L'approche d'apprentissage automatique pour l'optimisation de la robotique en essaim dans l'entrepôt automatisé servi par la communication 5G

par

Manh Tai HO

THÈSE PAR ARTICLES PRÉSENTÉE À L'ÉCOLE DE TECHNOLOGIE
SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION
DU DOCTORAT EN GÉNIE
Ph.D.

MONTRÉAL, LE 20 JUILLET 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CETTE THÈSE A ÉTÉ ÉVALUÉE

PAR UN JURY COMPOSÉ DE:

M. Mohamed Cheriet, directeur de thèse
Département de génie des systèmes

M. Kim Khoa Nguyen, codirecteur
Département de génie électrique

M. Aris Leivadeas, président du jury
Département de génie logiciel et des TI

M. Zbigniew Dziong, membre du jury
Département de génie électrique

M. Mohamed Faten Zhani, examinateur externe
Institut Supérieur d'Informatique et des Techniques de Communication - Université de Sousse

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 18 MAI 2023

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AVANT-PROPOS

Cette thèse est présentée en vue de l'obtention du diplôme de docteur en philosophie de l'École de technologie supérieure de l'Université du Québec. La recherche décrite ici a été menée sous la supervision du professeur Mohamed Cheriet au Département de génie des systèmes et du professeur Kim-Khoa Nguyen au Département de génie électrique du semestre été 2020 au semestre hiver 2023.

Le travail de cette thèse est original et structuré sous la forme d'une compilation d'articles qui ont été publiés ou soumis à de prestigieuses revues IEEE de premier plan dans le domaine de la gestion des services réseau et de la science et de l'ingénierie de l'automatisation. Les articles inclus dans cette thèse sont intégrés avec une grande fidélité pour assurer la conformité avec la structure et la forme des articles soumis et publiés. Pourtant, seules des modifications périphériques, par exemple, le cadrage, le repositionnement et la mise à l'échelle des figures ont été apportées dans le cadre des directives de thèse de l'École de technologie supérieure.

REMERCIEMENTS

Avant tout, je tiens à exprimer ma sincère gratitude envers mes estimés superviseurs, le professeur Mohamed Cheriet et le professeur Kim-Khoa Nguyen, pour leurs conseils et leur soutien inestimables. Je suis extrêmement reconnaissant envers eux pour avoir consacré leur temps, prodigué leurs conseils précieux et guidé mes recherches, contribuant ainsi à l'amélioration significative de cette thèse.

Je souhaite également exprimer ma gratitude envers mon comité de thèse pour ses commentaires fructueux et constructifs sur mon travail, qui ont grandement enrichi la qualité de cette thèse.

Mes remerciements s'adressent également à Dr. Anselme Ndikumana et à tous les membres du laboratoire Synchromedia pour leur généreux soutien tout au long de mon parcours à l'École de Technologie Supérieure.

Enfin, et surtout, je suis sans mots pour exprimer ma profonde gratitude envers ma famille pour leur amour, leurs encouragements et leur soutien infini, en particulier envers mes parents, Ho Ngoc Bich et Le Thi Chinh. Je souhaite adresser mes remerciements les plus sincères à ma femme, Le Thi Uyen Thy, et à ma fille, Ho Minh Hanh, qui ont été à mes côtés durant les moments difficiles avec un dévouement infaillible et sans faille.

L’approche d’apprentissage automatique pour l’optimisation de la robotique en essaim dans l’entrepôt automatisé servi par la communication 5G

Manh Tai HO

RÉSUMÉ

Le réseau sans fil de cinquième génération (5G) fournit des connexions à haut débit, à très faible latence et à haute fiabilité qui peuvent répondre aux exigences de l’Internet industriel des objets (IIoT) dans l’automatisation industrielle, en particulier pour le contrôle robotique. Dans l’entreposage intelligent, la robotique joue un rôle indispensable dans la réalisation de solutions logistiques intelligentes qui comprennent l’organisation, la planification, le contrôle et l’exécution intelligente du flux de marchandises/articles dans l’entrepôt. Les progrès récents des communications sans fil et des technologies de batterie permettent de remplacer de plusieurs travailleurs humains par des systèmes robotiques afin de réduire les coûts de main-d’œuvre, d’améliorer l’efficacité du travail en entrepôt et d’augmenter la fiabilité. Cependant, le déploiement de la robotique en essaim impose de nouveaux défis en termes de contrôle pour coordonner de nombreux types de ressources dans l’entrepôt afin de livrer les de services 5G pour la robotique et de planifier des tâches pour les robots.

En particulier, gestion efficace des ressources sans fil dans un réseau 5G hautement dynamique comme dans un entrepôt automatisé est un problème hautement difficile car l’extrême fiabilité et la faible latence avec une grande mobilité des robots ne sont pas résolvables efficacement par l’approche d’optimisation traditionnelle.

À cette fin, dans cette thèse, nous abordons conjointement les deux défis principaux d’un entrepôt automatisé : i)le provisionnement des services 5G et ii) le contrôle de la robotique en essaim. Les contributions principales de cette thèse sont les suivantes :

1. Tout d’abord, nous formulons le problème de provisionnement de services 5G pour servir la robotique en essaim dans l’entrepôt automatisé comme un clustering des multi-point coordonnés (CoMP) conjoints variables dans le temps et une formation de faisceaux de communication ultra-fiable à faible latence (URLLC) 5G. Les approches d’optimisation itératives traditionnelles ne sont pas efficaces pour résoudre ce problème non-convexe en temps réel à cause de leur temps de calcul élevé. Nous proposons ainsi un algorithme de clustering CoMP en utilisant la théorie des jeux combinée à la méthode d’apprentissage automatique Proximal Policy Optimization pour obtenir une solution stationnaire approximative à la solution optimale globale.
2. Deuxièmement, nous étudions le problème du contrôle des systèmes robotiques hétérogènes autonomes en essaim. Nous formulons un problème d’optimisation de contrôle de file d’attente non convexe à long terme pour minimiser la longueur de la file d’attente des tâches à traiter dans l’entrepôt. Les solutions traditionnelles basées sur des approches d’optimisation sont inefficaces pour gérer la nature stochastique du flux de marchandises/tâches et un grand nombre de robots dans le système. Ainsi, nous proposons un algorithme de planification de tâches basé sur l’apprentissage par renforcement profond (DRL) qui utilise la méthode

d'optimisation de politique proximale (PPO) pour trouver une politique de planification de tâches optimale. En raison de l'hétérogénéité du système, nous proposons un algorithme basé sur l'apprentissage fédéré pondéré proximal pour implémenter l'algorithme PPO décentralisé qui améliore la performance des agents PPO distribués qui sont déployés dans les différents entrepôts géographiquement distribués. Les résultats de notre démontrent l'efficacité de notre algorithme proposé par rapport aux méthodes existantes.

3. Enfin, nous proposons un modèle pour provisionner des services 5G et controller simultanéement la robotique en essaim dans un entrepôt automatisé. Nous visons à maximiser l'efficacité énergétique à long terme tout en respectant la contrainte de consommation d'énergie des robots et les exigences de communication ultra-fiable et à faible latence (URLLC) entre le contrôleur central et la robotique en essaim. Ce modèle d'optimisation est non-convexe puisque le taux réalisable et la probabilité d'erreur de décodage avec une courte longueur de bloc ne sont ni convexes ni concaves. Nous proposons une approche basée sur l'apprentissage par renforcement profond qui utilise la méthode du gradient de politique déterministe profond (DDPG) et le réseau neuronal convolutif (CNN) pour obtenir une politique de contrôle stationnaire optimale qui consiste en un certain nombre d'actions continues et discrètes. Les résultats expérimentaux montrent que notre algorithme DDPG multi-agent proposé surpasse les solutions existantes dans l'état de l'art en termes de probabilité d'erreur et d'efficacité énergétique.

Mots-clés: Provisionnement de services 5G, contrôle robotique, robotique en essaim, théorie de l'optimisation, apprentissage par renforcement profond, apprentissage fédéré

A machine learning approach for optimizing swarm robotics in 5G-enabled automated warehouses

Manh Tai HO

ABSTRACT

The fifth generation wireless network (5G) provides high-speed, low-latency and high-reliability connections that can meet the requirements of the Industrial Internet of Things (IIoT) in industrial automation, especially for robotic control. In intelligent storage, robotics play an essential role in achieving intelligent logistics solutions that include organization, planning, control and intelligent execution of goods/items flow in the warehouse. Recent advance in wireless communications and battery technologies make it possible to replace many human workers with robotic systems in order to reduce labor costs, improve warehouse work efficiency, and increase reliability. However, the deployment of swarm robotics poses new challenges in terms of control in order to coordinate many types of resources in the warehouse to deliver 5G services for robotics and plan tasks for robots.

In particular, efficient wireless resource management in a highly dynamic 5G network like in an automated warehouse is a challenging problem because extreme reliability and low latency with high mobility of robots are not efficiently solvable by traditional optimization approach.

To this end, in this thesis, we tackle the two main challenges of an automated warehouse simultaneously : i) provisioning 5G services and ii) controlling swarm robotics. The main contributions of this thesis are as follows :

1. Firstly, we formulate the problem of provisioning 5G services to serve swarm robotics in the automated warehouse as a joint clustering of coordinated multi-points (CoMP) and ultra-reliable low-latency communication (URLLC) 5G beamforming. Traditional iterative optimization approaches are not efficient in solving this non-convex real-time problem due to their high computational time. We thus propose a CoMP clustering algorithm using the combination of game theory and deep reinforcement learning Proximal Policy Optimization (PPO)method to obtain an approximate stationary solution to the global optimal solution.
2. Secondly, we study the problem of controlling autonomous heterogeneous robotic systems in the automated warehouse. We formulate a non-convex long-term queue control optimization problem to minimize the task queue length in the warehouse. Traditional solutions based on optimization approaches are not effective in managing the stochastic nature of the goods/tasks flow and a large number of robots in the system. Therefore, we propose a task scheduling algorithm based on the PPO method to find an optimal task planning policy. Due to the system's heterogeneity, we propose a federated proximal weighted learning algorithm to implement the decentralized PPO algorithm which improves the performance of distributed PPO agents deployed in different geographically distributed warehouses. Our simulation results demonstrate the effectiveness of our proposed algorithm compared to existing methods.

3. Finally, we propose a model for provisioning 5G services and controlling swarm robotics simultaneously in an automated warehouse. We aim to maximize long-term energy efficiency while meeting the energy consumption constraint of robots and the ultra-reliable low-latency communication (URLLC) requirements between the central controller and the swarm robotics. This optimization model is non-convex since the achievable rate and decoding error probability with short block length are neither convex nor concave. We propose a deep reinforcement learning approach that uses the Deep Deterministic Policy Gradient (DDPG) method and the Convolutional Neural Network (CNN) to obtain an optimal stationary control policy which consists of a number of continuous and discrete actions. The experimental results show that our proposed multi-agent DDPG algorithm outperforms existing solutions in the state-of-the-art in terms of error probability and energy efficiency.

Keywords: 5G service provisioning, robotic control, swarm robotics, optimization theory, deep reinforcement learning, federated learning

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ETS	École de Technologie Supérieure
5G	La cinquième génération (The fifth-generation)
WMS	Systèmes de gestion d'entrepôt (warehouse management systems)
RMFS	Systèmes d'accomplissement mobiles robotisés (robotic mobile fulfillment systems)
AVS/RS	Systèmes de stockage et de récupération autonomes basés sur des véhicules (autonomous vehicle-based storage and retrieval systems)
IIoT	Internet industriel des objets (Industrial Internet of Things)
AGV	Véhicules guidés automatisés (Automated Guided Vehicles)
AMR	Robots mobiles autonomes (Autonomous Mobile Robots)
URLLC	Communication ultra-fiable et à faible latence (Ultra-reliable and low-latency communication)
CoMP	Multipoint coordonné (Coordinated Multi-Point)
gNBs	Stations de base radio (radio base stations)
RAN	Réseau d'accès radio (Radio Access Network)
eMBB	Haut débit mobile amélioré (enhanced mobile broadband)
DRL	Apprentissage par renforcement profond (Deep Reinforcement Learning)
FL	Apprentissage par renforcement (Federated Learning)
PPO	Optimisation de la politique proximale (Proximal Policy Optimization)
DDPG	Gradient politique déterministe profond (Deep Deterministic Policy Gradient)
TRPO	Optimisation de la politique de région de confiance (Trust Region Policy Optimization)
SINR	Rapport signal sur interférence plus bruit (signal-to-interference-plus-noise ratio)

CSI	Informations sur l’état du canal (channel state information)
MDP	Processus décisionnel de Markov (Markov Decision Process)
RSRP	Puissance reçue du signal de référence (reference signal’s received power)
SGD	Descente de gradient stochastique (stochastic gradient descent)
HAR	Robotique autonome hétérogène (heterogeneous autonomous robotic)
PWFL	Apprentissage fédéré pondéré proximal (Proximal Weighted Federated Learning)
WFL	Apprentissage par renforcement profond fédéré pondéré (Weighted Federated Deep Reinforcement Learning)
DL	Apprentissage distribué (Distributed Learning)
CL	Apprentissage centralisé (Centralized Learning)
PFL	FedProx
FSF	Le serveur le plus rapide en premier (Fastest Server First)
CNN	Réseau de neurones convolutifs (convolutional neural network)
MAPF	Recherche de chemin multi-agents (Multi-agent path finding)
MIMO	Entrées multiples sorties multiples (multiple-input multiple-output)
MADDPG	Gradient de politique déterministe profond multi-agents (multi-agent deep deterministic policy gradient)
BW	Bande passante (Bandwidth)

LISTE DES SYMBOLES ET UNITÉS DE MESURE

$\mathbf{A}$,$\mathbf{a}$,a	Matrice, vecteur et scalaire
$\mathbf{A}^T$, $\mathbf{A}^{-1}$	Transposée et inverse d’une matrice
$\|.\|$	Norme
$\mathbb{E}$	Espérance (1er moment) d’une variable aléatoire
∇	Opérateur de gradient (dérivée du premier ordre)
$\mathcal{O}()$	Fonction de complexité de calcul
θ	Paramètres de réseau de la politique
ϕ	Paramètres de réseau de la valeur

INTRODUCTION

Ce chapitre fournit un survol de l'entrepôt automatisé et de ses défis. Nous discutons également l'architecture du réseau 5G dans l'entrepôt automatisé et la robotique en essaim. La motivation et la contribution de cette thèse sont également fournies.

0.1 Contexte générale

0.1.1 L'entrepôt automatisé

Pendant les dernières années, la logistique intelligente est devenue l'un des principaux facteurs de la " quatrième révolution industrielle " ou simplement " l'industrie 4.0 ". La logistique intelligente comprend l'organisation, la planification, le contrôle et l'exécution intelligents du flux de marchandises. L'entrepôt intelligent ou entrepôt automatisé, joue un rôle essentiel dans la logistique intelligente. Dans l'entrepôt intelligent, les robots sont déployés pour automatiser des tâches différentes. Aujourd'hui, les robots ont pu remplacer ou aider les humains dans toutes sortes de tâches, telles que la cueillette, l'emballage, le déplacement et le stockage d'articles. De plus, l'évolution des technologies mécaniques et de batterie d'énergie permet à augmenter le temps de service/temps de charge des robots mobiles ; ainsi, le remplacement des travailleurs humains par le système robotique peut réduire les coûts de main-d'œuvre, améliorer l'efficacité du travail en entrepôt et augmenter la fiabilité Liu, Wang, Wei, Liu & Liu (2020). Les technologies de détection avancées permettent aux robots d'obtenir des connaissances et une vision en temps réel de leur environnement. Par conséquent, les robots peuvent s'adapter dynamiquement aux changements de leur environnement. Un entrepôt intelligent nécessite un groupe de robots (appelé également une une robotique en essaim) réaliser une mission ou un groupe de missions. Par exemple, Amazon a employé plus de 200 000 robots dans leur 20 centres de distribution à travers le monde Koetsier (2022). Ce déploiement massive de la robotique en essaim impose de nouveaux défis aux systèmes de gestion d'entreposage intelligents. Par

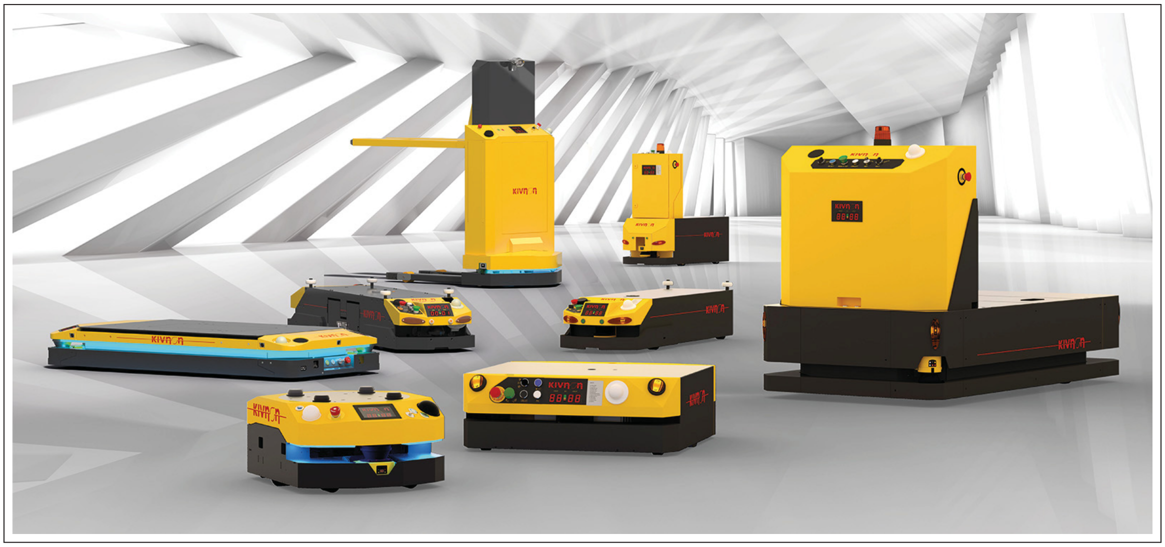


Figure 0.1 Les véhicules guidés automatisés (AGV) Kivnon (2023)

exemple, de nouvelles techniques de contrôle sont nécessaires pour coordonner de nombreux types de ressources dans un entrepôt automatisé afin de résoudre divers problèmes critiques tels que le provisionnement des services 5G pour servir la robotique en essaim, l'acheminement de multi-robots et la planification de tâches pour les robots.

0.1.2 Les besoins et les contraintes des robots dans l'entrepôt intelligent

Conçus pour augmenter la productivité, la précision et l'efficacité opérationnelle, les robots d'entrepôt sont devenus indispensables dans les entrepôts intelligents. Ces robots augmentent les performances de l'entrepôt en automatisant l'opération et les tâches répétitives, permettant ainsi aux travailleurs humains de se concentrer sur des tâches plus compliquées. Les types de robots principaux utilisés dans les entrepôts intelligents sont les suivants Kivnon (2023) :

a) Les véhicules guidés automatisés (AGV) aident à transporter les matériaux, les fournitures et les stocks dans les entrepôts. Les AGV sont utilisés dans les opérations pour remplacer les chariots élévateurs manuels ou les chariots de prélèvement. Certains AGV naviguent de manière autonome dans l'entrepôt suivant des itinéraires établis qui sont marqués par des fils, des bandes magnétiques, des pistes, des capteurs intégrés dans le sol ou d'autres guides physiques. D'autres AGV utilisent des caméras, le LIDAR, l'infrarouge et d'autres technologies avancées pour naviguer dans les espaces de travail, identifier les obstacles et éviter les collisions 6river (2020).

b) Les robots mobiles autonomes, ou AMR, sont des appareils mobiles qui peuvent naviguer de manière autonome dans un entrepôt ou un centre de distribution. Contrairement aux AGVs, les AMRs peuvent se déplacer dans l'entrepôt sans guidage externe puisque leur capteurs et cameras leur permettent de créer une carte numerique de leur environnement. Le système de gestion d'entrepôt (WMS) transmet les tâches aux robots, avec une cartographie numérique. Les capteurs, caméras et mécanismes de sécurité intégrés fournissent toutes les informations dont les robots ont besoin pour fonctionner.

Dans le système de gestion d'entrepôt, le contrôle robotique est un élément essentiel qui est responsable de l'optimisation de l'utilisation des ressources robotiques, améliorant ainsi la productivité globale de l'entrepôt. Un contrôle robotique innovant est inévitable pour faire face à un grand nombre de robots dans les entrepôts récents (par exemple, Amazon a déployé plus de 200 000 de robots opérant dans leurs entrepôts Koetsier (2022)) et à la forte demande sans précédent des clients Berthene (2022). Il est essentiel de développer un mécanisme de contrôle robotique puissant et adaptatif pour les entrepôts intelligents de nouvelle génération, capable d'allouer et de planifier efficacement les ressources robotiques pour la gestion des flux de marchandises et de commandes.

0.1.3 Provisionnement de services 5G pour les robots d'entrepôt

L'évolution récente de l'industrie manufacturière a vu le jour grâce majoritairement aux innovations des communications sans fil 5G, de la technologie d'automatisation et de l'intelligence artificielle. Techniquement, le principal avantage du réseau 5G est de fournir une plus grande fiabilité, une latence plus faible, une connectivité transparente et omniprésente avec un débit extrêmement élevé pour les connexions de type humain et machine Ho *et al.* (2019).

Cette nouvelle capacité de la 5G permet de créer des robots sans fil avec une plus grande agilité pour les usines intelligentes. Cela augmente énormément la productivité de l'entreprise car ces robots peuvent réaliser des tâches compliquées et dangereuses qui ne sont pas appropriées pour l'homme. De plus, le réseau 5G peut faciliter les tâches de synchronisation et de gestion des robots en déchargeant certains calculs réalisé par des robots sur les serveurs de périphérie. Cela permet d'augmenter considérablement le nombre de robots dans l'entrepôt intelligente, améliorant ainsi la productivité, sans avoir un impact plus important sur le coût de traitement Ho *et al.* (2019).

Le service de communication ultra-fiable et à faible latence (URLLC) fourni par le réseau sans fil 5G est une technique prometteuse pour répondre aux exigences strictes de la communication dans l'automatisation industrielle, e.g., la probabilité de perte de paquets de 10^{-9} et la disponibilité de 99,9999 % sont demandées pour controller le mouvement et des robots mobiles 3GPP (2018). Le réseau 5G fournit une connectivité massive et un service de communication haute performance dans l'industrie manufacturière par rapport aux générations de réseaux sans fil précédentes Jayaweera, Marasinghe, Rajatheva & Latva-Aho (2020). Cependant, la gestion efficace des ressources sans fil dans le réseau 5G, telles que la bande passante et la puissance de transmission, est un nouveau défi dans l'entrepôt, car l'extrême fiabilité et la faible latence nécessitent un nouveau modèle sophistiqué qui n'est gérable par les approches traditionnelles.

0.2 Motivation

0.2.1 Provisionnement de services 5G pour entrepôt automatisé

Comme souligné précédemment, le réseau sans fil 5G offre un service de communication ultra-fiable et à faible latence (URLLC) qui répond aux exigences strictes de l'automatisation des entrepôts. Cependant, il est extrêmement difficile de garantir une fiabilité et une connectivité massives dans l'environnement dynamique des entrepôts automatisés, où de nombreux robots à haute mobilité tels que les robots mobiles autonomes (AMR) et les véhicules guidés automatisés (AGV) coexistent.

Les approches d'optimisation itératives traditionnelles ne sont pas adaptées pour gérer efficacement cet environnement dynamique en temps réel, caractérisé par une grande mobilité des éléments. Bien que ces approches puissent converger vers une solution optimale, elles se heurtent à des problèmes de complexité et de temps de calcul élevés, les rendant inappropriées pour les applications critiques des entrepôts automatisés.

Pour surmonter ces défis, des approches basées sur l'apprentissage automatique, telles que le Deep Reinforcement Learning (DRL), se sont révélées prometteuses. Le DRL permet aux robots de prendre des décisions en temps réel en se basant sur des données sensorielles, et il est capable d'apprendre et de s'adapter de manière autonome à l'environnement complexe de l'entrepôt. Contrairement aux approches traditionnelles, le DRL peut gérer des espaces d'actions continus, ce qui est courant dans les problèmes de contrôle robotique.

L'utilisation du DRL offre la possibilité de prendre en compte la mobilité des éléments dans l'entrepôt automatisé, permettant aux robots de naviguer de manière autonome, de planifier des trajectoires efficaces et d'éviter les obstacles en temps réel. De plus, le DRL permet d'optimiser les opérations logistiques en prenant en compte la dynamique de l'environnement et en s'adaptant aux conditions changeantes.

En résumé, l'utilisation d'approches basées sur l'apprentissage automatique, comme le DRL, constitue une solution prometteuse pour relever les défis de l'automatisation des entrepôts. Ces approches permettent une adaptation en temps réel, une optimisation des performances opérationnelles et une gestion efficace de la mobilité des éléments dans un environnement dynamique.

0.2.2 Contrôle robotique dans l'entrepôt automatisé

L'évolution du commerce électronique, amplifiée par la demande sans précédent d'achats en ligne à l'ère du COVID, entraîne un flux constant de commandes pour les détaillants en ligne Berthene (2022). Cette augmentation du volume de commandes pose des défis complexes pour les systèmes robotiques dans les entrepôts automatisés, en particulier en raison de l'environnement stochastique et de la complexité résultant de la gestion simultanée de multiples commandes clients.

La gestion efficace de ce type d'environnement nécessite un nouveau type de système de contrôle robotique et de gestion robotique, capable de gérer la hétérogénéité des systèmes robotiques présents dans l'entrepôt. Cependant, les travaux actuels sur le contrôle robotique des entrepôts automatisés ne prennent pas pleinement en compte cette hétérogénéité, ce qui limite leur capacité à optimiser les performances et à s'adapter aux divers besoins et configurations des systèmes robotiques.

Il est essentiel de développer des approches de contrôle robotique qui capturent et exploitent pleinement l'hétérogénéité des systèmes robotiques dans l'entrepôt. Cela permettra d'optimiser les opérations logistiques, d'améliorer l'efficacité et de réduire les coûts. Des méthodes basées sur l'apprentissage automatique, telles que le Deep Reinforcement Learning (DRL), se sont avérées prometteuses pour relever ces défis en permettant aux robots de s'adapter de manière autonome aux différentes tâches et configurations du système.

En intégrant des approches de contrôle robotique avancées et des techniques d'apprentissage automatique, il est possible de créer des systèmes robotiques adaptatifs et polyvalents capables de gérer efficacement l'hétérogénéité des systèmes, tout en optimisant les performances globales de l'entrepôt. Cette approche permettra de répondre aux exigences croissantes du commerce électronique et de garantir une efficacité opérationnelle maximale dans les entrepôts automatisés modernes.

0.2.3 Apprentissage automatique pour l'automatisation industrielle

Dans un système d'entrepôt robotique autonome hétérogène (HAR), les approches traditionnelles de provision de services 5G et de planification des tâches sont inefficaces pour faire face à la nature stochastique du flux de marchandises et à la caractéristique hétérogène des multiples types de nouvelles générations de robots mobiles autonomes. Grâce à ses performances exceptionnelles dans le traitement des problèmes décisionnels stochastiques, l'apprentissage par renforcement profond (DRL) s'est révélé être une technique efficace pour développer des algorithmes adaptatifs dans divers domaines Sutton & Barto (2018); Lillicrap *et al.* (2015); Schulman, Wolski, Dhariwal, Radford & Klimov (2017).

Dans l'automatisation industrielle, les fonctions d'intelligence artificielle et d'apprentissage automatique nécessitent la collecte de données auprès de plusieurs éléments de fabricants et d'usines, ce qui soulève des problèmes de confidentialité. L'apprentissage fédéré (FL) McMahan, Moore, Ramage, Hampson & y Arcas (2017) est récemment apparu comme une technique prometteuse pour la formation de modèles dans des environnements distribués tout en protégeant la confidentialité des données en conservant les données de formation dans les appareils locaux. FL peut percevoir des méthodes d'apprentissage automatique décentralisées, en particulier DRL dans l'automatisation industrielle en permettant à des appareils géographiquement distribués de former leurs propres modèles sans divulguer de données sensibles et privées. Le modèle global acquiert les expériences de tous les agents DRL tout en protégeant éventuellement la sécurité et la confidentialité de chaque appareil.

FL est une technique prometteuse pour réaliser le DRL décentralisé grâce à son avantage dans la protection de la sécurité et de la confidentialité des participants Wang, Wang, Li, Leung & Taleb (2020e). De plus, en agrégeant les paramètres formés adéquats des participants, les performances de DRL peuvent être considérablement améliorées par rapport à d'autres techniques de formation centralisées et distribuées Wang *et al.* (2020e); Cao, Lien, Liang, Chen & Shen (2021), en particulier dans des environnements hétérogènes Wang *et al.* (2020d). Le DRL fédéré a montré son avantage dans de nombreux problèmes pratiques, par exemple, la mise en cache de périphérie mobile d'appareil à appareil Wang *et al.* (2020d), le contrôle d'accès utilisateur du réseau d'accès radio ouvert Cao *et al.* (2021), la gestion des données de l'Internet industriel des objets (IIoT) Zhang, Wang, Jiang & Han (2021), et gestion de l'énergie de la maison intelligente Lee & Choi (2020).

0.3 L'énoncé du problème

Comme mentionné précédemment, l'évolution des nouvelles générations de robots mobiles autonomes dans les entrepôts automatisés et la demande croissante des achats en ligne ont introduit de nombreux défis dans le déploiement du réseau 5G et le contrôle robotique associé. Le déploiement de la robotique en essaim pose de nouveaux défis aux systèmes d'entreposage intelligents, nécessitant des techniques de contrôle avancées pour coordonner efficacement les différentes ressources dans l'entrepôt.

Ces défis incluent des aspects tels que l'allocation efficace des ressources sans fil 5G, la planification de chemins pour les robots et la coordination des tâches multi-robots. Dans le cadre de cette thèse, notre objectif est de relever ces défis en proposant des solutions novatrices pour le déploiement du service 5G dans l'entrepôt automatisé et en développant des mécanismes de contrôle robotique efficaces.

Pour ce faire, nous nous concentrerons sur des problématiques clés telles que l'optimisation de la qualité de service du réseau 5G, la planification des trajets pour les robots mobiles autonomes et la coordination des tâches entre les robots. Nous visons à développer des algorithmes et des approches de contrôle innovants qui permettront d'optimiser les performances globales du système tout en répondant aux exigences spécifiques de l'entrepôt automatisé.

En résolvant ces problèmes de recherche, nous contribuerons à améliorer l'efficacité et la fiabilité du déploiement du service 5G dans l'entrepôt automatisé, ouvrant ainsi la voie à des opérations logistiques plus efficaces et à une meilleure gestion des ressources. Cette thèse offre une opportunité unique d'explorer les synergies entre la robotique avancée et les réseaux de communication 5G pour répondre aux défis actuels et futurs de l'automatisation des entrepôts.

Dans le but de relever les défis du déploiement du service 5G dans l'entrepôt automatisé ainsi que de fournir un provisionnement efficace du service 5G et un mécanisme de contrôle robotique, le problème de recherche à résoudre dans cette thèse peut être énoncé comme suit :

> « **Comment provisionner le service 5G pour le contrôle de la robotique en essaim dans un entrepôt automatisé d'une manière optimale en termes de la performance du réseau et des robots simultanément ?** »

Afin de répondre à l'énoncé du problème ci-dessus, nous détaillons davantage l'énoncé du problème en trois questions de recherche (RQ) comme suit :

0.3.1 Question de recherche RQ1

RQ1 : Comment concevoir un mécanisme de provisionnement de service 5G optimal dans un entrepôt automatisé ?

Les principaux défis liés à cette question sont :

1. Comment garantir une fiabilité extrêmement élevée du service 5G uRLLC avec un essaim de robots à haute mobilité et des obstructions physiques d'un entrepôt automatisé ?
2. Comment gérer les interférences entre la communication de plusieurs robots mobiles ?
3. Comment gérer les variations de fréquence radio hautement dynamiques des robots en mouvement ?

0.3.2 Question de recherche RQ2

RQ2 : Comment concevoir un mécanisme de contrôle robotisé optimal dans un entrepôt automatisé ?

Les principaux défis liés à cette question sont :

1. Comment gérer l'hétérogénéité du système robotique dans un entrepôt automatisé ?
2. Comment aborder le flux stochastique du bien et de l'ordre avec un système robotique hétérogène ?
3. Comment surmonter la complexité du système d'entrepôt à grande échelle ?

0.3.3 Question de recherche RQ3

RQ3 : Comment concevoir un mécanisme conjoint optimal de provisionnement de service 5G et de contrôle robotique qui optimise les performances du robot dans un entrepôt automatisé ?

Les principaux défis liés à cette question sont :

1. Comment programmer la tâche au robot qui respecte la contrainte de consommation énergétique du robot ?
2. Comment garantir une fiabilité extrêmement élevée du service 5G uRLLC avec un essaim de robots à haute mobilité ?
3. Comment optimiser la planification de trajectoire des robots pour accomplir leurs tâches spécifiques dans des contraintes de temps critiques tout en évitant simultanément les collisions potentielles avec d'autres robots ou objets dynamiques dans la zone de travail ?
4. Comment utiliser efficacement les ressources radio avec un nombre massif de robots mobiles ?

Notre question de recherche est illustrée dans la figure 0.2.

0.4 La structure de la thèse

Cette thèse basée sur des articles est organisée comme une compilation d'articles publiés ou soumis à des revues prestigieuses et de haut niveau de l'IEEE dans la gestion des services réseau et l'ingénierie des sciences de l'automatisation en réponse aux questions de recherche spécifiques énumérées ci-dessus. En excluant ce chapitre introductif, le reste de cette thèse est organisé comme suit :

- **Chapitre 1** passe en revue les travaux antérieurs liés à la portée de la recherche menée dans cette thèse.
- **Chapitre 2** définit les objectifs de cette thèse et présente la méthodologie adoptée pour répondre à chaque question de recherche.
- **Chapitres 3 à 5** présentent nos travaux publiés et soumis selon les objectifs de la thèse. Chaque chapitre consiste en un article qui cible l'une des trois questions de recherche de cette thèse. Chaque chapitre suit une structure spécifique dictée par la revue correspondante. Pourtant, seules des modifications périphériques (par exemple, le cadrage des figures, le

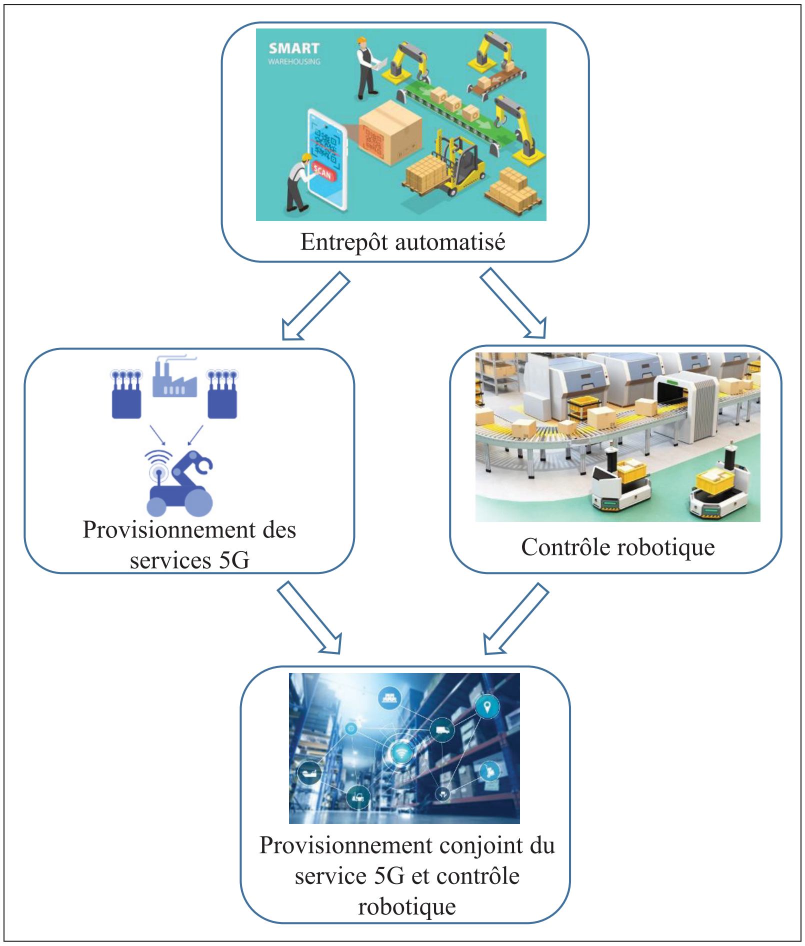

SMART
WAREHOUSING
Entrepôt automatisé
Provisionnement des services 5G
Contrôle robotique
Provisionnement conjoint du service 5G et contrôle robotique

Figure 0.2   Diagramme des questions de recherche

repositionnement et la mise à l'échelle) ont été apportées selon les directives de thèse de l'École de Technologie Supérieure.

- **Chapitre 6** fournit une discussion générale, qui détaille les points forts et les limites des solutions proposées.
- **Conclusion et travaux futurs** résume les contributions de cette thèse et énumère quelques recommandations d'améliorations et d'orientations futures pour s'appuyer sur cette thèse.

CHAPITRE 1

REVUE DE LITTÉRATURE

Ce chapitre propose une revue de l'état de l'art des travaux de recherche pertinents dans le domaine de cette thèse. Nous avons classé les travaux antérieurs liés à nos recherches en deux domaines principaux : i) le contrôle robotique dans l'entrepôt automatisé, et ii) la provision de services 5G pour l'entrepôt automatisé.

Il est important de souligner que le contrôle robotique dans l'entrepôt automatisé repose sur la provision de services 5G. Ces deux domaines sont étroitement liés par nature. Afin d'obtenir une solution optimale de contrôle robotique, il est essentiel de concevoir une solution optimale de provisionnement de services 5G.

Dans la première partie de cette revue, nous explorons les travaux existants sur le contrôle robotique dans l'entrepôt automatisé. Nous examinons les différentes approches de contrôle, telles que les algorithmes de planification de trajectoire, la coordination des robots, et la gestion des ressources. Nous identifions les avancées récentes, les défis actuels et les lacunes à combler dans ce domaine.

Dans la deuxième partie, nous nous concentrons sur la provision de services 5G pour l'entrepôt automatisé. Nous passons en revue les recherches sur l'optimisation de la qualité de service, la gestion des ressources sans fil, et la coordination des communications dans le contexte de l'entrepôt automatisé. Nous examinons également les approches de déploiement du réseau 5G et les solutions de connectivité pour les robots.

En comprenant l'état de l'art dans ces deux domaines, nous serons en mesure de proposer une solution de contrôle robotique optimale qui intègre efficacement la provision de services 5G. Cette approche holistique permettra d'exploiter pleinement les avantages des deux domaines et de relever les défis liés à l'automatisation de l'entrepôt de manière efficace et efficace.

1.1 Entrepôt robotisé automatisé

1.1.1 Systèmes de gestion d’entrepôt

Les systèmes robotiques de gestion d’entrepôt ont été bien étudiés pendant des décennies Roy, Krishnamurthy, Heragu & Malmborg (2013); Ekren, Heragu, Krishnamurthy & Malmborg (2012); Yuan & Gong (2017); Wang, Wu, Zheng & Chi (2020c); Ma *et al.* (2022); Lee & Murray (2019). Ces systèmes de gestion d’entrepôt comprennent des systèmes de stockage et de récupération autonomes basés sur des véhicules (AVS/RS) Roy *et al.* (2013); Ekren *et al.* (2012), des systèmes de traitement des commandes mobiles robotiques (RMFS) Yuan & Gong (2017); Wang *et al.* (2020c); Ma *et al.* (2022) et des systèmes de prélèvement robotisés autonomes. Roy *et al.* (2013) sont proposé un modèle de réseau de file d’attente semi-ouvert pour analyser les performances du système et évaluer les compromis de conception dans un AVS/RS. Ekren *et al.* (2012) sont présenté un modèle de réseau de file d’attente semi-ouvert à plusieurs serveurs à classe unique pour un AVS/RS. Yuan & Gong (2017) sont formulé deux modèles de réseau à file d’attente ouverte pour déterminer le nombre optimal de robots afin de minimiser le débit total dans un RMFS afin d’obtenir une gestion efficace des robots d’entreposage. Wang *et al.* (2020c) sont proposé un modèle de réseau de file d’attente ouvert pour RMFS modulaire avec des robots captifs d’allée pour les entrepôts logistiques de petite et moyenne taille. Le RMFS modulaire proposé est divisé en modules indépendants, chacun avec un poste de travail, un picker et des stations de pod dans la zone de picking et plusieurs allées indépendantes dans la zone de stockage. Ma *et al.* (2022) sont proposé une nouvelle politique de stockage dispersée nommée politique de stockage à corrélation dispersée basée sur la classification des produits pour atténuer le problème d’affectation du stockage des produits dans RMFS. Lee & Murray (2019) sont formulé un routage coordonné de deux types de robots autonomes hétérogènes, à savoir un robot de prélèvement et un robot de transport, les robots Fetch & Freight, afin de minimiser le temps nécessaire pour récupérer une collection d’articles provenant d’un entrepôt. Contrairement au système robotique Kiva d’Amazon (rebaptisé Amazon Robotics) Boysen, De Koster & Weidinger

(2019) qui transporte des racks entiers depuis/vers la zone de réapprovisionnement, les robots Fetch & Freight récupèrent des articles individuels dans l'entrepôt.

1.1.2 Contrôle robotique dans un entrepôt automatisé

Alors que le système logistique se développe continuellement avec le développement de l'automatisation robotique, la *planification des tâches* devient urgente dans le système d'entrepôt robotique moderne (RWS). Plusieurs travaux portent sur le problème d'ordonnancement dynamique des tâches de RWS Kim, Pais & Shen (2020); Bolu & Korçak (2021); Tang, Wang, Xue, Yang & Cao (2021); Yoshitake, Kamoshida & Nagashima (2019); Zou, Gong, Xu & Yuan (2017); Li, Barenji, Jiang, Zhong & Xu (2020c); Yang *et al.* (2021c); Roy, Nigam, de Koster, Adan & Resing (2019); Zhou, Shi, Wang & Yang (2014). Kim *et al.* (2020) sont proposé un algorithme heuristique efficace pour attribuer des éléments aux pods dans un RMFS. Bolu & Korçak (2021) sont proposé un modèle heuristique paramétrique pour le processus de sélection des tâches de commande pour un RMFS dans un environnement de simulation réaliste. Tang *et al.* (2021), un algorithme basé sur la critique d'acteur souple et la DRL hiérarchique a été proposé pour l'allocation de tâches de robot multi-logistique. Yoshitake *et al.* (2019) a proposé un nouveau système robotique utilisant un AGV pour la préparation des commandes dans les entrepôts logistiques qui planifie de manière flexible les tâches de transport des étagères d'inventaire et des étagères de tri à l'aide d'une méthode de planification holonique en temps réel. Zou *et al.* (2017) sont proposé une règle d'affectation basée sur les vitesses de traitement des postes de travail et a conçu un algorithme de recherche de voisinage pour trouver une règle d'affectation quasi optimale pour un RMFS. Li *et al.* (2020c) sont proposé un nouveau mécanisme de planification pour les problèmes multi-robots et d'allocation des tâches dans un système d'entrepôt intelligent avec plusieurs demandes simultanées de clients. Yang *et al.* (2021c) sont proposé une approche heuristique adaptative pour attribuer les tâches aux robots dans un entrepôt intelligent basé sur RMFS, en tenant compte de la dynamique du système, comme l'emplacement des robots et des pods, l'utilisation des bacs et l'âge des tâches. Roy *et al.* (2019) sont analysé un RMFS pour une zone de stockage unique et multiple avec une affectation de robots dédiés et groupés basée sur des modèles de réseau à file d'attente fermée multi-classes. Zhou *et al.* (2014) sont proposé un mécanisme d'équilibre heuristique pour assigner des tâches à robots qui se concentrent sur l'équilibrage de la charge de travail du robot tout en optimisant le temps de déplacement total du robot.

1.1.3 Approches basées sur l'apprentissage par renforcement profond pour le contrôle robotique dans un entrepôt automatisé

L'apprentissage par renforcement profond (DRL) a connu un succès remarquable dans la résolution de problèmes de prise de décision stochastiques, offrant ainsi des avancées significatives dans le domaine du contrôle robotique Luo, Ni, Tian & Cheng (2022); Bai, Yan, Pan & Guo (2021); Tan, Bejarano, Zhu, Ren & Nejat (2022); Han *et al.* (2022b); Karimi & Ahmadi (2021); Han *et al.* (2022a); Choi, Kim, Han, Oh & Kim (2022); Zhu, Li, Zhe & Zhang (2022). Le contrôle robotique est un défi complexe en raison de la nature complexe des systèmes physiques impliqués et de la nécessité de réaliser un contrôle en temps réel dans des dimensions étendues. C'est là que le DRL se démarque en tant qu'approche prometteuse pour le contrôle robotique, car il est capable d'apprendre directement des politiques de contrôle à partir des entrées sensorielles, sans dépendre de modèles explicites du système Mnih *et al.* (2015); Lillicrap *et al.* (2015).

