, indépendamment de la densité d'agents et de cibles. Pour conclure, l'approche MALOS démontre de solides performances en termes d'observation moyenne des cibles ainsi que dans la répartition de la distribution entre les cibles. Cependant, ces performances ne sont garanties que pour les paramètres sur lesquels le modèle a été spécifiquement formé, et elles tendent à se détériorer en cas de variations des paramètres. De plus, il est essentiel de souligner que cette méthode requiert un entraînement préliminaire sur un ensemble de données conséquent en amont.

La thèse introduit également deux contributions supplémentaires, en complément des approches de résolution du POP. La première contribution permet aux agents, ou aux méthodes, d'identifier efficacement les lieux ayant un potentiel intérêt à être visité parmi l'ensemble des cellules qui composent la carte d'oisiveté. La disposition des emplacements d'intérêt,

appelés "points d'intérêt dynamiques", s'ajuste en fonction de la configuration topographique de l'environnement et de la surface d'observation des agents. L'objectif est de maximiser les oisivetés perçues lors de la visite de ces points tout en garantissant une couverture complète de l'ensemble de l'environnement lors du positionnement de ces points d'intérêt. La seconde contribution a pour objectif d'optimiser des paramètres de la simulation, tels que le nombre d'agents ou la vitesse des cibles, selon les spécifications de l'utilisateur. Cette optimisation doit également respecter une ou plusieurs performances prédéfinies par ce même utilisateur. De cette manière, cet outil d'aide à la décision permet de dimensionner une mission selon les besoins de l'utilisateur, que ce soit sur les caractéristiques techniques des agents, telles que le rayon d'observation, de communication, ou la vitesse maximale atteignable, ou en ce qui concerne les particularités de la mission, avec le nombre de cibles, leurs vitesses, ou d'autres paramètres ajustables de la simulation.

Enfin, en adoptant une approche open-source pour l'ensemble de nos codes et de nos méthodes, nous avons visé la reproductibilité et la diffusion de notre recherche au sein de la communauté scientifique.

7.2 Discussion et ouverture

Mener un travail de thèse inclus la nécessité de prendre de nombreuses décisions et de choisir des axes de réflexion, ayant pour conséquence de délaisser d'autres pistes. Autrement dit, comme disait le poète André Gide : « Choisir, c'était renoncer pour toujours, pour jamais, à tout le reste, et la quantité nombreuse de ce reste demeurait préférable à n'importe quelle unité. » [45]

Au sein de cette dernière section, nous souhaitons revenir sur certaines de ces décisions, les discuter, parfois les critiquer, ainsi que parcourir les pistes prometteuses n'ayant pas eu l'occasion de prendre place dans nos recherches. Les possibilités d'axe de recherche sont nombreuses dans le domaine de l'intelligence artificielle, ou de l'intelligence augmentée, appliquée au monde des drones. Ce sentiment de toujours pouvoir aller plus loin dans la Recherche est décrit par l'écrivain et scientifique Isaac Asimov par les mots suivants : « La connaissance scientifique possède en quelque sorte des propriétés fractales : nous aurons beau accroître notre savoir, le reste - si infime soit-il - sera toujours aussi infiniment complexe que l'ensemble de départ. » [9]

7.2.1 Formalisme

Pour commencer, d'autres pistes ont été envisagées pour la création du formalisme encadrant le Problème de l'Observation appuyée par la Patrouille. Intuitivement, nous avons adopté une structure carrée pour la représentation des cellules afin d'inscrire les valeurs d'oisivetés. Cependant, dans le cas où l'observation des agents est apparentée à une géométrie circulaire, il serait intéressant de comparer l'utilisation de cette structure carrée avec une structure hexagonale, afin d'en déduire quelle forme permet de refléter au mieux les surfaces observées par les agents.

Les méthodes développées dans le cadre de cette thèse ont employé le formalisme du POP en supposant un environnement carré et exempt d'obstacles. Introduire davantage de complexité dans l'environnement serait utile pour mettre ces méthodes à l'épreuve et, le cas échéant, suggérer des améliorations. Par exemple, cela pourrait être accompli en créant un environnement dont la topographie s'approche davantage d'un terrain réel. Cette représentation inclurait ainsi des frontières non nécessairement rectilignes, ainsi que divers types d'obstacles à l'intérieur de l'environnement.

Enfin, comme élaboré dans l'introduction de cette thèse (cf. chapitre 1), effectuer un suivi des cibles tout en assurant une couverture continue de l'environnement trouve une application dans de nombreux scénarios de mission. L'objectif est également d'utiliser la problématique de la patrouille pour mener une recherche active de cibles et pour également répartir de manière plus équitable l'observation de celles-ci. Il convient de poursuivre la réflexion à ce sujet, car se limiter à la patrouille seule peut entraîner des limitations dans la recherche de cibles mobiles. En effet, une limitation significative peut survenir lorsqu'une cible se déplace exactement là où un agent a récemment effectué une visite. Dans de tels cas, l'oisiveté est considérée comme faible à cet endroit, ce qui réduit l'intérêt d'une nouvelle visite et donc de détecter la cible.

7.2.2 Méthodes

En ce qui concerne les méthodes d'apprentissage telles que MALOS, F2MARL ou encore FFRL, plusieurs possibilités d'amélioration sont envisageables. Tout d'abord, il serait intéressant d'explorer l'utilisation d'autres algorithmes d'apprentissage que le *Proximal Policy Optimization* (PPO) afin de comparer leurs performances en matière d'apprentissage. De plus, les techniques de recherche d'hyperparamètres sont en constante évolution, ce qui peut contribuer à améliorer les résultats de l'apprentissage des modèles. À titre d'exemple, on peut mentionner le *Population Based Bandits* (PB2) [93], un algorithme qui requiert

moins de ressources tout en produisant des résultats comparables, voire supérieurs à ceux du *Population Based Training algorithm* (PBT).