Une des grandes forces du RL par rapport aux autres techniques d'apprentissage automatique dans le domaine du contrôle robotique est sa capacité à gérer des espaces d'actions continus, qui sont couramment rencontrés dans les applications robotiques concrêtes. En effet, de nombreux problèmes de contrôle du monde réel nécessitent la manipulation d'actions continues, ce qui est également le cas du problème abordé dans cet thèse. Les méthodes de DRL permettent d'apprendre des politiques de contrôle continues en paramétrant la distribution des actions, ce qui permet à l'agent d'exprimer une distribution de probabilité sur l'espace des actions continues.

En explorant les travaux antérieurs, on constate que le DRL a été utilisé avec succès pour concevoir des algorithmes de contrôle adaptatifs pour les robots, tels que le DRL fédéré Luo *et al.* (2022) et le contrôle adaptatif multi-robot basé sur le DRL Bai *et al.* (2021). D'autres approches basées sur le DRL ont été proposées pour des problèmes spécifiques tels que l'exploration multi-robot décentralisée Tan *et al.* (2022), la navigation à l'aide de capteurs peu coûteux Han *et al.* (2022b), les tâches de contrôle cinématique Karimi & Ahmadi (2021), l'évitement de collision dans des scénarios encombrés Zhu *et al.* (2022), et l'apprentissage par renforcement multi-agent pour le contrôle coopératif de véhicules guidés automatisés (AGVs) dans des systèmes d'entrepôts Choi *et al.* (2022). Ces études démontrent l'efficacité du DRL dans le développement d'algorithmes de contrôle robotique adaptatifs et décentralisés pour une variété de tâches telles que la navigation, l'exploration et l'évitement de collision.

Le DRL a ouvert de nouvelles perspectives dans le domaine du contrôle robotique en permettant l'apprentissage de politiques de contrôle directement à partir des entrées sensorielles, ainsi que la gestion d'espaces d'actions continus. Cette approche offre des avantages significatifs pour résoudre les problèmes complexes de contrôle dans des environnements réels, ouvrant ainsi la voie à de nouvelles applications et possibilités dans le domaine de la robotique.

Des études antérieures ont démontré l'efficacité du DRL dans la conception d'algorithmes de contrôle adaptatifs pour les robots. Par exemple, le DRL fédéré Luo *et al.* (2022) a été utilisé pour développer des stratégies de contrôle adaptatif qui permettent aux robots d'apprendre et de s'adapter en temps réel à des environnements changeants. De même, le contrôle de formation adaptatif multi-robots basé sur le DRL Bai *et al.* (2021) a permis de créer des systèmes de robots capables de collaborer et de s'adapter collectivement pour accomplir des tâches complexes.

En dehors du domaine de la formation, le DRL a également été appliqué avec succès à d'autres problèmes de contrôle robotique. Par exemple, l'exploration multi-robots décentralisée Tan *et al.* (2022) a utilisé le DRL pour permettre à un groupe de robots de coopérer et d'explorer efficacement un environnement inconnu. De même, le contrôle de la navigation à l'aide de capteurs peu coûteux Han *et al.* (2022b) a exploité les avantages du DRL pour permettre aux robots de naviguer de manière autonome en utilisant des capteurs peu coûteux et disponibles dans des applications du monde réel.

Les tâches de contrôle cinématique Karimi & Ahmadi (2021) ont également bénéficié du DRL en permettant aux robots de maîtriser des mouvements complexes et précis. Le DRL a été utilisé pour apprendre des politiques de contrôle qui permettent aux robots d'effectuer des tâches cinématiques avec une grande précision et une efficacité accrue.

En ce qui concerne la sécurité, le DRL a été utilisé pour résoudre des problèmes d'évitement de collision dans des scénarios encombrés de robots Zhu *et al.* (2022). Les robots équipés de DRL ont été capables d'éviter les collisions de manière autonome en analysant les informations sensorielles et en prenant des décisions intelligentes en temps réel.

Enfin, l'apprentissage par renforcement multi-agent a été appliqué avec succès au contrôle coopératif de plusieurs véhicules guidés automatisés (AGV) dans les systèmes d'entrepôt Choi *et al.* (2022). Les AGV, grâce au DRL, peuvent coordonner leurs actions et collaborer efficacement pour optimiser les opérations d'entreposage et de récupération.

Ces études mettent en évidence la polyvalence et l'efficacité du DRL dans le développement d'algorithmes de contrôle robotique adaptatifs et décentralisés pour diverses tâches telles que la navigation, l'exploration, l'évitement de collision et le contrôle coopératif. Le DRL offre une approche prometteuse pour résoudre les défis complexes du contrôle robotique, ouvrant ainsi la voie à de nouvelles possibilités et avancées dans le domaine de la robotique.

1.1.4 Discussion

Les méthodes de contrôle robotique mentionnées précédemment ne capturent pas l'hétérogénéité des systèmes multi-entrepôts, qui peuvent comporter plusieurs systèmes robotiques hétérogènes. Par conséquent, il n'est pas suffisant ni efficace de déployer la même politique de contrôle robotique simultanément dans des entrepôts géographiquement dispersés.

Dans la pratique, l'hétérogénéité des systèmes est une considération essentielle pour les méthodes d'apprentissage automatique. En raison de l'hétérogénéité des systèmes multi-entrepôts, les données provenant de chaque entrepôt distribué peuvent être insuffisantes pour former de manière robuste un modèle pour chaque agent d'apprentissage par renforcement profond (DRL) de manière distribuée. D'autre part, transférer les données distribuées à un agrégateur coordonnateur (contrôleur central) pour construire un modèle global peut entraîner des coûts élevés en termes de communication et de retard, ainsi que des problèmes de confidentialité. De plus, en raison de l'hétérogénéité des données distribuées, telles que la taille et le format des données, le modèle global peut subir des déviations importantes et une divergence lors du processus de formation, rendant le transfert de données pour une formation centralisée inefficace.

En résumé, l'apprentissage fédéré (FL) s'est révélé être une méthode efficace pour gérer l'hétérogénéité des appareils participants et réduire l'écart de formation des modèles dans les systèmes hétérogènes. Avec FL, au lieu de transférer les données, seuls les modèles distribués sont agrégés, ce qui permet de préserver la confidentialité des données et de réduire les coûts de communication. Cette approche permet d'exploiter les avantages de la formation distribuée tout en obtenant une politique plus performante grâce à l'agrégation des modèles distribués. Des études antérieures ont montré que l'apprentissage fédéré peut être appliqué avec succès à des domaines tels que l'apprentissage en santé, l'apprentissage sur les appareils mobiles et les réseaux de capteurs.

En conclusion, l'utilisation de l'apprentissage fédéré peut être une solution efficace pour gérer l'hétérogénéité des systèmes multi-entrepôts et permettre une formation collaborative des modèles de contrôle robotique. Cette approche offre des avantages tels que la préservation de la confidentialité des données, la réduction des coûts de communication et la possibilité de former des politiques de contrôle plus performantes dans des environnements hétérogènes. La recherche continue dans ce domaine contribuera à améliorer les méthodes de contrôle robotique distribué et adaptatif dans les systèmes multi-entrepôts.

1.2 Fourniture de services 5G pour entrepôt automatisé

Le service de communication ultra-fiable et à faible latence (URLLC) fourni par le réseau sans fil 5G est en mesure de répondre aux exigences strictes de l'automatisation des usines, par ex. Probabilité de perte de paquets de 10^{-9} et disponibilité de 99,9999 % dans les cas d'utilisation du contrôle de mouvement et des robots mobiles 3GPP (2018). Cependant, garantir une fiabilité extrêmement élevée est un défi dans un environnement aussi dynamique d'un entrepôt automatisé avec des véhicules guidés automatisés (AGV) à haute mobilité.

La technique de communication multipoint coordonnée (CoMP) Marsch & Fettweis (2011) qui tire parti de la diversité spatiale promet d'atteindre l'URLLC en envoyant des flux de données en double sur divers chemins Khoshnevisan *et al.* (2019). Dans le scénario d'entrepôt automatisé, CoMP peut combiner les signaux de plusieurs stations de base radio afin que des communications hautement fiables puissent être obtenues avec les objets en mouvement, c'est-à-dire les robots avec des obstacles physiques, par ex. racks et étagères d'entrepôt. Avec les avantages que CoMP peut apporter au réseau sans fil, fournir une communication sans fil URLLC compatible CoMP dans l'Internet industriel des objets (IIoT) est particulièrement difficile en raison des variations de radiofréquence hautement dynamiques des objets en mouvement (tels que les robots mobiles) Khoshnevisan *et al.* (2019). Par conséquent, la conception d'un approvisionnement de service 5G qui satisfait les contraintes URLLC pour la connexion de robots massifs devient plus difficile et significativement différente de celle des systèmes de communication conventionnels.

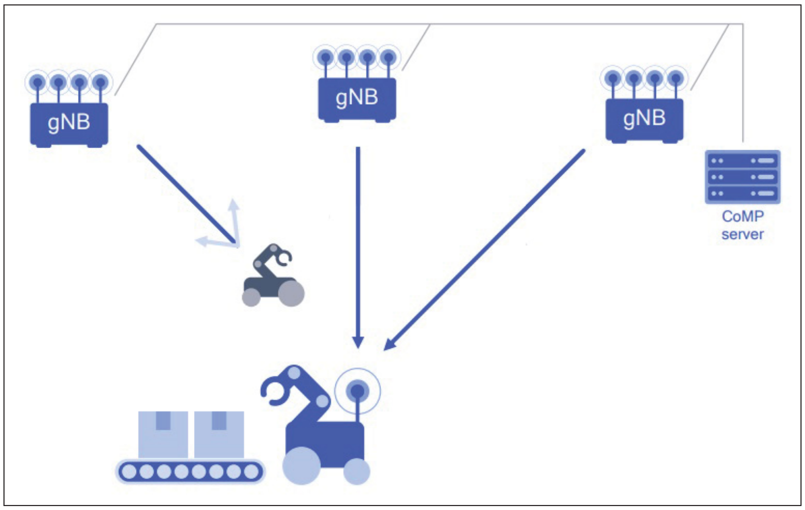


Figure 1.1 Coordinated Multi-Point (CoMP) dans l'automatisation industrielle Qualcomm (2019)

1.2.1 Service URLLC 5G compatible CoMP

Au cours des dernières années, de nombreux travaux tentent de coordonner la transmission CoMP pour améliorer le service URLLC dans le réseau 5G via la diversité spatiale. Dans Nasir, Tuan, Nguyen, Debbah & Poor (2020), Nasir *et al.* développent des algorithmes de suivi de chemin, qui génèrent une séquence de points réalisables améliorés pour résoudre le problème d'allocation des ressources et de conception de formation de faisceaux dans le régime de longueur de bloc courte pour URLLC. Yang *et al.* (2021a) formulent le problème de découpage RAN activé par CoMP pour le haut débit mobile amélioré multidiffusion (eMBB) et le multiplexage de services URLLC en rafale comme un problème d'optimisation à plusieurs échelles de temps dans le but de maximiser eMBB et Utilitaires de tranche URLLC, soumis à la bande passante totale du système et aux contraintes de puissance de transmission. Khan & Jacob (2020b) proposent un nouveau mécanisme de livraison de paquets, une stratégie de mise en file d'attente et une allocation de ressources temps-fréquence pour les URLLC compatibles CoMP dans l'architecture C-RAN. Dans Yang *et al.* (2020b), les auteurs étudient le découpage RAN activé par CoMP pour la de services URLLC et eMBB en rafale en dérivant la limite supérieure minimale de la bande passante réseau orchestrée pour la transmission du trafic URLLC afin de garantir la probabilité de blocage des paquets URLLC. Dans Khan & Jacob (2020a), les auteurs proposent un algorithme d'allocation de ressources heuristique pour les URLLC compatibles CoMP avec une communication par paquets courts en maximisant la disponibilité du CoMP. Dans Yang *et al.* (2021b), les auteurs proposent d'utiliser une méthode de direction alternée de multiplicateurs (ADMM) pour résoudre le problème d'optimisation des ressources du découpage RAN compatible CoMP pour l'Internet massif des objets (mIoT) et le multiplexage de services URLLC en rafales.

1.2.2 Approches basées sur l'apprentissage par renforcement profond pour le réseau 5G compatible CoMP

Pour surmonter les lacunes de la théorie d'optimisation traditionnelle pour le réseau 5G compatible CoMP, des travaux récents ont proposé d'utiliser l'apprentissage par renforcement profond (DRL) pour traiter des aspects importants de la communication CoMP tels que le regroupement et la conception de formation de faisceaux. Dans Al-Eryani, Akrout & Hossain (2020), un modèle DRL hybride combinant un gradient de politique déterministe profond (DDPG) et un modèle de réseau double Q profond (DDQN) est proposé pour regrouper les points d'accès et optimiser les vecteurs de formation de faisceaux afin de maximiser le débit de somme. Mismar, Evans & Alkhateeb (2019) sont proposé un algorithme basé sur le réseau Q profond (DQN) pour optimiser conjointement la formation de faisceaux, le contrôle de puissance et la coordination des interférences pour les porteurs de voix et les porteurs de données en sous-6 GHz et à ondes millimétriques dans le réseau sans fil 5G . Dans Ge, Liang, Joung & Sun (2020), les auteurs proposent un algorithme de coordination dynamique distribué de formation de faisceaux en liaison descendante basé sur la méthode DQN pour améliorer la capacité du système de ce canal d'interférence multi-entrée multi-sortie unique (MISO). Dans Wang, Peters, Liang & Hanzo (2020b), une méthode basée sur RL multi-agents est proposée pour résoudre le problème du regroupement de points de transmission/réception (TRP) centrés sur l'utilisateur et de l'association d'utilisateurs dans la technique multipoint coordonnée assistée par transmission (CoMP).

1.2.3 Discussion

Dans le contexte de la conception d'un approvisionnement de service 5G compatible CoMP (Coordinated Multi-Point), il est crucial de prendre en compte les contraintes strictes de l'URLLC (Ultra-Reliable and Low-Latency Communications). Contrairement aux systèmes de communication conventionnels, les environnements CoMP sont caractérisés par des variations dynamiques dans le temps et une grande mobilité des entités du réseau, telles que les robots mobiles.

Les approches d'optimisation itératives traditionnelles peuvent avoir des difficultés à gérer ces environnements dynamiques et mobiles, car elles sont souvent limitées par la complexité et le temps de calcul. Garantir la convergence vers une solution localement optimale peut être un défi, en particulier lorsque des contraintes strictes de latence et de fiabilité sont imposées.

C’est là que les approches basées sur l’apprentissage automatique, telles que le deep reinforcement learning (DRL), peuvent jouer un rôle important. Le DRL a la capacité d’apprendre et d’adapter des politiques de contrôle en fonction des données d’entrée et des objectifs spécifiques, ce qui peut être bénéfique dans les environnements CoMP. En utilisant des réseaux de neurones profonds, le DRL peut capturer des caractéristiques complexes des données d’entrée et générer des politiques de contrôle adaptatives en temps réel.

La conception d’un approvisionnement de service 5G compatible CoMP qui répond aux contraintes de l’URLLC peut bénéficier de l’utilisation du DRL. En permettant au système de s’adapter dynamiquement aux changements de l’environnement et de prendre des décisions basées sur des observations en temps réel, le DRL peut aider à garantir des performances optimales tout en respectant les contraintes de latence et de fiabilité requises.

Cependant, il convient de noter que l’utilisation du DRL dans les systèmes de communication présente également des défis. L’entraînement de modèles DRL nécessite généralement une quantité importante de données d’entraînement, ce qui peut être difficile à obtenir dans des environnements CoMP réels. De plus, l’exploration de l’espace des politiques de contrôle peut nécessiter des ressources de calcul significatives.

En résumé, l’utilisation du DRL dans la conception d’un approvisionnement de service 5G compatible CoMP peut offrir des avantages en termes de capacité d’adaptation et de prise de décision en temps réel. Cependant, il est important de prendre en compte les contraintes spécifiques de l’URLLC et les défis associés à l’entraînement et à l’optimisation des modèles DRL dans ces environnements. La recherche continue dans ce domaine contribuera à améliorer la conception et les performances des systèmes de communication CoMP dans des scénarios URLLC.

CHAPITRE 2

OBJECTIFS ET MÉTHODOLOGIE

Ce chapitre met en évidence les objectifs de recherche que nous visons à obtenir dans cette thèse, suivis de la méthodologie pour obtenir les objectifs de recherche présentés.

2.1 Objectif de recherche

Après avoir souligné l'urgence de développer des méthodes pour activer le service 5G dans l'entrepôt automatisé avec de nouvelles générations de robots mobiles autonomes, l'objectif principal de cette thèse est de développer les mécanismes de de service 5G et de contrôle robotique qui optimisent les performances des robots et de l'entrepôt. .

Nous affinons davantage l'objectif principal en objectifs plus spécifiques.

2.1.1 Objectif 1 :

O1 : Activer le service 5G pour le contrôle de la robotique en essaim dans un entrepôt automatisé.

Dans un entrepôt automatisé, les robots mobiles autonomes jouent un rôle crucial en se déplaçant le long des allées entre les blocs de rayonnages pour effectuer des tâches de stockage et de récupération de marchandises. Cependant, en raison de leur grande mobilité et des obstacles physiques présents dans l'environnement, les robots sont confrontés à des variations importantes du canal de communication radio. Ainsi, garantir une fiabilité extrêmement élevée devient un défi majeur, nécessitant une conception spécifique de l'approvisionnement en services 5G pour la transmission entre les stations de base et les robots mobiles, tout en respectant les contraintes d'ultra-fiabilité.

Notre premier objectif est de tirer parti de la diversité spatiale offerte par la technique de communication Coordinated Multi-Point (CoMP) pour concevoir un approvisionnement optimal des services 5G permettant de contrôler les robots avec une fiabilité maximale, en accord avec la contrainte URLLC. En exploitant les avantages de la CoMP, nous visons à surmonter les défis liés aux variations de canal et à garantir des performances de communication robustes et fiables pour soutenir les opérations fluides et efficaces des robots dans l'entrepôt automatisé.

2.1.2 Objectif 2 :

O2 : Développer un mécanisme de contrôle robotique pour les robots mobiles dans un entrepôt automatisé.

Dans un système d'entrepôt robotique mobile autonome hétérogène, les méthodes traditionnelles de gestion des robots d'entreposage se révèlent inefficaces pour faire face à la nature stochastique du flux de marchandises ainsi qu'à la diversité des multiples types de robots mobiles autonomes de nouvelle génération.

Notre deuxième objectif est d'exploiter les avancées de l'apprentissage fédéré et de l'apprentissage par renforcement profond pour gérer efficacement le système d'entrepôt robotique autonome hétérogène. Nous visons à développer une politique globale de planification des tâches optimale qui attribue de manière intelligente les différentes tâches d'entrepôt, telles que le stockage et la préparation des marchandises, aux robots de l'essaim robotique. En utilisant ces approches d'apprentissage avancées, nous cherchons à améliorer les performances des entrepôts répartis géographiquement au sein d'un système multi-entrepôts. Notre objectif est d'optimiser l'efficacité et la productivité de l'ensemble du système d'entrepôt, en exploitant les capacités de chaque robot et en les coordonnant de manière adaptative pour répondre aux besoins changeants de l'environnement d'entreposage.

2.1.3 Objectif 3 :

O3 : Développer un cadre conjoint d'approvisionnement de services 5G et de contrôle robotique dans un entrepôt automatisé qui optimise les performances des robots.

Fournissant un espace de stockage très dense, où un grand nombre de robots interagissent simultanément, il est essentiel de mettre en place un mécanisme de contrôle robotique efficace afin d'optimiser les performances des robots tout en garantissant une planification de trajectoire sans collision et en respectant les contraintes d'ultra-fiabilité du service 5G pour la connectivité entre les robots et la station de base.

Notre troisième objectif est de développer un mécanisme intégré d'approvisionnement de service 5G et de contrôle robotique pour les entrepôts automatisés. Nous cherchons à optimiser les performances globales du système robotisé en prenant en compte à la fois les aspects de contrôle robotique et les exigences d'ultra-fiabilité du service 5G. En intégrant ces deux éléments, nous visons à fournir une connectivité fiable et rapide entre les robots et la station de base, tout en assurant une coordination efficace des robots pour une planification de trajectoire optimisée et une exécution harmonieuse des tâches dans l'entrepôt automatisé.

Notre objectif de recherche est illustré dans la figure 2.1.

2.2 Méthodologie

Nous proposons trois méthodologies **M1**, **M2** et **M3** pour répondre respectivement à la question de recherche **RQ1**, **RQ2** et **RQ3** ainsi qu'à la objectif spécifique **O1**, **O2** et **O3**. Les trois méthodologies sont définies comme suit :

2.2.1 Méthodologie M1

La méthodologie **M1** répond à la question de recherche **RQ1** et à l'objectif spécifique **O1**. Dans cette méthodologie, nous proposons un algorithme de clustering CoMP basé sur la théorie des jeux combiné à la méthode d'optimisation de la politique proximale pour obtenir une solution stationnaire pour le problème de conception de formation de faisceaux de communication à faible latence ultra-fiable (URLLC) pour contrôler les robots mobiles dans un entrepôt automatisé. La méthodologie **M1** peut être résumée comme suit :

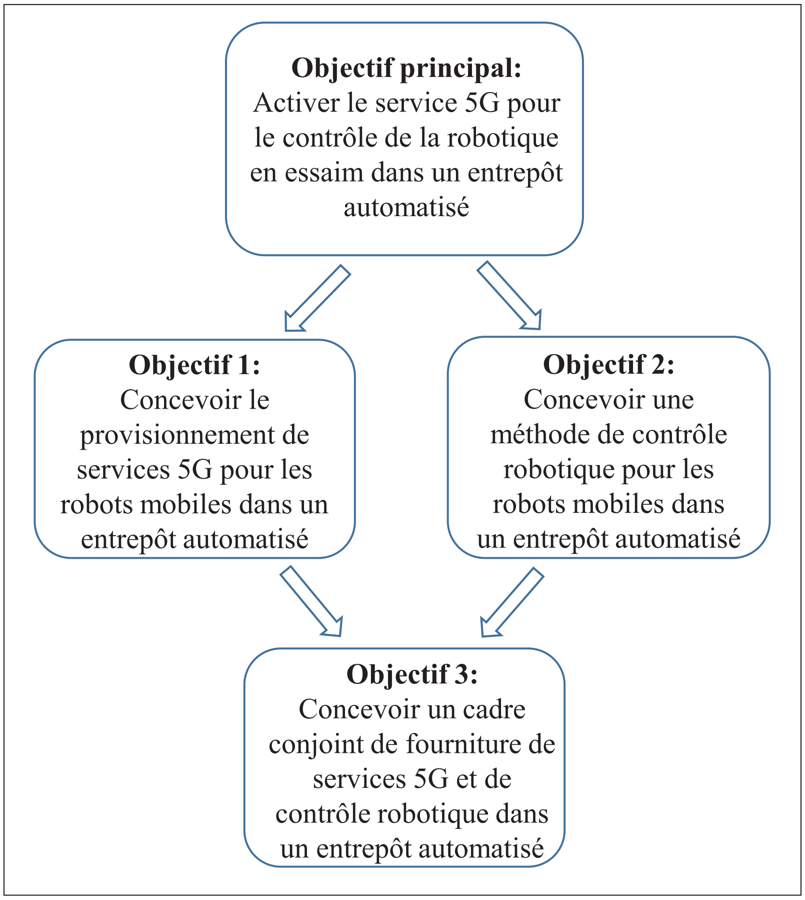

Objectif principal:
Activer le service 5G pour le contrôle de la robotique en essaim dans un entrepôt automatisé
Objectif 1:
Concevoir le provisionnement de services 5G pour les robots mobiles dans un entrepôt automatisé
Objectif 2:
Concevoir une méthode de contrôle robotique pour les robots mobiles dans un entrepôt automatisé
Objectif 3:
Concevoir un cadre conjoint de fourniture de services 5G et de contrôle robotique dans un entrepôt automatisé

Figure 2.1 Le schéma objectif

1. Nous formulons le problème variable dans le temps de la conception conjointe du clustering CoMP et de la formation de faisceaux pour la transmission URLLC 5G dans les applications d'automatisation industrielle.
2. Nous proposons un algorithme multi-agent basé sur l'optimisation de la politique proximale (PPO) pour obtenir une politique optimale de la conception de formation de faisceaux pour la transmission des gNBs.
3. Nous proposons ensuite un algorithme de clustering CoMP à faible complexité basé sur la théorie des jeux qui utilise les actions de l'algorithme multi-agent basé sur PPO pour obtenir un équilibre de Nash du jeu de clustering CoMP formulé parmi les AGV. À son tour, l'équilibre de Nash du jeu de clustering CoMP sera utilisé comme état du système pour former les agents de l'algorithme basé sur PPO.

4. Nous effectuons des simulations intensives pour démontrer l'efficacité de notre cadre proposé.

2.2.2 Méthodologie M2

La méthodologie **M2** répond à la question de recherche **RQ2** et à l'objectif spécifique **O2**. Dans cette méthodologie, nous proposons une méthode basée sur l'apprentissage fédéré pour mettre en œuvre l'algorithme d'apprentissage par renforcement décentralisé (DRL) qui permet d'obtenir une politique globale de planification optimale des tâches du système d'entrepôt distribué. Les agents DRL sont les planificateurs de tâches implémentés dans les postes de travail des entrepôts répartis géographiquement. La méthodologie **M2** se résume comme suit :

1. Nous formulons un problème d'ordonnancement des tâches pour un entrepôt automatisé avec un système robotique autonome hétérogène comme un problème d'optimisation du contrôle des files d'attente qui considère la nature stochastique du flux de tâches et l'hétérogénéité des robots mobiles autonomes. L'objectif est de minimiser la longueur de la file d'attente des tâches en attente de traitement dans chaque entrepôt.
2. Nous proposons un nouvel algorithme d'ordonnancement des tâches qui utilise une méthode d'optimisation de la politique proximale (PPO) pour trouver une politique d'ordonnancement des tâches optimale du problème formulé.
3. Nous proposons un algorithme Proximal Weighted Federated Learning (PWFL) pour implémenter un algorithme PPO décentralisé qui améliore les performances des agents PPO distribués géographiquement.
4. Nous effectuons des simulations intensives pour démontrer les performances du cadre proposé qui peut réduire efficacement le temps de service moyen des robots, le temps d'attente moyen et la longueur moyenne de la file d'attente des tâches par rapport aux méthodes existantes.

2.2.3 Méthodologie M3

La méthodologie **M3** répond à la question de recherche **RQ3** et à l'objectif spécifique **O3**. Dans cette méthodologie, nous proposons un cadre pour le contrôle de la robotique en essaim dans un entrepôt automatisé en gérant conjointement les ressources sans fil, la planification des tâches et le contrôle de mouvement tout en satisfaisant les contraintes d'ultra-fiabilité de la communication entre la station de base et les robots et la contrainte d'évitement de collision de le mouvement des robots. La méthodologie **M3** se résume comme suit :

1. Nous proposons un nouveau modèle de contrôle d'essaims robotiques dans un entrepôt combinant le modèle cinétique des robots et le modèle de communication URLLC du réseau 5G.
2. En combinant le modèle de mouvement des robots et le modèle de transmission du contrôleur central, nous formulons un problème d'optimisation pour maximiser l'efficacité énergétique moyenne à long terme du réseau 5G soumis à la contrainte énergétique limitée de l'essaim de robots et de l'ultra -contrainte de fiabilité de la transmission.
3. Nous utilisons une méthode d'apprentissage par renforcement profond pour obtenir la solution optimale pour le problème non convexe formulé, nous proposons un algorithme de gradient de politique déterministe profond multi-agent (MADDPG) basé sur une méthode de gradient de politique acteur-critique. Nous transformons les états du réseau en une image 3D en tant qu'entrée d'un réseau de neurones convolutifs qui est implémenté en tant que réseaux d'acteurs et de critiques dans notre modèle proposé. A chaque tranche de temps, une politique de contrôle optimal stationnaire est définie en tenant compte de l'état de l'environnement actuel dans lequel l'essaim de robots peut fonctionner sous les contraintes d'ultra-fiabilité de la communication entre le contrôleur central et l'essaim de robots.
4. Nous évaluons les performances de l'algorithme proposé avec des simulations poussées. Nous comparons la méthode proposée avec quatre lignes de base, à savoir la méthode du gradient de politique déterministe profond (DDPG), la limite optimale, la puissance de transmission maximale et l'allocation de bande passante fixe. Les résultats numériques

montrent que notre algorithme proposé surpasse les trois lignes de base tout en se rapprochant des performances de la ligne de base liée optimale.

2.3 Motivation de la Méthodologie

Les approches d'apprentissage en profondeur, telles que le deep reinforcement learning (DRL), offrent une alternative prometteuse pour le contrôle robotique dans des systèmes complexes et à grande échelle. Contrairement aux approches d'optimisation traditionnelles, le DRL permet à un modèle de réseau de prendre des décisions en temps réel en utilisant des données d'entrée sensorielles.

Dans le contexte du contrôle robotique, le DRL peut apprendre des politiques de contrôle directement à partir des entrées sensorielles, sans dépendre de modèles explicites du système. Cela permet au robot d'apprendre à réagir aux changements dans son environnement et d'adapter son comportement en conséquence. De plus, le DRL est capable de gérer des espaces d'actions continus, qui sont courants dans le contrôle robotique réel.

L'utilisation de l'apprentissage en profondeur dans le contrôle robotique présente plusieurs avantages. Tout d'abord, il permet une adaptation et une généralisation flexibles aux différents contextes et situations. Les modèles de réseau peuvent être entraînés sur une grande variété de données d'entrée, ce qui leur permet de s'adapter à différentes configurations de tâches et d'environnements. De plus, le DRL peut exploiter des architectures de réseau profond complexes pour capturer des caractéristiques et des relations plus fines dans les données d'entrée, ce qui peut conduire à des politiques de contrôle plus précises et efficaces.

Cependant, il est important de noter que l'utilisation du DRL dans le contrôle robotique n'est pas sans défis. L'apprentissage en profondeur nécessite souvent des ensembles de données volumineux et des ressources de calcul significatives, ce qui peut limiter son utilisation dans des environnements à contraintes énergétiques ou de calcul. De plus, l'entraînement d'un modèle de DRL peut être un processus long et complexe, nécessitant une exploration et une optimisation minutieuses des paramètres du réseau.

Malgré ces défis, le DRL continue de montrer des résultats prometteurs dans le contrôle robotique et ouvre de nouvelles perspectives pour la conception de politiques de contrôle adaptatives et efficaces dans des systèmes complexes à grande échelle. La recherche continue dans ce domaine contribuera à améliorer la performance, l'efficacité et la fiabilité des systèmes de contrôle robotique.

2.4 Contribution

La plupart des études mentionnés ci-dessus se concentrent sur des problématiques spécifiques liées aux communications Jayaweera *et al.* (2020); Ren, Pan, Deng, Elkashlan & Nallanathan (2019a); Pan, Ren, Deng, Elkashlan & Nallanathan (2019); Ren, Pan, Deng, Elkashlan & Nallanathan (2020a) ou à la planification des trajectoires et des tâches Chen, Li & Liu (2019); Willms & Yang (2006); Liu *et al.* (2020); Švancara, Vlk, Stern, Atzmon & Barták (2019); Li *et al.* (2020a); Han & Yu (2020); Hönig, Kiesel, Tinka, Durham & Ayanian (2019); Sartoretti *et al.* (2019); Wang, Liu, Li & Prorok (2020a); Li, Lin, Liu & Prorok (2020b); Rivière, Hönig, Yue & Chung (2020). Cependant, il est important de noter que ces deux aspects de la robotique sont intrinsèquement liés et ne doivent pas être abordés de manière isolée.

Dans le contexte du réseau 5G, l'absence d'une prise en compte simultanée de ces deux perspectives peut conduire à des solutions peu performantes et inefficaces. Ce problème devient critique dans la nouvelle ère des entrepôts intelligents, caractérisée par des exigences strictes en matière de communication URLLC, d'efficacité énergétique et de gestion d'entrepôt à grande échelle. Par conséquent, dans cet thèse, nous proposons un cadre novateur visant à garantir à la fois l'efficacité énergétique et l'efficacité spectrale dans le contrôle de la robotique en essaim, en gérant de manière conjointe les ressources sans fil, la planification des tâches et le contrôle des mouvements, tout en respectant les contraintes d'ultra-fiabilité et d'évitement des collisions. À notre connaissance, il s'agit de la première étude à aborder ces aspects de manière conjointe dans le domaine du contrôle de la robotique en essaim.

Dans le cadre de cette thèse, nous proposons un algorithme de contrôle robotique innovant qui combine les fonctionnalités de la robotique en nuage et de la robotique en essaim. Notre objectif principal est de superviser un groupe de robots ou un essaim de robots afin de réaliser des tâches spécifiques au sein d'un entrepôt automatisé. Pour ce faire, nous mettons en place une interaction entre les agents DRL multi-agents et les robots via une station de base 5G, qui peut être mise en œuvre soit au niveau Edge, soit dans le Cloud. L'intégration de la technologie en nuage et de la robotique en essaim ouvre ainsi la voie au développement de systèmes multi-robotiques avancés, qui se distinguent par leur efficacité énergétique accrue, leurs performances en temps réel optimisées et leur rentabilité globale.

L'évolution du commerce électronique, associée à une demande sans précédent d'achats en ligne à l'ère de la pandémie de COVID, engendre un flux incessant de commandes pour les détaillants en ligne Berthene (2022). Dans ce contexte, l'utilisation d'un système robotique avancé et hétérogène pour gérer plusieurs commandes clients simultanées crée un environnement complexe et stochastique, nécessitant ainsi un nouveau type de système de planification et de gestion. Les méthodes de planification des tâches mentionnées précédemment ne parviennent pas à capturer l'hétérogénéité présente dans les systèmes multi-entrepôts équipés de différents types de robots. Par conséquent, il est insuffisant et inefficace de déployer une politique de planification des tâches identique sur des entrepôts géographiquement dispersés. À notre connaissance, cette étude constitue la première tentative d'exploiter les techniques d'apprentissage fédéré avancées pour gérer l'hétérogénéité des agents DRL impliqués dans la planification de tâches robotiques distribuées.

2.5 Environnement expérimental

Dans cette thèse, pour effectuer la simulation et évaluer nos méthodes proposées, nous utilisons un ordinateur de bureau avec une configuration matérielle : Intel(R) Core(TM) i7-4790 CPU 3.60GHz et 16GB RAM. De plus, notre simulation est implémentée par Python version 3.7 comme langage de programmation et le puissant framework d'apprentissage automatique open source PyTorch version 1.2.0 principalement développé par le laboratoire AI Research de

Facebook. Pour implémenter des réseaux de neurones dans l'algorithme PPO et l'algorithme DDPG, nous utilisons le PyTorch.

CHAPITRE 3

CONVERGING GAME THEORY AND REINFORCEMENT LEARNING FOR INDUSTRIAL INTERNET-OF-THINGS

Manh Tai Ho[1] , Kim-Khoa Nguyen[1] , Mohamed Cheriet[1]

[1] École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article publié dans la revue IEEE « Transactions on Network and Service Management » le 29 août 2022. Identifiant d'objet digital : 10.1109/TNSM.2022.3202168

3.1 Résumé

Le réseau sans fil de cinquième génération (5G) fournit des connexions à haut débit, à très faible latence et à haute fiabilité qui peuvent répondre aux exigences de l'Internet industriel des objets (IIoT) dans l'automatisation des usines, en particulier pour le contrôle du mouvement des robots. Dans cet article, nous abordons la de services 5G dans un scénario d'entrepôt automatisé, où la robotique en essaim est contrôlée par un contrôleur industriel qui fournit des instructions de routage et de travail sur le réseau 5G. En tirant parti du multipoint coordonné (CoMP), nous formulons un problème de conception de formation de faisceaux de communication ultra-fiable à faible latence (URLLC) de clustering CoMP variable dans le temps pour contrôler les robots qui se déplacent dans l'entrepôt automatisé pour le stockage des marchandises avec les pistes de référence prévues . Les approches d'optimisation itératives traditionnelles ne sont pas pratiques dans un environnement sans fil aussi dynamique en raison du temps de calcul élevé. Nous proposons un algorithme de clustering CoMP de la théorie des jeux combiné à la méthode Proximal Policy Optimization pour obtenir une solution stationnaire proche de celle de l'algorithme de recherche exhaustive considéré comme la solution optimale globale.

Mots-clés : réseau 5G, IdO industriel, URLLC

3.2 Abstract

The fifth-generation (5G) wireless network provides high-rate, ultra-low latency, and high-reliability connections that can meet the Industrial Internet of Things (IIoT) requirements in factory automation, especially for robot motion control. In this paper, we address 5G service provisioning in an automated warehouse scenario, where swarm robotics is controlled by an industrial controller that provides routing and job instructions over the 5G network. Leveraging the coordinated multipoint (CoMP), we formulate a time-varying joint CoMP clustering and 5G ultra-reliable low-latency communication (URLLC) beamforming design problem to control the robots that move around the automated warehouse for goods storage with the planned reference tracks. Traditional iterative optimization approaches are impractical in such a dynamic wireless environment due to high computational time. We propose a game-theoretic CoMP clustering algorithm combined with the Proximal Policy Optimization method to obtain a stationary solution closed to that of the exhaustive search algorithm considered as the global optimal solution.

Keywords : 5G network, industrial IoT, URLLC

3.3 Introduction

The "Fourth Industrial Revolution" is considered the automation revolution thanks to the innovations of 5G wireless communications, automation technologies, and artificial intelligence. Ultra-reliable and low-latency communication (URLLC) service provided by the 5G wireless network is able to fulfill the stringent requirement of factory automation, e.g. 10^{-9} packet loss probability and 99.9999% availability in motion control and mobile robot use cases 3GPP (2018). However, guaranteeing extremely high reliability is challenging in such a dynamic environment of an automated warehouse with high mobility automated guided vehicles (AGV). Coordinated Multi-Point (CoMP) communication technique Marsch & Fettweis (2011) that leverages spatial diversity is promising to achieve URLLC by sending duplicate data streams over diverse paths Khoshnevisan *et al.* (2019). In the automated warehouse scenario, CoMP can combine the signals from multiple radio base stations (gNBs) so that highly dependable communications can be achieved to the moving objects, i.e., AGVs with the physical obstructions, e.g. warehouse racks and shelves.

Along with the advantages that CoMP can bring to the wireless network, providing CoMP-enabled URLLC wireless communication in the industrial Internet of Things (IIoT) is especially challenging due to highly dynamic radio frequency variations from moving objects (such as AGVs) Khoshnevisan *et al.* (2019). Therefore, designing a joint CoMP clustering and beamforming for transmission between gNBs and AGVs that satisfies the URLLC constraints becomes more difficult and significantly different from that in conventional communication systems.

3.3.1 Prior Works

During the past few years, plenty of works try to coordinate CoMP transmission to improve the URLLC service in the 5G network via spatial diversity. In Nasir *et al.* (2020), the authors develop path-following algorithms, which generate a sequence of improved feasible points to solve the problem of resource allocation and beamforming design in the short blocklength regime for URLLC. In Yang *et al.* (2021a), the authors formulate the CoMP-enabled RAN slicing problem for multicast enhanced mobile broadband (eMBB) and bursty URLLC service multiplexing as a multi-timescale optimization problem with a goal of maximizing eMBB and URLLC slice utilities, subject to total system bandwidth and transmit power constraints. In Khan & Jacob (2020b), the authors propose a novel packet delivery mechanism, queuing strategy, and time-frequency resource allocation for CoMP-enabled URLLC in C-RAN architecture. In Yang *et al.* (2020b), the authors investigate the CoMP-enabled RAN slicing for bursty URLLC and eMBB service provision by deriving the minimum upper bound of network bandwidth orchestrated for URLLC traffic transmission to guarantee the URLLC packet blocking probability. In Khan & Jacob (2020a), the authors propose a heuristic resource allocation algorithm for CoMP-enabled URLLC with short packet communication by maximizing the availability of the CoMP. In Yang *et al.* (2021b), the authors propose to use an alternating direction method of multipliers (ADMM) for solving the resource optimization problem of the CoMP-enabled RAN slicing for massive Internet of things (mIoT) and bursty URLLC service multiplexing.

To overcome the shortcomings of traditional optimization theory, recent works have proposed to use of deep reinforcement learning (DRL) to address important aspects of CoMP communication such as clustering and beamforming design. In Al-Eryani *et al.* (2020), a hybrid DRL model combining a deep deterministic policy gradient (DDPG) and a deep double Q-network (DDQN) model is proposed to cluster the access points and optimize the beamforming vectors to maximize the sum rate. The authors in Mismar *et al.* (2019) propose a deep Q-network (DQN)-based algorithm for jointly optimizing beamforming, power control, and interference coordination for voice bearers and data bearers in sub-6 GHz and millimeter-wave in 5G wireless network. In Ge *et al.* (2020) the authors propose a distributed dynamic downlink-beamforming coordination algorithm based on the DQN method to improve the system capacity of this multi-cell multi-input single-output (MISO) interference channel. In Wang *et al.* (2020b), a multi-agent RL-based method is proposed for solving the problem of user-centric transmission/reception point (TRP)-grouping and user-association in joint transmission aided coordinated multipoint (CoMP) technique.

The power of game theory in solving many engineering problems has been proven. Therefore, combining reinforcement learning and game theory has recently attracted the attention of scholars Shi *et al.* (2020); Chen *et al.* (2020). the authors Shi *et al.* (2020) propose a combination of the mean-field game (MFG) and DRL in which a DRL agent learns with the guidance of the Nash equilibrium solved by the MFG. The trust region policy optimization (TRPO) is applied to obtain the optimal solution to the problem modeled by MFG in Chen *et al.* (2020). Different from the existing works considering the combination of DRL and game theory, we propose a distributed framework in which the players of the game (i.e., AGVs) use the actions of the agents of the DRL (i.e., gNBs) to obtain a Nash equilibrium. In turn, the output of the game, i.e., the Nash equilibrium, is used as a network state to train the agents of the DRL model.

3.3.2 Motivation and Contribution

Most existing works which employ traditional iterative optimization approaches are unable to handle the time-varying dynamic environment with high mobility of the network entities which is the case in this paper. Traditional approaches can guarantee convergence to a locally optimal solution at the cost of complexity and computation time, which is not compatible with mission-critical applications. To the best of our knowledge, this is the first work that combines DRL and game theory to solve the high complexity problem of joint beamforming design and CoMP clustering in IIoT. In this paper, we propose a distributed low complexity game-theoretic CoMP clustering algorithm combined with the Proximal Policy Optimization (PPO) method to obtain an optimal solution for beamforming design for URLLC transmission between the gNBs and the AGVs in a highly dynamic environment of an automated warehouse application presented in Fig. 3.1. The main contributions of this paper can be summarized as follows :

- We formulate the time-varying problem of joint CoMP clustering and beamforming design for 5G URLLC transmission in industrial automation applications. The wireless channel condition is highly dynamic due to the high mobility of the AGVs in an automated warehouse scenario. Therefore, the traditional optimization approach is unable to handle the formulated problem in such a dynamic environment.
- We propose a multi-agent Proximal Policy Optimization (PPO) based algorithm to obtain an optimal policy of the beamforming design for the transmission of the gNBs.
- We then propose a low complexity game-theoretic CoMP clustering algorithm that uses the actions of the multi-agent PPO-based algorithm to obtain a Nash equilibrium of the formulated CoMP clustering game among AGVs. In turn, the Nash equilibrium of the CoMP clustering game will be used as a system state to train the agents of the PPO-based algorithm.
- The intensive simulation results demonstrate the effectiveness of our proposed framework in handling the interference caused by the increasing number of AGVs in the network.