Ensuite, il serait bénéfique d'étudier d'autres modèles que ceux employés par les méthodes d'apprentissage. Ajouter une dimension temporelle avec des modèles récurrents, tels que les réseaux LSTM [51] ou GRU [28], permettrait d'améliorer les stratégies des agents en considérant les observations et les actions passées. Par ailleurs, les modèles d'attention [124], permettant à l'agent de prendre une décision en se focalisant uniquement certaines parties spécifiques de l'observation, montre des performances intéressantes pour résoudre de nombreux problèmes d'apprentissage par renforcement.

En raison de contraintes temporelles, nous n'avons pas pu intégrer la fonction de génération des points d'intérêt dynamiques (DIP) dans les précédentes approches, que ce soit par champs de force ou par apprentissage. Cette intégration aurait permis aux agents de localiser rapidement les lieux à visiter, avec comme prévision une amélioration de l'efficacité de la patrouille multi-agents. Nous estimons également que l'efficacité de l'approche I-CMOMMT pourrait être améliorée en incorporant une stratégie de répartition des cibles entre les agents. Actuellement, les agents sont attirés vers les cibles en fonction de leur proximité. Ainsi, si deux agents suivent la même cible, et que la force de patrouille est négligeable, il serait avantageux qu'un seul des agents se charge de l'observation, laissant ainsi le deuxième agent libre pour rechercher d'autres cibles.

Bibliographie

[1] Martin ABADI et al. “TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems”. In : *CoRR* abs/1603.04467 (2016). arXiv : 1603.04467. URL : http://arxiv.org/abs/1603.04467.

[2] Attai Ibrahim ABUBAKAR et al. “A Survey on Energy Optimization Techniques in UAV-Based Cellular Networks : From Conventional to Machine Learning Approaches”. In : *Drones* 7.3 (2023). ISSN : 2504-446X. DOI : 10.3390/drones7030214. URL : https://www.mdpi.com/2504-446X/7/3/214.

[3] Tauhidul ALAM et al. “Distributed multi-robot area patrolling in adversarial environments”. In : *International Workshop on Robotic Sensor Networks*. 2015.

[4] Javier A ALCAZAR. “A Simple Approach to Multi-Predator Multi-Prey Pursuit Domain”. In : *Unifying Themes in Complex Systems*. 2011, p. 2-9. DOI : 10.1007/978-3-642-17635-7_1.

[5] Alessandro ALMEIDA et al. “Recent Advances on Multi-agent Patrolling”. In : sept. 2004, p. 474-483. ISBN : 978-3-540-23237-7. DOI : 10.1007/978-3-540-28645-5_48.

[6] Alessandro de Luna ALMEIDA et al. “Combining Idleness and Distance to Design Heuristic Agents for the Patrolling Task”. In : 2003.

[7] João P. B. ANDRADE et al. “Organization/fuzzy Approach to the CTO Problem”. In : *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. 2018, p. 444-449. DOI : 10.1109/BRACIS.2018.00083.

[8] Marcin ANDRYCHOWICZ et al. “What Matters In On-Policy Reinforcement Learning ? A Large-Scale Empirical Study”. In : juin 2020.

[9] Isaac ASIMOV. *I, Asimov : a Memoir*. New York, NY : Bantam, jan. 1920.

[10] Isaac ASIMOV. *The caves of steel*. New York, NY : Bantam Doubleday Dell Publishing Group, nov. 1991.

[11] Rashi ASWANI, Sai Krishna MUNNANGI et Praveen PARUCHURI. “Improving Surveillance Using Cooperative Target Observation”. In : *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI’17. San Francisco, California, USA : AAAI Press, 2017, p. 2985-2991.

[12] Ilario Antonio AZZOLLINI et al. *UAV-Based Search and Rescue in Avalanches using ARVA : An Extremum Seeking Approach*. 2021. arXiv : 2106.14514 [cs.RO].

[13] Jacopo BANFI et al. “An integer linear programming model for fair multitarget tracking in cooperative multirobot systems”. In : *Autonomous Robots* 43 (mars 2019). DOI : 10.1007/s10514-018-9735-4.

[14] Jacopo BANFI et al. “Fair Multi-Target Tracking in Cooperative Multi-Robot systems”. In : *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, p. 5411-5418. DOI : 10.1109/ICRA.2015.7139955.

[15] Pedram BEIGI, Mohammad Sadra RAJABI et Sina AGHAKHANI. *An Overview of Drone Energy Consumption Factors and Models*. 2022. arXiv : 2206.10775 [eess.SY].

[16] RICHARD BELLMAN. “A Markovian Decision Process”. In : *Journal of Mathematics and Mechanics* 6.5 (1957), p. 679-684. ISSN : 00959057, 19435274. URL : http://www.jstor.org/stable/24900506.

[17] Martin BENAVIDES, Fredrick FODRIE et David JOHNSTON. “Shark detection probability from aerial drone surveys within a temperate estuary”. In : *Journal of Unmanned Vehicle Systems* 8 (déc. 2019). DOI : 10.1139/juvs-2019-0002.

[18] Aurelie BEYNIER. “Cooperative Multiagent Patrolling for Detecting Multiple Illegal Actions under Uncertainty”. en. In : *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. San Jose, CA, USA : IEEE, nov. 2016, p. 9-16. ISBN : 978-1-5090-4459-7. DOI : 10.1109/ICTAI.2016.0013. URL : http://ieeexplore.ieee.org/document/7814573/.