The rest of the paper is organized as follows : Section 3.4 presents the system model and problem formulation. Section 3.5 presents an approach for user-centric CoMP clustering and the problem transformation. Section 3.6 introduces the Proximal Policy Optimization algorithm followed

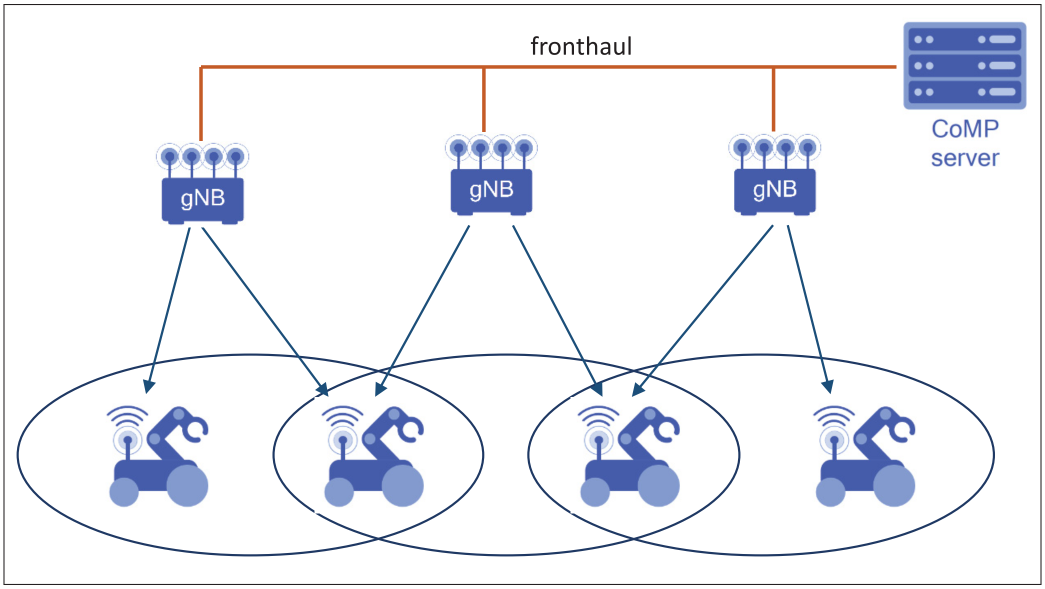


Figure 3.1 CoMP in factory automation

by Section 3.7 presents a game theoretic approach for CoMP clustering. Section 3.8 presents simulation results. Finally, Section 3.9 concludes the paper.

3.4 System Model and Problem Formulation

We consider an automated warehouse IIoT network with a set of B radio base stations (gNodeBs or gNBs) denoted as $\mathcal{B}$, each gNB with M-antennas, and a set of K single-antenna AGVs denoted as $\mathcal{K}$. The AGVs move around the warehouse for goods storage with planned reference tracks (Fig. 3.2). Each AGV traces its planned reference track. Each AGV can be served by a set of $B_k[t] < B$ gNBs at time t. The set $\mathcal{B}_k \subset \mathcal{B}$ consisting of B_k gNBs is the CoMP cluster of AGV k, represents the minimum number of gNBs which can provide 5G communications with the required reliability to AGV k. Note that, these CoMP clusters can be overlapped in which a gNB can be in different clusters that serve different AGVs.

Moreover, we denote $\mathcal{K}_j \subset \mathcal{K}$ as the set of AGVs that are served by gNB j. All gNBs are connected to a single CoMP server over optical fiber fronthaul links. The CoMP enables the distributed gNBs to collaborate and simultaneously serve all AGVs within the warehouse area. We assume all the gNBs are deployed on the ceiling of the warehouse. Let $\mathbf{q}_{\mathrm{gNB},j} = [x_{\mathrm{gNB},j}, y_{\mathrm{gNB},j}]$ denotes the coordinate of the gNB j and $z_{\mathrm{gNB},j}$ is the height of the gNB j.

The reference track is defined for each AGV k at each time step t as $X_k[t] = (\mathbf{q}_k[t], \theta_k[t])$ where $\mathbf{q}_k[t] = [x_k[t], y_k[t]]$ represents the spatial coordinates, and $\theta_k[t]$ is the orientation of the AGV. The control input $u_k[t] = \{v_k[t], \omega_k[t]\}$ sent from the controller implemented in the CoMP server to the k-th AGV consists of an intended translational velocity $v_k[t]$ and rotational velocity $\omega_k[t]$ at each time instant t. The AGV kinematic can be expressed as follows :

$$X_k[t+1] = X_k[t] + \Delta T \Theta_k[t] u_k[t], \tag{3.1}$$

where ΔT is the time slot duration and $\Theta_k[t]$ is given by :

$$\Theta_k[t] = \begin{bmatrix} \cos\theta_k[t] & 0 \\ \sin\theta_k[t] & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.2}$$

The distance between the gNB j and AGV k at time instant t is

$$d_{k,j}[t] = \sqrt{\left\|\mathbf{q}_{\mathrm{gNB},j} - \mathbf{q}_k[t]\right\|^2 + z_{\mathrm{gNB},j}^2} \tag{3.3}$$

The real-time position of the AGVs can be tracked by 5G positioning techniques such as Downlink-Time Difference Of Arrival (DL-TDOA), Downlink-Angle Of Departure (DL-AoD), Uplink-Relative Time Of Arrival (UL-RTOA), Uplink-Angle of Arrival (UL-AoA), etc Qualcomm (2021).

3.4.1 Communication Model

In reality, the channel state information (CSI) can be estimated by the CoMP through training the pilot sequences. Since the moving distance of an AGV in each time slot is substantially much

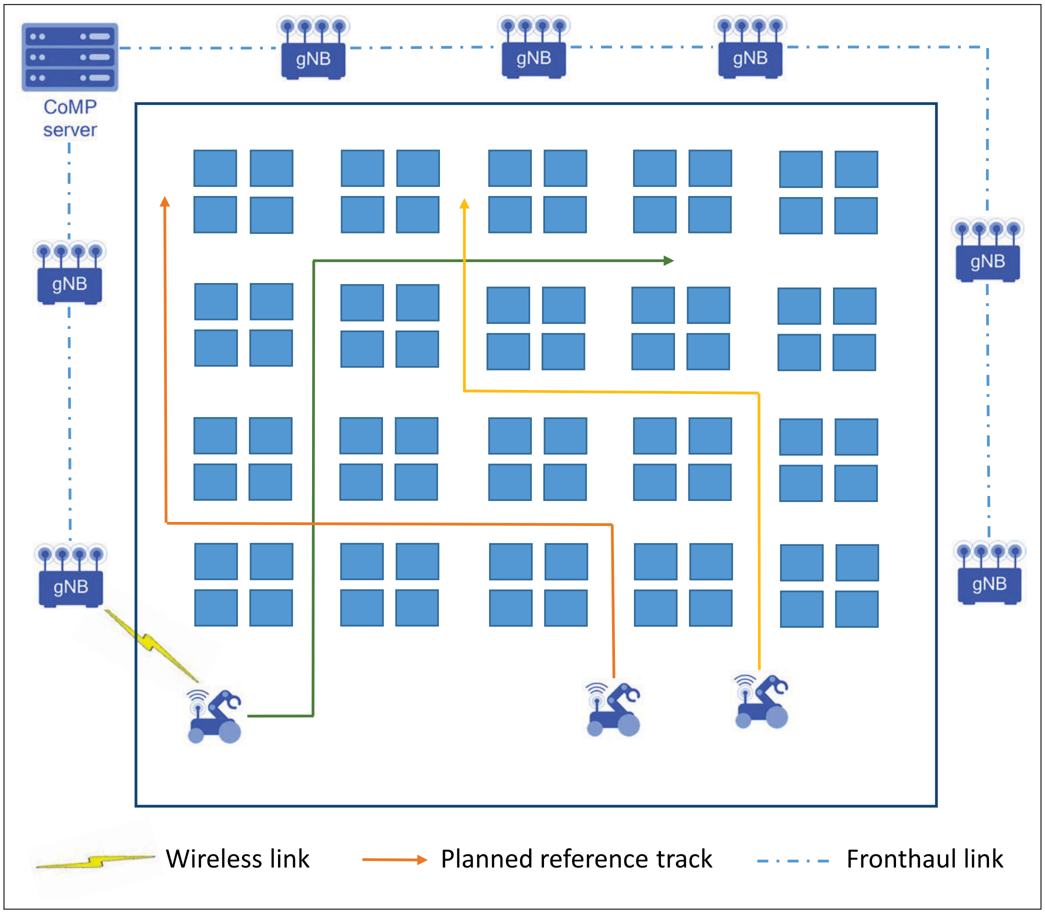


Figure 3.2 System model

smaller than the communication coverage of a gNB, we assume that CSI remains constant (fixed) within a slot but can vary across different time slots. Denote $\mathbf{w}_{k,j}$ as the transmit beamformer for the AGV k from the gNB j. Let s_k denote the complex data symbol for the AGV k and $\mathbb{E}\left[|\mathrm{s}_k|^2\right] = 1$, and $\sigma_k \sim \mathcal{CN}(0, \sigma_0^2)$ is the additive white Gaussian noise (AWGN) at the AGV k. The received signal y_k at AGV k can be expressed as [1]

$$\mathrm{y}_k = \underbrace{\sum_{j=1}^{B_k} \mathbf{h}_{k,j}^H \mathbf{w}_{k,j} \mathrm{s}_k}_{\text{Desired signal}} + \underbrace{\sum_{k' \neq k}^{K} \sum_{j=1}^{B_{k'}} \mathbf{h}_{k,j}^H \mathbf{w}_{k',j} \mathrm{s}_{k'}}_{\text{Interference}} + \sigma_k, \tag{3.4}$$

[1] $\mathbf{x}^H$ is denoted the conjugate transpose operator.

where $\mathbf{h}_{k,j} \in \mathbb{C}^{M\times 1}$ denotes the time-varying channel from the gNB j to the AGV k, and $\mathbf{h}_{k,j} = \sqrt{g_{k,j}}\tilde{\mathbf{h}}_{k,j}$ where $g_{k,j}$ accounts for the distance-based large-scale fading including path-loss component and shadow fading, and $\tilde{\mathbf{h}}_{k,j}$ is the small-scale fading vector associated with the channels between the gNB j and the AGV k. The large-scale fading channel gain $g_{k,j}$ between the gNB j and the AGV k can be expressed as :

$$g_{k,j} = \left(\frac{c}{4\pi f_c}\right)^2 \left(\frac{d_{k,j}}{d_0}\right)^{-\alpha_g}, \tag{3.5}$$

where f_c is the carrier frequency, c is the speed of light, $d_{k,j}$ is the distance between the gNB j and the AGV k, d_0 is a far field reference distance, and α_g is the path-loss exponent ($\alpha_g \in [2,6]$). We assume the small-scale fading from the gNB and the AGV follows the Nakagami-m fading model Simon & Alouini (1998). The probability density function of random variable $\tilde{h}^{(l)}_{k,j} \in \tilde{\mathbf{h}}_{k,j}$, the small-scale fading channel gain between the l-th antenna of eNB j and AGV k, can be expressed as Simon & Alouini (1998) :

$$f(z,m) = \frac{2m^m}{\Gamma(m)\Omega^m} z^{2m-1} \exp\left(-\frac{m}{\Omega}z^2\right), \tag{3.6}$$

where m is the fading parameter, $\Omega = \mathbb{E}\left[|\tilde{h}^{(l)}_{k,j}|^2\right]$, and $\Gamma(.)$ is the Gamma function. We assume that the CoMP server has knowledge of the instantaneous channel vectors $\{\mathbf{h}_{k,j}, \forall k \in \mathcal{K}, \forall j \in \mathcal{B}\}$.

The signal-to-interference-plus-noise ratio (SINR) and the Shannon achievable rate at the AGV k when using CoMP are given by Marsch & Fettweis (2011) :

$$\gamma_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) = \frac{\left|\sum_{j=1}^{B_k} \mathbf{h}^H_{k,j}\mathbf{w}_{k,j}\right|^2}{\sum_{k'\neq k}^{K}\left|\sum_{j=1}^{B_{k'}} \mathbf{h}^H_{k,j}\mathbf{w}_{k',j}\right|^2 + \sigma_k^2}, \tag{3.7}$$

$$\tilde{R}_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) = \log_2\left(1 + \gamma_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j})\right). \tag{3.8}$$

The maximum transmission rate to transmit D_k bits over n_k complex symbols in finite blocklength regime can be accurately approximated as Polyanskiy, Poor & Verdú (2010) :

$$R_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) = \tilde{R}_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) - \sqrt{\frac{V}{n_k}} \frac{Q^{-1}(\epsilon_k)}{\ln(2)} \geq \frac{D_k}{n_k}, \tag{3.9}$$

where ϵ_k is the decoding error probability, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du$ and Q^{-1} is the inverse of Q.

The achievable decoding error probability of the AGV k in terms of γ_k and n_k can be expressed as follows :

$$\epsilon_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) \leq Q\left(\frac{\ln(2)\left(\tilde{R}_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) - \frac{D_k}{n_k}\right)}{\sqrt{\frac{V}{n_k}}}\right). \tag{3.10}$$

where $V = 1 - \frac{1}{(1+\gamma_k)^2}$ is the channel dispersion. We assume that the packet size D_k and complex symbol n_k are the same for all gNBs in the set $\mathcal{B}_k$ corresponding to the AGV k.

From (3.10), it can be seen that, when the SINR $\gamma_k > 5$, the channel dispersion V can be accurately approximated by one $V \approx 1$, then the achievable decoding error probability can be rewritten as :

$$\epsilon_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) \leq Q\left(\ln(2)\sqrt{n_k}\left(\tilde{R}_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) - \frac{D_k}{n_k}\right)\right). \tag{3.11}$$

In low SINR regime (i.e., $\gamma_k < 5$), equation (3.11) can be considered the upper bound of the decoding error probability.

According to (3.10) and (3.7), the mathematical expression of the required beamformers $\{\mathbf{w}_{k,j}\}$ from the gNBs to AGV k that satisfies the decoding error probability ϵ_k requirements can be written as

$$\gamma_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) \geq \gamma_k^{th}(\epsilon_k), \tag{3.12}$$

where the SINR threshold γ_k^{th} is defined as follows :

$$\gamma_k^{th}(\epsilon_k) = \exp\left(\frac{D_k \ln(2)}{n_k} + \sqrt{\frac{V}{n_k}} Q^{-1}(\epsilon_k)\right) - 1. \tag{3.13}$$

3.4.2 Problem Formulation

At each time instant, depending on the real-time position $\{X_k[t]\}$ of the AGVs, the CoMP server dynamically performs the CoMP clustering by assigning the set $\mathcal{B}_k$ gNBs from the available B gNBs to each AGV. Then the corresponding optimal beamforming vectors are computed so that the SINR γ_k of the AGV k meets the ultra-reliability requirement. We consider the joint problem of CoMP clustering and beamforming design with the objective of sum-rate maximization for all AGVs subject to the URLLC constraint. Specifically, the joint problem in time slot t can be formulated as follows :

P1A :

$$\max_{\{\mathcal{B}_k\},\{\mathbf{w}_{k,j}\}} \sum_{k\in\mathcal{K}} R_k(\mathbf{w}_{k,j}[t], \mathbf{h}_{k,j}[t]) \tag{3.14a}$$

$$\text{subject to : } \gamma_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) \geq \gamma_k^{th}(\epsilon_k),\ \forall k \in \mathcal{K}, \tag{3.14b}$$

$$\sum_{k\in\mathcal{K}_j} \left\|\mathbf{w}_{k,j}[t]\right\|^2 \leq P_j,\ \forall j \in \mathcal{B}, \tag{3.14c}$$

$$\mathcal{B}_k[t] \subset \mathcal{B},\ \forall k \in \mathcal{K}. \tag{3.14d}$$

Constraint (3.14b) guarantees the reliability communication of AGV k, whereas (3.14c) sets a constraint on the total transmit power of gNB j. It can be seen that the problem **P1A** in (3.14) is non-convex combinatorial due to the non-convex objective function (3.14a), the URLLC constraint (3.14b), and the combinatorial constraint (3.14d).

We consider a second objective of max-min rate fairness for all AGVs subject to the URLLC constraint as follows :

P1B :

$$\max_{\{\mathcal{B}_k\},\{\mathbf{w}_{k,j}\}} \quad \min_{k\in\mathcal{K}} R_k(\mathbf{w}_{k,j}[t],\mathbf{h}_{k,j}[t]) \tag{3.15a}$$

$$\text{subject to :}\ \gamma_k(\mathbf{w}_{k,j},\mathbf{h}_{k,j}) \geq \gamma_k^{th}(\epsilon_k),\ \forall k\in\mathcal{K}, \tag{3.15b}$$

$$\sum_{k\in\mathcal{K}_j} \left\|\mathbf{w}_{k,j}[t]\right\|^2 \leq P_j,\ \forall j\in\mathcal{B}, \tag{3.15c}$$

$$\mathcal{B}_k[t]\subset\mathcal{B},\ \forall k\in\mathcal{K}. \tag{3.15d}$$

The max-min rate fairness in **P1B** improves the performance of the worst AGVs at the cost of total URLLC rate degradation, while the sum-rate maximization in **P1A** optimizes the total URLLC rate. Although max-min rate optimization can guarantee some fairness for users, it may not maximize the rate for all users. In practice, sum-rate maximization can be applied to the use-cases that require a high data rate for all users, while max-min rate optimization is rather applicable in the applications where users do not have minimum rate requirements. In other words, the max-min rate scheme is appropriate to reduce congestion in heavy traffic applications. For example, the AGVs have to perform compute-intensive tasks while suffering heavy traffic which can cause congestion in the network. In general, these two objective functions define the trade-offs between the network throughput and fairness. Therefore, they can be selected by the administrator depending on the use-case applications.

3.5 User-Centric CoMP Clustering

3.5.1 Problem Transformation

The beamformer variable of all gNBs that transmit to AGV k in cluster $\mathcal{B}_k$ $\mathbf{W}_k = \{\mathbf{w}_{k,j}, j\in\mathcal{B}_k\}$ is a matrix of $[M\times|\mathcal{B}_k|]$ continuous complex variables. Therefore, it is challenging to design joint clustering and beamforming solutions for all AGVs because these solutions consist of multiple matrices of continuous complex variables. In this paper, we propose using the codebook technique Mismar *et al.* (2019); Ge *et al.* (2020); Wang *et al.* (2009) so that the DRL agents can learn the transmit power and beam direction from a codebook instead of learning all the beamformer matrices for all AGVs.

The beamformer vector $\mathbf{w}_{k,j}$ from gNB j to AGV k can be decomposed into two separate parts as follows Ge *et al.* (2020) :

$$\mathbf{w}_{k,j}[t] = \sqrt{p_{k,j}[t]}\bar{\mathbf{w}}_{k,j}[t], \tag{3.16}$$

where $p_{k,j}[t] = \left\|\mathbf{w}_{k,j}[t]\right\|^2$ denotes the transmit power of gNB j to AGV k at time slot t that satisfies constraint (3.14c), and $\bar{\mathbf{w}}_{k,j}[t]$ represents the beam direction of the transmit beamformer $\mathbf{w}_{k,j}[t]$. The beam direction vector $\bar{\mathbf{w}}_{k,j}[t]$ represents the degree of angles of the transmit beams with values in the range of $[0, 2\pi)$.

We consider a codebook $C = [\mathbf{c}_q] \in \mathbb{C}^{M\times Q_{\text{code}}}$ composed of Q_{code} code vector $\mathbf{c}_q \in \mathbb{C}^{M\times 1}$. Each column of C is a code that specifies a beam direction. The element of the codebook matrix is designed as follows Ge *et al.* (2020)

$$c_{m,q} = \frac{1}{\sqrt{M}} \exp\left(i\frac{2\pi}{\Phi}\left\lfloor\frac{m\text{mod}(q+\frac{Q_{\text{code}}}{2}, Q_{\text{code}})}{Q_{\text{code}}/\Phi}\right\rfloor\right), \tag{3.17}$$

where $c_{m,q}$ refers to the phase shift of the nth antenna element in the qth code, Φ denotes the number of available phase values for each antenna element, and $\lfloor . \rfloor$ and mod$(.)$ represent the floor and modulo operations, respectively.

The problem **P1A** in (3.14) can be rewritten as follows :

P2A :

$$\max_{\{\mathcal{B}_k\},\{p_{k,j}\},\{\bar{\mathbf{w}}_{k,j}\}} \sum_{k\in\mathcal{K}} R_k(\mathbf{w}_{k,j}[t], \mathbf{h}_{k,j}[t]) \tag{3.18a}$$

$$\text{subject to : } \gamma_k(\mathbf{w}_{k,j}, \mathbf{h}_{k,j}) \geq \gamma_k^{th}(\epsilon_k), \ \forall k \in \mathcal{K}, \tag{3.18b}$$

$$\sum_{k\in\mathcal{K}_j} p_{k,j}[t] \leq P_j, \forall j \in \mathcal{B}, \tag{3.18c}$$

$$\mathcal{B}_k[t] \subset \mathcal{B}, \forall k \in \mathcal{K}, \tag{3.18d}$$

$$\bar{\mathbf{w}}_{k,j}[t] \in \mathcal{C}, \forall k \in \mathcal{K}, \forall j \in \mathcal{B}_k[t], \tag{3.18e}$$

The problem **P2A** is still difficult to solve because this kind of problem is NP-hard. To obtain the solution of problem **P2A**, we first decompose problem **P2A** into two subproblems, i.e., the CoMP clustering subproblem and beamforming design subproblem. In the following subsection, we propose a user-centric CoMP clustering algorithm to obtain the solution for the CoMP clustering subproblem.

3.5.2 User-Centric CoMP Clustering Algorithm

We design a user-centric clustering algorithm where each AGV k is served by a cluster of $\mathcal{B}_k$ gNBs. The cluster is defined on the reference signal's received power (RSRP) from gNBs. Adding gNBs to an existing cluster will increase the capacity of the cluster at the cost of additional complexity and signaling overhead. Therefore, it is important to balance CoMP efficiency and complexity. To minimize the signaling overhead in the fronthaul and CoMP server, we want to minimize the cluster size i.e., the number of coordinated gNBs per each cluster, while satisfying the SINR and URLLC constraints of each AGV. We determine a maximum number of gNBs in a cluster, i.e., $B_k \leq B_{\max}$ to balance the complexity against the CoMP efficiency trade-off.

Algorithme 3.1 User-centric CoMP Clustering

```
1  **Input** : $\{\mathbf{w}_{k,j}\}$, $\{\mathbf{h}_{k,j}\}$ $\forall k \in \mathcal{K}$ and $\forall j \in \mathcal{B}$, $B_{\max}$;
2  Initialize $\mathcal{B}_k = \emptyset\ \forall k \in \mathcal{K}$, $\rho_{\max}$;
3  **for** *AGV* $k \in \mathcal{K}$ **do**
4      **for** *gNB* $j \in \mathcal{B}$ **do**
5          Generate the sorted list $\mathcal{J}_{k,j+}$ and $\mathcal{J}_{k,j-}$ for each pair $(k, j)$;
6          Compute the downlink SIR-protection level $\rho_{k,j}$ for each pair $(k, j)$ ;
7      **end for**
8      Sort the list $\{\rho_{k,j}\}$ in the descend order.;
9      **for** $j \in \mathcal{B}$ **do**
10         $\mathcal{B}_k \longleftarrow j$ if $\rho_{k,j} \leq \rho_{\max}$ and $B_k \leq B_{\max}$;
11     **end for**
12 **end for**
13 **Output** : $\{\mathcal{B}_k\}$ $\forall k \in \mathcal{K}$;
```

Given the power allocation and codebook selection, the downlink SIR-protection level between the gNB j and AGV k is calculated as follows Wang *et al.* (2020b)

$$\rho_{k,j} = \frac{\mathbb{E}\left[\sum_{i\in\mathcal{J}_{k,j+}} \left|\mathbf{w}_{k,i}\mathbf{h}_{k,i}^{H}\right|^2\right]}{\mathbb{E}\left[\left|\mathbf{w}_{k,j}\mathbf{h}_{k,j}^{H}\right|^2 + \sum_{l\in\mathcal{J}_{k,j-}} \left|\mathbf{w}_{k,l}\mathbf{h}_{k,l}^{H}\right|^2\right]}, \tag{3.19}$$

where $\mathcal{J}_{k,j+}$ and $\mathcal{J}_{k,j-}$ denote the set of gNBs having higher and lower values of $\mathbb{E}\left[\left|\mathbf{w}_{k,i}\mathbf{h}_{k,i}^{H}\right|^2\right]$, $i \neq j$ than the gNB j, respectively.

The user-centric CoMP clustering algorithm is presented in Algorithm 3.1. Algorithm 3.1 works as a greedy algorithm in which each AGV greedily searches all gNBs that satisfy the criteria (line 8-11).

3.6 Proximal Policy Optimization

In this section, we propose a DRL-based framework to obtain the solution for the beamforming design subproblem by modeling the beamforming design subproblem as a Markov Decision Process (MDP).

3.6.1 System State, Action, and Reward Design

Consider an infinite-horizon discounted MDP, defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathbf{Pr}, r, \gamma)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathbf{Pr} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition probability $r : \mathcal{S} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. The MDP of beamforming design can be characterized as follows :

1. The network state at time t is defined by the tuple $\mathcal{S} = (\{\mathcal{B}_k[t-1]\}_{k\in\mathcal{K}}, \{\mathbf{h}_{k,j}[t]\}_{k\in\mathcal{K}, j\in\mathcal{B}})$ in which :
   - $\mathcal{B}_k[t-1], \forall k \in \mathcal{K}$ is the CoMP clustering at time $t-1$.
   - $\mathbf{h}_{k,j}[t], \forall k \in \mathcal{K}, \forall j \in \mathcal{B}$ is the channel state of all AGVs.
2. The action space at time t is the variables of problem **P2A** and defined by the tuple $\mathcal{A} = (\{p_{k,j}\}_{k\in\mathcal{K}, j\in\mathcal{B}}, \{\bar{\mathbf{w}}_{k,j}\}_{k\in\mathcal{K}, j\in\mathcal{B}})$. At each time t, the agent makes a decision of transmit power level and the corresponding codeword from gBNs to AGVs.
3. The reward is the signal from the environment to tell the agent how good the action is when it is executed. In the formulated problem, we aim to maximize the URLLC rate of the AGVs at each time slot. Naturally, the agent should take the transmission rates as its reward. However, if each gNB tries to maximize its transmission rate by increasing its transmit power, it can generate significant interference to other AGVs served by the other gNBs, hence, cannot satisfy the URLLC constraint (3.18b) for all AGVs. Therefore, the reward for each AGV at time t is designed as follows :

$$r_k[t] = \begin{cases} \kappa_1 R_k[t] - \kappa_2 \sum_{j\in\mathcal{B}_k} P_{k,j}, \\ \kappa_3, \text{ if (3.18b) does not satisfy} \end{cases} \tag{3.20}$$

   where the second term is the penalty for the gNBs for the exceeding transmit power in the cluster $\mathcal{B}_k$ and κ_1 and κ_2 are tunable scale coefficients. Moreover, κ_3 is a negative reward to penalize the agent if the URLLC constraint in (3.18b) cannot be satisfied. Reward for each agent j (i.e., gNB j) is formulated as follows :

a. Maximize sum-rate : Reward for each agent j (i.e., gNB j) is the sum of reward of all the AGVs served by gNB j :

$$r_j[t] = \sum_{k \in \mathcal{K}_j} r_k[t]. \tag{3.21}$$

b. Maximize minimum rate : Reward for each agent j (i.e., gNB j) is the minimum reward of the AGVs served by gNB j :

$$r_j[t] = \min_{k \in \mathcal{K}_j} r_k[t]. \tag{3.22}$$

3.6.2 Proximal Policy Optimization

Proximal policy optimization (PPO) Schulman *et al.* (2017) is a model-free, online, on-policy, policy gradient reinforcement learning method. This algorithm is a type of policy gradient training that alternates between sampling data through environmental interaction and optimizing a clipped surrogate objective function using stochastic gradient descent (SGD).

PPO alternatively constructs an unconstrained surrogate objective function to remove the incentive for large policy updates. PPO updates policies by taking multiple steps of (usually minibatch) SGD to maximize the objective

$$\theta^{(n+1)} = \arg\max_{\theta} \mathop{\mathrm{E}}_{s,a\sim\pi_{\theta_n}} \left[L(s, a, \theta_k, \theta) \right], \tag{3.23}$$

where L is given in (3.24). $\pi_\theta(a|s)$ is new parameterized policy trying to seek the optimal parameter vector θ, and $\pi_{\theta_n}(a|s)$ is the old policy. Here, ϵ is a small hyperparameter presenting how far the new policy is allowed to go from the old policy. The advantage function $A^{\pi_{\theta_n}}(s, a)$ can be calculated by

$$A^{\pi_{\theta_n}}(s, a) = Q^{\pi_{\theta_n}}(s, a) - V^{\pi_{\theta_n}}(s), \tag{3.27}$$

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_n}(a|s)} A^{\pi_{\theta_n}}(s,a),\ \ \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_n}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_n}}(s,a)\right), \tag{3.24}$$

$$\theta^{(n+1)} = \arg\max_\theta \frac{1}{|\mathcal{D}_n|\Delta T} \sum_{\tau\in\mathcal{D}_n} \sum_{t=0}^{\Delta T} \min\left(\frac{\pi_\theta({}_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t,a_t), g(\epsilon, A^{\pi_{\theta_k}}(s_t,a_t))\right); \tag{3.25}$$

$$\boldsymbol{\phi}^{(n+1)} = \arg\min_\phi \frac{1}{|\mathcal{D}_n|\Delta T} \sum_{\tau\in\mathcal{D}_n} \sum_{t=0}^{\Delta T} \left[V_{\boldsymbol{\phi}^{(n)}}(\boldsymbol{s}[t]) - r(\boldsymbol{s}[t], \boldsymbol{a}[t])\right]^2; \tag{3.26}$$

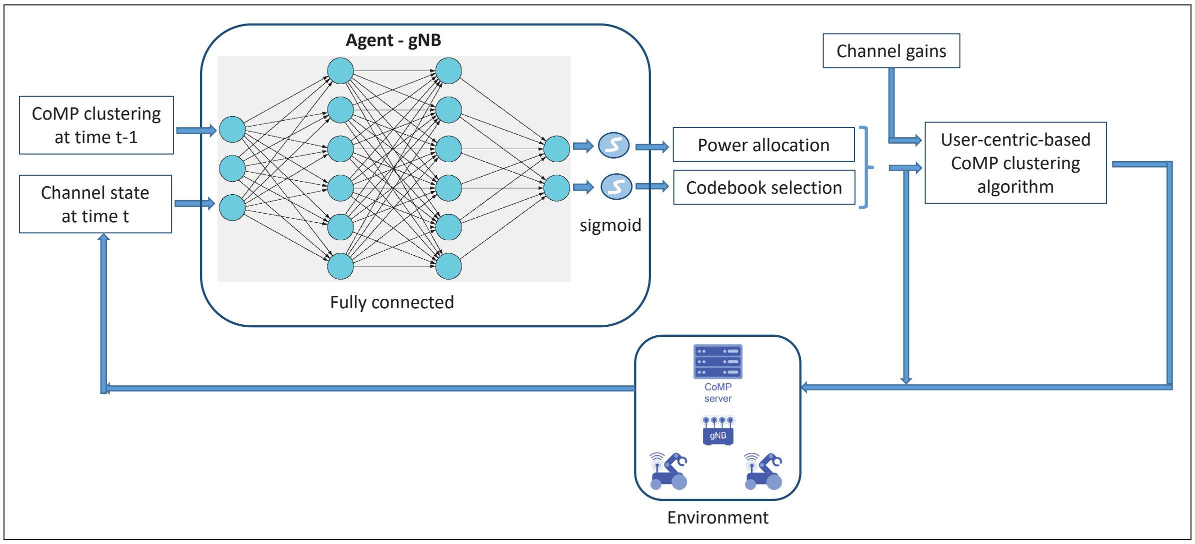


Figure 3.3 Joint CoMP clustering and beamforming design framework

where $Q^{\pi_{\theta_n}}(s,a)$ is the action-value function estimated by samples, and $V^{\pi_{\theta_k}}(s)$ is the approximation of the state-value function.

$$Q^{\pi_{\theta_n}}(s_t,a_t) = \mathbb{E}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right]. \tag{3.28}$$

The PPO algorithm is presented in Algorithm 3.2 and illustrated in Fig. 3.3. Each agent (i.e., gNB) collects a minibatch of transitions by running the current policy to produce the beamforming actions including the transmit power and beam direction to its connected AGVs (line 4). Each agent computes advantage estimates and updates the policy by maximizing the

Algorithme 3.2 PPO-based beamforming design

```
1 Initialize policy parameter $\theta^{(0)}$, initialize value function parameters $\phi^{(0)}$ for each gNB
    agent;
2 for $n = 0, 1, 2, ..$ iterations do
3     for each gNG agent do
4         Collect a minibatch of $D$ transitions $\mathcal{D}_n = \{s_i, a_i, r_i, s_{i+1}\}_{i=0:D-1}$ by running
            policy $\pi_\theta$;
5         Compute advantage estimates $\hat{A}(s_t, a_t)$ based on the current value function
            $V_{\phi^{(n)}}(s_t)$;
6         Update the policy by maximizing the PPO-clip objective in (3.25) where

              $g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1-\epsilon)A & A < 0. \end{cases}$      (3.29)

            Fit value function by regression on MSE in (3.26) ;
7     end for
8 end for
```

PPO-clip objective with the minibatch of transitions (line 5,6). Then, each agent trains the value functions by regression on mean-squared error (MSE) (line 7). The steps are repeated until the agents' policies converge to stationary policies.

3.7 Game Theoretic-Based CoMP Clustering

The user-centric CoMP clustering Algorithm 3.1 in Section 3.5 is based on the downlink SIR-protection criteria which is the maximum average SIR that an AGV can potentially achieve by removing the number of gNBs with lower signal strength from the original cluster Wang *et al.* (2020b). However, this criteria does not consider the interference that affects other AGVs. This means that the current user-centric clustering mechanism is selfish and does not guarantee an equilibrium for all AGVs. The game theoretic-based solution can obtain equilibria for all AGVs.

There is an increasing interest in applying game theory to design self-organized, distributed cooperative clustering Guidolin, Badia & Zorzi (2014); Abdelhakam, Elmesalawy, Mahmoud & Ibrahim (2018). In the game-theoretic approach, a payoff function is introduced to formulate the CoMP gain and cost trade-off for forming CoMP clusters.

3.7.1 Clustering Game Formulation

We consider a CoMP clustering game in which each AGV is a player trying to select a set of serving gNBs to maximize its payoff. We define the action of each player AGV as follow :

$$a_{k,j} = \begin{cases} 1, \text{ if AGV } k \text{ selects gNB } j \\ 0, \text{ otherwise.} \end{cases}$$

The clustering game can be formulated as follows :

Definition 3.1. *The CoMP clustering game is a tuple* $\mathcal{G} = (\mathcal{K}, \{a_{k,j}\}, \{\mathcal{P}_{k,j}\})$ *where*

1. *Player set : set of AGV* $\mathcal{K}$*.*
2. *Strategy : the strategy of each player is defined as decisions on choosing a set of gNBs to be served* $\boldsymbol{A} = \{\boldsymbol{a}_j\}_{j\in\mathcal{B}}$, $\boldsymbol{a}_j = \{a_{k,j}\}_{k\in\mathcal{K},j\in\mathcal{B}}$ *to maximize its payoff function.*
3. *Payoff function : The payoff of player* k *is given by*

$$\mathcal{P}_k(\boldsymbol{A}) = \sum_{j\in\mathcal{B}} \mathcal{P}_{k,j}(\boldsymbol{a}_j), \tag{3.30}$$

$$\mathcal{P}_{k,j}(\boldsymbol{a}_j) = \frac{\left(a_{k,j}\left|\boldsymbol{w}_{k,j}\boldsymbol{h}_{k,j}^H\right|^2\right)^\alpha}{\sum_{l\in\mathcal{K}}\left(a_{l,j}\left|\boldsymbol{w}_{l,j}\boldsymbol{h}_{l,j}^H\right|^2\right)^\alpha} - \xi a_{k,j} \sum_{l\neq k\in\mathcal{K}} \left|\boldsymbol{w}_{k,j}\boldsymbol{h}_{l,j}^H\right|^2, \tag{3.31}$$

where α *and* ξ *are positive. The first term of the payoff function presents the percentage allocated power of gNB* j *to AGV* k*. The second term presents the total interference caused by the transmission from gNB* j *to AGV* k*. The payoff function indicates that the utility and the total interference each AGV incurs will vary inversely according to the increasing number of AGVs connected to the same gNB.*

In the following subsections, we transform the game $\mathcal{G}$ into a mean-field game and analyze the Nash equilibrium.

3.7.2 Mean Field Approximation for CoMP Clustering

When the system becomes large, traditional game-theoretic analysis is computationally inefficient because every single action of every player should be taken into account. A mean-field game is proposed to tackle the dimensionality difficulty of the traditional game by taking the statistical mean-field distribution instead of tracking the action of each player Hanif, Tembine, Assaad & Zeghlache (2015).

Denote the weight by $\omega_{k,j} = |\mathbf{w}_{k,j}\mathbf{h}_{k,j}^H|^2$, we define the mean-field as a weighted α-norm of all the actions as follows :

$$m_j = \left(\frac{1}{K}\sum_{k\in\mathcal{K}} \left(\omega_{k,j}a_{k,j}\right)^{\alpha}\right)^{\frac{1}{\alpha}}, \forall j \in \mathcal{B}. \tag{3.32}$$

The payoff function in (3.31) can be rewritten as follows :

$$\begin{aligned} \mathcal{P}_{k,j}(a_{k,j}, m_{j,-k}) &= \frac{1}{K}\left(\frac{\omega_{k,j}a_{k,j}}{m_j}\right)^{\alpha} - \xi\mathcal{I}_{k,j}a_{k,j}, \\ &= \frac{\left(\omega_{k,j}a_{k,j}\right)^{\alpha}}{(K-1)m_{j,-k}^{\alpha} + \left(\omega_{k,j}a_{k,j}\right)^{\alpha}} - \xi\mathcal{I}_{k,j}a_{k,j}, \end{aligned} \tag{3.33}$$

where $\mathcal{I}_{k,j} = \sum_{l\neq k\in\mathcal{K}} \left|\mathbf{w}_{k,j}\mathbf{h}_{l,j}^H\right|^2$, and

$$\begin{aligned} m_{j,-k}^{\alpha} &= \frac{1}{K-1}\sum_{j\neq k} \left(\omega_{l,j}a_{l,j}\right)^{\alpha} \\ &= \frac{K}{K-1}\left(m_j^{\alpha} - \frac{\left(\omega_{k,j}a_{k,j}\right)^{\alpha}}{K}\right). \end{aligned} \tag{3.34}$$

The payoff function of a player has the following properties :

- The payoff function depends only on the player’s action $a_{k,j}$ and the mean field m_j.
- The payoff is discontinuous when there is no connection to the gNB j, i.e., $\sum_{k\in\mathcal{K}} \left(\omega_{k,j}a_{k,j}\right)^{\alpha} = 0$.

3.7.3 Equilibrium for Clustering Game

In this section, we characterize the mean-field equilibrium of the formulated game.

Definition 3.2. *An action vector* $\boldsymbol{a}_j^{NE} = \{a_{k,j}^{NE}\}_{k\in\mathcal{K}}$ *is said to be a Nash equilibrium if no player can improve its payoff by unilaterally deviating its action from the Nash equilibrium, such that :*

$$\mathcal{P}_{k,j}(a_{k,j}^{NE}, m_{j,-k}) \geq \mathcal{P}_{k,j}(a_{k,j}, m_{j,-k}),\ \ a_{k,j} \in (0,1), \forall k.$$

Theorem 1. *There exists at least one Nash equilibrium for the game* $\mathcal{G}$.

Proof : We consider the case there is at least one AGV connected to a gNB so that the payoff function is smooth, continuous and differential. If there is no AGV in the coverage of an gNB, the game $\mathcal{G}$ simply excludes such gNB out of the strategy of the players, i.e., AGVs.

The first and second derivative with respect to $a_{k,j}$ can be written as follows :

$$\begin{aligned}\frac{\partial \mathcal{P}_{k,j}(a_{k,j}, m_j)}{\partial a_{k,j}} &= \frac{1}{K}\left[\frac{\alpha\omega_{k,j}^{\alpha}a_{k,j}^{\alpha-1}m_j^{\alpha} - \left(\omega_{k,j}a_{k,j}\right)^{\alpha}\left(\frac{\alpha}{K}a_{k,j}^{\alpha-1}\right)}{m_j^{2\alpha}}\right] - \xi\mathcal{I}_{k,j} \\ &= \frac{\alpha\omega_{k,j}^{\alpha}a_{k,j}^{\alpha-1}}{K}\left[\frac{m_j^{\alpha} - \frac{a_{k,j}^{\alpha}}{K}}{m_j^{2\alpha}}\right] - \xi\mathcal{I}_{k,j},\end{aligned} \tag{3.35}$$

$$\begin{aligned}\frac{\partial^2 \mathcal{P}_{k,j}(a_{k,j}, m_j)}{\partial {a_{k,j}}^2} &= \frac{\alpha\omega_{k,j}^{\alpha}}{K}\big(m_j^{\alpha} - \frac{a_{k,j}^{\alpha}}{K}\big) \times \left[\frac{(\alpha-1)a_{k,j}^{\alpha-2}m_j^{2\alpha} - a_{k,j}^{\alpha-1}\frac{2\alpha}{K}a_{k,j}^{\alpha-1}m_j^{\alpha}}{m_j^{4\alpha}}\right] \\ &= \frac{\alpha\omega_{k,j}^{\alpha}}{K}\big(m_j^{\alpha} - \frac{a_{k,j}^{\alpha}}{K}\big)\frac{a_{k,j}^{\alpha-2}}{m_j^{3\alpha}}\left[(\alpha-1)m_j^{\alpha} - \frac{2\alpha}{K}a_{k,j}^{\alpha}\right].\end{aligned} \tag{3.36}$$

For $0 \leq \alpha \leq 1$, the second derivative of the payoff function with respect to $a_{k,j}$ is negative, then the payoff is concave with respect to own-action $a_{k,j}$. Therefore, there exists at least one Nash equilibrium for the game $\mathcal{G}$.

We consider an asymmetric game in which all the players have asymmetric strategies in equilibrium whenever it exists. In other words, each AGV has its own clustering strategy which is different from other AGVs. However, due to the complicated structure of the payoff function, deriving a closed-form asymmetric Nash equilibrium is not trivial. Instead, we propose an iterative form that converges to mean-field equilibrium. As the number of players tends to infinity, the mean-field equilibrium asymptotically converges to Nash equilibrium.

Definition 3.3. *Mean field best response of player k given the actions of other players given by*

$$\boldsymbol{Br}(a_{k,j}, m_j) = \arg\max_{a_{k,j}} \left[\frac{1}{K} \left(\frac{\omega_{k,j} a_{k,j}}{m_j} \right)^{\alpha} - \xi \mathcal{I}_{k,j} a_{k,j} \right]. \tag{3.37}$$

Theorem 2. *The iterative best response updates converge to Nash equilibrium*

$$a_{k,j}(\tau+1) = \lambda(\tau) \boldsymbol{Br}(a_{k,j}(\tau), m_j(\tau)) + (1 - \lambda(\tau)) a_{k,j}(\tau), \tag{3.38}$$

where τ represents the iterations and $\lambda(\tau)$ is a step size and

$$\boldsymbol{Br}(a_{k,j}(\tau), m_j(\tau)) = \left[K \left(m_j^{\alpha}(\tau) - \frac{K \xi \mathcal{I}_{k,j} m_j^{2\alpha}(\tau)}{\alpha \omega_{k,j}^{\alpha} a_{k,j}^{\alpha-1}(\tau)} \right) \right]^{\frac{1}{\alpha}}, \tag{3.39}$$

$$m_j(\tau+1) = \lambda(\tau) \left[\frac{a_{k,j}^{\alpha}(\tau)}{K} + \frac{K \xi \mathcal{I}_{k,j} m_j^{2\alpha}(\tau)}{\alpha \omega_{k,j}^{\alpha} a_{k,j}^{\alpha-1}(\tau)} \right]^{\frac{1}{\alpha}} + (1 - \lambda(\tau)) m_j(\tau). \tag{3.40}$$

Démonstration. Since $\mathbf{Br}(a_{k,j}(\tau), m_j(\tau))$ is obtained by setting the first derivative of the payoff function in (3.35) equals zero, then it is the unique solution.

The iterative best response update (3.38) has the form of Ishikawa (Mann) iteration Ishikawa (1974). It was proven in Ishikawa (1974) that, with a vanishing learning rate, i.e., $\lambda(\tau) > 0$, $\sum_{\tau} \lambda(\tau) = \infty$, and $\sum_{\tau} \lambda^2(\tau) < \infty$, the iterative best response update (3.38) converges strongly to a fixed point which is a unique Nash equilibrium. □

A distributed game-based CoMP clustering is presented in Algorithm 3.3. After receiving the beamforming information from gNBs, each AGV updates its strategy by the iterative best response equation and the approximated mean-field value without knowledge of other AGVs'

Algorithme 3.3 Distributed Game-based CoMP Clustering

```
Initialize $a_{k,j}(0)$ and $m_j(0)$ $\forall k \in \mathcal{K}, j \in \mathcal{B}$;
All gNBs broadcast their beamforming profiles $\mathbf{w}_{k,j}$;
repeat
    Each AGV $k$ updates its strategy $a_{k,j}(\tau)$ according to (3.38) and (3.39);
    Update mean field according to (3.40);
until $|a_{k,j}(\tau + 1) - a_{k,j}(\tau)| \leq \varepsilon$;
```

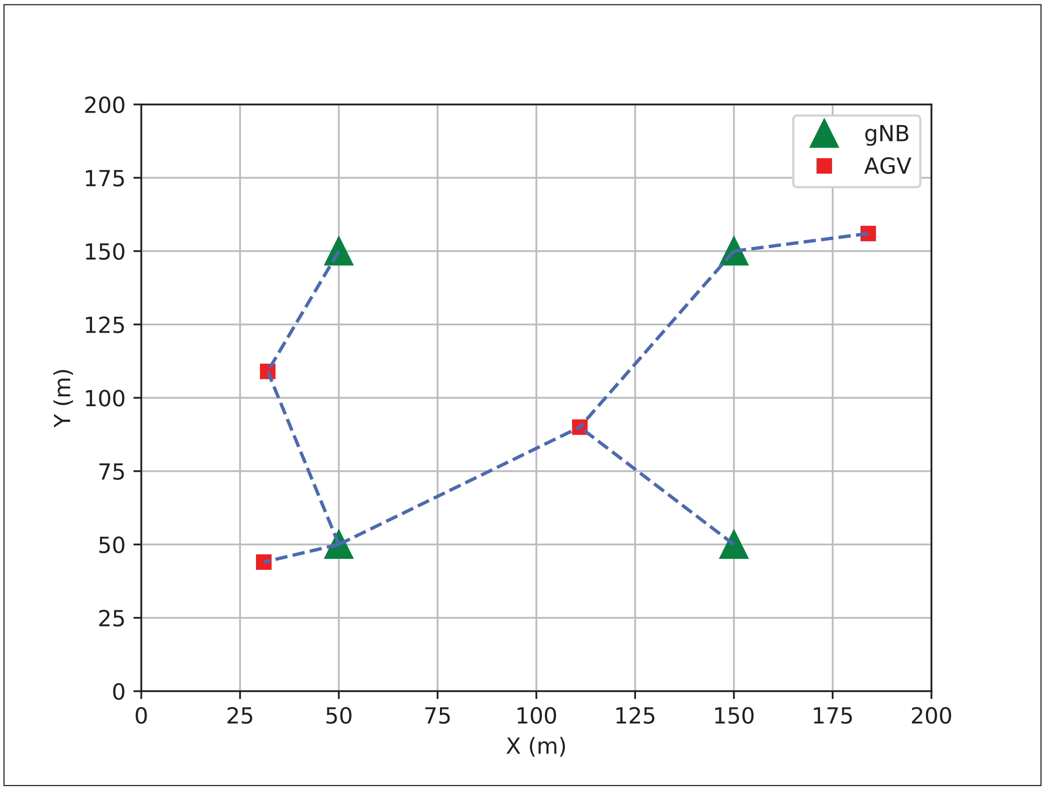


Figure 3.4 CoMP clustering illustration. The dash lines present the association between the gNBs and the AGVs. Each AGV has its own CoMP cluster indicated by the set of associated gNBs

actions. Therefore, this method can reduce the message exchange overhead and complexity of the algorithm.

Tableau 3.1 PPO hyperparameters setting

Parameter	**Value**
Policy network	128, relu, 128, relu, 128, relu, tanh
Value network	128, relu, 128, relu, 128, relu, linear
Step size	$1e-3$
Batch size	20
Discount factor	0.995
Epsilon clip	0.1

3.7.4 Complexity

In practice, PPO usually is implemented in Actor-Critic framework in which the policy network is implemented as an actor and the value function is implemented as a critic network. The computational complexity of the PPO-based algorithm can be calculated based on the complexity related to the training of the actor and critic neural networks. Let L^{actor} and L^{critic} denote the number of fully connected layers of the actor network and critic network, respectively. The computational complexity of the PPO-based algorithm is $O(\sum_{l=0}^{L^{\text{actor}}-1} u_l^{\text{actor}} u_{l+1}^{\text{actor}} + \sum_{l=0}^{L^{\text{critic}}-1} u_l^{\text{critic}} u_{l+1}^{\text{critic}})$ Qiu, Hu, Chen & Zeng (2019), where u_l^{actor} and u_l^{critic} are the unit numbers in the l-th layers of actor network and in the l-th layer of critic network, respectively. Here, u_0^{actor} and u_0^{critic} represent the input sizes of actor network and critic network, respectively. The input size of the actor and critic networks in our model is $u_0^{\text{actor}} = u_0^{\text{critic}} = |\mathcal{B}| + M \times |\mathcal{B}| \times |\mathcal{K}|$ where M is the number of antennas of each gNB, $|\mathcal{K}|$ is the number of AGVs, and $|\mathcal{B}|$ is the number of gNBs. It can be seen that, the computation complexity of the PPO-based algorithm increases according to the network state, i.e., the number of AGVs and the number of gNBs.

The computational complexity of Algorithm 3.3 is a polynomial function of the number of iterations of the iterative best response update (3.38), i.e., $O(2T \times |\mathcal{K}| \times |\mathcal{B}|)$ where T is the number of iterations. It is proved in Theorem 2 that T is finite and in the simulation, we see that the number of iterations T of the best response update (3.38) is around 5.

3.8 Simulation Results

3.8.1 Simulation Setting

We perform extensive simulations to evaluate the performance of our proposed design in terms of the sum URLLC rate in (3.9), i.e., the objective of the optimization problem **P1**. We vary the number of AGVs in the range of [2-20] in a 200×200 meters square automated warehouse. There are 4 gNBs each with 4 antennas so that they can fully cover the area and provide service to the AGVs as depicted in Fig. 3.4. At the beginning of each episode, the central controller generates a uniformly distributed destination for each AGV and the AGV follows the shortest path from its starting point to its destination. The velocity of each AGV follows a Gaussian distribution $\mathcal{N}(5, 2)$ with a mean 5 m/s and a standard deviation of 2. The carrier frequency is 6 GHz with 2 MHz bandwidth. The pathloss exponent is set to 3.76, the noise power spectral density is set to -174 dBm/Hz and the decoding error probability is set to 10^{-9}. The data packet size is 20 bytes and channel blocklength is 512 symbols Ren, Pan, Deng, Elkashlan & Nallanathan (2020b).

To implement the neural networks, we employ the powerful open-source machine learning framework PyTorch version 1.2.0 primarily developed by Facebook's AI Research lab. The programming language is Python 3.7 on a desktop computer with hardware configuration : Intel(R) Core(TM) i7-4790 CPU 3.60GHz and 16GB RAM. The hyperparameters of our proposed PPO Algorithm 3.2 is presented in Table 3.1. However they are not chosen arbitrarily but should be related to the network parameters. For example, we choose the number of AGVs in the range of [2-20]. Therefore, the number of hidden layers is relatively small and is tuned in the range of 1 to 3, and the number of hidden units is from 64 to 128.

We compare our proposed joint CoMP clustering and beamforming design scheme (denoted as 'PPO-Game') with four benchmark schemes as follows :

- 'DDPG-Game' : This baseline is the multi-agent off-policy deep deterministic policy gradient Lillicrap *et al.* (2015); Qiu *et al.* (2019) combined with the distributed game theoretic based

CoMP clustering Algorithm 3.3. We investigate whether on-policy or off-policy gradient method outperforms in a dynamic environment as in a robotic network.

- ‘PPO-Heuristic’ : The user-centric CoMP clustering in the Algorithm 3.1 works as a greedy algorithm where each AGV selfishly searches a set of serving gNBs satisfies downlink protection criteria and without considering interference caused to other AGVs.
- ‘EXHAUST’ : We use the exhaustive search method over the Euclidean space $\mathcal{K} \times \mathcal{B} \times \mathcal{C} \times P$. The ‘EXHAUST’ baseline is considered the optimal solution for the formulated problem.
- ‘RANDOM’ : The CoMP clustering, transmit power, and beam direction (codebook) are randomly selected at each time slot.

The performance of the proposed scheme and the baseline is evaluated through the following metrics :

1. URLLC rate : This metric is the URLLC rate in (3.9).
2. Outage probability : This metric is the percentage of the solutions that do not satisfy the URLLC constraint (3.14b) and (3.18b).
3. Complexity : This metric is the computation complexity of each baseline, and signaling overhead.

Note that, except Fig. 3.7e, in all other Figures, the ‘PPO-Game’ scheme is simulated with the ‘Sum-rate’ objective.

3.8.2 Results Analysis

Fig. 3.4 illustrates the CoMP cluster of each AGV created by our proposed ‘PPO-Game’ scheme. The set of gNBs associated with an AGV forms a CoMP cluster of that AGV. It can be observed that the CoMP cluster of each AGV is different from the others depending upon the channel condition and beamforming of each gNB.

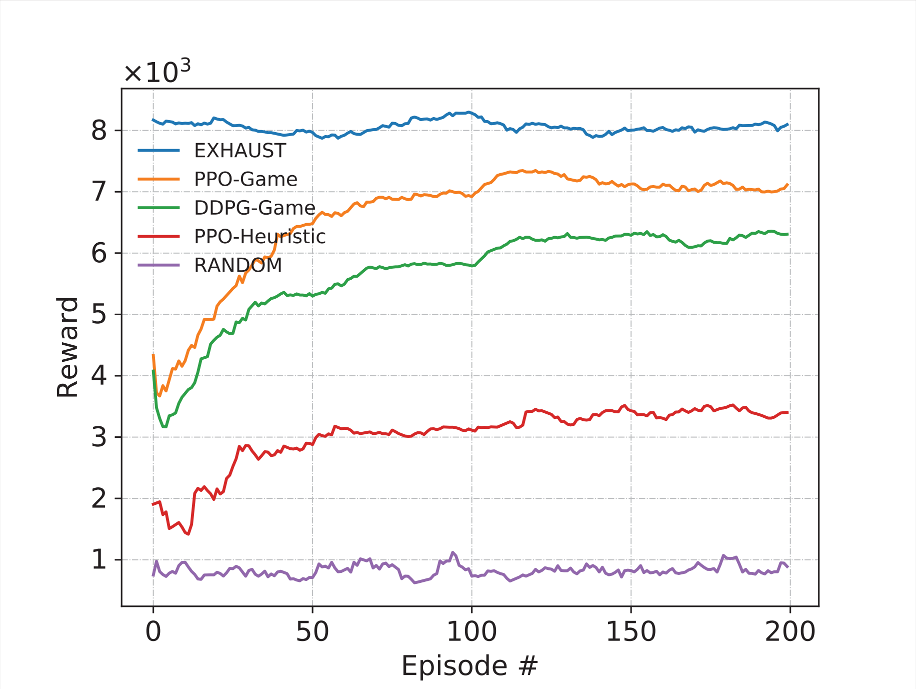


Figure 3.5 Accumulative reward

3.8.2.1 Convergence Performance

Fig. 3.5 shows the convergence of the accumulative reward of our proposed scheme and four benchmark schemes over 200 episodes (each with hundreds of time steps). The 'EXHAUST' scheme achieves the highest reward while the 'RANDOM' experiences the worst performance. Our proposed scheme 'PPO-Game' improves gradually over the episodes and converges to a fairly stable situation in approximately 150 episodes. It can be observed that our proposed scheme 'PPO-Game' significantly outperforms the 'PPO-Heuristic' baseline and reaches a stable reward close to the 'EXHAUST' baseline which is the optimum. Moreover, the 'DDPG-Game' baseline has a similar convergence behavior but converges to a lower value compared to the 'PPO-Game' scheme. In a stable environment, the off-policy DDPG-based algorithm may outperform the on-policy PPO-based algorithm due to the sample efficiency characteristic of the DDPG method. However, in a highly dynamic environment, which is the case in this paper, the DDPG method may cause sudden failures due to the exploration noise, resulting in instabilities during training due to the sensitivity to the model hyperparameters Henderson *et al.* (2018). Whilst the on-policy

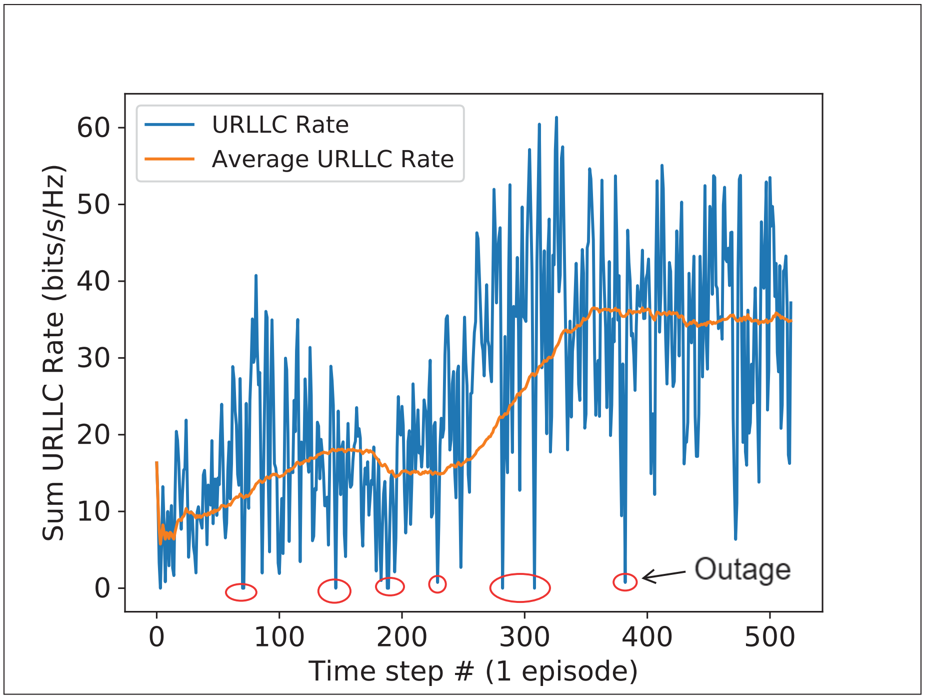


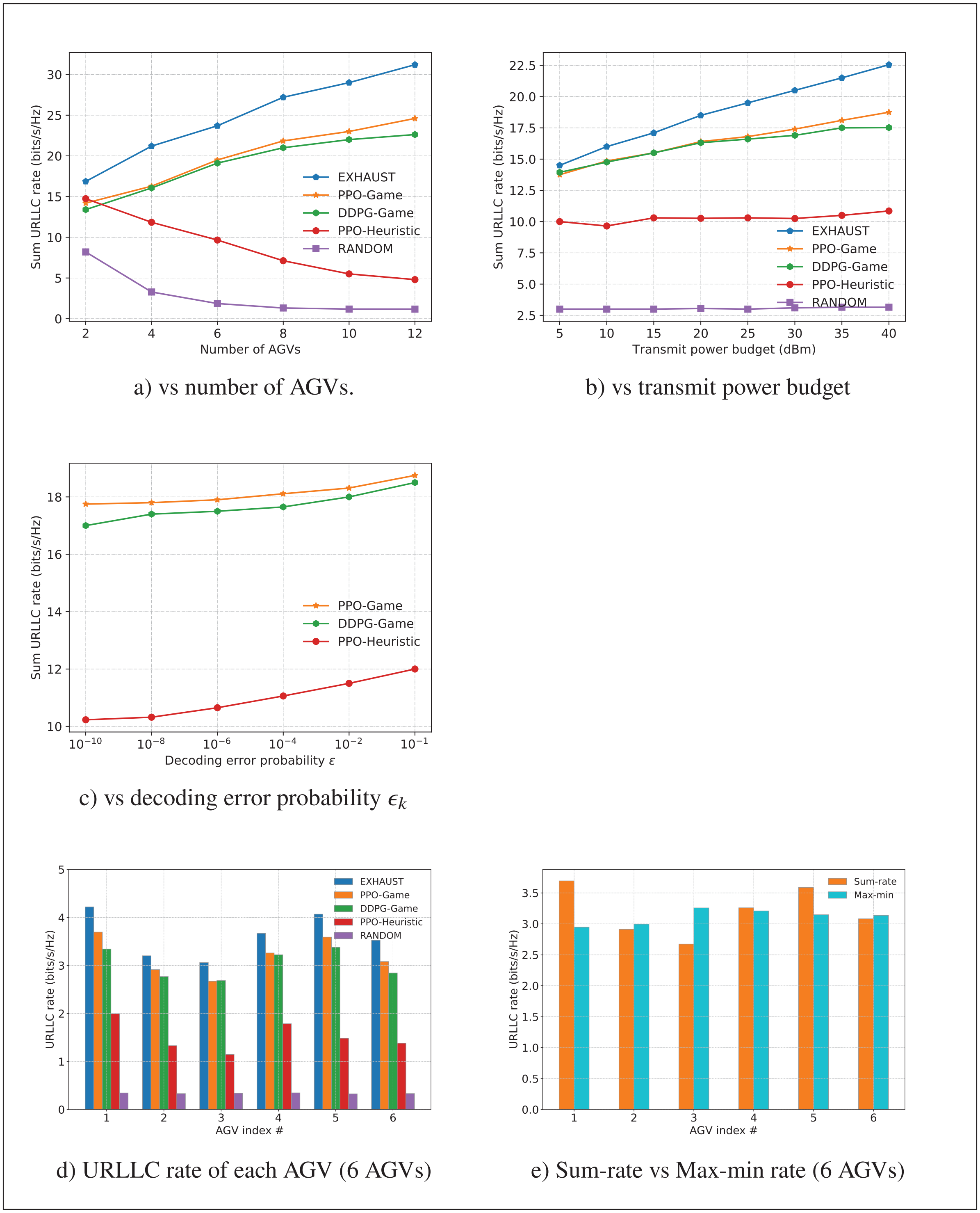
Figure 3.6 URLLC rate performance during one episode with around 500 time steps

PPO method monotonically improves the policy and guarantees the new policy after the gradient step is not too different than before Ho, Nguyen & Cheriet (2021a).

Fig. 3.6 shows the sum URLLC rate within an episode of around 500 time steps. We can observe that there are some time steps at which the outage happens, i.e., the AGVs do not satisfy the URLLC constraint (3.14b) and (3.18b) in problem **P1A** and **P2A**. Such outage happens when the AVGs fail to obtain the Nash equilibrium for the CoMP clustering game, or the PPO agents produce a beamforming profile that does not satisfy the URLLC constraint. For example, when an AGV is at the edge of a gNB's coverage but not in any other gNBs' coverage. The outage performance is analyzed in detail with different scenarios in Fig 3.8.

3.8.2.2 URLLC Rate Performance

Fig. 3.7a depicts the sum URLLC rates of the five schemes versus the number of AGVs. The proposed scheme 'PPO-Game' baseline can handle the interference caused by the overlapping



a) vs number of AGVs.

b) vs transmit power budget

c) vs decoding error probability ϵ_k

d) URLLC rate of each AGV (6 AGVs)

e) Sum-rate vs Max-min rate (6 AGVs)

Figure 3.7 URLLC rate performance

clusters. Therefore, when the number of AGVs increases, the sum URLLC rate of all AGVs increases. A similar increasing trend can be seen with the 'DDPG-Game' baseline. This result implies an effective adaptation of the game-theoretic CoMP clustering according to the increasing number of AGVs.

Moreover, we can see that when the number of AGVs in the network is small, the performance gap between PPO and DDPG methods is relatively small. However, when the number of AGVs becomes larger, the performance gap between these two policy gradient methods is more significant. When the number of AGVs is 12, the 'PPO-Game' achieves a sum URLLC rate of 24.65 bits/s/Hz compared to 31.2 bits/s/Hz of the 'EXHAUST' baseline, in other words, a performance of approximately 78.5% compared to the optimal solution. On the other hand, the 'PPO-Heuristic' baseline experiences a noticeable decrease of sum URLLC rate when the number of AGVs increases. The poor performance of the 'PPO-Heuristic' baseline can be explained by the fact that the interference is not managed in this clustering algorithm, even though both 'PPO-Game' and 'PPO-Heuristic' schemes implement the same PPO-based beamforming algorithm. The user-centric CoMP clustering in the 'PPO-Heuristic' baseline works as a greedy algorithm in which each AGV greedily searches all the possible serving gNBs that satisfy the SIR criteria while ignoring the interference that may cause to the other AGVs. Therefore, the more AGVs in the network, the more interference each AGV incurs, and the poorer network is.

Fig. 3.7b plots the sum URLLC rates versus the transmit power budget. It can be seen that the sum URLLC rates of the 'PPO-Game' scheme, 'DDPG-Game' and 'EXHAUST' baseline increase along with the increase in the transmit power budget whereas the sum URLLC rate of the 'PPO-Heuristic' baseline is nearly constant. This result again confirms the 'PPO-Game' scheme can manage interference better than the 'PPO-Heuristic' baseline. When the transmit power increases the interference also increases, hence, an interference adaptive scheme would be beneficial.

Moreover, it can be observed that when the transmit power budget is small, the performances of 'PPO-Game' and 'DDPG-Game' schemes are almost identical. However, when the transmit power budget increases, the performance gap between ‘PPO-Game’ and ‘DDPG-Game’ schemes increase significantly.

In Fig. 3.7c, we draw the sum URLLC rates of three schemes ‘PPO-Game’, ‘DDPG-Game’ and ‘PPO-Heuristic’ versus the decoding error probability $\epsilon = \epsilon_k, \forall k$. For all considered schemes, the sum URLLC rate is a monotonically increasing function of the decoding error probability. This is because the inverse error function $Q^{-1}(\epsilon)$ is a monotonically decreasing function of ϵ. However, as can be observed, the impact of the decoding error probability on the URLLC rate is minor. As we can see in (3.9), the second term can be interpreted as a penalty on the rate in order to guarantee the decoding error probability in a finite blocklength regime. This penalty is relatively small compared to the Shannon capacity, i.e., the first term in (3.9). Moreover, both ‘PPO-Game’ and ‘DDPG-Game’ schemes significantly outperform the ‘PPO-Heuristic’ scheme, and the observed performance gain is similar to the simulation scenarios with the transmit power budget (3.7b) and the number of AGVs (3.7a).

Fig. 3.7d presents URLLC rate of each AGV (6 AGVs) in five schemes. It can be seen that the ‘EXHAUST’ scheme has the highest URLLC rate for all AGVs compared to all other schemes and the ‘RANDOM’ baseline has the lowest URLLC rate for all AGVs. The ‘PPO-Game’ scheme has a slightly higher URLLC rate than that of the ‘DDPG-Game’ scheme. A significant improvement for all AGVs obtained by the ‘PPO-Game’ and ‘DDPG-Game’ schemes compared to the ‘PPO-Heuristic’ scheme can be explained by the fact that all the AGVs can find the equilibrium in the CoMP clustering game, and the DRL agents can produce beamforming profiles that adapt to the changing of the network state of each AGV.

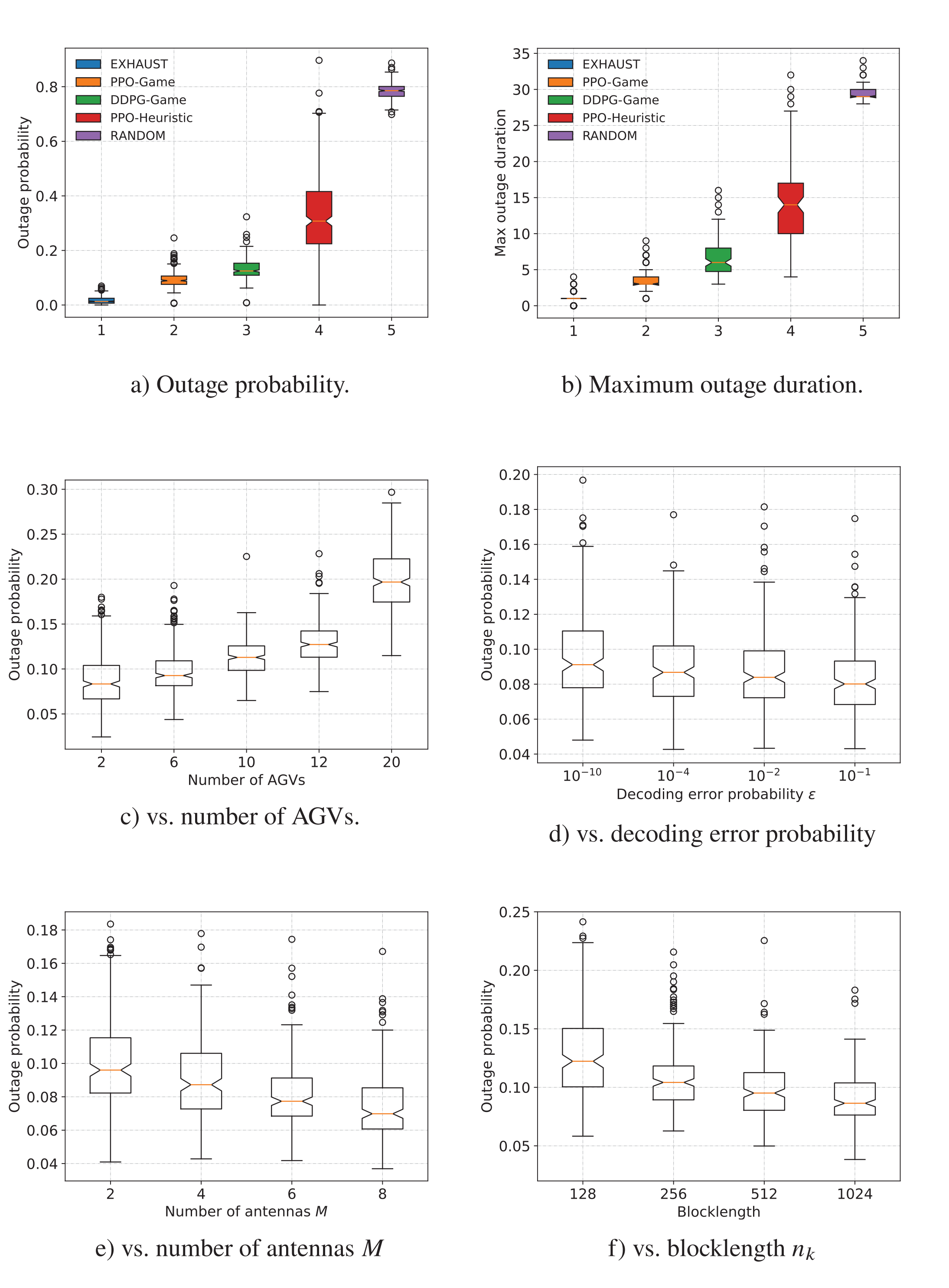
In Fig. 3.7e, we compare the URLLC rate performance of our proposed ‘PPO-Game’ scheme with two different objective functions, i.e., maximize sum-rate (denoted as ‘Sum-rate’) and maximize minimum rate (denoted as ‘Max-min’). It can be seen that with the ‘Max-min’ objective the proposed ‘PPO-Game’ scheme can achieve better fairness compared to the ‘Sum-rate’ objective. However, the ‘Sum-rate’ objective provides a higher total throughput (sum URLLC rate) of all the AGVs than the ‘Max-min’ objective. More specifically, the ‘Max-min’ objective achieves a better URLLC rate for the worst user, i.e., AGV number 3, than the ‘Sum-rate’ objective, but achieves a lower URLLC rate for AGVs number 1 and number 5. Note that, with the equilibrium obtained in the CoMP clustering game our proposed ‘PPO-Game’ scheme can achieve certain fairness compared to the ‘EXHAUST’ scheme as shown in Fig. 3.7d.

3.8.2.3 Outage Probability Performance

Fig 3.8a shows the outage probability of all schemes with 5 AGVs for all episodes. As expected, the ‘EXHAUST’ baseline has the lowest outage probability at around the median value of 2% while the ‘RANDOM’ baseline has the highest outage probability at around 78%. The ‘PPO-Game’ scheme has a lower outage probability than that of the ‘DDPG-Game’ scheme, at around 9% and 13%, respectively. The ‘PPO-Heuristic’ baseline has the median value of outage probability at around 30% but it has the widest range of outage probability value compared to all other schemes, which is from 0% to 72% with some outliers over 90%. In the worst case, the ‘PPO-Heuristic’ baseline can obtain a poor performance as the ‘RANDOM’ baseline. This result once again confirms our proposed scheme ‘PPO-Game’ can obtain a comparable performance compared to the exhaustive search algorithm which can be considered an optimal solution.

Fig. 3.8b compare the maximum continuous outage duration of five schemes with 10 AGVs for all episodes. Similarly to the outage probability shown in Fig 3.8a, the ‘EXHAUST’ scheme has the lowest outage duration at around the median value of 1 time step. In contrast, the ‘RANDOM’ baseline has the highest outage duration at around the median value of 29 time steps. The ‘PPO-Game’ scheme has the outage duration value close to the ‘EXHAUST’ scheme at around 3 time steps, while the outage duration of the ‘DDPG-Game’ and ‘PPO-Heuristic’ schemes are two times and five times higher than that of the ‘PPO-Game’ scheme, respectively. This result again confirms the performance gap between the ‘PPO-Game’ and ‘DDPG-Game’ schemes is more significant when the number of AGVs increases.

We investigate the performance in terms of outage probability of our proposed scheme ‘PPO-Game’ with the variation of the number of AGVs, decoding error probability ϵ, the number of



a) Outage probability.

b) Maximum outage duration.

c) vs. number of AGVs.

d) vs. decoding error probability

e) vs. number of antennas M

f) vs. blocklength n_k

Figure 3.8 Outage probability performance

antennas M, and blocklength n_k, in Fig. 3.8c, Fig. 3.8d, Fig. 3.8e, and Fig. 3.8f, respectively. The outage probability values are collected over 300 running episodes, each episode is with hundreds of time steps.

In Fig. 3.8c, it can be observed that with the fixed wireless resource, i.e., system bandwidth, and a number of serving gNBs in the network, the more number of AGVs the more probability the AGVs incur outage. The outage probability increases gradually with a small number of AGVs but increases dramatically when the number of AGVs is sufficiently large. Therefore, to assure the URLLC can be achieved when a large number of AGVs operates in the network, it is important to guarantee enough wireless resources and a number of serving gNBs.

In Fig. 3.8d, we plot the outage probability of our proposed scheme with the different values of the decoding error probability requirement. As we can see, a tighter reliability requirement and a higher outage probability will be obtained. However, the increase of the outage probability when we decrease the decoding error probability requirement is not significant. For example, when the decoding error probability requirement is 10^{-1}, the outage probability median value is 0.08 or 8% but when the decoding error probability requirement decreases to 10^{-10} the outage probability only increases to around 0.09 or 9%.

In Fig. 3.8e, the more number of antennas in each gNBs, the lower outage probability we can achieve. This is because we can obtain a better SINR value with a higher number of antennas, therefore, a lower outage probability will be incurred. A similar trend can be observed in Fig. 3.8f when we increase the blocklength. A higher blocklength value, a lower outage probability we can obtain. However, we cannot increase more blocklength values to achieve a better outage probability performance since the achievable URLLC rate will be saturated when the blocklength value exceeds 1024.

3.8.2.4 Complexity Performance

In Fig. 3.10, we compare the complexity in terms of computation time in milliseconds of four schemes except for the ‘RANDOM’ baseline. It is obvious the ‘EXHAUST’ baseline has the

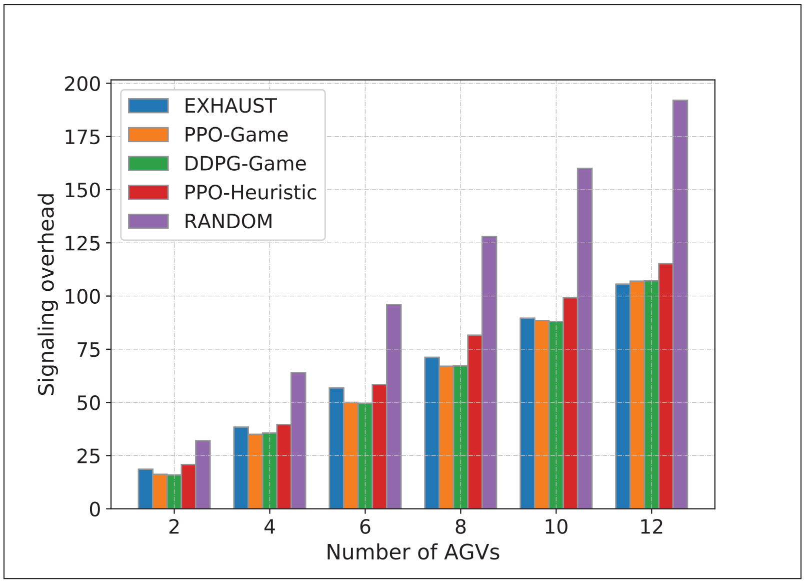


Figure 3.9 Signaling overhead

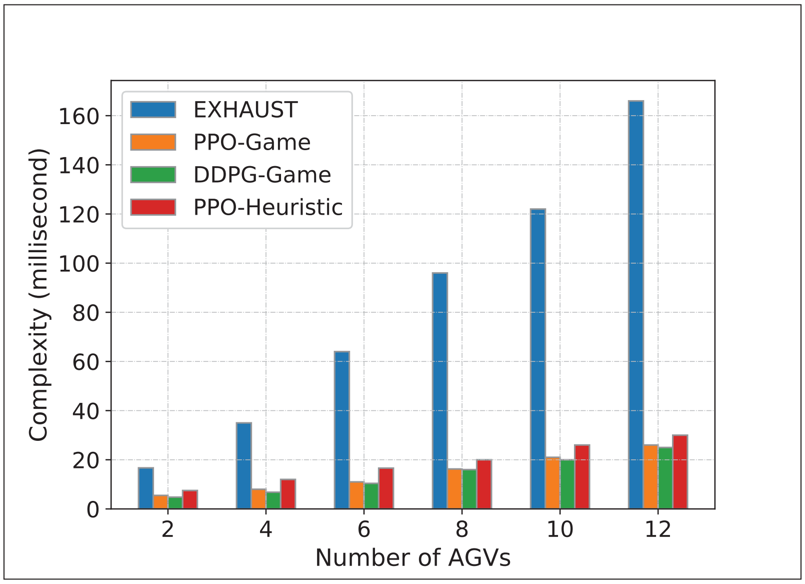


Figure 3.10 Complexity comparison

highest complexity compared to all other schemes. The complexity of the ‘EXHAUST’ baseline increases significantly with the number of AGVs in the network, whereas the complexities of the ‘PPO-Game’, ‘DDPG-Game’, and ‘PPO-Heuristic’ schemes increase slightly. For example, when the number of AGVs is 2, the complexity of the ‘EXHAUST’ baseline is about 3 times higher than that of the ‘PPO-Game’, ‘DDPG-Game’, and ‘PPO-Heuristic’ schemes. However, when the number of AGVs in the network is 12, the complexity of the ‘EXHAUST’ baseline is about 6 times higher than the other schemes. Furthermore, while the ‘PPO-Game’ and ‘DDPG-Game’ schemes have similar complexity, the ‘PPO-Heuristic’ scheme has a higher complexity than that of the ‘PPO-Game’ and ‘DDPG-Game’ schemes. This result can be explained by the fact that the PPO and DDPG algorithms are implemented by the actor-critic method, i.e., using two neural networks to implement the policy network and value network separately. Moreover, as stated in Section III.B and Section V.D, we see that the heuristic user-centric CoMP clustering algorithm (Alg.3.1) has a slightly higher complexity than that of the Game-based CoMP clustering algorithm (Alg.3.3.)

Fig. 3.9 depicts the signaling overhead of five schemes. The signaling overhead is calculated based on the number of connections between the AGVs and gNBs and the 5G-RRC (Radio Resource Control) connection setup procedure, i.e., 8 messages over a 5G-RRC connection 3GPP (2022). In this simulation, we omit the messages sent periodically from AGVs to the gNBs in order to provide information about the channel state. It can be seen that the ‘RANDOM’ baseline generates high signaling overhead, while the ‘PPO-Game’ and ‘DDPG-Game’ schemes result in the lowest signaling overhead. The signaling overhead of the ‘PPO-Heuristic’ scheme increases more rapidly than the ‘PPO-Game’ and ‘DDPG-Game’ schemes. This result confirms the CoMP clustering game is more efficient than the greedy heuristic user-centric CoMP clustering

3.9 Conclusion

This paper has presented the joint CoMP clustering and beamforming problem for URLLC in an automated warehouse IIoT network. By combining a low complexity game-theoretic based CoMP clustering algorithm and the Proximal Policy Optimization method, we proposed an effective interference management framework that is suitable for a dynamic environment and can obtain performance approximated to the optimum and outperforms the user-centric CoMP clustering baseline.

CHAPITRE 4

FEDERATED DEEP REINFORCEMENT LEARNING FOR TASK SCHEDULING IN HETEROGENEOUS AUTONOMOUS ROBOTIC SYSTEM

Manh Tai Ho[1] , Kim-Khoa Nguyen[1] , Mohamed Cheriet[1]

[1] École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article publié dans l'IEEE « Transactions on Automation Science and Engineering » le 14 novembre 2022. Identifiant d'objet digital : 10.1109/TASE.2022.3221352

4.1 Résumé

La robotique autonome joue un rôle central dans la logistique intelligente où les robots peuvent remplacer ou aider les humains dans toutes sortes de tâches, telles que la cueillette, le déplacement et le stockage d'articles. Dans cet article, nous étudions le problème de la planification des tâches dans les entrepôts automatisés avec des systèmes robotiques autonomes hétérogènes (HAR). Nous formulons un problème d'optimisation de contrôle de file d'attente non convexe à long terme pour minimiser la longueur de la file d'attente des tâches à traiter dans l'entrepôt. Les solutions traditionnelles de planification des tâches basées sur des approches d'optimisation sont inefficaces pour gérer la nature stochastique du flux de marchandises/tâches et un grand nombre de robots dans le système en raison de leur coût de calcul. Nous proposons un algorithme de planification de tâches basé sur l'apprentissage par renforcement profond (DRL) qui utilise la méthode d'optimisation de politique proximale (PPO) pour trouver une politique de planification de tâches optimale. En raison de l'hétérogénéité du système, nous proposons un algorithme basé sur l'apprentissage fédéré pondéré proximal pour implémenter un algorithme PPO décentralisé qui améliore les performances des agents PPO distribués qui sont déployés dans les postes de travail des entrepôts géographiquement distribués. Les résultats de la simulation démontrent l'amélioration des performances de notre algorithme proposé par rapport aux méthodes existantes.

4.2 Note aux praticiens

La planification des tâches pour les essaims robotiques dans les entrepôts intelligents est importante pour le commerce électronique. Les solutions de pointe se sont concentrées sur la planification efficace des tâches pour les systèmes robotiques homogènes en utilisant des techniques d'apprentissage automatique mises en œuvre dans les systèmes de gestion d'entrepôt (WMS). Cependant, la planification des tâches pour un système robotique autonome hétérogène (HAR) n'a pas été entièrement étudiée jusqu'à présent. Cet article fournit un algorithme complet de planification des tâches pour les systèmes HAR qui exploite des techniques innovantes d'apprentissage par renforcement profond et d'apprentissage fédéré. L'algorithme proposé peut être déployé dans les entrepôts géographiquement distribués d'une entreprise de commerce électronique et facilement intégré dans le WMS pour contrôler de manière optimale le fonctionnement du système HAR avec des flux stochastiques de marchandises/tâches dans l'entreposage intelligent.

Mots clés : Entreposage intelligent, automatisation industrielle, apprentissage fédéré, apprentissage par renforcement profond, robotique autonome hétérogène

4.3 Abstract

Autonomous robotics play a central role in smart logistics where robots can replace or aid humans in all kinds of tasks, such as items picking, moving, and storing. In this paper, we investigate the problem of task scheduling in automated warehouses with heterogeneous autonomous robotic (HAR) systems. We formulate a long-term non-convex queueing control optimization problem to minimize the queue length of tasks to be processed in the warehouse. Traditional task scheduling solutions based on optimization approaches are inefficient in handling the stochastic nature of the goods/tasks flow and a large number of robots in the system due to their computational cost. We propose a deep reinforcement learning (DRL) based task scheduling algorithm that employs the proximal policy optimization (PPO) method to find an optimal task scheduling policy. Due to the heterogeneity of the system, we propose a proximal weighted federated learning-based

algorithm for implementing a decentralized PPO algorithm that improves the performance of the distributed PPO agents that are deployed in the workstations at the geographically distributed warehouses. The simulation results demonstrate the performance improvement of our proposed algorithm compared to the existing methods.

4.4 Note to Practitioners

Task scheduling for robotic swarms in smart warehouses is substantial for e-commerce. State-of-the-art solutions have focused on efficient task scheduling for homogeneous robotic systems using machine learning techniques implemented in the warehouse management systems (WMS). However, task scheduling for a heterogeneous autonomous robotic (HAR) system has not fully been investigated so far. This article provides a comprehensive task scheduling algorithm for HAR systems that leverages innovative deep reinforcement learning and federated learning techniques. The proposed algorithm can be deployed in the geographically distributed warehouses of an e-commerce company and easily integrated into the WMS to optimally control the operation of the HAR system with stochastic goods/tasks flows in the smart warehousing.

Keywords : Smart warehousing, industrial automation, federated learning, deep reinforcement learning, heterogeneous autonomous robotic

4.5 Introduction

In the smart warehousing, the culmination of warehouse automation and smart logistics, robotics play an indispensable role in attaining efficient automation solutions. Smart logistics includes intelligent organization, planning, control, and execution of goods/items flow in the warehouse. Recent advances in wireless communications and battery technologies prolong the serving time of robotics ; thus, replacing human workers by robotic systems can reduce labor costs, enhance warehouse working efficiency, and improve reliability Wen, He & Zhu (2018). Cloud robotics has emerged as a new technology for path planning and task scheduling in robotic systems Chen *et al.* (2018). It enables robotic systems to be endowed with powerful capability whilst reducing costs through cloud technologies.

In the warehouse management system (WMS), task scheduling is a vital component that is responsible for optimizing the robotic resources utilization, hence, improving the overall productivity of the warehouse. Innovating scheduling is inevitable to cope with a very large number of robots in recent warehouses (for example, Amazon has deployed over 200, 000 robots operating in their warehouses Koetsier (2022)) and the unprecedented high demand of the customers Berthene (2022). It is a crucial requirement to develop a powerful and adaptive task scheduling algorithm for the next generation smart warehouses that can efficiently allocate and schedule the robotic resources for goods and order flows management.

However, in a heterogeneous autonomous robotic (HAR) warehouse system, the traditional approaches to warehousing robots managements are inefficient to tackle the stochastic nature of the goods flow and the heterogeneous characteristic of the multiple types of new generations of autonomous mobile robots. Thanks to its outstanding performance in handling stochastic decision-making problems, deep reinforcement learning (DRL) has been shown as an effective technique in developing adaptive algorithms in various domains Sutton & Barto (2018); Lillicrap *et al.* (2015); Schulman *et al.* (2017).

In industrial automation, artificial intelligence and machine learning functions require data collecting from multiple elements of manufacturers and factories which raises privacy concerns. Federated learning (FL) McMahan *et al.* (2017) has recently emerged as a promising technique for model training in distributed environments while protecting data privacy by keeping the training data in the local devices. FL can perceive decentralized machine learning methods, especially DRL in industrial automation by allowing geographically distributed devices to train their own models without disclosing any sensitive and private data. The global model acquires the experiences of all the DRL agents while eventually protecting the security and privacy of each device.

FL is a promising technique to realize the decentralized DRL thanks to its advantage in protecting the security and privacy of the participants Wang *et al.* (2020e). Moreover, by aggregating the adequate trained parameters of the participants, the performance of DRL can be significantly enhanced compared to other centralized and distributed training techniques Wang *et al.* (2020e); Cao *et al.* (2021), especially in heterogeneous environments Wang *et al.* (2020d). Federated DRL has shown its advantage in many practical problems, for example, device-to-device mobile edge caching Wang *et al.* (2020d), user access control of open radio access network Cao *et al.* (2021), data management of industrial Internet of Things (IIoT) Zhang *et al.* (2021), and energy management of smart home Lee & Choi (2020).