[19] James C. BEZDEK, Robert EHRLICH et William FULL. “FCM : The fuzzy c-means clustering algorithm”. In : *Computers and Geosciences* 10.2 (1984), p. 191-203. ISSN : 0098-3004. DOI : https://doi.org/10.1016/0098-3004(84)90020-7. URL : https://www.sciencedirect.com/science/article/pii/0098300484900207.

[20] J. BLANK et K. DEB. “Pymoo : Multi-Objective Optimization in Python”. In : *IEEE Access* 8 (2020), p. 89497-89509.

[21] Greg BROCKMAN et al. *OpenAI Gym*. 2016. arXiv : 1606.01540 [cs.LG].

[22] Jamy CHAHAL, Assia BELBACHIR et AMAL EL FALLAH SEGHROUCHNI. “I-CMOMMT : A multiagent approach for patrolling and observation of mobile targets with a continuous environment representation”. In : *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE* (juill. 2021). DOI : 10.18293/SEKE2021-135.

[23] Jamy CHAHAL, Assia BELBACHIR et Amal El Fallah SEGHROUCHNI. “Dynamic Interest Points : A Formalism to Identify Areas to Patrol within a Continuous Environment”. In : *56th Hawaii International Conference on System Sciences, HICSS 2023, Maui, Hawaii, USA, January 3-6, 2023*. Sous la dir. de Tung X. BUI. ScholarSpace, jan. 2023, p. 6853-6862. URL : https://hdl.handle.net/10125/103464.

[24] Jamy CHAHAL, Amal El Fallah SEGHROUCHNI et Assia BELBACHIR. “A decision-making architecture for observation and patrolling problems using machine learning”. In : *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*. Juill. 2021, p. 426-431. DOI : 10.1109/IIAI-AAI53430.2021.00074.

[25] Jamy CHAHAL, Amal El Fallah SEGHROUCHNI et Assia BELBACHIR. “A Force Field Reinforcement Learning Approach for the Observation Problem”. In : *Intelligent Distributed Computing XIV, 14th International Symposium on Intelligent Distributed Computing, IDC 2021, Virtual Event, 16-18 September 2021*. T. 1026. Studies in Computational Intelligence. Springer, sept. 2021, p. 89-99. DOI : 10.1007/978-3-030-96627-0_9.

[26] Yann CHEVALEYRE. “Le problème multi-agents de la patrouille”. working paper or preprint. Nov. 2006. URL : https://hal.archives-ouvertes.fr/hal-00115783.

[27] Yann CHEVALEYRE, François SEMPÉ et Geber RAMALHO. “A Theoretical Analysis of Multi-Agent Patrolling Strategies.” In : jan. 2004, p. 1524-1525. DOI : 10.1109/AAMAS.2004.34.

[28] Kyunghyun CHO et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv : 1406.1078 [cs.CL].

[29] Jui-Sheng CHOU et al. “Optimal path planning in real time for dynamic building fire rescue operations using wireless sensors and visual guidance”. In : *Automation in Construction* 99 (2019), p. 1-17. ISSN : 0926-5805. DOI : https://doi.org/10.1016/j.autcon.2018.11.020. URL : https://www.sciencedirect.com/science/article/pii/S0926580518307143.

[30] Hoang Nam CHU et al. “Swarm Approaches for the Patrolling Problem, Information Propagation vs. Pheromone Evaporation”. en. In : *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*. Patras, Greece : IEEE, oct. 2007, p. 442-449. ISBN : 978-0-7695-3015-4. DOI : 10.1109/ICTAI.2007.80. URL : http://ieeexplore.ieee.org/document/4410318/.

[31] Leonardo COSTA et al. “Comparative Study of Neural Networks Techniques in the Context of Cooperative Observations”. In : *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*. Salvador : SBC, 2019, p. 563-574. DOI : 10.5753/eniac.2019.9315. URL : https://sol.sbc.org.br/index.php/eniac/article/view/9315.

[32] Thayanne França DA SILVA et al. “Smart targets to avoid observation in CTO problem”. In : *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS* 4.May (2019), p. 1958-1960. ISSN : 9781510892002.

[33] Philip M. DAMES. “Distributed multi-target search and tracking using the PHD filter”. en. In : *Autonomous Robots* 44.3-4 (mars 2020), p. 673-689. ISSN : 0929-5593, 1573-7527. DOI : 10.1007/s10514-019-09840-9. URL : http://link.springer.com/10.1007/s10514-019-09840-9.

[34] Alessandro DE LUNA ALMEIDA et al. “Recent advances on multi-agent patrolling”. In : *17th Brazilian Symposium on Artificial Intelligence*. Lecture Notes in Computer Science 3171 (2004), p. 474-483. DOI : 10.1007/978-3-540-28645-5_48.

[35] K. DEB et al. “A fast and elitist multiobjective genetic algorithm : NSGA-II”. In : *IEEE Transactions on Evolutionary Computation* 6.2 (2002), p. 182-197. DOI : 10.1109/4235.996017.

[36] Marwan DHUHEIR et al. “Deep Reinforcement Learning for Trajectory Path Planning and Distributed Inference in Resource-Constrained UAV Swarms”. en. In : *IEEE Internet of Things Journal* (2022). arXiv :2212.11201 [cs], p. 1-1. ISSN : 2327-4662, 2372-2541. DOI : 10.1109/JIOT.2022.3231341. URL : http://arxiv.org/abs/2212.11201.

[37] Yingying DING et Yan HE. “Flexible formation of the multi-robot system and its application on CMOMMT problem”. In : *CAR 2010 - 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics* 1.1 (2010), p. 377-380. DOI : 10.1109/CAR.2010.5456820.

[38] Yingying DING et al. “P-CMOMMT algorithm for the cooperative multi-robot observation of multiple moving targets”. In : *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* 2 (2006), p. 9267-9271. DOI : 10.1109/WCICA.2006.1713794.