In practical deployment, systems heterogeneity is a critical concern of the machine learning methods. Due to the heterogeneity of the multi-warehouse system, the data at each distributed warehouse may be insufficient to train a robust model for each DRL agent in a distributed training manner. On the other hand, transferring the distributed data to the coordinating aggregator (central controller) to build a global model may be costly in terms of communication and delay and raise the privacy issues. Moreover, due to the heterogeneity of the distributed data, e.g, data size, and data format, the global model may suffer a severe deviation and divergence in the training process, making transferring data for a centralized training becomes an inefficient solution. In short, a distributed training obtains a local policy with a low performance, whereas, a centralized training provide a better policy compared to a distributed training but is costly and infeasible. By aggregating only the distributed models but not the data, FL has been shown as an effective method to handle the heterogeneity of the participating devices and reduce the deviation of model training in heterogeneous systems McMahan *et al.* (2017); Li *et al.* (2018).

To overcome the aforementioned challenges, in this paper, we propose a federated learning-based method for implementing the decentralized reinforcement learning algorithm that achieves a global optimal task scheduling policy and improves the performance of the distributed DRL agents compared to the state-of-the-art centralized and distributed learning schemes. The DRL agents are the task schedulers implemented in the workstations at the geographically distributed warehouses. The policy gradient methods (e.g., Trust Region Policy Optimization

(TRPO), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO)) have been proven their successfulness in the discrete and continuous control compared to value-based methods Lillicrap *et al.* (2015), Henderson *et al.* (2018). Moreover, as shown in Schulman *et al.* (2017), PPO outperforms the other methods in almost all continuous control environments. Therefore, in this paper, we use PPO method to implement our DRL agents. The main contributions of this paper can be summarized as follows :

1. We formulate a task scheduling problem for autonomous warehouses with a heterogeneous autonomous robotic system as a queueing control optimization problem that considers the stochastic nature of the tasks flow and the heterogeneity of the autonomous robots. The objective is to minimize the queue length of tasks that are waiting to be processed in each warehouse.
2. We propose a novel task scheduling algorithm that employs a Proximal Policy Optimization (PPO) method to find an optimal task scheduling policy of the formulated problem.
3. We propose a Proximal Weighted Federated Learning (PWFL) algorithm for implementing a decentralized PPO algorithm that improves the performance of the geographically distributed PPO agents.
4. Extensive simulations demonstrate that the proposed PWFL framework can effectively reduce the average service time of the robots, average waiting time, and average queue length of the tasks compared with the existing methods.

The rest of the paper is organized as follows : Section 4.7 presents the system model and problem formulation. Section 4.8 introduces the Proximal Policy Optimization algorithm followed by Section 4.9 that presents a Proximal Weighted Federated Learning algorithm. Section 4.10 presents simulation results. Finally, Section 4.11 concludes the paper.

4.6 Related Work

Robotic warehouse management systems have been well studied for decades Roy *et al.* (2013); Ekren *et al.* (2012); Yuan & Gong (2017); Wang *et al.* (2020c); Ma *et al.* (2022); Lee & Murray (2019). These warehouse management systems includes autonomous vehicle-based storage and retrieval systems (AVS/RS) Roy *et al.* (2013); Ekren *et al.* (2012), robotic mobile fulfillment systems (RMFS) Yuan & Gong (2017); Wang *et al.* (2020c); Ma *et al.* (2022), and autonomous robots picking systems Lee & Murray (2019). Roy *et al.* Roy *et al.* (2013) proposed a semi-open queueing network model to analyze system performance and evaluate design trade-offs in an AVS/RS. Ekren *et al.* Ekren *et al.* (2012) presented a single-class multiple-server semi-open queueing network model for an AVS/RS. Yuan *et al.* Yuan & Gong (2017) formulated two open queue network models to determine the optimal number of robots to minimize the total throughput in an RMFS to achieve effective warehousing robots management. Wang *et al.* Wang *et al.* (2020c) proposed an open queueing network model for modular RMFS with aisle-captive robots for small and medium-sized logistics warehouses. The proposed modular RMFS is partitioned into independent modules, each with a workstation, a picker, and pod stations in the picking area and several independent aisles in the storage area. Ma *et al.* Ma *et al.* (2022) proposed a new scattered storage policy named scattered-correlation storage policy based on the commodity classification for mitigating the commodity storage assignment problem in RMFS. Lee *et al.* Lee & Murray (2019) formulated a coordinated routing of two types of heterogeneous autonomous robots, i.e., a picker robot and a transport robot, Fetch & Freight robots, to minimize the time required to retrieve a collection of items from within a warehouse. In contrast to the Amazon's Kiva robot system (renamed as Amazon Robotics) Boysen *et al.* (2019) which transport entire racks from/to the replenishment area, the Fetch & Freight robots retrieve individual items in the warehouse.

As the logistics system continuously expands with the development of robotic automation, *task scheduling* becomes urgent in the modern robotic warehouse system (RWS). Several works focus on the dynamic task scheduling problem of RWS Kim *et al.* (2020); Bolu & Korçak (2021); Tang *et al.* (2021); Yoshitake *et al.* (2019); Zou *et al.* (2017); Li *et al.* (2020c); Yang *et al.* (2021c); Roy *et al.* (2019); Zhou *et al.* (2014). Kim *et al.* Kim *et al.* (2020) proposed an efficient heuristic algorithm for assigning items to pods in a RMFS. Bolu *et al.* Bolu & Korçak (2021) proposed a parametric heuristic model for the order task selection process for an RMFS within a realistic simulation environment. In Tang *et al.* (2021), a soft actor-critic and hierarchical DRL based algorithm was proposed for multi-logistic robot task allocation. Yoshitake *et al.* Yoshitake *et al.* (2019) proposed a new robotic system using an AGV for order picking in logistics warehouses that flexibly schedules transport tasks of both inventory shelves and sorting shelves using a real-time holonic scheduling method. Zou *et al.* Zou *et al.* (2017) proposed an assignment rule based on handling speeds of workstations and design a neighborhood search algorithm to find a near optimal assignment rule for a RMFS. Li *et al.* Li *et al.* (2020c) proposed a novel scheduling mechanism for multi-robot and tasks allocation problems in an intelligent warehouse system with simultaneous multiple customer demands. Yang *et al.* Yang *et al.* (2021c) proposed an adaptive heuristic approach to assign generated tasks to robots in an RMFS-based smart warehouse, considering system dynamics such as the location of robots and pods, utilization of totes, and age of the tasks. Roy *et al.* Roy *et al.* (2019) analyzed a RMFS for single and multiple storage zone with both dedicated and pooled robot assignment based on multi-class closed queueing network models. Zhou *et al.* Zhou *et al.* (2014) proposed a heuristic balance mechanism to assign tasks to robots that focus on balancing robot workload while optimizing total robot travel time.

The evolution of e-commerce with the unprecedented-high demand for online shopping in the COVID era creates an endless stream of orders to the online retailers Berthene (2022). The advanced heterogeneous robotic system with multiple simultaneous customer orders creates a complicated and more stochastic environment that needs a new type of scheduling and management system. The aforementioned task scheduling methods do not capture the heterogeneity of the multi-warehouse system with multiple heterogeneous robotic systems. Therefore, it is insufficient and ineffective to deploy the same task scheduling policy simultaneously to geographically distributed warehouses. To the best of our knowledge, this is the first work that leverage the advanced federated learning to handle the heterogeneous of the distributed robotic task scheduling DRL agents.

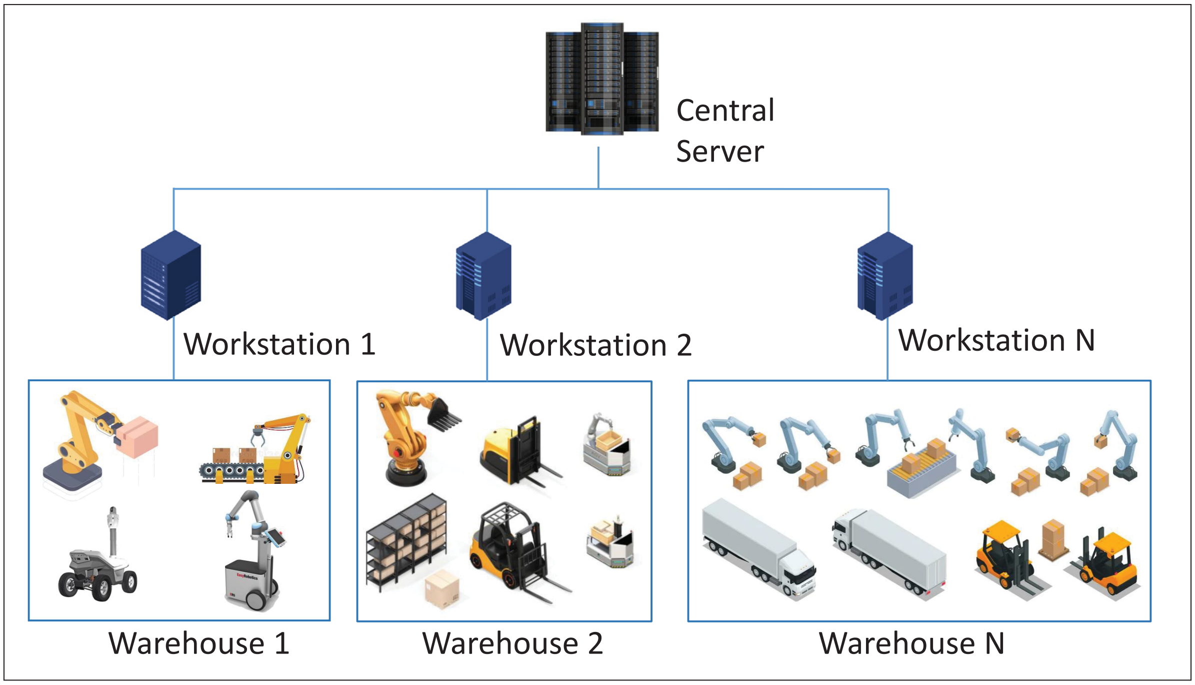


Figure 4.1 Illustration of a warehouse management system

4.7 System Model and Problem Formulation

4.7.1 Warehouse System Model

We consider a warehouses system of an e-commerce company consisting of a set $\mathcal{M}$ of M warehouses which are geographically distributed as shown in Fig. 4.1. Each warehouse can be a Distribution Center (DC) or a Fulfillment Center (FC) with the main focus is storing and delivering final goods, executing order receipt, checking, labeling, packing, and shipment to the consumer, for example, Amazon Fulfillment Center da Costa Barros & Nascimento (2021). Each warehouse is operated by a set $\mathcal{K}_m$ of K_m heterogeneous autonomous robots that perform the tasks, i.e., storage and retrieval items in the warehouse. The robots are heterogeneous with different mobility capabilities, i.e., automated guided vehicles (AGVs), autonomous mobile robots (AMRs), collaborative mobile robots (cobots). These mobile robots can autonomously navigate through a warehouse using sensors, cameras, and safety mechanisms to build a digital map of their environment and freely move around the warehouse without external guidance. The warehouse management system (WMS) feeds tasks to the robots via wireless communication, i.e., a 5G wireless network, and the digital mapping provides all the information they need to accomplish their tasks. We assume that each warehouse has a WMS implemented in a workstation. All the workstations are connected to a central server of the company.

Within each warehouse, the items are stored on the racks that are arranged back-to-back, and a set of racks forms a block of racks. To facilitate robot movements to the racks and reduce the blockage of the robots in a large-scale warehouse, each block of racks is surrounded by aisles as shown in Fig 4.2. There are a picking station and a delivery station at the corners of the warehouse where the items arrive and depart from the warehouse.

Each warehouse operates as follows : when an item or an order arrives, a corresponding storage or retrieval task is requested. If a storage task is scheduled for a robot, the robot has to pick up the item at the picking station and store the item at a predefined location on the racks. If a robot is assigned a retrieval task, the robot will travel to the location where the item is stored to bring the item to the delivery station via the shortest available path. A robot can be anyplace inside the warehouse and ready for a new task if it is free, i.e., in an idle state after finishing a task. We consider single-command operations in which each robot only performs a single task at every time slot [2].

The assumptions for analysis the task scheduling queuing system are summarized as follows :

- Each robot performs a single task at every time slot.
- Each robot can only run in four directions : forward, backward, left, and right.
- All robots are assigned tasks equally, there is no priority between the robots.
- The travel velocity of a robot remains constant within a time slot and robot acceleration/deceleration effects are ignored.
- Each robot is equipped with sensors for autonomous navigation, and they can avoid collision.
- Both picking and delivery station have sufficient space for robots to pick up and drop items simultaneously.

[2] We leave the multiple tasks operations scenario in which the robot has to visit multiple locations to pick or retrieve items as our future direction.

4.7.2 Task Scheduling Queueing System

In this paper, we apply time-driven decision-making, where the decision period is defined by the length of a time slot. This means that the arrival items/orders will be stored first in the queue of tasks and then be scheduled periodically every time slot [3]. The objective of this paper is to find an optimal task scheduling policy that minimizes the long-run average queue length of tasks in the warehouse. The time slot can be defined depending on the use-case of the warehouse system, i.e., the scale of the warehouses, the service time of the robots, and the arrival rate of items/orders.

The task scheduling process in a warehouse can be modeled as a queueing system. An arrival task (storage or retrieval task) will be entered in a queue of tasks and waiting to be serviced. The queue is emptied according to the first-come-first-serve (FCFS) discipline, and there is no preemptive priority between storage and retrieval tasks. We assume the storage and retrieval tasks follow the Poisson process. The location of each item is predefined following a discretely uniform distribution. Each robot in a warehouse acts as a server. The mobility patterns of the robots are different, hence the service times of the robots are different. The service time of a robot is the sum of the time to travel to the item location and the time to pick and place the item. The service time of a robot is typically not exponentially distributed. Therefore, we consider a general distribution for the service times of the robots.

A task scheduling queueing system with Poisson arrival, general service time, and multiple servers can be characterized as an $M/G/K$ queue, where M stands for the memoryless Poisson arrival, G represents general service time, and K is the number of robots employed. Unfortunately, the exact expressions for mean task queue length and waiting time in an $M/G/K$ queue are not available. Finding an optimal task scheduling policy for such heterogeneous multi-server queueing system with general distribution service times is not trivial even with a small number of robots (servers). However, numerous approximations do exist. The first approximation for the

[3] The proposed task scheduling method in this paper can also be extended to the event-driven decision-making, i.e., when the tasks flow is low, then whenever an item/order arrives, the agent could assign this task to the available robots.

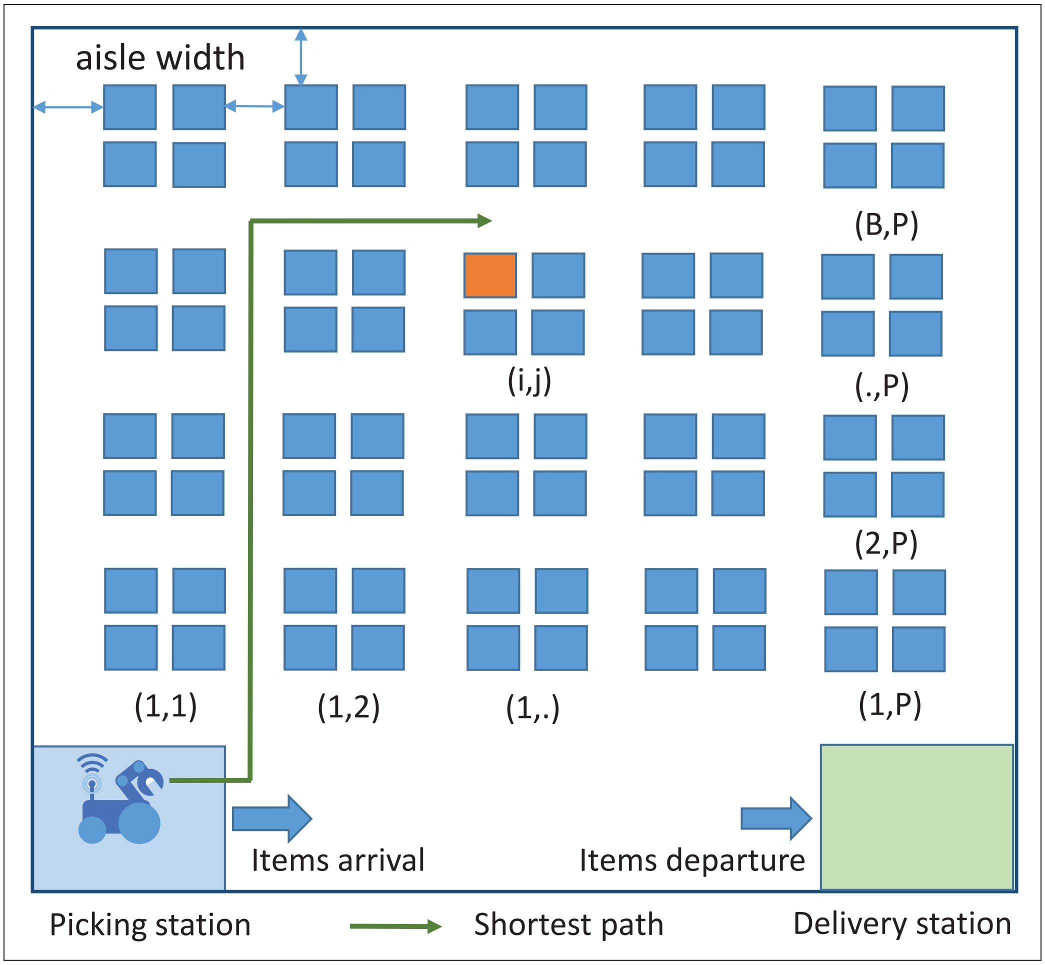


Figure 4.2 Illustration of a warehouses

mean waiting time for an $M/G/K$ queue was given by Lee and Longton Lee & Longton (1959) as follows :

$$\mathbb{E}[W^{M/G/K}] \approx \left(\frac{C^2+1}{2}\right)\mathbb{E}[W^{M/M/K}], \tag{4.1}$$

where $\mathbb{E}[W^{M/M/K}]$ is the mean waiting time with exponentially distribution with the same mean service time $\mathbb{E}[X]$ as in the $M/G/K$ system. $C^2 = var(X)/(\mathbb{E}[X])^2$ is the squared coefficient of variation (SCV) of service time variable X.

In this paper, we use the above approximation to formulated the task scheduling problem as presented in the following subsection.

4.7.3 Task Scheduling Problem Formulation

The expected service time of any robot is the sum of the expected time to travel to the picking station, to the location of the item, and/or to the delivery station. The travel time of the robots may involve some additional time due to the congestion or blocking by other robots. However, the phenomenon of the robots' congestion is very sophisticated and difficult to model mathematically. Therefore, in this paper, the expected travel time of the robots is calculated by the expected travel distance via the shortest path. The shortest path from the picking station to the block (i, j) and the shortest path from the block (i, j) to the picking station or delivery station is calculated as follows :

$$d_{i,j} = i(Rl_s + a) + j(Sw_s + a) + a, \tag{4.2}$$

where a is the width of an aisle, l_s and w_s are the length and width of a rack, respectively. R and S are the numbers of racks in a row and in a column of a block, respectively. The expected travel distance of a robot is given as follows :

$$\begin{aligned}\mathbb{E}[d_{i,j}] &= \frac{1}{PB}\sum_{i=1}^{P}\sum_{j=1}^{B} d_{i,j} \\ &= \frac{1}{PB}\sum_{i=1}^{P}\sum_{j=1}^{B}[i(Rl_s + a) + j(Sw_s + a) + a] \\ &= \frac{P+1}{2}(Rl_s + a) + \frac{B+1}{2}(Sw_s + a) + a,\end{aligned} \tag{4.3}$$

where P and B are the numbers of rows and columns of blocks in the warehouse, respectively. The shortest path from block (i, j) to block (i', j') is calculated as :

$$d_{i,j\rightarrow i',j'} = |i - i'|(Rl_s + a) + |j - j'|(Sw_s + a). \tag{4.4}$$

The expected travel distance from block (i, j) to block (i', j') is given as :

$$\mathbb{E}[d_{i,j\rightarrow i',j'}] = \frac{P}{2}(Rl_s + a) + \frac{B}{2}(Sw_s + a). \tag{4.5}$$

The expected travel time of robot k is given by :

$$\mathbb{E}[X_k] = \frac{1}{\bar{v}_k}\Bigg[\mathbb{E}[d_{i,j}] + \mathbb{E}[d_{i,j \to i',j'}]\Bigg], \tag{4.6}$$

where $\bar{v}_k$ is the average velocity of the robot k. The service rate of robot k is calculated using the expected travel time $\mu_k = \frac{1}{\mathbb{E}[X_k]}$.

The upper bound for mean queue length of a heterogeneous multi-server queue $M/M/K$ is given by Bertsekas & Gallager (2021) :

$$\mathbb{E}[Q] = p_0 \frac{\left(\sum_{k=1}^{K} \mu_k\right)^K}{\prod_{k=1}^{K}\sum_{j=1}^{k} \mu_j} \frac{\rho^{K+1}}{(1-\rho)^2}, \tag{4.7}$$

where p_0 is the probability the queue is empty and can be expressed as follows :

$$p_0 = \left[\sum_{k=1}^{K-1} \frac{\lambda^k}{\prod_{j=1}^{k}\sum_{i=1}^{j} \mu_i} + \frac{\lambda^K}{(1-\rho)\prod_{k=1}^{K}\sum_{j=1}^{k} \mu_j}\right]^{-1}, \tag{4.8}$$

where $\rho = \frac{\lambda}{\mu}$, $\mu = \sum_{k=1}^{K} \mu_k$, $\lambda = \lambda_I + \lambda_O$, and λ_I and λ_O are the arrival rate of storage task and retrieval task, respectively.

The service policy π_m of the queueing system in warehouse m is the task scheduling policy that assigns arrival tasks to the robots at each decision time. The objective is to find an optimal service policy π_m^* that minimize the long-run average queue length of tasks in the system

$$\min_{\pi_m} \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[Q_m(t)] \tag{4.9a}$$

$$\text{s.t. } \lambda \leq \mu \tag{4.9b}$$

Constraint (4.9b) is the stable constraint for the queueing system.

The formulated problem in (4.9) is a stochastic long-term optimization problem that is difficult to handle by traditional optimization approaches. Lyapunov optimization Neely (2010) could be an appropriate approach for this problem. However, Lyapunov optimization requires multiple steps to approximate to find the optimal solution of the short-term problem which results in increased computational complexity. Moreover, in the modern warehouse system with thousand robots, solving a sequence of optimization problems with thousand variables at every time slot by traditional optimization approaches is extremely costly. These hurdles can be overcome through deep learning approaches in which a well-trained network model can make decisions in a real-time manner with a complex and large-scale system Luong *et al.* (2019); Zhang, Patras & Haddadi (2019a). In the following section, we propose a deep reinforcement learning approach to obtain an optimal service policy for the formulated queueing system.

4.8 Proximal Policy Optimization

In this section, we formulate our task scheduling process as a Markov Decision Process (MDP). Then, we propose a Proximal Policy Optimization-based algorithm which is an on-policy policy gradient DRL to obtain an optimal policy for the formulated MDP.

The queue length dynamic can be formulated as follows :

$$Q_m(t+1) = \Big[Q_m(t) - I_m(t) - O_m(t)\Big]^+ + \Lambda_m^I(t+1) + \Lambda_m^O(t+1), \tag{4.10}$$

where $I_m(t)$ and $O_m(t)$ are the numbers of storage and retrieval tasks that are successfully processed at the previous decision period. $\Lambda_m^I(t+1)$ and $\Lambda_m^O(t+1)$ are the arrival numbers of storage and retrieval tasks at the current decision time.

The task scheduling problem at a warehouse m can be characterized as a MDP as follows :

4.8.1 System State, Action, and Reward Design

Consider an infinite-horizon discounted MDP, defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathbf{Pr}, r, \gamma)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathbf{Pr} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the transition probability $r : \mathcal{S} \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor.

4.8.1.1 State space

The system state at time slot $t+1$ is defined by the tuple $\mathcal{S}_m = (Q_m(t), I_m(t), O_m(t), \Lambda_m^I(t+1), \Lambda_m^O(t+1), \{\mathbf{h}_{m,k}(t)\}_{k\in\mathcal{K}_m})$ in which $h_{m,k}[t]$ is the current status of robot k in warehouse m, i.e., free or busy.

4.8.1.2 Action space

An action scheduled to a robot (not currently busy) is either a task (storage or retrieval) or a command allowing the robot to be idle until the next decision even if there are still awaiting jobs in the queue. The action space at time $t+1$ is defined by the tuple $\mathcal{A}_m = (\{a_{m,k}(t+1)\}_{m\in\mathcal{M},k\in\mathcal{K}_m})$, indicating the task scheduling to the robots in the set $\mathcal{K}_m$ in the warehouse m at time $t+1$.

4.8.1.3 Reward function

Designing the reward function is the most challenging part of reinforcement learning. The immediate reward at each time step should reflect the performance of the system. In the formulated problem, we aim to minimize the queue length Q_m of the waiting storage and retrieval tasks. As a result, the reward function is proportional to the number of accomplished tasks at the decision time and inverse proportional to the queue length. The reward for each agent at time t as is designed as follow :

$$r_m(t) = \kappa_1(I_m(t) + O_m(t)) + \kappa_2/Q_m(t) + \kappa_3/X_m(t), \tag{4.11}$$

where $X_m(t)$ is the service time of the robots at time t. Furthermore, because accomplished tasks and service time have different units (i.e., number of tasks and minutes), we use κ_1, κ_2, and κ_3 as the tunable positive coefficients to adjust the impact of the accomplished tasks, the waiting tasks, and the service time, respectively. Since the queue of tasks depends proportionally on the accomplished tasks, we can set κ_1 and κ_2 equally while fine-tuning the coefficient of the service time κ_3. Putting more weight on the inverse of service time, i.e., increase κ_3, can reduce the service time, and hence increase accomplished tasks. However, we cannot arbitrarily increase κ_3 as the service time is bounded by the capability of each robot.

4.8.2 Proximal Policy Optimization

Proximal policy optimization (PPO) Schulman *et al.* (2017) is a model-free, online, on-policy, policy gradient reinforcement learning method. This algorithm is a type of policy gradient training that alternates between sampling data through environmental interaction and optimizing a clipped surrogate objective function using stochastic gradient descent (SGD) Henderson *et al.* (2018). PPO alternatively constructs an unconstrained surrogate objective function to remove the incentive for large policy updates. PPO updates policies of the agents by taking multiple steps of (usually minibatch) SGD to maximize the objective :

$$\theta_m^{n+1} = \arg\max_{\theta_m} \mathop{\mathbb{E}}_{s,a\sim\pi_{\theta_m^n}} \left[L(s_m, a_m, \theta_m^n, \theta_m)\right], \tag{4.12}$$

$$\begin{aligned} &L(s_m, a_m, \theta_m^n, \theta_m) = \\ &\min\left(\frac{\pi_{\theta_m}(a_m|s_m)}{\pi_{\theta_m^n}(a_m|s_m)} A^{\pi_{\theta_m^n}}(s_m, a_m), \ \ \text{clip}\left(\frac{\pi_{\theta_m}(a|s)}{\pi_{\theta_m^n}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_m^n}}(s_m, a_m)\right), \end{aligned} \tag{4.13}$$

$$\begin{aligned} &\theta_m^{n+1} = \\ &\arg\max_{\theta_m} \frac{1}{|\mathcal{D}_m^n|T} \sum_{\tau\in\mathcal{D}_m^n} \sum_{t=0}^{T} \min\left(\frac{\pi_{\theta_m}({}_t|s_t)}{\pi_{\theta_m^n}(a_t|s_t)} A^{\pi_{\theta_m^n}}(s_m(t), a_m(t)), g(\epsilon, A^{\pi_{\theta_m^n}}(s_m(t), a_m(t)))\right); \end{aligned} \tag{4.14}$$

$$\boldsymbol{\phi}_m^{n+1} = \arg\min_{\phi_m} \frac{1}{|\mathcal{D}_m^n|T} \sum_{\tau\in\mathcal{D}_m^n} \sum_{t=0}^{T} \left[V_{\boldsymbol{\phi}_m^n}(s_m(t)) - r_m(s_m(t), a_m(t))\right]^2; \tag{4.15}$$

where L is given in (4.13). $\pi_{\theta_m}(a_m|s_m)$ is new parameterized policy of the agent m that tries to seek the optimal parameter vector θ_m and $\pi_{\theta_m^k}(a_m|s_m)$ is the old policy. ϵ is a small hyperparameter implying how far away the new policy is allowed to go from the old one. The advantage function $A^{\pi_{\theta_m^n}}(s_m, a_m)$ can be calculated by

$$A^{\pi_{\theta_m^n}}(s_m, a_m) = Q^{\pi_{\theta_m^n}}(s_m, a_m) - V^{\pi_{\theta_m^n}}(s_m), \tag{4.16}$$

where $Q^{\pi_{\theta_m^n}}(s_m, a_m)$ is the action-value function estimated by samples, and $V^{\pi_{\theta_m^n}}(s_m)$ is the approximation of the state-value function.

$$Q^{\pi_{\theta_m^n}}(s_m(t), a_m(t)) = \mathbb{E}\left[\sum_{l=0}^{\infty} \gamma^l r_m(s_m(t+1))\right]. \tag{4.17}$$

The PPO algorithm is presented in Algorithm 4.1. Each workstation (i.e., PPO agent) collects information of its warehouse including the number of items and orders in the queue, the number of the arrival items and orders at the decision time, and the status of each robot (idle or busy) (line 3) [4]. The workstation schedules the tasks for the robots in its warehouse based on the decision made by the policy network (line 4).Then the workstation computes advantage estimates and updates the policy by maximizing the PPO-clip objective with the minibatch of transitions (lines 5-7). The workstation trains the value functions by regression on mean-squared error (MSE) (line 8). These steps are repeated until the task scheduling policy converges to an optimal policy.

4.9 Federated Learning for Model Aggregation

In this section, we propose an FL-based method to implement a decentralized PPO algorithm proposed in Section 4.8.

4.9.1 Vanilla Federated Learning

The vanilla Federated Learning technique was first proposed in McMahan *et al.* (2017) named Federated Averaging (FedAvg) in which at each update round, a subset $\bar{M} \leq M$ of agents are selected and averaging at the central server for the number of epochs. In FL, choosing the number of local epochs training for each communication round of global update is crucial since it affects the convergence of the FedAvg algorithm. In our scenario, the warehouses are heterogeneous, hence, a large number of local epochs may lead each local agent towards the

[4] Due to uncertainty, information for the agent could be incomplete. We assume prediction mechanisms to fulfill the incomplete information, as well as the uncertainty of goods/tasks flow. However, this paper focuses on task assignments, so such prediction will be presented in our future work.

Algorithme 4.1 PPO algorithm for task scheduling in automated warehouse system

```
Initialize policy parameters $\{\theta_m^{(0)}\}$, initialize value function parameters $\{\phi_m^{(0)}\}$ for agent $m$;
for $n = 0, 1, 2, ..$ iterations do
    for each workstation $m \in \mathcal{M}$ do
        Workstation $m$ collects information of its warehouse including the number of
          items and orders in the queue, the number of the arrival items and orders at the
          decision time, and the status of each robot (idle or busy);
        Workstation $m$ schedules the tasks for the robots based on the decision of the
          policy network $\pi_{\theta_m}(a_m|s_m)$;
        Collect a minibatch of $D_m$ transitions $\mathcal{D}_m^n = \{s_i, a_i, r_i, s_{i+1}\}_{i=0:D_m-1}$;
        Compute advantage estimates $\hat{A}(s_m(t), a_m(t))$ based on the current value
          function $V_{\phi_m^n}(s_m(t))$;
        Update the policy by maximizing the PPO-clip objective in (4.14) where

            $g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1-\epsilon)A & A < 0. \end{cases}$

          Fit value function by regression on mean-squared error in (4.15);
    end for
end for
```

local optimal as opposed to the global objective, which potentially results in a divergence. However, a larger number of local epochs can reduce communication costs, which can improve the overall convergence speed in communication-constrained conditions. On the other hand, a smaller number of local epochs may not allow the local agent to complete training within a given communication round and hence, therefore, significantly reducing the global performance. Therefore, it is crucial to tune the optimization hyperparameters of FL properly McMahan *et al.* (2017); Li *et al.* (2018). In the simulation, we continue to fine-tune the number of the local updates for each communication round until our global model converges and reaches good performance in terms of average reward and average queue length.

Proposition 1. *At each communication round in the FedAvg algorithm, the global model parameters are aggregated by averaging the received parameters from selected agents as*

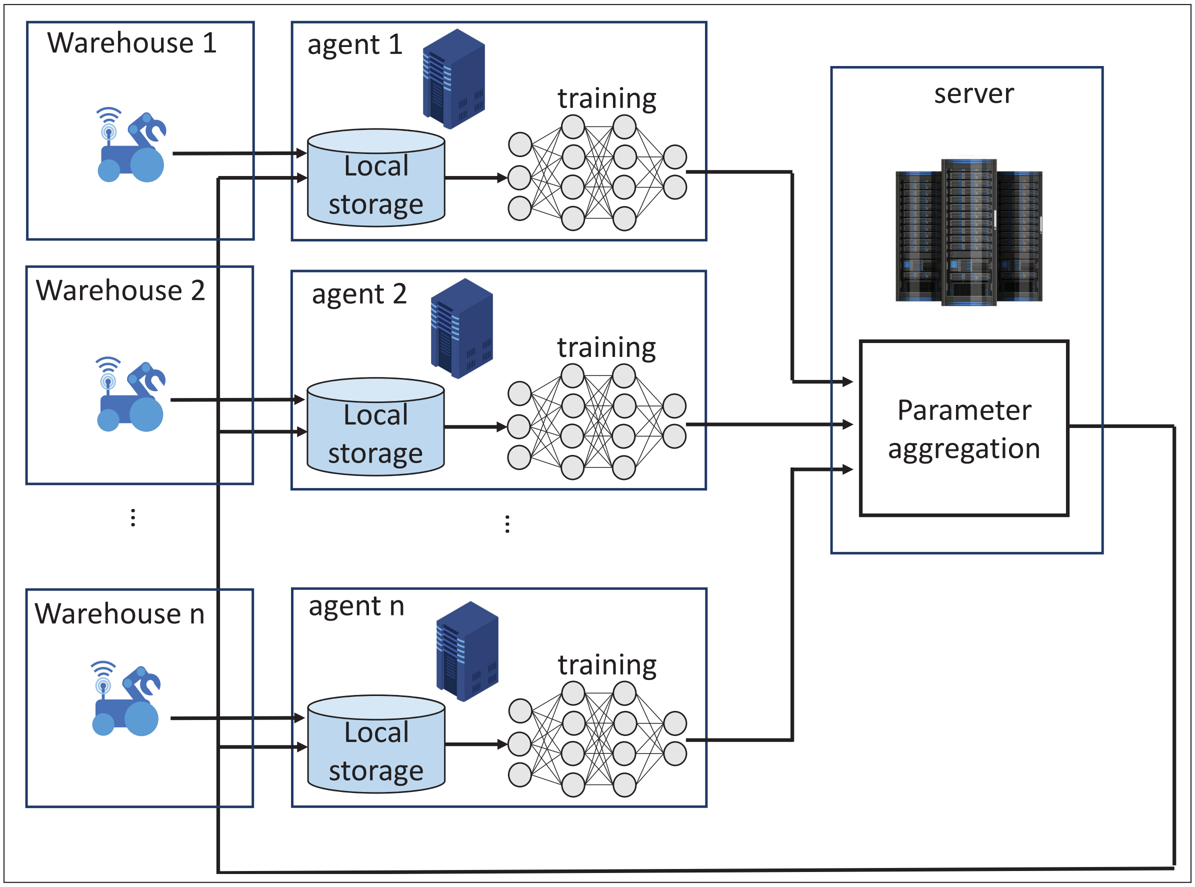


Figure 4.3 Federated learning framework

follows :

$$
\begin{aligned}
\theta_G^{(n+1)} &= \frac{1}{\overline{M}} \sum_{m \in \overline{\mathcal{M}}} \theta_m^{(n+1)}, \\
\boldsymbol{\phi}_G^{(n+1)} &= \frac{1}{\overline{M}} \sum_{m \in \overline{\mathcal{M}}} \boldsymbol{\phi}_m^{(n+1)},
\end{aligned}
\tag{4.18}
$$

where $\overline{\mathcal{M}} \subset \mathcal{M}$ is the set of the selected agents at the communication round $n+1$

4.9.2 Proximal Weighted Federated Learning

Due to the heterogeneity of the agents (i.e., heterogeneous warehouses), it is not practical to aggregate each local parameter equally. In this section, we propose a weighted FL technique based on the proximal aggregation technique in Li *et al.* (2018). The weighted federated aggregation problem is formulated as follow :

$$
\begin{aligned}
&\max_{\boldsymbol{\theta}_G} h(\theta_G) = \sum_{m\in\overline{\mathcal{M}}} \omega_m H_m(\theta_G),\\
&\min_{\boldsymbol{\phi}_G} g(\phi_G) = \sum_{m\in\overline{\mathcal{M}}} \omega_m G_m(\phi_G),
\end{aligned}
\tag{4.19}
$$

where $H_m(\theta_m^n)$ and $G_m(\phi_m^n)$ are the local functions of the policy network and value network in the right hand side of (4.14) and (4.15), respectively. θ_G, and $\boldsymbol{\phi}_G$ are the global parameters. ω_m is a weight factor for agent m to measure the contribution of agent m to the global model.

In the proposed Proximal Weighted Federated Learning (PWFL) algorithm, a proximal term, which is the Euclidean norm of the global parameters of each communication round and the local parameters at each local training phase, is added to the local functions of each agent. The local parameters of each agent are trained as follows :

$$
\theta_m^{n+1} \approx \arg\max_{\theta_m} H_m(\theta_m^n) + \frac{\mu}{2}\left\|\theta_G - \theta_m^n\right\|^2, \tag{4.20}
$$

$$
\boldsymbol{\phi}_m^{n+1} \approx \arg\min_{\phi_m} G_m(\phi_m^n) + \frac{\mu}{2}\left\|\boldsymbol{\phi}_G - \boldsymbol{\phi}_m^n\right\|^2, \tag{4.21}
$$

Proposition 2. *In the PWFL algorithm, the global model parameters are aggregated by averaging the received parameters from selected agents as follows :*

$$
\begin{aligned}
\theta_G^{(n+1)} &= \frac{1}{\overline{M}} \sum_{m\in\overline{\mathcal{M}}} \omega_m^{(n+1)} \theta_m^{(n+1)},\\
\boldsymbol{\phi}_G^{(n+1)} &= \frac{1}{\overline{M}} \sum_{m\in\overline{\mathcal{M}}} \omega_m^{(n+1)} \boldsymbol{\phi}_m^{(n+1)},
\end{aligned}
\tag{4.22}
$$

where the local parameters θ_m^{n+1} and $\boldsymbol{\phi}_m^{n+1}$ are locally trained according to (4.20) *and* (4.21)*, respectively.*

The weight of each agent $\omega_m^{(n+1)}$ at each communication round is calculated as follows :

$$
\omega_m = softmax(\mathbf{R}), \tag{4.23}
$$

Algorithme 4.2 PWFL for automated warehouse system

```
Initialize policy parameters $\{\theta_G^{(0)}\}$, initialize value function parameters $\{\phi_G^{(0)}\}$ for the global model;
Initialize number of local update epochs $\Omega$;
for each communication round do
    Central server selects a subset of $\overline{M}$ workstations at random;
    Central server sends $\theta_G^{(n)}$ and $\boldsymbol{\phi}_G^{(n)}$ to all selected agents;
    Each workstation $m$ updates its parameters $\theta_m^{(n+1)}$ and $\boldsymbol{\phi}_m^{(n+1)}$ according to (4.20) and (4.21) and sends back to the central server;
    Central server aggregates the global parameters $\theta_G^{(n+1)}$ and $\boldsymbol{\phi}_G^{(n+1)}$ according to (4.22);
end for
```

where $\mathbf{R} = [\{\overline{r}_m(t)\}_{m \in \mathcal{M}}]$ is the vector of the average reward of the selected agents during the local updates phase.

The PWFL algorithm is presented in Algorithm 4.2.

4.9.3 Complexity and Convergence Analysis

The computational complexity of the PPO-based algorithm can be calculated based on the complexity related to the training of the policy and value networks. Let L^{P} and L^{V} denote the number of fully connected layers of the policy network and value network, respectively. The computational complexity of Algorithm 4.1 is $O(\sum_{l=0}^{L^{\mathrm{P}}-1} u_l^{\mathrm{P}} u_{l+1}^{\mathrm{P}} + \sum_{l=0}^{L^{\mathrm{V}}-1} u_l^{\mathrm{V}} u_{l+1}^{\mathrm{V}})$, where u_l^{P} and u_l^{V} are the unit numbers in the l-th layers of policy network and in the l-th layer of value network, respectively.

The convergence of the proposed PWFL algorithm can be proved by the measure of dissimilarity between the distributed agents (i.e., warehouses) in the federated aggregation round as follows Li *et al.* (2018)

Theorem 3. *Assume the local functions H_m and G_m are non-convex, L-Lipschitz smooth and are B-dissimilar that $\mathbb{E}[\|\nabla H_m(\theta_G)\|^2] \geq \|h(\theta_G)\|^2 B^2$ and $\mathbb{E}[\|\nabla G_m(\phi_G)\|^2] \leq \|g(\phi_G)\|^2 B^2$.*

Then the proposed Alg. 4.2 converge with some $\epsilon > 0$ after T federated aggregation rounds that $\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}[\left\|\nabla H_m(\theta_G^{(t)})\right\|^2] \leq \epsilon$ *and* $\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}[\left\|\nabla G_m(\phi_G^{(t)})\right\|^2] \leq \epsilon$.

Démonstration. A similar proof can be found at Li *et al.* (2018). □

4.9.4 Practical Implementation Discussion

In smart logistic, we can easily have thousands of robots per warehouse resulting in an unpredictable long training time until the algorithm converges to an optimal control policy. A feasible technique is to pre-train the agents offline in a simulated environment. Then, the offline pre-trained models can be deployed onto a real-world warehouse system. If the number of robots in the real warehouse is larger than the number of robots present in the pre-trained model, the workstation, i.e., the agent can select a subset of robots to schedule tasks at each time step. However, in reality, pre-training a DRL model with thousands of robots is extremely expensive in terms of training resource (i.e., GPU).

Generally, training in a simulator is simpler than in the real-world scenarios which is much more complicated. The differences between the simulator and real world introduces a reality gap Jakobi, Husbands & Harvey (1995); Rusu *et al.* (2017); Chebotar *et al.* (2019); Tzeng *et al.* (2020); Dai *et al.* (2022). Normally, the simulator-trained agents do not directly transfer to the real-world scenarios. There are several methods to address this sim-to-real gap in the real-world robotic system, for example, fine-tuning the simulator-trained agents in the real-world applications Rusu *et al.* (2017), combining adapting system identification and dynamics domain randomization using real world data to reduce the reality gap Chebotar *et al.* (2019), and performing domain adaptation for transferring from the simulated data to the real-world physics Tzeng *et al.* (2020). Digital twin can be considered as an efficient approach to fill the reality gap in which the real system is synchronized with the simulation Tao, Zhang, Liu & Nee (2018). Then, the neural network will be trained on a digital twin of the real system with simulated data was parameterized and fine-tuned with real parameters from the real system.

For the federated learning implementation, after deploying a pre-trained model to the workstations in a warehouse, the distributed DRL agents progressively improve their performance by training their local models with the real data collected from the warehouse. Then, these agents will send their local parameters to the central controller through a wired or wireless network to aggregate the global model while keeping all the local data privately. Then, the central controller sends the global model back to the workstations to update the local models. This federated learning process can continue until a pre-defined termination criterion is met (e.g., the maximum number of aggregation rounds, or a model accuracy threshold).

4.10 Simulation Results

4.10.1 Simulation Setting

We consider an e-commerce company consisting of three warehouses. The system setting is given in Table 4.2. The arrival of storage and retrieval tasks follow a Poisson distribution with the parameter set in the range of $[1-6]$ tasks per minute. The location of each task is uniformly distributed within the blocks of racks in each warehouse. The robots' velocity at each time slot follows a uniform distribution between the minimum and maximum velocity values which are varied across warehouses.

System heterogeneity setting : The number of rows and number of columns of blocks in each warehouse are uniformly distributed in the range $[10-20]$. To train the agents, the arrival rates of the tasks (storage and retrieval task) are set to 3, 4, and 5 tasks per minute in the first, second, and third warehouses, respectively. The maximum speed of the robots in the first, second, and third warehouses are set to 1.5, 2.5, and 3 m/s, respectively.