[39] Yehuda ELMALIACH, Noa AGMON et Gal A. KAMINKA. “Multi-Robot Area Patrol under Frequency Constraints”. In : *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, p. 385-390. DOI : 10.1109/ROBOT.2007.363817.

[40] Alessando FARINELLI, Luca IOCCHI et Daniele NARDI. “Distributed on-line dynamic task assignment for multi-robot patrolling”. In : *Autonomous Robots* 41 (août 2017). DOI : 10.1007/s10514-016-9579-8.

[41] Kailun FENG, Shiwei CHEN et Weizhuo LU. “Machine learning based construction simulation and optimization”. In : *Proceedings - Winter Simulation Conference* 2018-Decem (2019), p. 2025-2036. ISSN : 08917736. DOI : 10.1109/WSC.2018.8632290.

[42] J. FERBER. *Les systèmes multi-agents : vers une intelligence collective*. Paris : InterEditions, 1995.

[43] F. FERNÁNDEZ, D. BORRAJO et L. PARKER. “A Reinforcement Learning Algorithm in Cooperative Multi-Robot Domains”. In : *Journal of Intelligent and Robotic Systems* 43 (2005), p. 161-174.

[44] M. GARCIA, Antidio VIGURIA et Anibal OLLERO. “Dynamic Graph-Search Algorithm for Global Path Planning in Presence of Hazardous Weather”. In : *Journal of Intelligent and Robotic Systems* 69 (jan. 2013). DOI : 10.1007/s10846-012-9704-7.

[45] Andre GIDE. *Les nourritures terrestres/Les nouvelles nourritures*. fr. Paris, France : Gallimard, mai 1973.

[46] Arnaud GLAD et al. “Self-Organization of Patrolling-Ant Algorithms”. In : *2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*. 2009, p. 61-70. DOI : 10.1109/SASO.2009.39.

[47] Nikolaus HANSEN et Anne AUGER. “CMA-ES : Evolution Strategies and Covariance Matrix Adaptation”. In : *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*. GECCO ’11. Dublin, Ireland : Association for Computing Machinery, 2011, p. 991-1010. ISBN : 9781450306904. DOI : 10.1145/2001858.2002123.

[48] Peter E. HART, Nils J. NILSSON et Bertram RAPHAEL. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In : *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), p. 100-107. DOI : 10.1109/TSSC.1968.300136.

[49] Dewan Tariq HASAN et al. “On evaluation of patrolling and signalling schemes to prevent poaching in green security games”. en. In : *Intelligent Systems with Applications* 14 (mai 2022), p. 200083. ISSN : 26673053. DOI : 10.1016/j.iswa.2022.200083. URL : https://linkinghub.elsevier.com/retrieve/pii/S2667305322000230.

[50] Katsuya HATTORI et Toshiharu SUGAWARA. “Effective Area Partitioning in a Multi-Agent Patrolling Domain for Better Efficiency :” en. In : *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*. SCITEPRESS - Science et Technology Publications, 2021, p. 281-288. ISBN : 978-989-758-484-8. DOI : 10.5220/0010241102810288. URL : https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010241102810288.

[51] Sepp HOCHREITER et Jürgen SCHMIDHUBER. “Long Short-term Memory”. In : *Neural computation* 9 (déc. 1997), p. 1735-80. DOI : 10.1162/neco.1997.9.8.1735.

[52] L. HONG et Guangxin JIANG. “Offline Simulation Online Application : A New Framework of Simulation-Based Decision Making”. In : *Asia-Pacific Journal of Operational Research* 36 (déc. 2019), p. 1940015. DOI : 10.1142/S0217595919400153.

[53] Siyi HU et al. *MARLlib : A Scalable Multi-agent Reinforcement Learning Library*. 2023. arXiv : 2210.13708 [cs.LG].

[54] Haomiao HUANG et al. “Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers”. In : *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011, p. 4835-4840. DOI : 10.1109/CDC.2011.6161237.

[55] Max JADERBERG et al. *Population Based Training of Neural Networks*. 2017. arXiv : 1711.09846 [cs.LG].

[56] Meghdeep JANA, Leena VACHHANI et Arpita SINHA. “A deep reinforcement learning approach for multi-agent mobile robot patrolling”. In : *International Journal of Intelligent Robotics and Applications* 6 (mai 2022). DOI : 10.1007/s41315-022-00235-1.

[57] Khaled JARRAH et al. “Flight Time Optimization and Modeling of a Hybrid Gasoline-Electric Multirotor Drone : An Experimental Study”. In : *Aerospace* 9.12 (2022). ISSN : 2226-4310. DOI : 10.3390/aerospace9120799. URL : https://www.mdpi.com/2226-4310/9/12/799.

[58] Shiyuan JIN et Zhihua QU. “A heuristic task scheduling for multi-pursuer multi-evader games”. In : *2011 IEEE International Conference on Information and Automation*. 2011, p. 528-533. DOI : 10.1109/ICINFA.2011.5949050.

[59] George KARYPIS et Vipin KUMAR. “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs”. In : *SIAM Journal on Scientific Computing* 20.1 (1998), p. 359-392. DOI : 10.1137/S1064827595287997. eprint : https://doi.org/10.1137/S1064827595287997. URL : https://doi.org/10.1137/S1064827595287997.

[60] Marvin KASTNER et al. “Integrated Simulation-Based Optimization of Operational Decisions at Container Terminals”. In : *Algorithms* 14.2 (2021). ISSN : 1999-4893. DOI : 10.3390/a14020042.

[61] Chihiro KATO et Toshiharu SUGAWARA. “Decentralized Area Partitioning for a Cooperative Cleaning Task”. en. In : *PRIMA 2013 : Principles and Practice of Multi-Agent Systems*. Sous la dir. de David HUTCHISON et al. T. 8291. Series Title : Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013, p. 470-477. ISBN : 978-3-642-44926-0 978-3-642-44927-7. DOI : 10.1007/978-3-642-44927-7_36. URL : http://link.springer.com/10.1007/978-3-642-44927-7_36.

[62] Asif KHAN, Bernhard RINNER et Andrea CAVALLARO. “Cooperative Robots to Observe Moving Targets : Review”. In : *IEEE Transactions on Cybernetics* PP (déc. 2016), p. 1-12. DOI : 10.1109/TCYB.2016.2628161.

[63] Ines KHOUFI, Anis LAOUITI et Cedric ADJIH. “A Survey of Recent Extended Variants of the Traveling Salesman and Vehicle Routing Problems for Unmanned Aerial Vehicles”. In : *Drones* 3.3 (2019). ISSN : 2504-446X. DOI : 10.3390/drones3030066. URL : https://www.mdpi.com/2504-446X/3/3/66.

[64] David KLAŠKA et al. *Minimizing Expected Intrusion Detection Time in Adversarial Patrolling*. en. arXiv :2202.01095 [cs]. Fév. 2022. URL : http://arxiv.org/abs/2202.01095.

[65] Andreas KOLLING. "Multi-robot pursuit-evasion". Thèse de doct. 2009. URL : https://escholarship.org/uc/item/6nm6c1gh.

[66] Andreas KOLLING et Stefano CARPIN. "Cooperative observation of multiple moving targets : An algorithm and its formalization". In : *International Journal of Robotics Research* 26.9 (2007), p. 935-953. ISSN : 02783649. DOI : 10.1177/0278364907080424.

[67] Alfred KORZYBSKI. *Une carte n'est pas le territoire : Prolégomènes aux systèmes non-aristotéliciens et à la sémantique générale*. fr. Sept. 2015.

[68] Fabrice LAURI et François CHARPILLET. "Ant Colony Optimization applied to the Multi-Agent Patrolling Problem". In : *IEEE Swarm Intelligence Symposium 2006*. Indianapolis, Indiana, United States : IEEE, mai 2006. URL : https://inria.hal.science/inria-00119485.

[69] Fabrice LAURI et Abderrafiaa KOUKAM. "A two-step evolutionary and ACO approach for solving the multi-agent patrolling problem". In : *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. 2008, p. 861-868. DOI : 10.1109/CEC.2008.4630897.

[70] Fabrice LAURI et Abderrafiaa KOUKAM. "Robust Multi-agent Patrolling Strategies Using Reinforcement Learning". In : *Swarm Intelligence Based Optimization*. Sous la dir. de Patrick SIARRY, Lhassane IDOUMGHAR et Julien LEPAGNOT. Cham : Springer International Publishing, 2014, p. 157-165. ISBN : 978-3-319-12970-9.

[71] Xianfeng LI et al. "Profit-Driven Adaptive Moving Targets Search with UAV Swarms". In : *Sensors* 19.7 (2019). ISSN : 1424-8220. DOI : 10.3390/s19071545. URL : https://www.mdpi.com/1424-8220/19/7/1545.

[72] Xiaohui LI et Li XING. "Use of Unmanned Aerial Vehicles for Livestock Monitoring based on Streaming K-Means Clustering∗∗This work was supported by the Australian Research Council." In : *IFAC-PapersOnLine* 52.30 (2019). 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019, p. 324-329. ISSN : 2405-8963. DOI : https://doi.org/10.1016/j.ifacol.2019.12.560. URL : https://www.sciencedirect.com/science/article/pii/S2405896319324796.

[73] Zhenhua LI et al. "Evolution strategies for continuous optimization : A survey of the state-of-the-art". In : *Swarm and Evolutionary Computation* 56 (2020), p. 100694. ISSN : 2210-6502. DOI : https://doi.org/10.1016/j.swevo.2020.100694.

[74] Eric LIANG et al. *RLlib : Abstractions for Distributed Reinforcement Learning*. 2018. arXiv : 1712.09381 [cs.AI].

[75] Xin LIU et al. "Fair energy-efficient resource optimization for multi-UAV enabled Internet of Things". In : *IEEE Transactions on Vehicular Technology* (2022).

[76] Stuart P. LLOYD. "Least Squares Quantization in PCM". In : *IEEE Transactions on Information Theory* 28.2 (1982), p. 129-137. ISSN : 15579654. DOI : 10.1109/TIT.1982.1056489.

[77] Ryan LOWE et al. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In : *Neural Information Processing Systems (NIPS)* (2017).

[78] Sean LUKE et al. “Tunably decentralized algorithms for cooperative target observation”. en. In : *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS ’05*. The Netherlands : ACM Press, 2005, p. 911. ISBN : 978-1-59593-093-4. DOI : 10.1145/1082473.1082611. URL : http://portal.acm.org/citation.cfm?doid=1082473.1082611.

[79] Steven MACENSKI et al. “Robot Operating System 2 : Design, architecture, and uses in the wild”. In : *Science Robotics* 7.66 (2022), eabm6074. DOI : 10.1126/scirobotics.abm6074. URL : https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[80] R.P.s MAHLER. “Multitarget Bayes Filtering via First-Order Multitarget Moments”. In : *Aerospace and Electronic Systems, IEEE Transactions on* 39 (nov. 2003), p. 1152-1178. DOI : 10.1109/TAES.2003.1261119.

[81] Jean-Samuel MARIER, Camille BESSE et Brahim CHAIB-DRAA. “Solving the Continuous Time Multiagent Patrol Problem”. In : mai 2010, p. 941-946. DOI : 10.1109/ROBOT.2010.5509608.