The PPO hyperparameters setting is given in Table I. However, they are not chosen arbitrarily but should be link to the network parameters. As an example, we choose the number of robots in each warehouse in the range of [8-10]. Therefore, the number of hidden layers is turned in the range of 1 to 3, and the number of hidden units is from 64 to 128. To implement the

Tableau 4.1 PPO hyperparameters setting

Parameter	**Value**
Policy network	128, tanh, 128, tanh, 128, tanh
Value network	128, tanh, 128, tanh, 128, linear
Learning rate	$1e-4$
Batch size	20
Discount factor	0.995
ϵ clip	0.3

Tableau 4.2 Simulation parameters

Parameter	**Value**
Number of warehouses	3
Number of robots in each warehouse	[6-10]
Width of an aisle	2 m
Length of a rack	2.4 m
Width of a rack	1.2 m
Number of racks in a row of a block	2
Number of racks in a column of a block	2
Number of rows of blocks	[10-20]
Number of column of blocks	[10-20]

neural networks, we employ the powerful open-source machine learning framework PyTorch version 1.2.0 primarily developed by Facebook's AI Research lab. We use Python 3.7 as the programming language on a desktop computer with hardware configuration : Intel(R) Core(TM) i7-4790 CPU 3.60GHz and 16GB RAM.

To evaluate the performance of our proposed Proximal Weighted Federated Learning framework presented in Algorithm 4.2 (denotes as 'PWFL'), six benchmarks are presented as baselines for comparisons as follows :

- **Distributed Learning** (denotes as 'DL') Cao *et al.* (2021) : In the 'DL' algorithm, each workstation trains its PPO agent model with its own local experiences independently. There is neither interaction nor information exchange between the workstations.
- **Centralized Learning** (denotes as 'CL') Cao *et al.* (2021) : In this baseline, no workstation trains its local PPO agent model, but sends all the local data to the central server. The central

server trains the global model with the collected data from all the workstations. Then all the workstations update their local agent model by the global model trained by the central server.

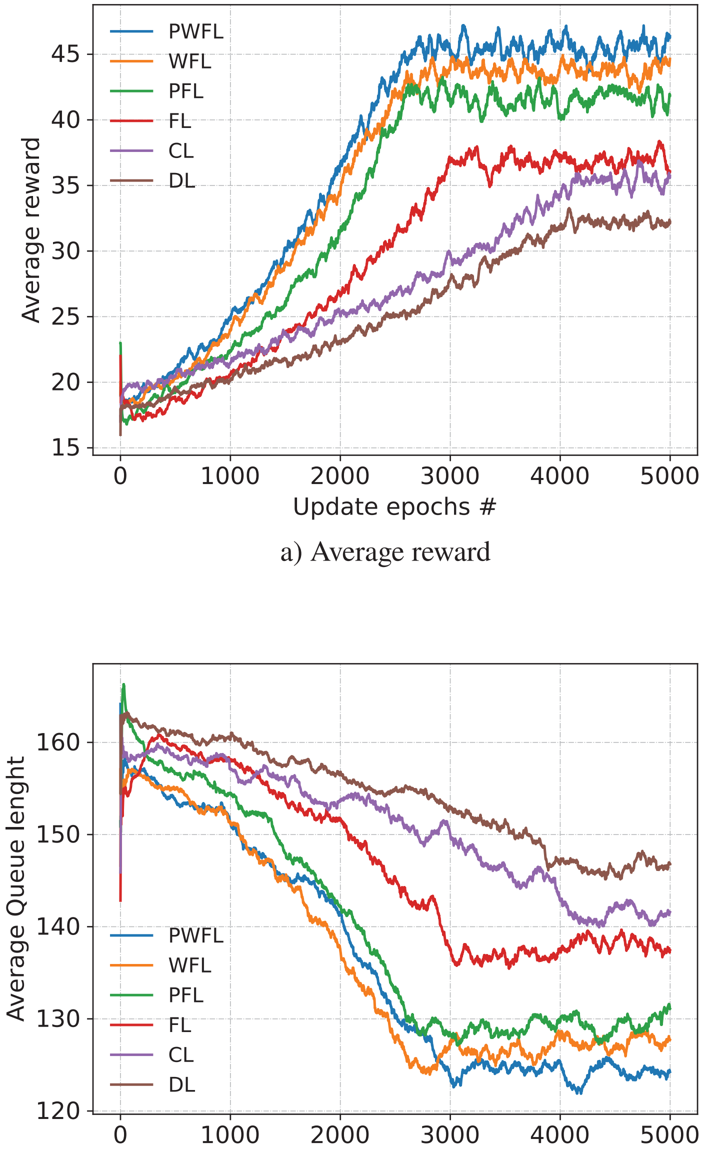
- **FedAvg** (denotes as 'FL') : This baseline is the vanilla Federated Learning framework proposed by McMahan in 2017 McMahan *et al.* (2017).
- **FedProx** (denotes as 'PFL') Li *et al.* (2018) : A generalization and re-parameterization of 'FedAvg' to tackle heterogeneity in federated networks. In this baseline, the local parameters are updated by adding a proximal term in the local function of each agent to effectively limit the impact of variable local updates.
- **Weighted Federated Deep Reinforcement Learning** (denotes as 'WFL') Wang *et al.* (2020d) : Each agent's parameters will be weighted by its average accumulated reward value at the time of aggregation by the central server.
- **Fastest Server First** (denotes as 'FSF') Nourbakhsh & Turner (2022) : a simple greedy policy commonly implemented in practice, which always assigns each arriving task to an available (i.e., idle) robot which can handle this task most rapidly.

4.10.2 Results Analysis

4.10.2.1 Convergence analysis

Fig. 4.4 shows the training curves that present the convergence of the accumulative rewards (Fig. 4.4a) and the average queue length (Fig. 4.4b) for the six learning schemes. We only plot the same agent in each scheme (i.e., agent 3) for comparison.

Fig. 4.4a indicates that all the learning schemes tend to converge gradually to the optimal policy after around 3000 to 5000 update epochs. The PPO algorithm is on-policy which has a higher variance and less sample efficient than the off-policy method. Therefore, it will need more samples to converge to an optimal policy. Moreover, we can observe that the four federated DRL-based schemes converge faster and obtain a better performance compared to the centralized and distributed DRL-based schemes. More specifically, the proposed 'PWFL' scheme achieves the highest performance and the 'DL' scheme converges to the lowest value in terms of reward,

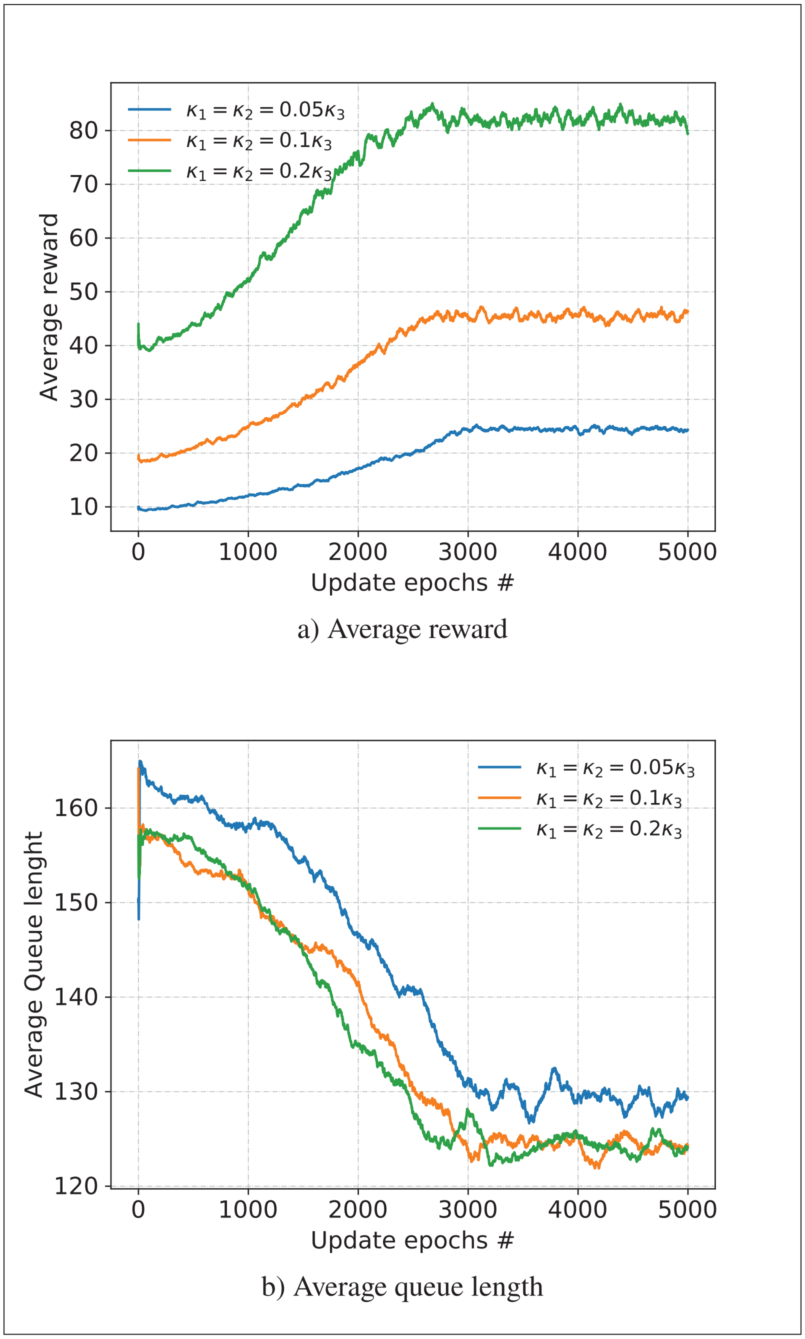


a) Average reward

b) Average queue length

Figure 4.4 Average reward (a) and average queue length (b) of different learning schemes with the number of update epochs (each with 20 episodes)

as shown in Fig. 4.4a. The ‘DL’ scheme trains each agent independently without interaction between the agents, whereas the ‘CL’ scheme trains a single model at the central server with the



a) Average reward

b) Average queue length

Figure 4.5 Average reward (a) and average queue length (b) with different reward setting

collected data from all the agents. However, the training data is distributed in a heterogeneous nature in the network. Each warehouse generates its own local data which is non-identically distributed among the warehouses. Therefore, the ‘CL’ scheme may suffer from high variance

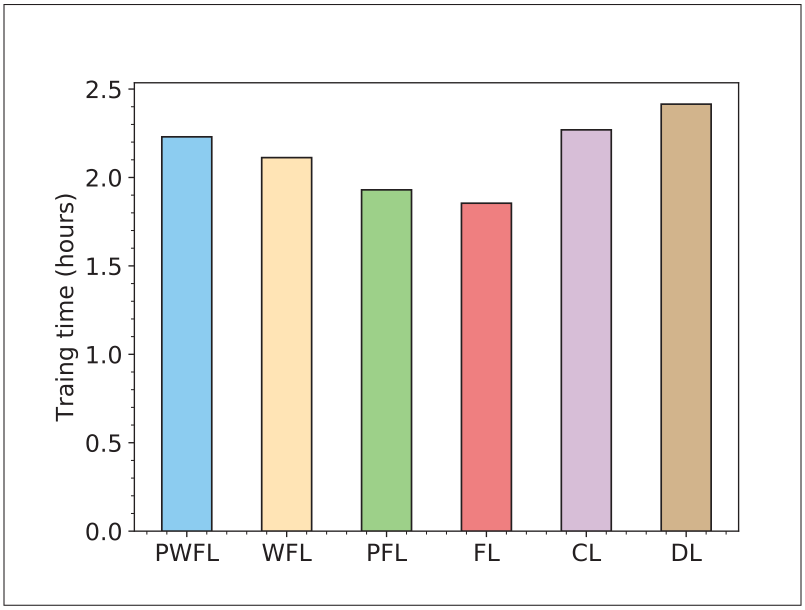
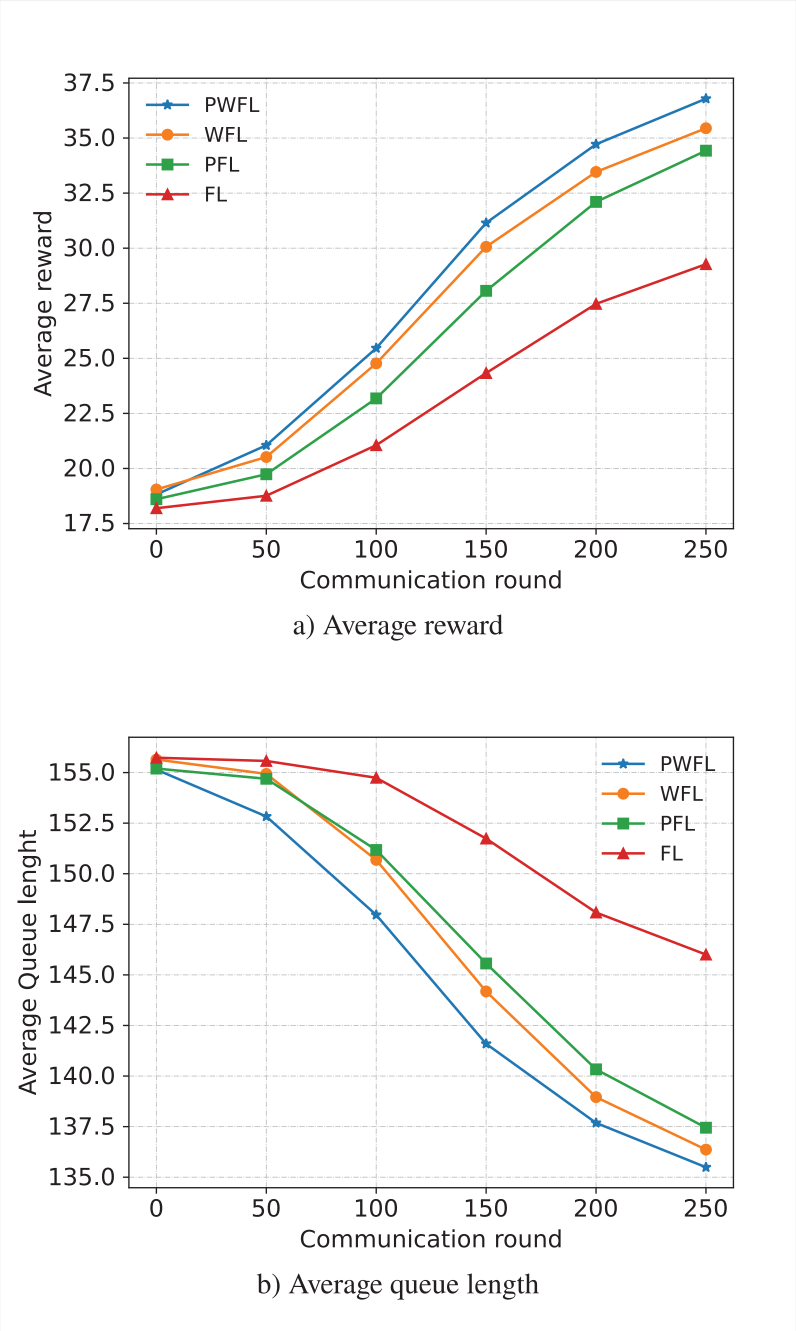


Figure 4.6 Training time (in hours) of different learning methods

resulting in poor performance compared to the ‘FL’ based schemes in which a central server continuously aggregates the local trained models of distributed agents to improve the global model.

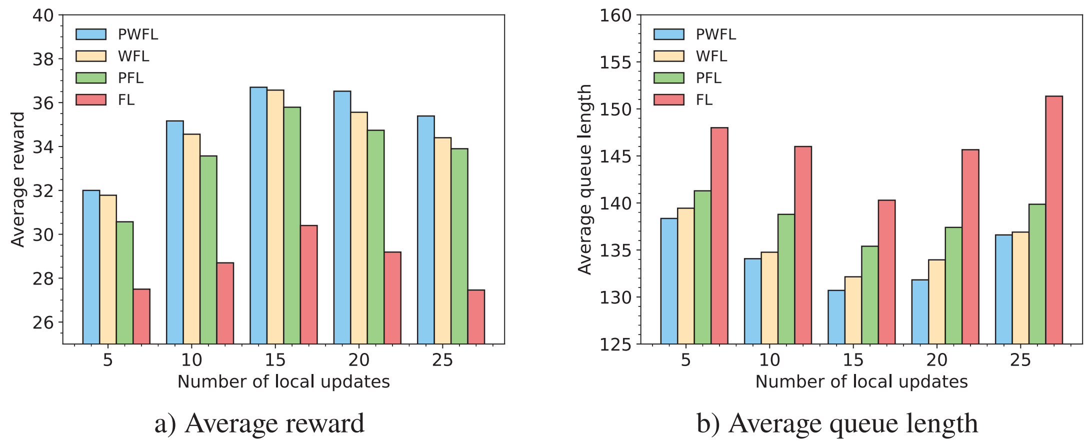
Fig. 4.4b shows the convergence behavior of the mean queue length, i.e., our objective function. It can be observed that the average queue lengths of all learning schemes decrease gradually to a stable value presenting the efficiency of the training process in minimizing the queue length of tasks waiting to be processed in the warehouse. Particularly, the proposed ‘PWFL’ scheme is the most efficient task scheduling scheme that converges to the shortest queue compared to all other schemes, and the ‘DL’ scheme converges to the longest queue of tasks. This result is consistent with the results indicated in Fig. 4.4a with the average reward value. Additionally, from Fig. 4.4a and Fig. 4.4b, even though the vanilla ‘FL’ algorithm outperforms the ‘DL’ and ‘CL’ schemes, the improvement gain is not significant compared to the performance gain of the three schemes ‘PWFL’, ‘WFL’, and ‘PFL’. Furthermore, the three schemes ‘PWFL’, ‘WFL’, and ‘PFL’ present a significant convergence improvement relative to ‘FL’ in heterogeneous environments.



a) Average reward

b) Average queue length

Figure 4.7 Average reward (a) and average queue length (b) of the federated DRL based schemes with the number of communication (aggregation) round (each with 100 episodes)



a) Average reward

b) Average queue length

Figure 4.8 Average reward (a) and average queue length (b) of the federated DRL based schemes with different number of local updates per communication (aggregation) round

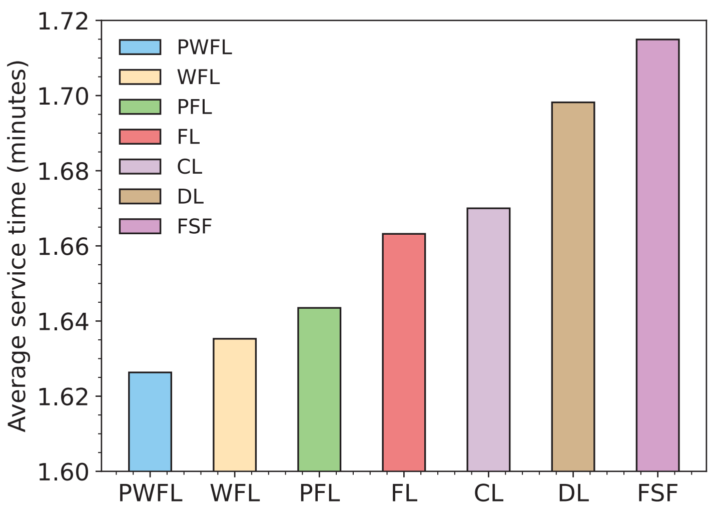
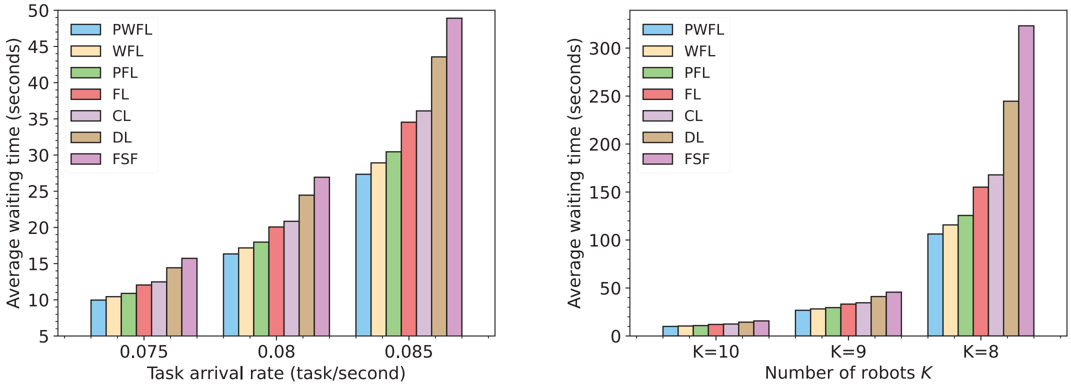


Figure 4.9 Average service time (average time of a general robot to finish a general task) of different task scheduling schemes

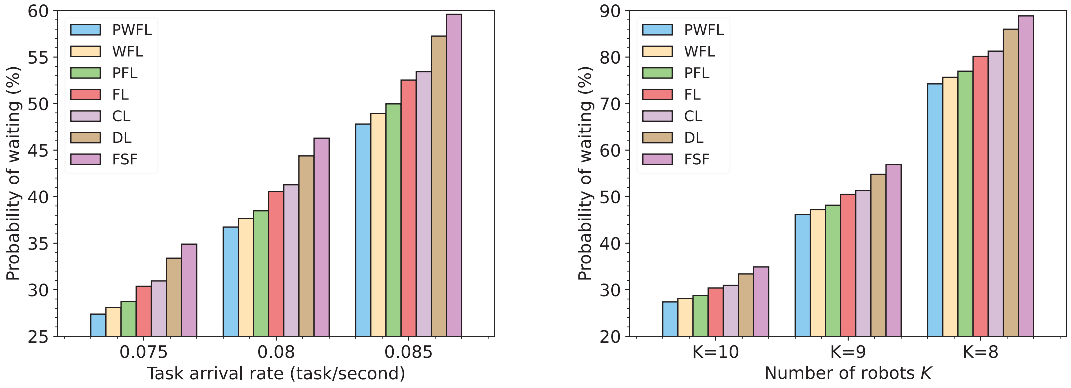
Fig.4.5a and Fig. 4.5b illustrate the effect of different reward setting on the performance in terms of immediate reward and average queue length of the proposed ‘PWFL’ scheme. Fig.4.5a shows



a) vs. arrival rate

b) vs. number of robots

Figure 4.10 Average waiting time of different learning schemes



a) vs. arrival rate

b) vs. number of robots

Figure 4.11 Probability of waiting of different learning schemes

that with different reward setting, the agent can obtain different immediate reward value over the training process. For example, we set the reward coefficients with the ratio $\kappa_1 = \kappa_2 = 0.05\kappa_3$ and increase the coefficient κ_3, i.e., the coefficient of the service time of the robot that the agent selects in the decision duration. It is obvious that the agent can obtain a higher immediate reward if the coefficient κ_3 increases since the reward increases linearly with κ_3 as shown in Fig.4.5a. However, when the ratio of the coefficients exceeds $\kappa_1 = \kappa_2 = 0.1\kappa_3$ the network performance in terms of average queue length becomes saturated because of the bounded capability of the robots as shown in Fig. 4.5b.

We compare the training time of the different learning methods in Fig. 4.6. For a fair comparison, we use the same desktop computer with the hardware configuration Intel(R) Core(TM) i7-4790 CPU 3.60GHz and 16GB RAM. The training time is measured over 5000 update epochs, each epoch is with 20 episodes. It can be observed that the 'FL' learning method converges in the shortest time to a global policy for all agents, while the 'DL' learning method requires the longest training time. These results can be explained by the fact that the local data at each distributed warehouse may be insufficient to train a robust model for an agent. Although the 'DL' baseline converges to a local policy, the performance of the model is incomparable to all other learning methods, i.e., centralized training method with combined data, and federated training method with parameter aggregation. Moreover, the vanilla federated learning method 'FL' provides the lowest training time because it equally aggregates the parameter of each agent without putting weight on each agent, i.e. the 'WFL' learning method, or adding a proximal term, i.e., the 'PFL' learning methods, or our proposed learning methods 'PWFL'.

4.10.2.2 Effects of the communication rounds and number of local updates

Fig. 4.7 shows the learning curves for the FL-based schemes with the number of communication (aggregation) rounds. We can observe a monotonic increase in terms of average reward (Fig. 4.7a) and a monotonic decrease in terms of average queue length (Fig. 4.7b) of all the FL-based schemes when the number of communication rounds increases. From Fig. 4.7a, when the number of communication rounds is 250, the 'PWFL' scheme achieves an average reward of 36.6 compared to 29.3 given by the 'FL' scheme, which presents an improvement of 24.9%. Similarly, an improvement of 20.8% and 17.4% can be observed in the case of the 'WFL' and 'PFL' schemes compared to the 'FL' scheme, respectively.

Fig. 4.8 shows the effects of the number of local updates on the performance in terms of average reward (Fig. 4.8a) and average queue length (Fig. 4.8b) of all the FL-based schemes. The number of local epochs plays an important role in federated learning Li *et al.* (2018). From Fig. 4.8a and Fig. 4.8b, all the federated DRL-based schemes get their best performance at around 15 to 20 local updates per communication round.

4.10.2.3 System Performance Analysis

Fig. 4.9 shows the average service time, i.e., the average time that a robot accomplishes a general task. While the greedy 'FSF' experiences the lowest performance, the proposed 'PWFL' outperforms all the baseline schemes. Indeed, the 'WFL' scheme can be viewed as a special case of our proposed 'PWFL' when we set the value of μ in (4.20) and (4.21) equal to 0. From Fig. 4.10 and Fig. 4.11, the proposed 'PWFL' is the most efficient task scheduling scheme when the task arrival rate increases and the number of robots in the warehouse decreases. For example, when the task arrival rate (a general task) is 0.085 task per second, the reduction in terms of average waiting time of the proposed 'PWFL' scheme, 'WFL' and 'PFL' baselines compared to the greedy 'FSF' baseline are 44%, 40.8%, and 37.7%, respectively. Whereas, when the number of robots in the warehouse is 8, the reduction in terms of average waiting time of the proposed 'PWFL' scheme, 'WFL' and 'PFL' baselines compared to the greedy 'FSF' baseline are 67.2%, 64.4%, and 61.1%, respectively.

4.11 Conclusion

In this paper, we have investigated the task scheduling problem for a heterogeneous autonomous robotic system in automated warehouses. The problem of task scheduling is formulated as a queueing control problem. We propose a deep reinforcement learning-based algorithm to obtain an optimal task scheduling policy of the formulated problem. Then, we propose a Proximal Weighted Federated Learning (PWFL) to implement a decentralized DRL algorithm in which each warehouse in the system is operated by a Proximal Policy Optimization (PPO) agent implemented in a workstation to schedule tasks for the robots in each warehouse, and a central server acts as a global model aggregator. The simulation results demonstrate the improvement of our proposed PWFL task scheduling scheme compared to prior work in terms of average queue length, average waiting time, and probability of waiting.

CHAPITRE 5

ENERGY EFFICIENCY DEEP REINFORCEMENT LEARNING FOR URLLC IN 5G MISSION-CRITICAL SWARM ROBOTICS

Manh Tai Ho[1] , Kim-Khoa Nguyen[1] , Mohamed Cheriet[1]

[1] École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article soumis à la revue IEEE « Transactions on Network and Service Management » le 05 Mars 2023 (Révision mineure).

5.1 Résumé

Le réseau 5G fournit des connexions à haut débit, à très faible latence et à haute fiabilité pour prendre en charge les robots mobiles sans fil avec une agilité accrue pour l’automatisation des usines. Dans cet article, nous abordons le problème du contrôle robotique en essaim pour les applications robotiques critiques dans un scénario d’entrepôt automatisé basé sur une grille. Notre objectif est de maximiser l’efficacité énergétique à long terme tout en répondant à la contrainte de consommation d’énergie des robots et aux exigences de communication ultra-fiable et à faible latence (URLLC) entre le contrôleur central et la robotique en essaim. Le problème du contrôle de la robotique en essaim dans le régime URLLC est formulé comme un problème d’optimisation non convexe puisque le taux réalisable et la probabilité d’erreur de décodage avec une longueur de bloc courte ne sont ni convexes ni concaves en bande passante et en puissance de transmission. Nous proposons une approche basée sur l’apprentissage par renforcement profond (DRL) qui utilise la méthode du gradient de politique déterministe profond (DDPG) et le réseau neuronal convolutif (CNN) pour obtenir une politique de contrôle stationnaire optimale qui consiste en un certain nombre d’actions continues et discrètes. Les résultats numériques montrent que notre algorithme DDPG multi-agent proposé surpasse les lignes de base en termes de probabilité d’erreur de décodage et d’efficacité énergétique.

Mots clés : Réseau 5G, robotique en essaim, automatisation industrielle, URLLC, apprentissage par renforcement profond

5.2 Abstract

5G network provides high-rate, ultra-low latency, and high-reliability connections in support of wireless mobile robots with increased agility for factory automation. In this paper, we address the problem of swarm robotics control for mission-critical robotic applications in an automated grid-based warehouse scenario. Our goal is to maximize long-term energy efficiency while meeting the energy consumption constraint of the robots and the ultra-reliable and low latency communication (URLLC) requirements between the central controller and the swarm robotics. The problem of swarm robotics control in the URLLC regime is formulated as a nonconvex optimization problem since the achievable rate and decoding error probability with short blocklength are neither convex nor concave in bandwidth and transmit power. We propose a deep reinforcement learning (DRL) based approach that employs the deep deterministic policy gradient (DDPG) method and convolutional neural network (CNN) to achieve a stationary optimal control policy that consists of a number of continuous and discrete actions. Numerical results show that our proposed multi-agent DDPG algorithm outperforms the baselines in terms of decoding error probability and energy efficiency.

Key words : 5G network, swarm robotics, cloud robotics, automated warehouse, industrial automation, URLLC, deep reinforcement learning

5.3 Introduction

In recent years, smart logistics have become one of the main factors in the “Fourth Industrial Revolution” or simply “Industry 4.0”. Smart logistics include the intelligent organization, planning, control, and execution of goods flow. Smart warehouse, the culmination of warehouse automation, is an important service in smart logistics. In the smart warehouse, robots play an indispensable role in attaining efficient automation solutions. Indeed, robots can replace or aid humans in all kinds of tasks, such as items picking, packing, moving, and storing Ho, Nguyen, Nguyen & Cheriet (2021b). Moreover, the improvement in motion and battery technologies helps to increase the serving time/charging time of the mobile robots ; thus, replacing human workers with the robotic system can reduce labor costs, improve warehouse working efficiency, and increase reliability Liu *et al.* (2020). The advanced sensing technologies allow the robots to obtain real-time knowledge and vision about their surrounding environment. Therefore, the robots can adapt to the change of the dynamic environment. A smart warehouse requires a group of robots, or swarm robotics to handle a mission or group of missions. For instance, Amazon has employed $45,000$ robots around 20 distribution centers worldwide in its product line Wen *et al.* (2018). The deployment of swarm robotics imposes new challenges to smart warehousing systems. For example, new control techniques are required to coordinate many kinds of resources in a smart warehouse to address various critical issues such as multi-robot path planning and multi-robot task scheduling.

Path planning is one of the fundamental problems in robotics, which aims to determine conflict-free paths between the start and target positions of robots to complete their specific tasks. Conflict-free path planning for the multi-robot systems in large dynamic environments is challenging since the robots have to optimize their path to accomplish their specific tasks in critical time constraints while simultaneously avoiding potential collision with other robots or dynamic objects in the working area Zhang, Luo, Wang, Liu & Liu (2019b). Furthermore, path planning in the dynamic environment, such as a working area with moving objects or dynamic orders and arrival item flows, requires an expensive computational cost in terms of complexity and resource utilization to obtain adaptive and effective solutions in a real-time manner.

Cloud robotics is considered a promising architecture for determining the competent solutions for path planning and task scheduling in multi-robot systems Chen *et al.* (2018). It enables robot systems to be endowed with powerful capability whilst reducing costs through cloud technologies. To efficiently utilize resources in the cloud for path planning and task scheduling, powerful and adaptive algorithms are required. Recent studies have been done with respect to path planning and task scheduling, unfortunately, the communication problem in cloud robotics has not fully been investigated so far Chen *et al.* (2019); Willms & Yang (2006); Švancara *et al.* (2019); Li *et al.* (2020a); Han & Yu (2020); Hönig *et al.* (2019); Sartoretti *et al.* (2019); Wang *et al.* (2020a); Li *et al.* (2020b); Rivière *et al.* (2020).

Indeed, communication is a non-negligible constituent in cloud robotics, that ensures the seamless connectivity and the success of information exchange between the cloud and robots, and hence, guarantees the accuracy and effectiveness of the robot's operation. Ultra-reliable and low-latency communication (URLLC) service provided by 5G wireless network is a promising technique to fulfill the stringent requirement of the communication in industrial automation, e.g. 10^{-9} packet loss probability and 99.9999% availability in motion control and mobile robot use cases 3GPP (2018). The 5G networks can provide massive connectivity and high-performance communication service in the manufacturing industry compared to the prior wireless network generations Ren *et al.* (2020a); Jayaweera *et al.* (2020); Ren *et al.* (2019a); Pan *et al.* (2019); Ren, Pan, Deng, Elkashlan & Nallanathan (2019b). The key advantage of 5G wireless networks is improving reliability, lower latency, seamless and ubiquitous connectivity at an extremely high throughput both for human and machine-type connections. However, the efficient management of wireless resources in 5G network, such as bandwidth and transmit power, is still challenging since the extreme reliability and low latency requires a new sophisticated model which is not trivial to handle by the traditional optimization approaches.

Considering the aforementioned important aspects in robotics, in this paper, we formulate the smart warehouse management problem in which path planning and task scheduling for the swarm robotics are considered simultaneously with the wireless resource allocation in the 5G network so that the ultra-reliability constraints of the communication between the swarm robotics and the central edge controller is guaranteed.

5.3.1 Prior Works

Multi-agent path finding (MAPF) have attracted a wide attention from academic and industry Liu *et al.* (2020); Švancara *et al.* (2019); Li *et al.* (2020a); Han & Yu (2020); Hönig *et al.* (2019). The authors in Chen *et al.* (2018) discussed the key technical issues in cloud robotics such as real-time and low latency, energy-efficiency on both transmission and computing aspects, intermittent connectivity, big data processing, and MAPF. From the communication perspective, the authors in Jayaweera *et al.* (2020) consider the URLLC in factory automation where transmit power and blocklength are optimized to reduce the total transmission energy consumption. In Ren *et al.* (2020a), the authors adopt massive multiple-input multiple-output (MIMO) to support the wireless transmission for industrial applications. Then, a joint optimization of the pilot and payload transmission power under two well-known beamforming schemes maximum-ratio combining (MRC) and zero-forcing (ZF) is formulated to maximize the achievable uplink data rate. The authors in Chen *et al.* (2019) consider the coordinated path planning of multi-robot in the intelligent warehouse. In Liu *et al.* (2020), the warehousing environment is partitioned into several sectors, and then the sector-level robot path is generated in the time-expended sector graph. Both centralized and decentralized methods are considered to ensure the traffic flow equilibrium among the sectors, in which collision-free paths can be found by local cooperative A* algorithm, incorporated with the conflict-based searching strategy. In Li *et al.* (2020a), a lifelong MAPF in large-scale warehouses is decomposed into a sequence of Windowed MAPF instances, and which can then be solved using a generalized Multi-Label A* algorithm to find a sequence of locations. In Hönig *et al.* (2019), the Action Dependency Graph (ADG) is introduced to capture the action-precedence relationships of a MAPF solution in warehouses.

Learning-based approaches have increasingly been used to address online path planning in dynamic environment Sartoretti *et al.* (2019); Wang *et al.* (2020a); Li *et al.* (2020b); Rivière *et al.* (2020). In Sartoretti *et al.* (2019), the authors propose a hybrid framework for decentralized MAPF that combines reinforcement learning and imitation learning from an expert centralized MAPF planner. Wang *et al.* (2020a) propose a learning-based technique that exploits environmental spatio-temporal information, thus outperforms re-planning strategies. Particularly, their local observation including the location of free cell, static obstacle, and the dynamic obstacle is transformed into three-channel RGB data, and then 3D CNN is employed to capture the spatial information. Then, LSTM is applied to the output of 3D CNN to further extract the temporal information. By allowing message-dependent attention, the authors in Li *et al.* (2020b) consider the Graph Neural Networks (GNNs) for decentralized path planning at large system scales of mobile robots. In another approach, Rivière *et al.* (2020) introduces Global-to-Local Autonomy Synthesis (GLAS), which is a combination of the centralized planning and decentralized controller, for addressing the MAPF problem. Specifically, a global planner is created in a centralized way to generate trajectories, and then deep imitation learning is used to learn an online decentralized policy that imitates the expert (i.e, the global policy) from the dataset extracted from the generated trajectories.

Deep reinforcement learning (DRL) has achieved remarkable success in handling stochastic decision-making problems and has been applied to various domains, including robotic control Luo *et al.* (2022); Bai *et al.* (2021); Tan *et al.* (2022); Han *et al.* (2022b); Karimi & Ahmadi (2021); Han *et al.* (2022a); Choi *et al.* (2022); Zhu *et al.* (2022). Robotic control is a challenging problem due to the complexity of the physical system and the need for high-dimensional control in real-time. DRL has emerged as a promising approach for robotic control due to its ability to learn control policies directly from sensory inputs without relying on explicit models of the system Mnih *et al.* (2015); Lillicrap *et al.* (2015). The advantage of RL over other machine learning techniques in robotic control is its ability to handle continuous action spaces, which are common in robotics. Many real-world control problems involve continuous actions, and this is the case in the problem addressed in this paper. DRL methods can learn continuous control policies by parameterizing the action distribution, which allows the agent to output a probability distribution over the continuous action space. Prior studies have employed DRL to design adaptive control algorithms for robots, such as federated DRL Luo *et al.* (2022) and DRL-based multi-robot adaptive formation control Bai *et al.* (2021). Other DRL-based approaches have been proposed for decentralized multi-robot exploration Tan *et al.* (2022), navigation using low-cost sensors Han *et al.* (2022b), kinematic control tasks Karimi & Ahmadi (2021), collision avoidance in robots' congested scenarios Zhu *et al.* (2022), and multi-agent reinforcement learning for cooperative multi Automated Guided Vehicles (AGVs) control in warehouse systems Choi *et al.* (2022). These studies highlight the effectiveness of DRL in developing adaptive and decentralized robotic control algorithms for various tasks, such as navigation, exploration, and collision avoidance.

In this paper, we propose a robotic control algorithm that employs both the functionalities of cloud robotics and swarm robotics. Specifically, we aim to control a group of robots or a robot swarm to perform tasks in an automated warehouse, where multi-agents DRL interact with the robots through a 5G base station, implemented either at the Edge or in the Cloud. The integration of cloud technology and swarm robotics offers a feasible avenue to develop multi-robotic systems, which are equipped with enhanced energy efficiency, real-time performance, and cost-effectiveness.

5.3.2 Motivation and Contribution

Most of the aforementioned works address the technical issues from either the communications Jayaweera *et al.* (2020); Ren *et al.* (2019a); Pan *et al.* (2019); Ren *et al.* (2020a) or path planning and task scheduling Chen *et al.* (2019); Willms & Yang (2006); Liu *et al.* (2020); Švancara *et al.* (2019); Li *et al.* (2020a); Han & Yu (2020); Hönig *et al.* (2019); Sartoretti *et al.* (2019); Wang *et al.* (2020a); Li *et al.* (2020b); Rivière *et al.* (2020) perspectives. However, these two perspectives of robotics are coupled in nature, therefore, they should not be solved separately. In 5G network, the lack of consideration of both perspectives simultaneously can result in a low performance and inefficient solution. This issue becomes critical in the new era of smart warehouse with stringent requirements on URLLC communication, energy-saving, and large-scale warehouse management. Therefore, in this paper, we propose a framework to provide both energy efficiency and spectrum efficiency in swarm robotics control by jointly managing wireless resources, task scheduling, and the motion control while satisfying the constraints of ultra-reliability and collision avoidance. To the best of our knowledge, this is the first work jointly considers these aspects in swarm robotics control.

The main contributions of this paper can be summarized as follows :

- We propose a new model for robotic swarm control in a smart warehouse combining the kinetic model of the robots and the URLLC communication model of the 5G network. By combining the motion model of the robots and the transmission model of the central controller, we formulate an optimization problem to maximize the long-term average energy efficiency of 5G network subject to the limited energy constraint of the robot swarm and the ultra-reliability constraint of the transmission.

- Leveraging deep reinforcement learning to obtain the optimal solution for the formulated nonconvex problem, we propose a multi-agent deep deterministic policy gradient (MADDPG) algorithm based on an actor-critic policy gradient method. We transform the network states into a 3D image as an input of a convolutional neural network which is implemented as actor and critic networks in our proposed model. At each time slot, a stationary optimal control policy is defined considering the current environment state through which the robot swarm can operate under the ultra-reliability constraints of the communication between the central controller and the robot swarm.
- We evaluate the performance of the proposed algorithm with extensive simulations. We compare the proposed method with four baselines, namely, deep deterministic policy gradient (DDPG) method, optimal bound, maximum transmit power, and fixed bandwidth allocation. The numerical results show our proposed algorithm outperforms the three baselines whilst approaches the performance of the optimal bound baseline.

The rest of the paper is organized as follows : Section 5.4 presents the system model and problem formulation. Section 5.5 introduces the multi-agent deep reinforcement learning approach and the proposed MADDPG-based algorithm for robot swarm control followed by Section 5.6 presents the performance evaluation of our proposed algorithm. Section 3.8 presents extensive simulation results. Finally, section 5.8 concludes the paper.

5.4 System Model and Problem Formulation

Consider a robot network in an automated grid-based warehouse in which a set of K robots is served by a central controller in a grid map as shown in Fig. 5.1. The grid resolution is set based on the size of robots $\Delta a \times \Delta a$. Each grid is considered to be a node that is either free or occupied by a robot. The robots operate in a time-slot based system. Each robot can move in four directions left, right, forward, backward.

Individually, the robot is not intelligent, it does not make decisions by itself. The robots' actions are all coordinated by the central controller. The central controller controls the communication between itself and the robotic swarm by managing the wireless resources so that the 5G constraints of ultra-reliability are satisfied. Moreover, the central controller is in charge of controlling the mobility of the robots by making decisions on the velocity and orientation of each robot. The hierarchical nature of multi-robot systems, which are often operated through supervisory control in unprotected communication channels, renders them more vulnerable to cyber-threats. To mitigate these risks, we assume in this paper that the central controller communicates with the robots over dedicated control channels.

In the warehouse, there is a picking station where the ordered items arrive to be stored. There are N ($N < K$) picking slots in the picking station, and the position of a picking slot in the grid is denoted as $\phi_n, n \in N$.

The robots will be assigned suitable slots to pick up items at the picking station and move toward the items' destinations. At the destination, the robot drops the item in a crate that is stored in a huge vertical stack at each grid. This storage system is more space-time efficient than a traditional warehouse where items are scattered around on distant shelves. The robots need to plan the shortest collision-free paths from the starting points to the target points for efficiently managing the arrival items at the picking stations.

Each robot is equipped an on-board battery and needs to be recharged after exhaustion. The robots will be recharged at the charging station. There are $M(M < K)$ charging slots in the charging station, and the position of a charging slot is denoted as $\phi_m, m \in M$. When the battery level goes down to a certain minimum threshold level, the robot will move to the charging station to recharge until the battery is full before picking a new item. The charging time of each robot is T_c. Let $E_{\max}$ and $\bar{E}_k[t]$ denote the full battery level and the available energy of the robot $k \in K$ at time t, respectively.