[82] Stefan MARKOV et Stefano CARPIN. “A cooperative distributed approach to target motion control in multirobot observation of multiple targets”. In : *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007, p. 931-936. DOI : 10.1109/IROS.2007.4399159.

[83] Yuya MARUYAMA, Shinpei KATO et Takuya AZUMI. “Exploring the performance of ROS2”. In : *Proceedings of the 13th ACM SIGBED International Conference on Embedded Software (EMSOFT)*. 2016, p. 1-10.

[84] Talita MENEZES, Patricia TEDESCO et Geber RAMALHO. “Negotiator Agents for the Patrolling Task”. In : oct. 2006, p. 48-57. ISBN : 978-3-540-45462-5. DOI : 10.1007/11874850_9.

[85] Yaniv OSHRAT, Noam AGMON et Sarit KRAUS. “Adversarial Fence Patrolling : Non-Uniform Policies for Asymmetric Environments”. In : *AAAI Conference on Artificial Intelligence*. 2020.

[86] Mehdi OTHMANI-GUIBOURG. “Supervised learning for distribution of centralised multiagent patrolling strategies”. Theses. Sorbonne Université, déc. 2019. URL : https://hal.archives-ouvertes.fr/tel-02876729.

[87] Mehdi OTHMANI-GUIBOURG, Amal EL FALLAH-SEGHROUCHNI et Jean-Loup FARGES. “Decentralized Multi-agent Patrolling Strategies Using Global Idleness Estimation”. en. In : *PRIMA 2018 : Principles and Practice of Multi-Agent Systems*. Sous la dir. de Tim MILLER et al. T. 11224. Series Title : Lecture Notes in Computer Science. Cham : Springer International Publishing, 2018, p. 603-611. ISBN : 978-3-030-03097-1 978-3-030-03098-8. DOI : 10.1007/978-3-030-03098-8_47. URL : http://link.springer.com/10.1007/978-3-030-03098-8_47.

[88] Mehdi OTHMANI-GUIBOURG et al. “Multi-agent patrolling in dynamic environments”. en. In : *2017 IEEE International Conference on Agents (ICA)*. Beijing, China : IEEE, juill. 2017, p. 72-77. ISBN : 978-1-5386-0768-8. DOI : 10.1109/AGENTS.2017.8015305. URL : http://ieeexplore.ieee.org/document/8015305/.

[89] Mehdi William OTHMANI-GUIBOURG, Amal EL FALLAH SEGHROUCHNI et Jean-Loup FARGES. "LSTM Path-Maker : une stratégie à base de réseau de neurones LSTM pour la patrouille multiagent". In : *Revue Ouverte d'Intelligence Artificielle* 3.3-4 (2022), p. 345-372. DOI : 10.5802/roia.34. URL : https://hal.science/hal-03649519.

[90] Aydano PAMPONET MACHADO et al. "Multi-Agent Movement Coordination in Patrolling". In : *First Workshop on Agents in Computer Games*. 2002.

[91] Selina PAN et al. "Pursuit, evasion and defense in the plane". In : *2012 American Control Conference (ACC)*. 2012, p. 4167-4173. DOI : 10.1109/ACC.2012.6315389.

[92] Lynne E. PARKER. "Distributed algorithms for multi-robot observation of multiple moving targets". In : *Autonomous Robots* 12 (2002), p. 231-255. ISSN : 09295593. DOI : 10.1023/A:1015256330750.

[93] Jack PARKER-HOLDER, Vu NGUYEN et Stephen ROBERTS. *Provably Efficient Online Hyperparameter Optimization with Population-Based Bandits*. 2021. arXiv : 2002.02518 [cs.LG].

[94] Luciano C. A. PIMENTA et al. "Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks". en. In : *Algorithmic Foundation of Robotics VIII*. Sous la dir. de Bruno SICILIANO et al. T. 57. Series Title : Springer Tracts in Advanced Robotics. Berlin, Heidelberg : Springer Berlin Heidelberg, 2009, p. 85-99. ISBN : 978-3-642-00311-0 978-3-642-00312-7. DOI : 10.1007/978-3-642-00312-7_6. URL : http://link.springer.com/10.1007/978-3-642-00312-7_6.

[95] David PORTUGAL, Micael S. COUCEIRO et Rui P. ROCHA. "Applying Bayesian learning to multi-robot patrol". In : *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2013, p. 1-6. DOI : 10.1109/SSRR.2013.6719325.

[96] David PORTUGAL, Luca IOCCHI et Alessandro FARINELLI. "A ROS-Based Framework for Simulation and Benchmarking of Multi-robot Patrolling Algorithms". In : *Robot Operating System (ROS) : The Complete Reference (Volume 3)*. Sous la dir. d'Anis KOUBAA. Cham : Springer International Publishing, 2019, p. 3-28. ISBN : 978-3-319-91590-6. DOI : 10.1007/978-3-319-91590-6_1. URL : https://doi.org/10.1007/978-3-319-91590-6_1.

[97] David PORTUGAL et Rui ROCHA. "Cooperative multi-robot patrol with Bayesian learning". In : *Autonomous Robots* 40 (juin 2016), p. 929-953. DOI : 10.1007/s10514-015-9503-7.

[98] David PORTUGAL et Rui ROCHA. "MSP Algorithm : Multi-Robot Patrolling Based on Territory Allocation Using Balanced Graph Partitioning". In : *Proceedings of the 2010 ACM Symposium on Applied Computing*. SAC '10. Sierre, Switzerland : Association for Computing Machinery, 2010, p. 1271-1276. ISBN : 9781605586397. DOI : 10.1145/1774088.1774360. URL : https://doi.org/10.1145/1774088.1774360.