5.4.1 Trajectory and Task Model

The position of robot k at time t is $[x_k[t], y_k[t]]$. The mapping from the realtime location of a robot to a grid index is defined as $\Phi(x_k[t], y_k[t]) : \mathbb{R}^2 \rightarrow \mathbb{N}$. Two robots cannot occupy the

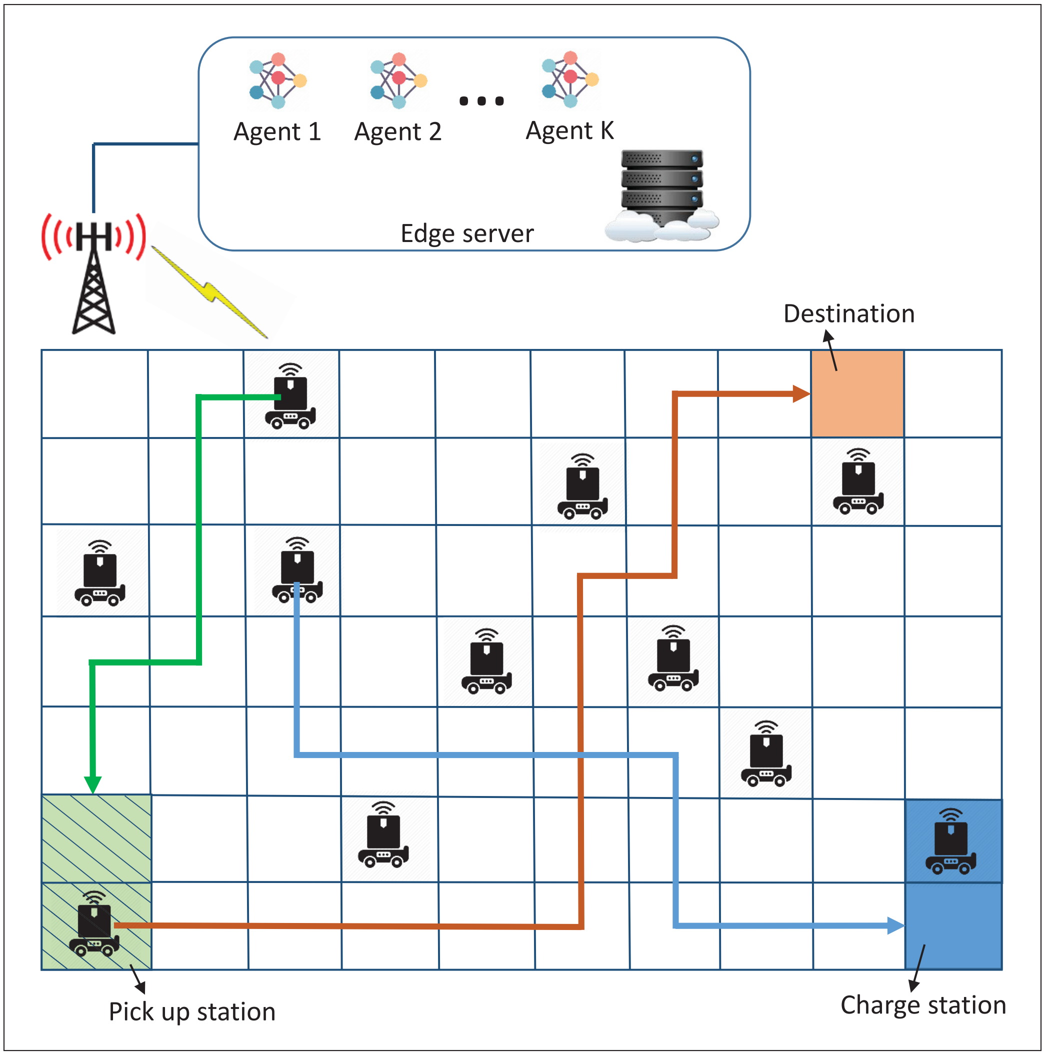


Figure 5.1 Illustration of an automated warehouse consisting of a central controller and a set of mobile robots operating on a grid-based storage system

same position at the same time. The collision avoidance constraint between any two robots at every time slot can be written as

$$\Phi(x_k[t], y_k[t]) \neq \Phi(x_{k'}[t], y_{k'}[t]), \forall k \neq k' \in K. \tag{5.1}$$

There are generally multiple decisions to be made in an intelligent warehouse system. For example, task assignment, path planning, resource allocation, etc. In the task assignment problem, the central controller assigns the task by transmitting the ordered items information to the robot, i.e., the location of the item at the picking station and its destination location. Then, the robot moves to the picking station for picking up the item and brings it to the item's destination.

Let the number of items to pick up before time t be $L^{\mathsf{re}}[t-1]$ and the arrival items at time t is $l^{\mathsf{ar}}[t]$. We define $\varpi_{k,n}$ as the task assigning variable

$$\varpi_k^n[t] = \begin{cases} 1, & \text{if the item at slot } n \text{ of the picking station is assigned to robot } k \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\phi_n^{\text{item}}[t]$ denote the destination position of the item at the picking slot n at time t. Note that, a robot is assigned task only if it is in an idle state, i.e., after completing item placement or charging. Then, the number of remaining items at the picking stations after time t is

$$L^{\mathsf{re}}[t] = L^{\mathsf{re}}[t-1] + l^{\mathsf{ar}}[t] - \sum_k \sum_n \varpi_k^n[t]. \tag{5.2}$$

We define $\vartheta_{k,m}$ as the charging slot assigning variable

$$\vartheta_k^m[t] = \begin{cases} 1, & \text{if charging slot } m \text{ is assigned to robot } k \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

5.4.2 Kinematics Model

Let $\mathbf{q}_k[t] = [x_k[t], y_k[t], \theta_k[t]]$ denote the robot k coordinates, where x_k and y_k denote the linear coordinates, and θ_k is the orientation. Robot kinematics can be represented by

$$\dot{\mathbf{q}}_k = \begin{bmatrix} \dot{x}_k[t] \\ \dot{y}_k[t] \\ \dot{\theta}_k[t] \end{bmatrix} = \begin{bmatrix} \cos\theta_k[t] & 0 \\ \sin\theta_k[t] & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_k[t] \\ w_k[t] \end{bmatrix}, \tag{5.3}$$

where v and w denote the linear and angular velocities which are used as the control inputs. The equations-of-motion of the robot are thus given by the following :

$$\begin{cases} v_k[t] = \sqrt{\dot{x}_k^2[t] + \dot{y}_k^2[t]} \\ w_k[t] = \dot{\theta}_k[t]. \end{cases} \tag{5.4}$$

5.4.3 Energy Model

5.4.3.1 Motion

The energy consumption for motion of the robot consists two main parts : the energy transformed into robot kinetic energy and the energy to overcome traction resistance. The kinetic energy of robot k can be expressed as follows : Liu & Sun (2013)

$$\begin{aligned} E_k^{\text{kinetic}} &= \frac{1}{2}m_k v_k[t]^2 + \frac{1}{2}I_k w_k[t]^2 \\ &= \int_t d\left(\frac{1}{2}m_k v_k[t]^2\right) + d\left(\frac{1}{2}I_k w_k[t]^2\right) \\ &= \int_t \left(m_k v_k[t] a_k^{\text{L}}[t] + I_k w_k[t] a_k^{\text{A}}[t]\right) dt \end{aligned} \tag{5.5}$$

where m_k and I_k denote the mass and the moment of inertia of robot k, respectively; while a_k^{L} and a_k^{A} denote the linear and angular accelerations, respectively. We assume that there is no slippage between the robot wheels and rails.

The energy consumption for traction resistance can be derived as follows Liu & Sun (2013) :

$$\begin{aligned} E_k^{\text{resistance}} &= \int_t \mu m_k g(|v_k[t] - b w_k[t]| + |v_k[t] + b w_k(t|)dt \\ &= 2\mu m_k g \int_t \max |v_k[t]|, |b w_k[t]| dt \end{aligned} \tag{5.6}$$

where g denotes gravitational acceleration, b denotes the shaft spacing of the wheels on both sides, and $m_k g$ denotes all the weight and payload of the robot. μ denotes the rolling friction coefficient which depends on the type of ground surface (i.e., rails).

5.4.3.2 Total Energy Consumption

The total energy consumption for the motion of the robot is expressed as follows :

$$E_k = E_k^{\text{kinetic}} + E_k^{\text{resistance}} + E_k^{\text{other}}, \tag{5.7}$$

where $E_k^{\text{other}} = \int_t P_s dt$ denotes energy consumption of non-mechanical components such as sensors, microcontroller, and the power consumption P_s can be modeled by a constant.

Due to the dynamic of the environment (i.e., movement of all robots at every time slot), the traveling time of a robot from its current position to the picking station, or from the picking station to the item's destination, and moving toward the charging station is unknown in advance. Therefore, the trajectory of each robot is not predefined. Instead, at each time slot, the controller will make the decision on moving direction and moving speed for each robot according to the current state of the environment.

Let E^{picking}, $E^{\text{destination}}$, and E^{charging} denote the energy consumption of a robot for moving from the current position to the picking station, for moving from picking station to the destination, and for moving from the current position to the charging station, respectively. The total energy consumption for traveling from the current position to the picking station, from the picking station to the destination of an item, and from the current position to the charging station is given as follow :

$$\begin{aligned} E_k^{\text{total}}(\varpi_k^n[t], \vartheta_k^m[t], v_k[t], w_k[t]) &= E_k^{\text{picking}} + E_k^{\text{destination}} + \mathbf{1}_{[E[t] \leq E_{\min}]}(E_k^{\text{charging}}) \\ &= \mathbf{1}_{[\bar{E}_k[t] > E_{\min}]} \sum_{n \in N} \varpi_k^n[t] \left(\sum_{t=t_0}^{\Delta T_k^{\text{p}}} E_k[t] + \sum_{t=t_0}^{\Delta T_k^{\text{d}}} E_k[t] \right) + \mathbf{1}_{[\bar{E}_k[t] \leq E_{\min}]} \left(\sum_{m \in M} \vartheta_k^m[t] \sum_{t=t_0}^{\Delta T_k^{\text{c}}} E_k[t] \right) \end{aligned} \tag{5.8}$$

where $E_k[t]$ is the energy for motion, traction resistance, and other in (5.7) over time t. ΔT_k^{p} is the number of required moving steps for the robot to travel from its current position to the picking station. ΔT_k^{d} is the number of required moving steps for the robot to travel from picking station to the destination position of the item. ΔT_k^{c} is the number of required moving steps for the robot to travel from its current position to the charing station. Here, ΔT_k^{p}, ΔT_k^{d}, and ΔT_k^{c} are not known by the controller in advance due to the dynamic of the environment. $\bar{E}_k[t] = E_{\max} - E_k^{\mathrm{total}}[t]$ is the available battery level of the robot at time t. $E_{\min}$ is the minimum energy level that requires the robot to recharge its battery.

5.4.4 Communication Model

The controller needs to transmit data packets to the robots to control the movement and operation of the robots. The packet size for robot k in the interval from time instant t to time instant $t + 1$ is denoted as D_k bits. The interval between two time instants is assumed to be small enough so that the distances between the central controller and robots in all mini-time instants in this interval can be considered as unchanged. Thus, let $d_k[t]$ be the distance between the central controller and robot k at time instant t, which is given by

$$d_k[t] = \sqrt{(H_0 - H_k)^2 + (x_0 - x_k[t])^2 + (y_0 - y_k[t])^2} \tag{5.9}$$

where H_0 and H_k are the height of the base station and the total height of the robot and the grid, respectively, and (x_0, y_0) is the central controller coordinate.

Taking into account the potential impact of various channel-related impairments and the spatial arrangement of the base station and the robots, in this paper, we consider a wireless communication system under two distinct types of fading. Firstly, large-scale fading, which accounts for path loss and shadowing effects, and secondly, small-scale fading, which pertains to the rapid fluctuations of the received signal strength over relatively short distances and time-frames Tse & Viswanath (2005). Let $h_k[\tilde{t}]$ denote the channel fading coefficients from the central controller to robot k in the mini-time instant $\tilde{t}$. Then, $h_k[t]$ can be calculated as

$$h_k[\tilde{t}] = \sqrt{\varrho_k[t]}\tilde{h}_k[\tilde{t}] \tag{5.10}$$

where $\varrho_k[t] = \varrho_0 d_k^{-\tilde{\alpha}}[t]$ accounts for the large-scale fading effects such as path loss and shadowing, ϱ_0 is the path loss at the reference distance, and $\tilde{\alpha}$ is the path loss exponent. $\tilde{h}_k[\tilde{t}]$ is generally a complex valued random variable with $\mathbb{E}[|\tilde{h}_k[\tilde{t}]|^2] = 1$ accounting for the small-scale fading Tse & Viswanath (2005).

We consider the system bandwidth W_c which is divided into multiple basic frequency band units with bandwidth W_0. Each robot is assumed to operate in different frequency band units and the total frequency bandwidth allocated to the kth robot is denoted as $W_k = \alpha_k W_0$, where α_k denotes the number of bandwidth units allocated to the kth robot. The total bandwidth allocated to all the robots should be no larger than the system bandwidth W_c.

Based on Polyanskiy *et al.* (2010), for a simple point-to-point communication system with finite blocklength $l_k = \alpha_k B_0 T_d$, a lower bound on maximum achievable data rate (bits/s) can be accurately approximated by :

$$\tilde{R}_k[\tilde{t}] \approx \alpha_k W_0 \left[\log_2\left(1 + \gamma_k[\tilde{t}]\right) - \sqrt{\frac{V}{\alpha_k W_0 T_d}} \frac{Q^{-1}(\varepsilon_k)}{\ln 2} \right], \tag{5.11}$$

where ε is decoding error probability, T_d is transmission duration. In URLLC, the transmission duration T_d is extremely small, which is shorter than the channel coherence time. $\gamma_k[\tilde{t}] = \tilde{p}_k[\tilde{t}]|h_k[\tilde{t}]|^2/\alpha_k W_0 \sigma_k^2$ denotes the signal-to-noise ratio (SNR) at the robot k, and p_k is the downlink transmit power of the BS to robot k. $Q^{-1}(\cdot)$ is the inverse function $Q(x) = \frac{1}{\sqrt{2\mu}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt$, and V is the channel dispersion Polyanskiy *et al.* (2010) given by

$$V = 1 - \frac{1}{\left(1 + \frac{\tilde{p}_k[\tilde{t}]|h_k[\tilde{t}]|^2}{\sigma_k^2}\right)^2} \tag{5.12}$$

where V can be approximated by 1 when the received SNR is higher than 5 dB. As seen in (5.11), when the blocklength l approaches infinity, the data rate R will approach $\log_2(1+\gamma)$ which is the classic Shannon capacity. The second term in (5.11) can be interpreted as a penalty on the rate in order to guarantee the decoding error probability ε for a finite blocklength l_k.

In this problem, we consider that $\tilde{p}_k[\tilde{t}] = p_k[t], \forall \tilde{t} \in [t, t+1]$, then the average rate in interval $[t, t+1]$ is given as

$$R_k[t] = \int \tilde{R}_k[\tilde{t}]\phi_{|h_k[\tilde{t}]|}(|h_k[\tilde{t}]|)d|h_k[\tilde{t}]| \tag{5.13}$$

where $\phi_{|h_k[\tilde{t}]|}(|h_k[\tilde{t}]|) = \frac{|h_k[\tilde{t}]|}{\sigma}\exp\left(-\frac{|h_k[\tilde{t}]|^2}{2\sigma}\right)$.

To transmit a packet of size D using $l_k[t] = \alpha_k[t]B_0T_d$ symbols (coding rate $D/l_k[t]$), the decoding error probability at the robot k can be obtained from (5.11) as

$$\varepsilon_k[t] = Q(f(\gamma_k[t], \alpha_k[t], D)) \tag{5.14}$$

where $f(\gamma_k[t], \alpha_k[t], D) = \ln 2\sqrt{\frac{l_k[t]}{V}}(\log_2(1+\gamma_k[t]) - \frac{D}{l_k[t]})$. The values of ε_k in (5.14) are illustrated in Fig. 5.2 where SNR γ_k is fixed. It can be seen that the decoding error probability increases with the packet size but decreases with the blocklength.

For the latency requirement of a URLLC transmission, we assume that the packet arrival process follows Poisson distribution with the arrival rate λ_k. The total latency can be calculated by the transmission delay T_{tr}, queueing delay T_q and processing delay T_{pc} Yang *et al.* (2020a) :

$$T_{Lat} = T_{tr} + T_q + T_{pc}, \tag{5.15}$$

where the transmission delay of the packet with size D can be calculated by $T_{tr} = D/R_k$, where R_k is the data rate given in (5.13) and T_q can be calculated by assuming an M/M/1 queueing model at the transmitter.

The low latency constraint of a URLLC transmission requires each packet to be successfully transmitted in a given time period denoted as $T_{\max}$. The latency constraint can be guaranteed by

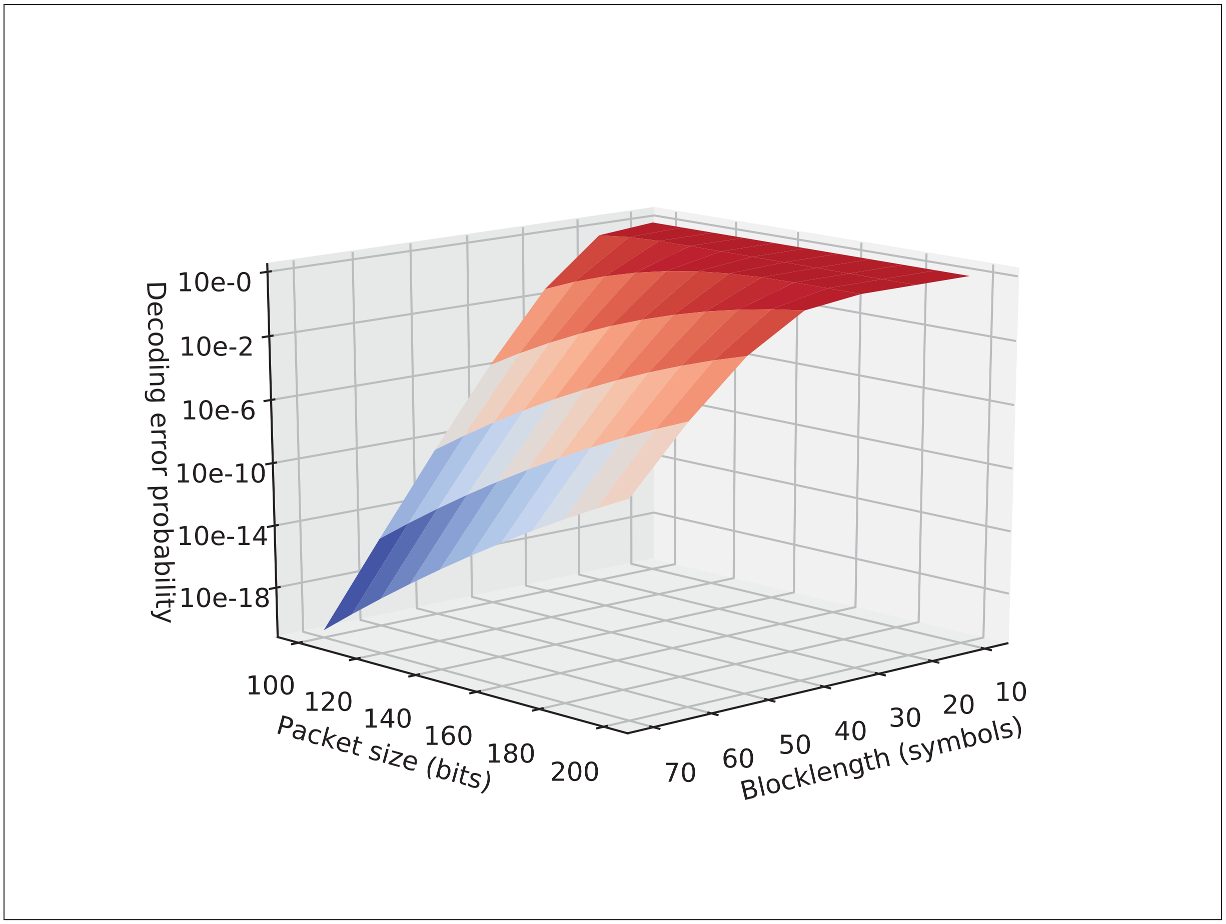


Figure 5.2 Demonstration of decoding error probability vs blocklength and packet size

controlling the latency outage probability under a certain threshold :

$$T_{Lat} \leq T_{\max} \tag{5.16}$$

5.4.5 Energy Efficiency

In this paper, the energy efficiency of the robot is conceptualized as the quotient of the average achievable URLLC rate of the robot and the total energy consumption by the robot and the controller, represented as the sum of the transmit power of the controller and the energy consumed by the robot in motion.

$$\eta[t] = \sum_{k\in K} \frac{R_k(1-\varepsilon_k[t])}{p_k[t] + E_k^{\text{total}}[t]} \tag{5.17}$$

5.4.6 Problem Formulation

Given the current positions and on-board energy of the robot swarm, we formulate the problem of maximization of long-term EE subject to the extreme reliability constraint of mission critical robotic network.

$$\max_{\substack{\boldsymbol{\varpi},\boldsymbol{\vartheta},\boldsymbol{v},\boldsymbol{w}\\ \boldsymbol{\alpha},\boldsymbol{p}}} \ \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} \mathbb{E}[\eta[t]] \tag{5.18a}$$

s.t. : (5.16)

$$E_k^{\text{total}}[t] \le E_{\max}, \forall k \in K, \tag{5.18b}$$

$$\varepsilon_k[t] \le \varepsilon_{\max}, \forall k \in K, \tag{5.18c}$$

$$\sum_{k\in K} \varpi_k^n[t] \le 1, \sum_{n\in N} \varpi_k^n[t] \le 1, \forall k \in K, \forall n \in N, \tag{5.18d}$$

$$\sum_{k\in K} \vartheta_k^m[t] \le 1, \sum_{n\in N} \vartheta_k^m[t] \le 1, \forall k \in K, \forall m \in M, \tag{5.18e}$$

$$\sum_{k\in K} \alpha_k[t] B_0 \le W_c, \tag{5.18f}$$

$$\Phi(x_k[t], y_k[t]) \ne \Phi(x_{k'}[t], y_{k'}[t]), \forall k \ne k' \in K, \tag{5.18g}$$

$$\sum_{k\in K} \varpi_k^n[t]\Phi(x_k[\Delta T_k^{\text{p}}], y_k[\Delta T_k^{\text{p}}]) = \phi_n, \forall n \in N, \tag{5.18h}$$

$$\sum_{k\in K} \varpi_k^n[t]\Phi(x_k[\Delta T_k^{\text{d}}], y_k[\Delta T_k^{\text{d}}]) = \phi_n^{\text{item}}[t], \tag{5.18i}$$

$$\sum_{k\in K} \vartheta_k^n[t]\Phi(x_k[\Delta T_k^{\text{c}}], y_k[\Delta T_k^{\text{c}}]) = \phi_m, \forall m \in M, \tag{5.18j}$$

where the energy constraint over the operation time for each robot is given in (5.18b), whereas constraint (5.18c) imposes the extremely high reliability for each robot. Constraints (5.18d) and

(5.18e) state that at most one robot is assigned at most one task at each time slot. Constraint (5.18f) restricts the total bandwidth allocated to all the robots not exceed the system bandwidth. Constraints (5.18h),(5.18i), and (5.18j) enforce the robots reaching the destination of the assigned task whereas constraint (5.18g) guarantees collision avoidance among all the robots involved in the operation.

Remark 1 :

1. The problem (5.18) is non-convex and nonlinear combinatorial because the achievable coding rate expression in (5.11) exhibits neither convexity nor concavity with respect to the transmit power, and the Q-function in (5.14) is also non-convex with respect to the transmit power and resource allocation. Therefore, conventional optimization techniques cannot solve this problem.
2. It is extremely difficult, if not impossible, to obtain the globally optimal solution to problem (3.14) with a polynomial-time algorithm. Therefore, we will focus on developing a low-complexity algorithm to compute stationary optimal solutions rather than a globally optimal solution.

5.5 DRL Based Robot Swarm Control

In this section, we propose a DRL-based algorithm to obtain the solution for the robot swarm control problem by modeling the formulated problem as a Markov Decision Process (MDP). We consider a MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma, \rho_0)$. ($\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, and $\gamma \in (0, 1)$ is the discount factor. The dynamics or transition distribution are denoted as $p(s'|s, a)$, the initial state distribution as $\rho_0(s)$, and the reward function as $r(s, a)$. The dynamics $p(s'|s, a)$ are assumed to be unknown. The goal of reinforcement learning is to find the optimal policy π^* that maximizes the expected sum of discounted rewards, denoted by

$$\pi^* = \underset{\pi}{\text{argmax}}\ \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s[t], a[t]) \right] \tag{5.19}$$

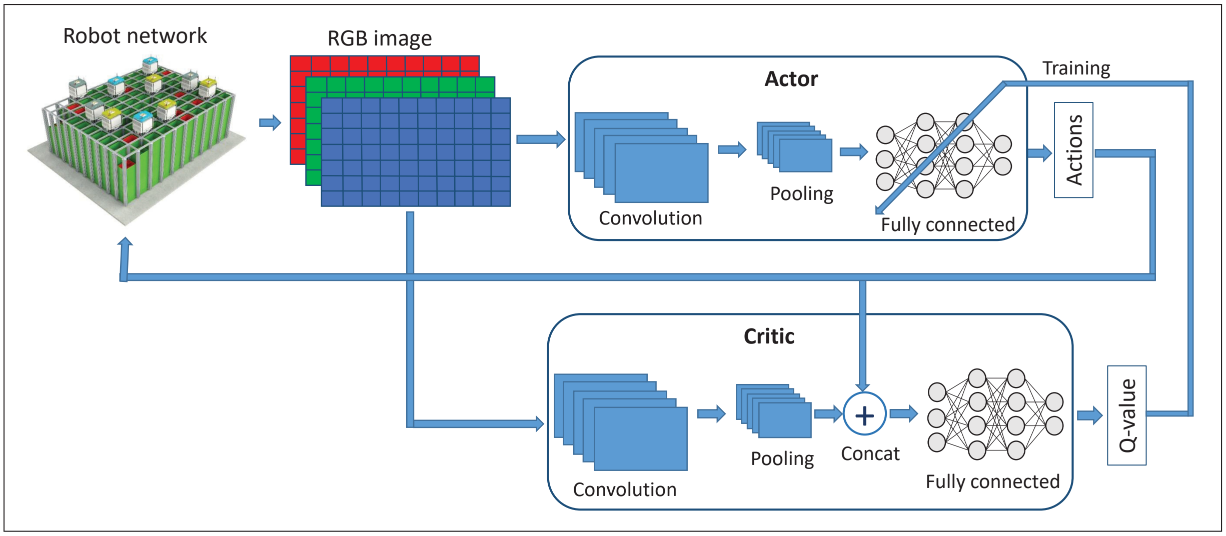


Figure 5.3 Deep deterministic policy gradient

5.5.1 Preliminaries

In Deep Q-Network (DQN), action-value function $Q(\boldsymbol{s}[t], \boldsymbol{a}[t])$ is approximated by $Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}^Q)$, i.e., $Q(\boldsymbol{s}, \boldsymbol{a}) \approx Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}^Q)$ where $\boldsymbol{\theta}^Q$ is the set of weights of the DQN which is updated in each epoch by minimizing the loss

$$L(\boldsymbol{\theta}^Q) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\in\mathcal{D}}\left[\left(y[t] - Q\big(\boldsymbol{s}[t], \boldsymbol{a}[t]|\boldsymbol{\theta}^Q\big)\right)^2\right], \tag{5.20}$$

where the update target $y[t]$ is defined as

$$y[t] = r[t] + \gamma \underset{\boldsymbol{a}\in\mathcal{A}}{\text{argmax}} Q\big(\boldsymbol{s}[t+1], \boldsymbol{a}|\boldsymbol{\theta}^Q\big), \tag{5.21}$$

and $r[t]$ presents the accumulated reward, and $\gamma \in [0, 1]$ is a time-varying learning rate. $\mathcal{D}$ presents the replay buffer of experiences that are stored throughout training.

Considering the continuous problem formulated in (3.14), the deterministic policy gradient (DPG) method Sutton, McAllester, Singh & Mansour (2000) can be employed. Assume $\mu(\boldsymbol{s})$ be the differentiable deterministic policy : $\mu(\boldsymbol{s}) : \mathcal{S} \to \mathcal{A}$. The actor-critic framework is a well-known approach to be employed for DPG that consists of two parts : an actor and a critic

Lillicrap *et al.* (2015), the actor represents the policy whereas the critic is used to estimate the value function (e.g., the Q-function). The basic idea of the actor-critic method is to maintain the parameterized actor function $\mu(\boldsymbol{s}|\boldsymbol{\theta}^{\mu})$ to derive the best action from a given state, and a critic function $Q(\boldsymbol{s}, \mathbf{a}|\boldsymbol{\theta}^{Q})$ to model the correlation between Q values and state-action pairs. $\boldsymbol{\theta}^{\mu}$ and $\boldsymbol{\theta}^{Q}$ are the weights of the actor network and critic network, respectively. The above DQN can be used to implement the critic function $Q(\boldsymbol{s}, \mathbf{a}|\boldsymbol{\theta}^{Q})$. The actor network can be updated using the gradient of the policy $\mu(\boldsymbol{s}|\boldsymbol{\theta}^{\mu})$ with respect to the actor parameters $\boldsymbol{\theta}^{\mu}$:

5.5.2 State and Action Space

5.5.2.1 State Space

As formulated in Section 5.4.4, the decoding error probability is related to the channel fading coefficients between the central controller and robot swarm. The channel fading coefficient can be computed based on the distance between the robot and the central controller. Moreover, the robot can only perform its assigned task if its onboard energy is available. As a result, the network state includes the robots' energy level, which provides information to the DRL agent in making decisions on task assignments.

The network state at time slot t can be characterized by :

- Current location of the robots $\mathbf{B}[t] = \{[x_k[t], y_k[t]]|\forall k \in K\}$
- The distance between the robot swarm and the BS at time t, $\mathbf{D}[t] = \{d_k[t]|\forall k \in K\}$.
- The available on-board energy of all robots at time t, $\bar{\mathbf{E}}[t] = \{\bar{E}_k|\forall k \in K\}$.
- The number of remaining items at the picking stations after time instant t, $L^{\text{re}}[t]$

The state vector can be described as $\boldsymbol{s}[t] = (\mathbf{B}[t], \mathbf{D}[t], \bar{\mathbf{E}}[t], L^{\text{re}}[t]) \in \mathcal{S}$. Since the grid map is not composed only of the grids occupied by the robots but also of the free grids that the robots can move in, we transform the network state $\boldsymbol{s}_1[t] = (\mathbf{B}[t], \mathbf{D}[t], \bar{\mathbf{E}}[t])$ into three channels graphic (RGB data) to compose the current input of the DRL model (Fig. 5.3).

It should be noted that each robot's action involves not only the direction of movement based on the occupation state of the grid, but also task scheduling and resource allocation to ensure reliable transmission. This, in turn, depends on the robot's energy level and distance to the controller. To address this, we have transformed the network state into a tensor of three channels that represent the global system state as an image-like tensor. By using this representation, we can utilize convolutional neural networks (CNNs), which have proven to be effective in image classification tasks. Each channel contains information specific to agents and the environment, and the features extracted by the CNN from the channels enable the agent to make accurate decisions. The three channels can be described as follows :

- Red channel : contains the occupation state of the grid (i.e., occupied by a robot or a free grid)
- Green channel : contains the energy level of the robot at the grid
- Blue channel : contains the distance between the robot and the central controller.

5.5.2.2 Action Space

The action $a[t]$ of the robotic network consists of three parts :

- Task assignment : $\{\boldsymbol{\varpi}[t], \boldsymbol{\vartheta}[t]\}$ If a robot is in idle state, it will be assigned to an item at the picking station, or if the robot's energy is exhaust, it will have to move to the charging station. The task assignment should satisfy the constraints (5.18d) and (5.18e).
- Robot movement : $\{\boldsymbol{v}[t], \boldsymbol{w}[t]\}$ Each robot can move to its adjacent grids which are not occupied by any other robot. If the integrated sensors detect a potential collision due to the other robot moves toward the same grid, the robot will stop its move. The linear and angular velocities that control each robot should not exceed the maximum values.
- Resource allocation : $\{\boldsymbol{\alpha}[t], \boldsymbol{p}[t]\}$ The central controller allocates bandwidth and power to transmit a packet to each robot such that the URLLC requirements are satisfied, i.e., constraint (3.14d). The total bandwidth allocated to all the robots should not exceed the system bandwidth W_c as stated in constraint (5.18f)

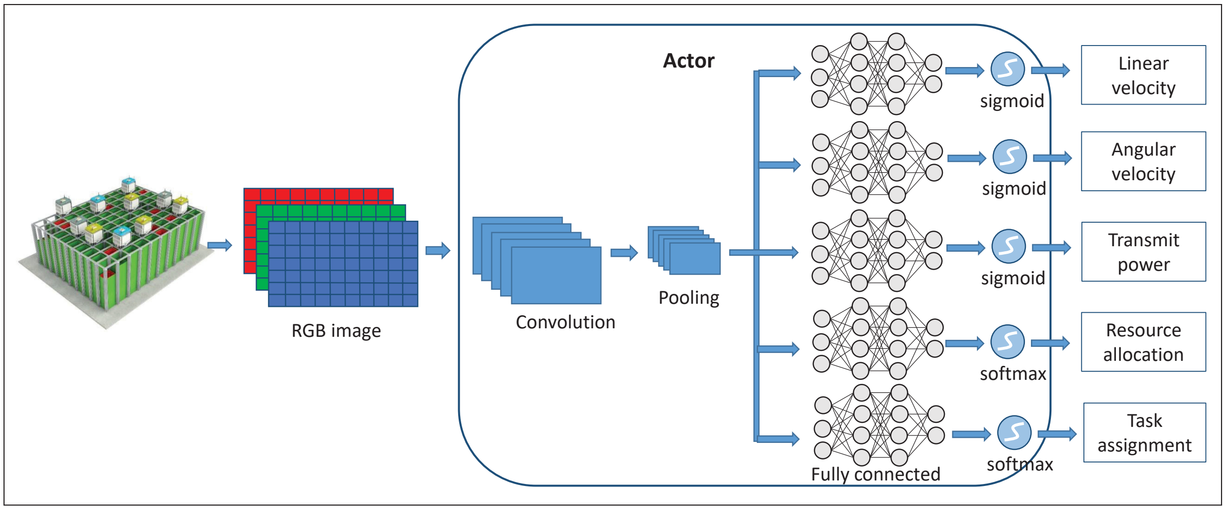


Figure 5.4 Single-agent DDPG with multiple outputs CNN, each output is a vector of K elements (actions)

5.5.3 Reward Design

In reinforcement learning, the reward should be related to the objective function. Our reward design aims to two targets : minimizing the average long-term transmission decoding error probability and helping the robots accomplish their tasks.

5.5.3.1 Reach Destination Reward

A straightforward idea to encourage the robot to move toward its current task's destination is to add a positive reward when the robot reaches its destination. However, these reward values are very sparse which means the robot would be rewarded if and only if it reaches the destination. To accomplish a task, each robot can move thousands of steps. Therefore, there is only one signal indicating whether a robot reaches the destination every thousand time slots. Moreover, the initial policy is randomly generated, and the robots would arrive at their assigned task's destination with a probability of almost zero. Consequently, the agent could be difficult to learn through earning reach-destination rewards. In order to deal with the sparse reach-destination reward issue, we define a reward component that encourages the robot k moving toward the destination as follow

$$r_k^{\text{des}}[t] = \begin{cases} \frac{\kappa_1^{\text{des}}}{(d_k^{\text{des}}[t])^{\epsilon_1^{\text{des}}}}, \text{ if } 0 < d_k^{\text{des}}[t] \leq d_{\max} \\ \kappa_2^{\text{des}} d_k^{\text{des}}[t], \text{ if } d_k^{\text{des}}[t] > d_{\max} \\ \kappa_3^{\text{des}}, \qquad \text{if } d_k^{\text{des}}[t] = 0 \\ \kappa_4^{\text{des}}, \qquad \text{if robot reaches maximum moving step} \end{cases} \tag{5.22}$$

where $d_k^{\text{des}}[t]$ denotes the distance between robot k's current location and its destination at time slot t, κ_1^{des} and κ_3^{des} are the positive reward constant coefficients, κ_2^{des} is a negative reward constant coefficient to prevent the robot moves far away from its destination position, whereas κ_4^{des} is a negative constant to punish nonarrival . ϵ_1^{des} is a positive coefficient to express the impact level of the distance $d_k^{\text{des}}[t]$ to the reward. When the robot moves in a direction toward its destination that makes the distance shorter, the reward becomes higher. On the other hand, if the distance between the robot and its destination larger than a maximum value, it will receive a negative reward proportional to that distance.

5.5.3.2 Penalty Reward

We set penalty rewards to punish the actions that violate the constraints

$$r_k^{\text{pen}}[t] = \begin{cases} \kappa_1^{\text{pen}}, \text{ if } \varepsilon_k[t] > \varepsilon_{\max} \\ \kappa_2^{\text{pen}}, \text{ if } \Phi(x_k[t], y_k[t]) = \Phi(x_{k'}[t], y_{k'}[t]) \\ \kappa_3^{\text{pen}}, \text{ if } T_{Lat}[t] > T_{\max} \end{cases} \tag{5.23}$$

where κ_1^{pen}, κ_2^{pen} and κ_2^{pen} are negative constants specifying penalty rewards for the violation of reliability constraint (3.14c), collision constraint (5.18g), and latency constraint (5.16), respectively.

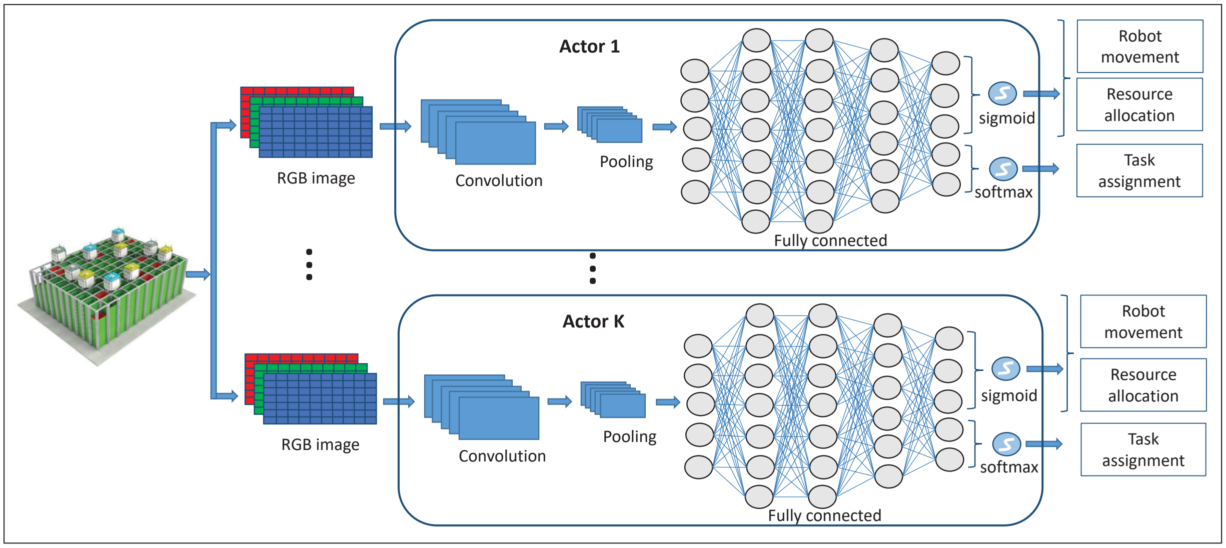


Figure 5.5 Multi-agent DDPG with multiple outputs convolutional neural network (CNN)

In summary, the reward of robot k at time t can be formulated as

$$r_k[t] = r_k^{\text{des}}[t] + r_k^{\text{pen}}[t] \tag{5.24}$$

5.5.4 Multi-Agent Deep Deterministic Policy Gradient Algorithm

To achieve the design goal, a multi-agent DDPG (MADDPG) actor-critic architecture has been adopted as shown in Fig 5.3. In this section, we propose to use an extension of actor-critic policy gradient methods where the critic is augmented with extra information about the policies of all other agents.

Specifically, a set of actors $\boldsymbol{\mu} = \{\mu_1(\boldsymbol{o}_1|\boldsymbol{\theta}_1^{\boldsymbol{\mu}}), ..., \mu_k(\boldsymbol{o}_k|\boldsymbol{\theta}_k^{\boldsymbol{\mu}}), ..., \mu_M(\boldsymbol{o}_M|\boldsymbol{\theta}_M^{\boldsymbol{\mu}})\}$ of all the K robots are considered. Then, the gradient of the expected return for agent k, i.e., robot k, is given by

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}_k^{\mu}} J]\mu(\boldsymbol{\theta}_m^{\boldsymbol{\mu}}) \\ &\approx \mathbb{E}_{(\boldsymbol{s},\mathbf{a},\boldsymbol{E},\boldsymbol{s}')\in\mathcal{D}}\left[\nabla_{\boldsymbol{\theta}_k^{\mu}} Q_k(\boldsymbol{s}[t], \mathbf{a}[t]|\boldsymbol{\theta}_k^{\boldsymbol{Q}})|_{\boldsymbol{a}_k[t]=\mu_k(\boldsymbol{o}_k[t]|\boldsymbol{\theta}_k^{\mu})}\right] \\ &= \mathbb{E}_{(\boldsymbol{s},\mathbf{a},\boldsymbol{E},\boldsymbol{s}')\in\mathcal{D}}\left[\nabla_{\boldsymbol{a}_k} Q_k(\boldsymbol{s}[t], \mathbf{a}[t]|\boldsymbol{\theta}_k^{\boldsymbol{Q}})|_{\boldsymbol{a}_k[t]=\mu_k(\boldsymbol{o}_k[t]|\boldsymbol{\theta}_k^{\mu})} \times \nabla_{\boldsymbol{\theta}_k^{\mu}} \mu_k(\boldsymbol{o}_k[t]|\boldsymbol{\theta}_k^{\mu})\right], \end{aligned} \tag{5.25}$$

where $Q_k(\boldsymbol{s}[t], \mathbf{a}[t]|\boldsymbol{\theta}_k^{\boldsymbol{Q}})$ represents the critic function that takes as input the network state $\boldsymbol{s}$, the actions of all agents, i.e., $\mathbf{a}[t] = \{\boldsymbol{a}_1[t], \boldsymbol{a}_2[t], ..., \boldsymbol{a}_M[t]\} \in \mathcal{A}$, and outputs the Q-value for agent k, i.e., robot k. The critic function of the agent k, i.e., $Q_k(\boldsymbol{s}[t], \mathbf{a}[t]|\boldsymbol{\theta}_k^{\boldsymbol{Q}})$, can be updated by minimizing the loss function as follows

$$\mathcal{L}(\boldsymbol{\theta}_k^{\boldsymbol{Q}}) = \mathbb{E}_{(\boldsymbol{s},\mathbf{a},\boldsymbol{E},\boldsymbol{s}')\in\mathcal{D}}\left[\left(y_k[t] - Q_k\left(\boldsymbol{s}[t], \mathbf{a}[t]|\boldsymbol{\theta}_k^{\boldsymbol{Q}}\right)\right)^2\right], \tag{5.26}$$

$$y_k[t] = r_k[t] + \gamma Q'_k\left(\boldsymbol{s}[t+1], \mathbf{a}[t+1])|\boldsymbol{\theta}_k^{\boldsymbol{Q}'}\right), \tag{5.27}$$

where $(\boldsymbol{s}, \mathbf{a}, \boldsymbol{E}, \boldsymbol{s}') \in \mathcal{D}$ represents the sample from the experience replay memory $\mathcal{D}$ recording experiences of all agents. $r_k[t]$ denotes the accumulated reward of robot k. μ'_k and $Q'_k()$ represent the actor and critic target networks of agent k with the weights $\boldsymbol{\theta}_k^{\boldsymbol{\mu}'}$ and $\boldsymbol{\theta}_k^{\boldsymbol{Q}'}$, respectively.

The proposed algorithm for multi-agent robot swarm control using MADDPG is presented in Algorithm 1.

5.5.5 Network Architecture

The actor network of each agent takes the 3D image as input and outputs the action. We utilize a convolutional neural network (CNN) to extract features from the input tensor and to output the action by a fully connected (FC) layer (Fig. 5.5). The actor network is comprised of two 2D convolutional layers followed by a FC layer. This layer is designed as shown in Fig. 5.5. The FC layer has a number of hidden layers and an output layer with different activation functions for each type of control action, i.e., continuous and discrete actions. We use the activation function sigmoid to output the control actions of robot movement and resource allocation which is continuous actions. In the field of continuous control policy gradient reinforcement learning, where actions are taken continuously within a range (a, b), a probability distribution is generally defined and utilized to select actions, with each action being associated with a probability density. The output layers of the network parameterize this probability distribution, thus making it possible to use activation functions like sigmoid, tanh, or other bounded functions.

The bandwidth allocation variable is relaxed to be a continuous variable. Moreover, for the task assignment action which is a discrete action, we adopt the activation function softmax. The resulting output vector represents a probability distribution of the predicted actions, indicating that the actions remain continuous. Subsequently, the binary action is chosen based on this probability during the interaction with the environment.

The critic network takes the 3D image state and action from the actor network as the input, and outputs the action-value function $Q_k(\boldsymbol{s}[t], \mathbf{a}[t]|\boldsymbol{\theta}_k^{\boldsymbol{Q}})$. The convolutional layers of the critic network is similar to the one in actor network. However, the input of the FC layer in the critic network is concatenated with the action which is the output of the actor network (Fig. 5.3).