[99] David PORTUGAL et Rui ROCHA. "Multi-Robot Patrolling Algorithms : Examining Performance and Scalability". In : *Advanced Robotics* 27 (fév. 2013), p. 325-336. DOI : 10.1080/01691864.2013.763722.

[100] David PORTUGAL et Rui P. ROCHA. “Distributed multi-robot patrol : A scalable and fault-tolerant framework”. In : *Robotics and Autonomous Systems* 61.12 (2013), p. 1572-1587. ISSN : 0921-8890. DOI : https://doi.org/10.1016/j.robot.2013.06.011. URL : https://www.sciencedirect.com/science/article/pii/S0921889013001206.

[101] Cyril POULET. “Coordination dans les systèmes multi-agents : Le problème de la patrouille en système ouvert”. Thèse de doct. 2013. URL : http://www.theses.fr/2013PA066152.

[102] Cyril POULET, Vincent CORRUBLE et Amal EL FALLAH SEGHROUCHNI. “Auction-Based Strategies for the Open-System Patrolling Task”. In : *15th International Conference on Principles and Practice of Multi-Agent Systems* (2012), p. 92-106. DOI : 10.1007/978-3-642-32729-2_7.

[103] Cyril POULET, Vincent CORRUBLE et Amal EL FALLAH-SEGHROUCHNI. “Travailler en équipe : le choix social appliqué au problème de la patrouille multi-agents”. In : *journées francophones sur les systèmes multi-agents (JFSMA ’12)*. Honfleur, France, oct. 2012. URL : https://hal.science/hal-00753752.

[104] Morgan QUIGLEY. “ROS : an open-source Robot Operating System”. In : *IEEE International Conference on Robotics and Automation*. 2009.

[105] Cyril ROBIN. “Modèles et algorithmes pour systèmes multi-robots hétérogènes : application à la patrouille et au suivi de cible”. Theses. INSA de Toulouse, juin 2015. URL : https://theses.hal.science/tel-02094404.

[106] Laura von RUEDEN et al. “Combining Machine Learning and Simulation to a Hybrid Modelling Approach : Current and Future Directions”. In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12080 LNCS (2020), p. 548-560. ISSN : 16113349. DOI : 10.1007/978-3-030-44584-3_43.

[107] Fabrice SAFFRE et al. “Monitoring and Cordoning Wildfires with an Autonomous Swarm of Unmanned Aerial Vehicles”. en. In : *Drones* 6.10 (oct. 2022), p. 301. ISSN : 2504-446X. DOI : 10.3390/drones6100301. URL : https://www.mdpi.com/2504-446X/6/10/301.

[108] Pablo A. SAMPAIO, Geber RAMALHO et Patrícia TEDESCO. “The gravitational strategy for the timed patrolling”. In : *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* 1 (2010), p. 113-117. ISSN : 10823409. DOI : 10.1109/ICTAI.2010.24.

[109] Amir SANI. “Machine Learning for Decision Making”. Theses. Université de Lille 1, mai 2015.

[110] H. SANTANA et al. “Multi-agent patrolling with reinforcement learning”. In : *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.* 2004, p. 1122-1129.

[111] John SCHULMAN et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv : 1707.06347 [cs.LG].

[112] Vourchteang SEA, Ayumi SUGIYAMA et Toshiharu SUGAWARA. *Frequency-based multi-agent patrolling model and its area partitioning solution method for balanced workload*. T. 10848 LNCS. Springer International Publishing, 2018, p. 530-545. ISBN : 9783319930305. DOI : 10.1007/978-3-319-93031-2_38.

[113] Esmaeil SERAJ, Andrew SILVA et Matthew GOMBOLAY. *Multi-UAV Planning for Cooperative Wildfire Coverage and Tracking with Quality-of-Service Guarantees*. Juin 2022. DOI : 10.48550/arXiv.2206.10544.

[114] C. E. SHANNON. "A Mathematical Theory of Communication". In : *Bell System Technical Journal* 27.3 (1948), p. 379-423. DOI : https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[115] Efrat SLESS, Noa AGMON et Sarit KRAUS. "Multi-Robot Adversarial Patrolling : Facing Coordinated Attacks". In : *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*. AAMAS '14. Paris, France : International Foundation for Autonomous Agents et Multiagent Systems, 2014, p. 1093-1100. ISBN : 9781450327381.

[116] Efrat SLESS LIN, Noa AGMON et Sarit KRAUS. "Multi-robot adversarial patrolling : Handling sequential attacks". In : *Artificial Intelligence* 274 (2019), p. 1-25. ISSN : 0004-3702. DOI : https://doi.org/10.1016/j.artint.2019.02.004. URL : https://www.sciencedirect.com/science/article/pii/S0004370219300475.

[117] Ayumi SUGIYAMA, Vourchteang SEA et Toshiharu SUGAWARA. "Effective Task Allocation by Enhancing Divisional Cooperation in Multi-Agent Continuous Patrolling Tasks". In : *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. 2016, p. 33-40. DOI : 10.1109/ICTAI.2016.0016.

[118] Ayumi SUGIYAMA, Lingying WU et Toshiharu SUGAWARA. "Improvement of Multi-agent Continuous Cooperative Patrolling with Learning of Activity Length". In : *International Conference on Agents and Artificial Intelligence*. 2019.

[119] Fredrik SVANSTRÖM, Fernando ALONSO-FERNANDEZ et Cristofer ENGLUND. "Drone Detection and Tracking in Real-Time by Fusion of Different Sensing Modalities". In : *Drones* 6.11 (2022). ISSN : 2504-446X. DOI : 10.3390/drones6110317. URL : https://www.mdpi.com/2504-446X/6/11/317.

[120] J. K TERRY et al. "PettingZoo : Gym for Multi-Agent Reinforcement Learning". In : *arXiv preprint arXiv :2009.14471* (2020).

[121] Justin K TERRY et al. *Revisiting Parameter Sharing In Multi-Agent Deep Reinforcement Learning*. 2021. arXiv : 2005.13625 [cs.LG].

[122] Paolo TESTOLINA et al. "Enabling simulation-based optimization through machine learning : A case study on antenna design". In : *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings* (2019). DOI : 10.1109/GLOBECOM38437.2019.9013240. arXiv : 1908.11225.

[123] Claude F. TOUZET. "Distributed Lazy Q-Learning for Cooperative Mobile Robots". In : *International Journal of Advanced Robotic Systems* 1.1 (2004), p. 1. DOI : 10.5772/5614. eprint : https://doi.org/10.5772/5614. URL : https://doi.org/10.5772/5614.

[124] Ashish VASWANI et al. *Attention Is All You Need*. 2017. arXiv : 1706.03762 [cs.CL].

[125] Christian Schroeder de WITT et al. *Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge ?* 2020. arXiv : 2011.09533 [cs.AI].

[126] Annie WONG et al. *Multiagent Deep Reinforcement Learning : Challenges and Directions Towards Human-Like Approaches*. 2021. arXiv : 2106.15691 [cs.LG].

[127] Kyle Hollins WRAY et Benjamin B. THOMPSON. “An application of multiagent learning in highly dynamic environments”. In : *AAAI Workshop - Technical Report* WS-14-09 (2014), p. 42-48. ISSN : 9781577356707.

[128] Lingying WU, Ayumi SUGIYAMA et Toshiharu SUGAWARA. “Energy-Efficient Strategies for Multi-Agent Continuous Cooperative Patrolling Problems”. In : *Procedia Computer Science* 159 (2019). Knowledge-Based and Intelligent Information and Engineering Systems : Proceedings of the 23rd International Conference KES2019, p. 465-474. ISSN : 1877-0509. DOI : https://doi.org/10.1016/j.procs.2019.09.201. URL : https://www.sciencedirect.com/science/article/pii/S1877050919313833.

[129] Yiqing XU, Jiaming LI et Fuquan ZHANG. “A UAV-Based Forest Fire Patrol Path Planning Strategy”. In : *Forests* 13.11 (2022). ISSN : 1999-4907. DOI : 10.3390/f13111952. URL : https://www.mdpi.com/1999-4907/13/11/1952.

[130] Chuanbo YAN et Tao ZHANG. “Multi-robot patrol : A distributed algorithm based on expected idleness”. In : *International Journal of Advanced Robotic Systems* 13 (nov. 2016). DOI : 10.1177/1729881416663666.

[131] Peng YAN, Tao JIA et Chengchao BAI. “Searching and Tracking an Unknown Number of Targets : A Learning-Based Method Enhanced with Maps Merging”. In : *Sensors* 21.4 (2021). ISSN : 1424-8220. DOI : 10.3390/s21041076. URL : https://www.mdpi.com/1424-8220/21/4/1076.

[132] Keisuke YONEDA, Chihiro KATO et Toshiharu SUGAWARA. “Autonomous Learning of Target Decision Strategies without Communications for Continuous Coordinated Cleaning Tasks”. In : *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02*. WI-IAT ’13. USA : IEEE Computer Society, 2013, p. 216-223. ISBN : 9780769551456. DOI : 10.1109/WI-IAT.2013.112. URL : https://doi.org/10.1109/WI-IAT.2013.112.

[133] Keisuke YONEDA et al. “Learning and relearning of target decision strategies in continuous coordinated cleaning tasks with shallow coordination1”. en. In : *Web Intelligence* 13.4 (nov. 2015), p. 279-294. ISSN : 24056464, 24056456. DOI : 10.3233/WEB-150326.

[134] Chao YU et al. *The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games*. 2021. arXiv : 2103.01955 [cs.LG].

Patrouille multi-drones et observation de cibles mobiles

RÉSUMÉ

Cette thèse aborde l'association des problèmes d'observation et de patrouille en contexte multi-agents. L'observation vise à maximiser le nombre de cibles mobiles observées au cours du temps, tandis que la patrouille a pour objectif de visiter le plus fréquemment un ensemble de lieux dans l'environnement. Nous nommons ce dilemme entre exploration (recherche de nouvelles cibles via la patrouille) et exploitation (maximisation de l'observation des cibles) le Problème de l'Observation appuyée par la Patrouille (POP). Nous proposons la résolution de ce problème par l'emploi : d'un champ de forces (I-CMOMMT), de l'apprentissage par renforcement (FFRL, F2MARL) ainsi que supervisé (MALOS). Les expériences sont simulées avec des drones en tant qu'agents et des robots terrestres mobiles comme cibles, puis sont validées sur de vrais drones en volière. Deux autres contributions permettent l'identification de lieux d'intérêt à patrouiller, ainsi que l'optimisation des paramètres de mission.

Multi-drones patrol and observation of mobile targets

ABSTRACT

This thesis addresses the combination of observation and patrol problems in a multi-agent context. Observation aims to maximize the number of mobile targets observed over time, while patrol aims to visit a set of locations in the environment as frequently as possible. We refer to this dilemma between exploration (finding new targets through patrol) and exploitation (maximizing target observation) as the Observation and Patrolling Problem (OPP). We propose to solve this problem using force-field (I-CMOMMT), reinforcement learning (FFRL, F2MARL), and supervised learning (MALOS). The experiments are simulated with drones as agents and mobile ground robots as targets, and then are validated with real drones in an aviary. Two additional contributions enable the identification of places of interest to patrol and the optimization of mission parameters.

Jamy Chahal, Thèse de doctorat, soutenue le 30 novembre 2023
EDITE de Paris, Sorbonne Université, CNRS, LIP6, F-75005