5.6 Performance Evaluation

5.6.1 Optimal Bound Resource Allocation

Given the location of each robot, the SNR threshold for guaranteeing the decoding error probability when transmitting a packet of size D can be derived from (5.11) as follows

$$\gamma_k^{th} = \exp\left[\frac{D \ln 2}{\alpha_k W_0 T_d} + \frac{Q^{-1}(\varepsilon_k)}{\sqrt{\alpha_k W_0 T_d}}\right] - 1, \tag{5.28}$$

Algorithme 5.1 Multi-Agent Robot Swarm Control

```
Initialize critic network $Q_m(\boldsymbol{s}, \mathbf{a}|\boldsymbol{\theta}_m^{\boldsymbol{Q}})$, and actor network $\mu_m(\boldsymbol{o}_m|\boldsymbol{\theta}_m^{\boldsymbol{\mu}})$, $\forall m \in \mathcal{M}$;
Initialize target network $Q'_m$ and $\mu'_m$ with weights $\boldsymbol{\theta}_m^{\boldsymbol{Q}'} \leftarrow \boldsymbol{\theta}_m^{\boldsymbol{Q}}$ and $\boldsymbol{\theta}_m^{\boldsymbol{\mu}'} \leftarrow \boldsymbol{\theta}_m^{\boldsymbol{\mu}}$, $\forall m \in \mathcal{M}$;
Initialize replay memory $\mathcal{D}$ and the mini-batch $\tilde{\mathcal{D}}$;
while True do
    for agent k=1 to K do
        Base on global state $\boldsymbol{s}[t]$, agent $k$ chooses an action $\boldsymbol{a}_k[t] = \mu_m(\boldsymbol{s}[t]|\boldsymbol{\theta}_k^{\boldsymbol{\mu}}) + \epsilon$;
          The central controller transmits control packet to robot $k$ with $p_k[t]$ and $\alpha_k[t]$
          according to $\boldsymbol{a}_k[t]$;
        Robot $k$ moves to nearby grid with $v_k[t]$, $w_k[t]$ and $\varpi_k^n[t]$, $\vartheta_k^n[t]$ according to
          $\boldsymbol{a}_k[t]$;
        Sample a random mini-batch $\tilde{\mathcal{D}}[t] \subseteq \mathcal{D}$;
        Update critic $\boldsymbol{\theta}_m^{\boldsymbol{Q}}$ by minimizing the loss in (5.26);
        Update actor $\boldsymbol{\theta}_m^{\boldsymbol{\mu}}$ using gradient (5.25);
        $\boldsymbol{\theta}_k^{\boldsymbol{Q}'} \leftarrow \kappa\boldsymbol{\theta}_k^{\boldsymbol{Q}} + (1-\kappa)\boldsymbol{\theta}_k^{\boldsymbol{Q}'}$;
        $\boldsymbol{\theta}_k^{\boldsymbol{\mu}'} \leftarrow \kappa\boldsymbol{\theta}_k^{\boldsymbol{\mu}} + (1-\kappa)\boldsymbol{\theta}_k^{\boldsymbol{\mu}'}$;
    end for
    Central controller observe network state $\boldsymbol{s}[t+1]$;
    Store $(\boldsymbol{s}[t], \mathbf{a}[t], \boldsymbol{r}[t], \boldsymbol{s}[t+1])$ in $\mathcal{D}$;
    Update $t \leftarrow t+1$
end while
```

where $V \approx 1$ is applied. Then, we can obtain the expression of the transmit power threshold as follows

$$p_k = \frac{\alpha_k W_0 \sigma_k^2 \gamma_k^{th}}{|h_k|^2}, \tag{5.29}$$

which is a function of bandwidth allocation α_k

$$\begin{aligned} & f(\alpha_k W_0) = \\ & \frac{\alpha_k W_0 \sigma_k^2}{|h_k|^2} \left\{ \exp\left[\frac{D \ln 2}{\alpha_k W_0 T_d} + \frac{Q^{-1}(\varepsilon_k)}{\sqrt{\alpha_k W_0 T_d}} \right] - 1 \right\}. \end{aligned} \tag{5.30}$$

Function $f(\alpha_k W_0)$ is non-convex with respect to bandwidth allocation variable α_k. From constraint (5.18f), the optimal bandwidth allocation policy satisfies

$$f(\alpha_k W_0) \leq p_{\max}, \tag{5.31}$$

Property 1 : $f(\alpha_k W_0)$ *is strictly concave in* α_k *when* $0 < \alpha_k \leq \alpha_k^*$, *where* α_k^* *is the unique solution that minimize* $f(\alpha_k W_0)$

Proof : Please refer to Appendix A in Sun *et al.* (2018).

5.6.2 Computational Complexity

The computational complexity of the proposed algorithm can be calculated based on the complexity of training the adopted CNNs. The complexity of a CNN is a linear summation of all convolutional layers and fully connected layers. The total complexity of all convolutional layers is $O(\sum_{l=1}^{d} n_{l-1}s_l^2 n_l m_l^2)$, where l is the index of the convolutional layer, and d is the depth (number of convolutional layers). n_l is the number of filters (also known as 'width') in the l-th layer. n_{l-1} is also known as the number of input channels of the l-th layer. s_l is the spatial size (length) of the filter, m_l is the spatial size of the output feature map. This time complexity applies to both training and testing time.

The number of multiplications through the fully connected layer i with L layers is given by $n_i = I_i h_1 + \sum_{l=1}^{L-1} h_l h_{l+1}$, where I_i is the size of the input layer of the fc i, and h_l is the number of channels of the l-th hidden layer of the fc i. The complexity of each fc is $O(n_i)$. In the proposed DDPG-CNN algorithm, we use five fc layers to predict five different outputs of the network, each with different input and output dimension. Therefore, the complexity of the feed-forward propagation and back-propagation for one sample is $O(\sum_{i=1}^{5} n_i)$, where i is the index of fc i-th as shown in Fig. 5.4. The total complexity of the actor network in DDPG-CNN algorithm is $O\big(\sum_{l=1}^{d} n_{l-1}s_l^2 n_l m_l^2 + \sum_{i=1}^{5}(I_i h_1 + \sum_{l=1}^{L-1} h_l h_{l+1})\big)$. A similar complexity is applied for the critic network.

In the proposed MADDPG-CNN, each agent's actor network includes a number of convolutional layers and a single fc. The total complexity of all actor networks in MADDPG-CNN algorithm is $O\big(\sum_{k=1}^{K}\big(\sum_{l=1}^{d} n_{l-1}s_l^2 n_l m_l^2 + Ih_1 + \sum_{l=1}^{L-1} h_l h_{l+1}\big)\big)$, where I is the size of the input layer of the fc in each actor network of each agent.

Tableau 5.1 Simulation parameters

Parameter	**Value**
Available bandwidth W_c	[1-6] MHz
Maximum transmit power	50 mW
BS height	10 m
Pathloss exponent	2.3
Noise power spectral density	-173 dBm/Hz
Decoding error probability requirement	10^{-9}
Latency requirement	1 ms
Transmission delay duration	100 μs
Maximum speed of robot	0.5 m/s
Robot acceleration	0.3 m/s^2
Capacity of the on-board battery	200 Wh
Robot mass	9 kg
Robot size	[0.5×0.5] m
Item mass	[1-5] kg
Grid size	[20×20]×0.5 m
Rolling friction coefficient	0.051
Moment of inertia	0.16245 kgm^2
Time slot	100 ms

5.7 Simulation Results

5.7.1 Simulation Setting

In this section, we present the simulations to evaluate the performance of our proposed design. We consider a grid-based warehouse scenario with one central controller (i.e, 5G BS) which controls a set of [2-20] robots. The adopted system parameters are given in Table 5.1.

The proposed MADDPG-CNN algorithm (denote as 'MADDPG-CNN') parameters are provided in Table 5.2. To implement this algorithm, we employ the powerful open-source software library, i.e., Tensorflow which is developed by the Google Brain team. In the simulation, we continue fine-tuned hyperparameters manually such as the number of hidden layers, the number of hidden units, learning rate, etc, until our model achieves a good performance.

For comparisons, six schemes are simulated as baselines as follows :

Tableau 5.2 CNN's hyperparameters setting

Parameter	**Value**
Actor first conv2D layer	[4, 4, 3, 16]
Actor second conv2D layer	[4, 4, 16, 32]
Actor FC input layer	[3 ×3 × 32, 256]
Actor FC hidden layers	[256×128]
Actor FC output layer	[128×4]
Actor FC output layer	[128×2]
Critic first conv2D layer	[4, 4, 3, 16]
Critic second conv2D layer	[4, 4, 16, 32]
Critic FC input layer	[3 ×3 × 32 + 4 +2, 256]
Critic FC hidden layers	[256×128]
Critic FC output layer	[128×1]
Number of color channels	3
Replay memory	500000
Batch size	256
Learning rate actor	5e-4
Learning rate critic	1e-3

1. 'DDPG-CNN' : Our proposed single-agent DDPG presented in Fig. 5.4.
2. 'MBPO' : Model-Based Policy Optimization presented in Janner, Fu, Zhang & Levine (2019).
3. 'DQN' : Deep Q-Network as robotic control benchmark Mnih *et al.* (2015); Lv, Zhang, Ding & Wang (2019); Zhang, Sun, Barth & Ma (2020); Han, Liu, Sun & Zhang (2019); Karimi & Ahmadi (2021).
4. 'Optimal' : This baseline is the optimal bandwidth allocation that minimize (5.30) and satisfies (5.31).
5. 'Max power' : In this baseline, the central controller transmits with maximum power.
6. 'Fix BW' : In this baseline, the central controller transmits with fixed bandwidth allocation to the robots, i.e., equal bandwidth allocation.

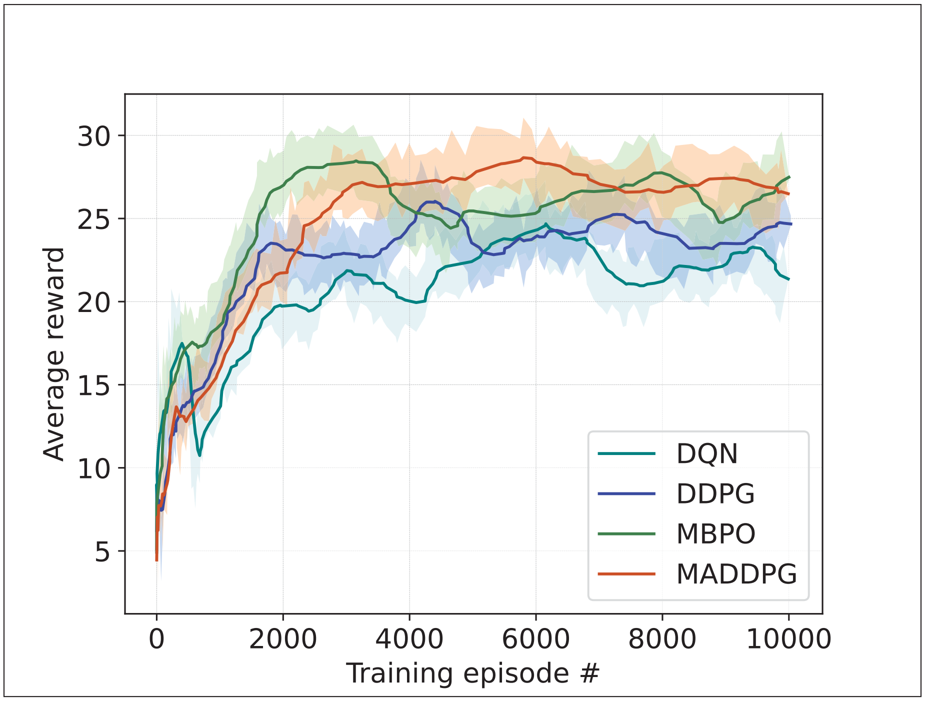


Figure 5.6 Average reward

5.7.2 Results Analysis

Fig. 5.6 displays the training curves for the accumulative reward of the four DRL-based approaches considered in this study. Our simulation results indicate that while our proposed algorithm ‘MADDPG’ exhibits comparable asymptotic performance to the ‘MBPO’ algorithm, and outperforms the ‘DQN’ algorithm by a significant margin. It is worth noting that, the ‘DQN’ algorithm is designed for discrete action spaces, while our task involves continuous control action. To address this, we utilize the ‘DDPG’ algorithm for the continuous control action. While ‘DQN’ can perform well on certain discrete tasks, it falls short when it comes to complex robotic control tasks, such as controlling a simulated robot arm Lillicrap *et al.* (2015). In the ‘DDPG-CNN’ scheme, a single agent is in charge of making decisions for all robots ; thus, a large number of hyperparameters is required to achieve a good decision for all robots in the smart warehouse. On the other hand, in the ‘MADDPG-CNN’ scheme, multiple agents are considered, and each agent makes the decision to control its corresponding robot. Moreover, in our scenario, all agents share their observations as well as their actions, which represents a cooperative scheme.

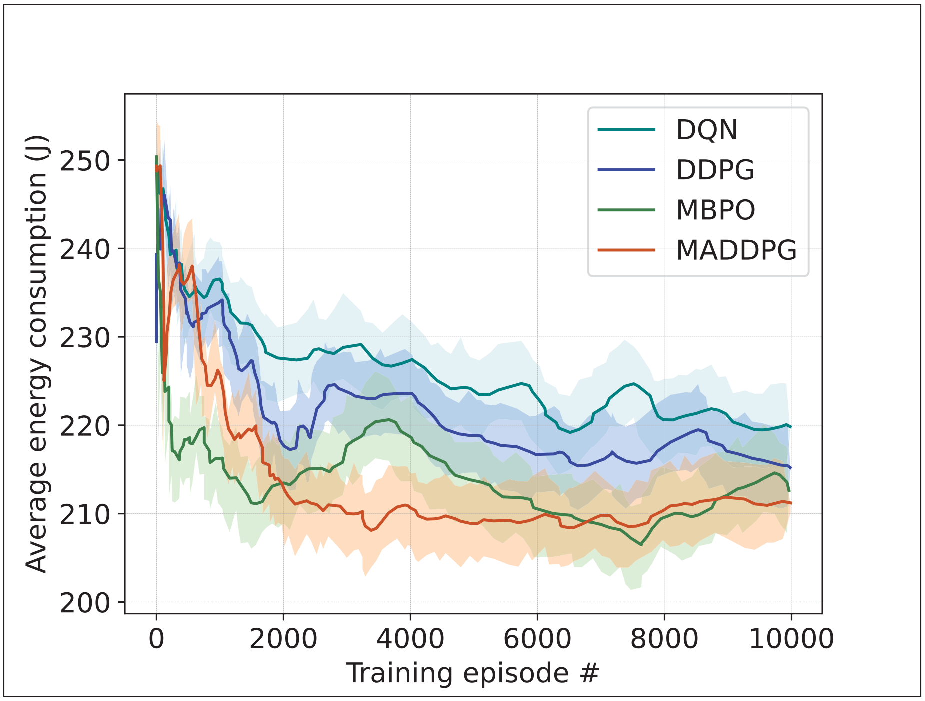


Figure 5.7 Average energy consumption

Therefore, each agent in the 'MADDPG-CNN' scheme can be effectively trained with a smaller number of hyperparameters than that of 'DDPG-CNN'. Thus, the combination of high-accurate predicting agents in the 'MADDPG-CNN' scheme returns better results as we can see in Fig. 5.6. Moreover, the

Fig. 5.7 depicts the average energy of robots consumed in four DRL-based approaches. It can be observed that while all DRL-based algorithms show their effectiveness in decreasing the energy consumption of robots, the 'MADDPG' and 'MBPO' algorithms exhibit lower energy consumption than 'DDPG' and 'DQN'. This outcome can be attributed to their reward function, which has been designed to incentivize the robots to follow efficient paths, consequently reducing energy consumption during movement.

Fig. 5.8 illustrates the energy consumption of a robot over two tasks (with more than 1000 moving steps). As seen a 9 kg robot when a robot moving with no load (i.e., moving to picking

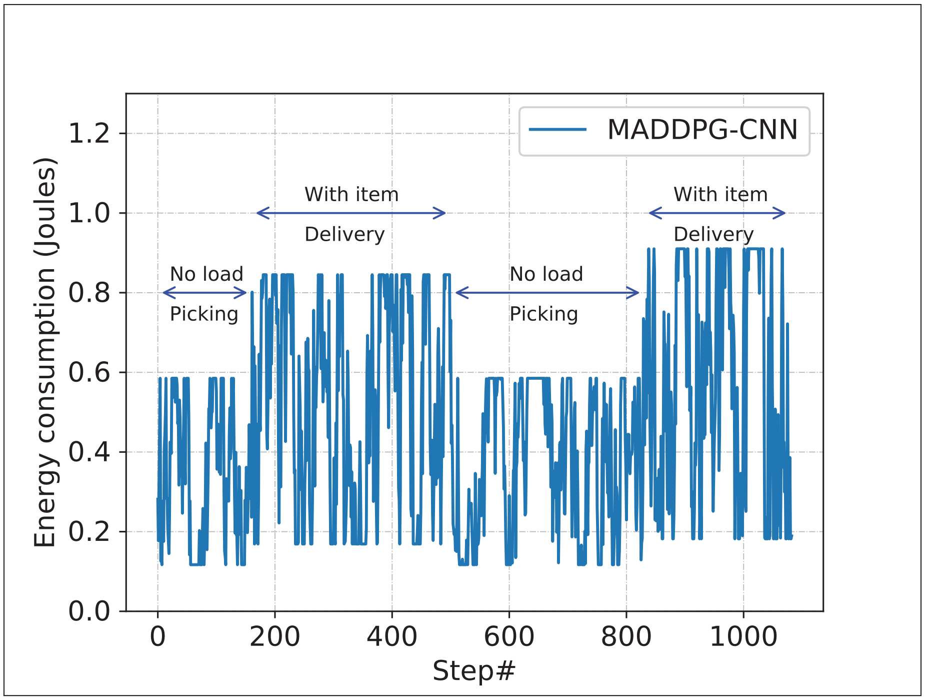


Figure 5.8 Energy consumption of a robot weighting 9kg in two different tasks

station) consumes around 0.6 J per moving step. When carrying an item, the robot consumes more energy depending on the mass of the item.

We next vary the number of robots, the available bandwidth, the packet length, and the transmission duration delay to compare the performance in terms of decoding error probability between our proposed 'MADDPG-CNN' algorithm and the 'DDPG-CNN' baseline. Fig. 5.9 depicts the decoding error probability of the two schemes with a different number of robots and different system bandwidth. It can be observed that the proposed 'MADDPG-CNN' algorithm can achieve a lower decoding error probability compared to the baseline 'DDPG-CNN' in most cases. Moreover, the decoding error probability increases when the number of robots increases in both schemes. However, to assure the ultra-reliability constraint the system bandwidth can be increased when the number of robots increases. This is because when the system bandwidth increases, the decoding error probability decreases so that the minimum decoding error probability threshold can be satisfied.

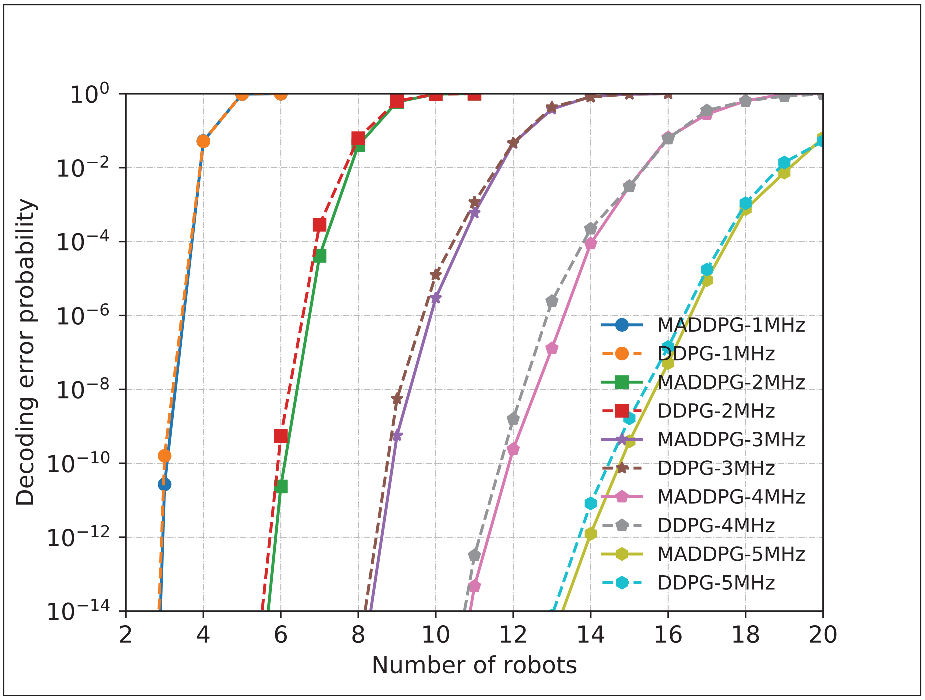


Figure 5.9 Decoding error probability vs number of robot with different system bandwidth and transmission duration $T_d = 100\mu s$, packet size $D = 120$ bits

Fig. 5.10 shows the decoding error probability when we vary the number of robots with a different transmission duration of T_d. Similar curves can be seen as in Fig. 5.9. However, an interesting observation about the trade-off between latency and reliability can be seen in Fig. 5.10. Indeed, higher latency can result in a low decoding error probability and vice versa. The decision of high reliability and low latency depends on the type of application. In our model, the operation of the robots is critical, therefore, the transmission between the central controller and the robots requires rather high reliability than low latency. A careful setting in the transmission duration and decoding error probability thresholds should be considered such that the system can effectively achieve the targeted latency and reliability.

Fig. 5.11 shows the decoding error probability when we vary the packet size. As we can see in Fig.5.2, the decoding error probability increases when the packet size increases. The results in Fig. 5.11 again confirm this correlation between the decoding error probability and the packet

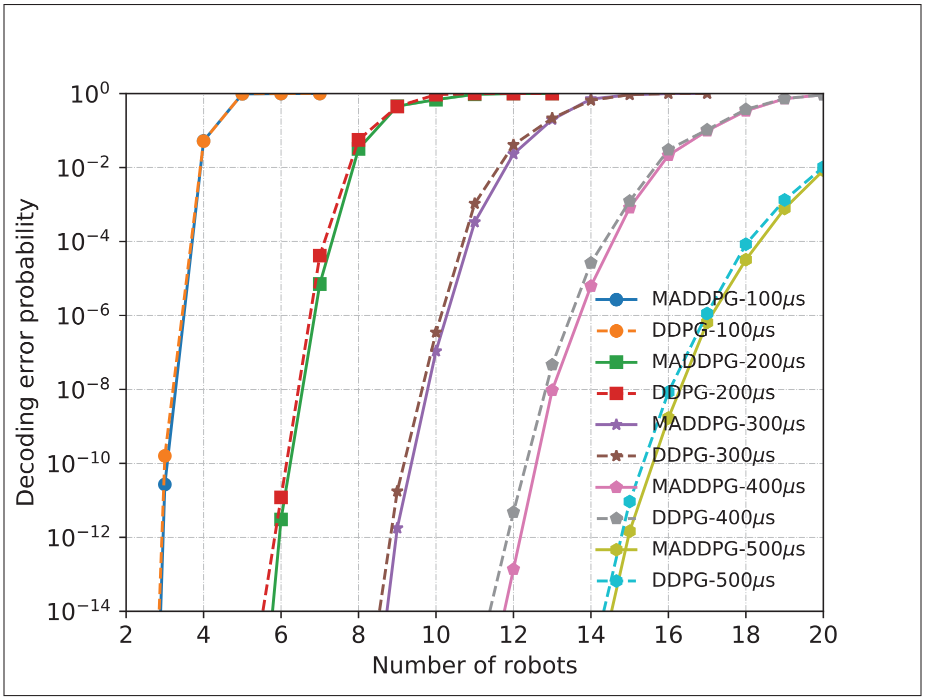


Figure 5.10 Decoding error probability vs number of robot with different transmission duration T_d, system bandwidth $W_c = 1$ MHz, packet size $D = 120$ bits

size. In the URLLC regime, it is important to transmit a short packet size to guarantee the reliability constraint.

Fig. 5.12 shows the energy efficiency with different values of bandwidth when the packet size $D = 120$bits. As shown, the energy efficiency of all schemes decreases when the system bandwidth increases. However, the ‘MADDPG-CNN’ scheme is better than the ‘DDPG-CNN’ baseline and close to the ‘Optimal’ baseline. For example, when the system bandwidth is 3 MHz, the energy efficiency of the ‘MADDPG-CNN’ scheme is 16.9 bits/Hz/Joules compared to 15.3 bits/Hz/Joules of the ‘DDPG-CNN’ baseline, which represents a 10.45% improvement. Furthermore, in a high energy efficient region, i.e. at the points with system bandwidth $W_c = 1, 2, 3$ MHz, the ‘MADDPG-CNN’ and ‘DDPG-CNN’ schemes significantly outperform both ‘Max power’ and ‘Fix BW’ baselines. In a low energy efficient region, i.e., in high system bandwidth,

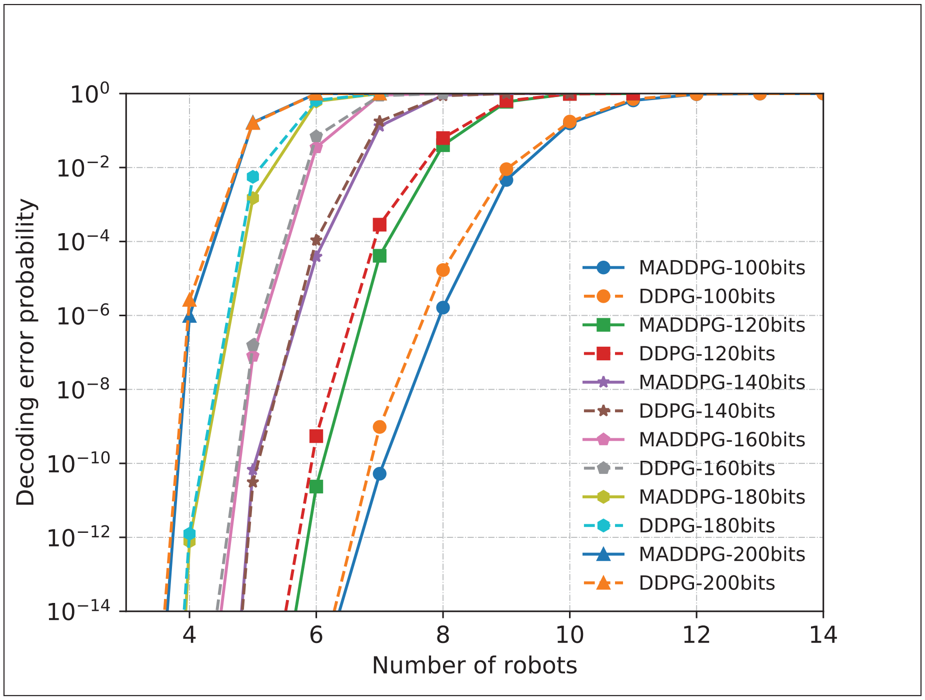


Figure 5.11 Decoding error probability vs number of robot with different packet size, transmission duration $T_d = 100\mu s$, system bandwidth $W_c = 2$ MHz

the 'Max Power' baseline is close to the 'MADDPG-CNN' and 'DDPG-CNN' schemes, and these three schemes are much better than the Fix BW' baseline.

Fig. 5.13 illustrates the variation of energy efficiency with the different packet sizes. We can see the energy efficiency of all schemes increases when the packet size increases. However, it should be noted that when we increase the packet size the decoding error probability also increases as shown in Fig. 5.11. Moreover, it can be seen that the 'MADDPG-CNN' scheme outperforms the 'DDPG-CNN' baseline and achieves a significant improvement compared to both 'Max power' and 'Fix BW' baselines. For example, when the packet size is 200 bits, the energy efficiency of the 'MADDPG-CNN' scheme, the 'DDPG-CNN' baseline, and 'Max power' and 'Fix BW' baselines are 21.4, 19.7, and 13.3 bits/Hz/Joules, respectively, which represent 8.6% and 60.9% improvements.

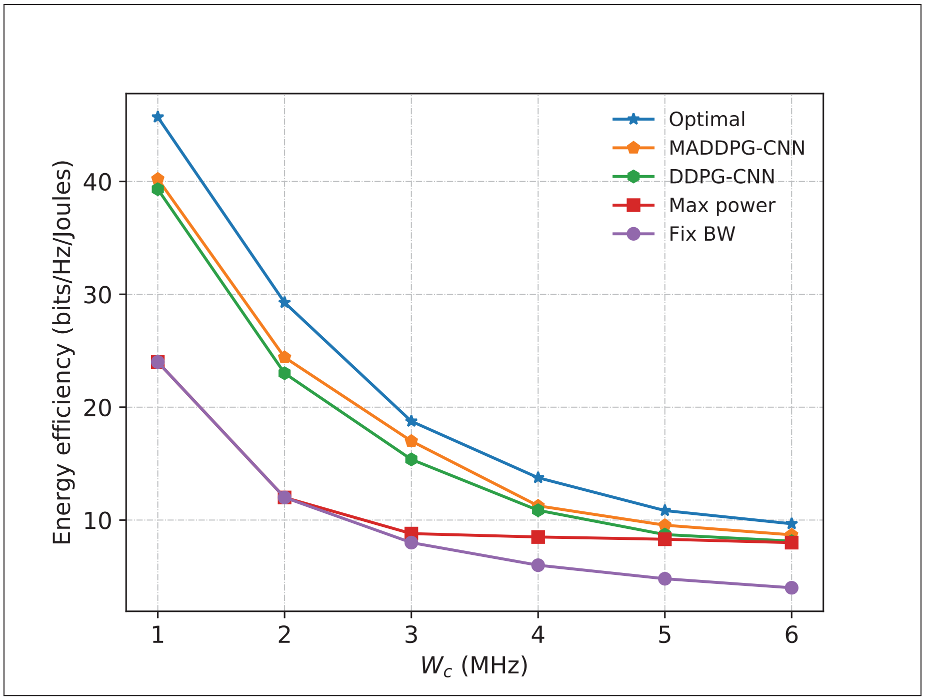


Figure 5.12 Energy efficiency with system bandwidth and $D = 120$ bits

5.8 Conclusion

In this paper, we proposed a deep reinforcement learning-based approach for robot swarm control in 5G mission-critical robotic networks. Such a swarm can be deployed in an automated grid-based warehouse scenario with URLLC requirements between the central controller and the robot swarm. We design a multi-agent DRL-based algorithm that employs the deep deterministic policy gradient (DDPG) method and convolutional neural network (CNN) to achieve a stationary optimal control policy that consists of a number of continuous and discrete control variables. Our formulated problem takes into account the kinematic energy consumption model of the robots as well as the short blocklength communication model between the base station and the robot swarm. Numerical results reveal the proposed multi-agent DDPG-based (MADDPG) algorithm outperforms the single-agent DDPG-based baseline and achieves a significant improvement compared to the 'Max power' and 'Fix BW' baselines in terms of energy efficiency.

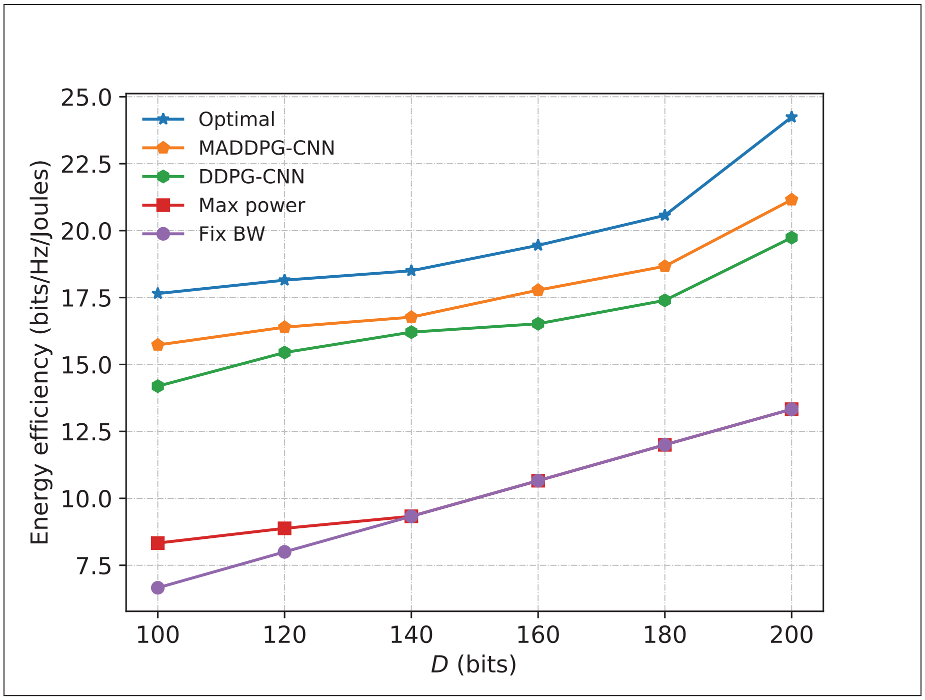


Figure 5.13 Energy efficiency with packet size and W_c = 3 MHz

CHAPITRE 6

DISCUSSION GÉNÉRALE

Même nos méthodes proposées surpassent les travaux existants en termes de performances réseau, de performances robotiques et de performances d'entrepôt. Il y a des limites qui doivent être améliorées dans les travaux futurs.

Dans la première méthode proposée pour la de services 5G dans un entrepôt automatisé en rack, nous supposons que les stations de base et le serveur CoMP ont une connaissance parfaite des informations sur l'état du canal (CSI) des robots mobiles. En pratique, étant donné que les conditions du canal varient en raison de la mobilité des AGV, le CSI instantané peut être estimé par une approche populaire dite séquence d'apprentissage (ou séquence pilote) qui introduit des erreurs d'estimation. Ainsi, le CSI instantané peut être erroné et plutôt obsolète, ce qui a des effets néfastes sur le rapport signal sur bruit (SNR) de bout en bout. Cela pourrait être considéré comme les limites de notre hypothèse. De plus, étant donné que la distance de déplacement d'un AGV dans chaque créneau temporel est sensiblement beaucoup plus petite que la couverture de communication d'un gNB, nous supposons que le CSI reste constant (fixe) dans un créneau et varie entre différents créneaux.

Dans la deuxième méthode proposée pour le contrôle robotique dans un entrepôt automatisé basé sur des racks, nous formulons le temps de service de la base du robot en supposant que chaque robot n'effectue qu'une seule tâche à chaque tranche de temps. Cela signifie que chaque robot ne porte/transporte qu'un seul article à chaque créneau horaire. Ceci peut être considéré comme la limite de ce travail. Pour l'opération de tâches multiples, c'est-à-dire que chaque robot peut transporter plusieurs articles à la fois, puis le robot doit visiter plusieurs endroits pour ramasser ou récupérer des articles. Dans le problème de prélèvement ou de récupération d'articles multiples, le robot doit optimiser son trajet afin de minimiser son temps de déplacement ou de service. Par conséquent, le temps de service du robot devient différent et beaucoup plus compliqué que notre conception actuelle qui rend la formulation du problème sophistiquée. Ce scénario sera considéré dans notre orientation future.

Dans la méthode proposée pour le contrôle robotique dans un entrepôt automatisé, la probabilité d'erreur de décodage augmentera lorsque le nombre de robots dans l'entrepôt augmentera, car nous supposons que tous les robots partagent la même ressource spectrale. Pour assurer la contrainte d'ultra-fiabilité, la bande passante du système peut être augmentée lorsque le nombre de robots augmente. Ceci est dû au fait que lorsque la bande passante du système augmente, la probabilité d'erreur de décodage diminue de sorte que le seuil minimum de probabilité d'erreur de décodage peut être satisfait. Cependant, la ressource spectrale 5G est coûteuse, nous ne pouvons pas augmenter arbitrairement la bande passante du système. Par conséquent, cela peut être considéré comme notre limite.

CONCLUSION ET RECOMMANDATIONS

7.1 Conclusion

Dans cette thèse, nous avons proposé trois cadres pour la de services 5G et le contrôle robotique dans un entrepôt automatisé.

Tout d'abord, nous avons présenté le problème conjoint de regroupement et de formation de faisceaux CoMP pour la de services 5G dans un entrepôt automatisé. En combinant un algorithme de clustering CoMP basé sur la théorie des jeux à faible complexité et la méthode d'optimisation de la politique proximale, nous avons proposé un cadre de gestion des interférences efficace qui convient à un environnement dynamique et peut obtenir des performances approchées à l'optimum et surpasser la ligne de base de clustering CoMP centrée sur l'utilisateur. .

Deuxièmement, nous avons étudié le problème de contrôle robotique pour un système robotique autonome hétérogène dans un entrepôt automatisé. Le problème de l'ordonnancement des tâches est formulé comme un problème de contrôle des files d'attente. Nous proposons un algorithme basé sur l'apprentissage par renforcement profond pour obtenir une politique d'ordonnancement des tâches optimale du problème formulé. Ensuite, nous proposons un Proximal Weighted Federated Learning (PWFL) pour implémenter un algorithme DRL décentralisé dans lequel chaque entrepôt du système est exploité par un agent Proximal Policy Optimization (PPO) implémenté dans un poste de travail pour planifier les tâches des robots dans chaque entrepôt, et un serveur central agit comme un agrégateur de modèles global. Les résultats de la simulation démontrent l'amélioration de notre schéma de planification des tâches PWFL proposé par rapport aux travaux antérieurs en termes de longueur moyenne de la file d'attente, de temps d'attente moyen et de probabilité d'attente.

Enfin, nous avons proposé une approche basée sur l'apprentissage par renforcement profond pour le contrôle robotique dans un entrepôt automatisé basé sur une grille. La robotique en essaim est déployée dans un scénario d'entrepôt automatisé basé sur une grille avec des exigences URLLC entre le contrôleur central et l'essaim de robots. Nous concevons un algorithme basé sur DRL multi-agents qui utilise la méthode du gradient de politique déterministe profond (DDPG) et le réseau neuronal convolutif (CNN) pour obtenir une politique de contrôle stationnaire optimale qui consiste en un certain nombre de variables de contrôle continues et discrètes. Notre problème formulé prend en compte le modèle de consommation d'énergie cinématique des robots ainsi que le modèle de communication à courte longueur de bloc entre la station de base et l'essaim de robots. Les résultats numériques révèlent que l'algorithme proposé basé sur le DDPG multi-agent (MADDPG) surpasse la ligne de base basée sur le DDPG à agent unique et réalise une amélioration significative par rapport aux lignes de base "Max power" et "Fix BW" en termes d'efficacité énergétique.

7.2 Reconnaissance

Ce travail a été soutenu par Mitacs, Ciena et ENCQOR dans le cadre de la subvention IT13947.

7.3 Publication

7.3.1 Publication de revue

1. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet. "Energy Efficiency Deep Reinforcement Learning for URLLC in 5G Mission-Critical Swarm Robotics." IEEE Transactions on Network and Service Management (2022) (Minor Revision).
2. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet. "Federated Deep Reinforcement Learning for Task Scheduling in Heterogeneous Autonomous Robotic System." IEEE Transactions on Automation Science and Engineering (2022) (Dans la presse), DOI : 10.1109/TASE.2022.3221352.

3. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet. "Converging game theory and reinforcement learning for industrial Internet-of-Things." IEEE Transactions on Network and Service Management (2022) (Dans la presse), DOI : 10.1109/TNSM.2022.3202168.
4. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet. "UAV control for wireless service provisioning in critical demand areas : A deep reinforcement learning approach." IEEE Transactions on Vehicular Technology 70, no. 7 (2021) : 7138-7152, DOI : 10.1109/TVT.2021.3088129.
5. Tai Manh Ho et Kim-Khoa Nguyen, "Joint Server Selection, Cooperative Offloading and Handover in Multi-Access Edge Computing Wireless Network : A Deep Reinforcement Learning Approach," in IEEE Transactions on Mobile Computing, vol. 21, no. 7, pp. 2421-2435, 1 July 2022, DOI : 10.1109/TMC.2020.3043736.

7.3.2 Publication de la conférence

1. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet, "Optimized Task Offloading in UAV-Assisted Cloud Robotics," IEEE International Conference on Communications (ICC), 28 May – 01 June 2023 Rome, Italy.
2. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet, "Collaborative Game Theory and Deep Learning Closed-loop Automation In O-RAN 5G Network Slicing For Smart Grid Applications," IEEE International Conference on Communications (ICC), 28 May – 01 June 2023 Rome, Italy.
3. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet. "Federated Deep Reinforcement Learning for Task Scheduling in Heterogeneous Autonomous Robotic System." 2022 IEEE Global Communications Conference (GLOBECOM). IEEE, 2022.
4. Tai Manh Ho, Kim-Khoa Nguyen, and Abdo Shabah. "Self-organizing internet of multi-RAT robotic things mesh network." 2022 14th IFIP Wireless and Mobile Networking Conference (WMNC). IEEE, 17-19 October 2022, Sousse, Tunisia .

5. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet. "Game Theoretic Reinforcement Learning Framework For Industrial Internet of Things." 2022 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2022, DOI : 10.1109/WCNC51071.2022.9771864.
6. Tai Manh Ho, Kim-Khoa Nguyen, et Mohamed Cheriet. "Energy-aware Control Of UAV-based Wireless Service Provisioning." 2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021, DOI : 10.1109/GLOBECOM46510.2021.9685905.
7. Tai Manh Ho, Ti Ti Nguyen, Kim-Khoa Nguyen, et Mohamed Cheriet. "Deep reinforcement learning for URLLC in 5G mission-critical cloud robotic application." In 2021 IEEE Global Communications Conference (GLOBECOM), IEEE, 2021, DOI : 10.1109/GLOBECOM46510.2021.9685978.
8. Tai Manh Ho et Kim-Khoa Nguyen. "Deep Q-learning for joint server selection, offloading, and handover in multi-access edge computing." ICC 2021-IEEE International Conference on Communications. IEEE, 2021, DOI : 10.1109/ICC42927.2021.9500353.

BIBLIOGRAPHIE

3GPP. (2018). Study on Communication for Automation in Vertical domains (Release 16). Dans *TR 22.804 V16.2.0*. Repéré à https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3187.

3GPP. (2022). 3GPP Radio Resource Control (RRC) protocol specification (Release 17) . Dans *TS 38.331 V17.0.0 (2022-03)*. Repéré à https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3197.

6river. (2020). What is warehouse robotics ? [Format]. Repéré à https://6river.com/what-is-warehouse-robotics/.

Abdelhakam, M. M., Elmesalawy, M. M., Mahmoud, K. R. & Ibrahim, I. I. (2018). A cooperation strategy based on bargaining game for fair user-centric clustering in cloud-RAN. *IEEE Communications Letters*, 22(7), 1454–1457.

Al-Eryani, Y., Akrout, M. & Hossain, E. (2020). Multiple Access in Cell-Free Networks : Outage Performance, Dynamic Clustering, and Deep Reinforcement Learning-Based Design. *IEEE Journal on Selected Areas on Communications.*, 39(4), 1028–1042.

Arracking. (2021). Types of Industrial Racking for the Warehouse : Classification and characteristics [Format]. Repéré à https://www.ar-racking.com/en/news-and-blog/storage-solutions/storage-racking-solutions/types-of-industrial-racking-for-the-warehouse-classification-and-characteristics.

Bai, C., Yan, P., Pan, W. & Guo, J. (2021). Learning-based multi-robot formation control with obstacle avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 11811–11822.

Bassoy, S., Farooq, H., Imran, M. A. & Imran, A. (2017). Coordinated multi-point clustering schemes : A survey. *IEEE Commun. Sur. Tutor.*, 19(2), 743–764.

Berthene, A. (2022). Coronavirus pandemic adds 219 billion to US ecommerce sales in 2020-2021. Repéré le 2022-03-15 à https://www.digitalcommerce360.com/article/coronavirus-impact-online-retail/.

Bertsekas, D. & Gallager, R. (2021). *Data networks*. Athena Scientific.

Bolu, A. & Korçak, Ö. (2021). Adaptive Task Planning for Multi-Robot Smart Warehouse. *IEEE Access*, 9, 27346–27358.

Boysen, N., De Koster, R. & Weidinger, F. (2019). Warehousing in the e-commerce era : A survey. *European Journal of Operational Research*, 277(2), 396–411.

Businessinsider. (2018). Inside Ocado's new warehouse where thousands of robots zoom around a grid system to pack groceries [Format]. Repéré à https://www.businessinsider.com/inside-ocado-new-warehouse-where-robots-do-the-work-uk-andover-2018-5.

Cao, Y., Lien, S.-Y., Liang, Y.-C., Chen, K.-C. & Shen, X. (2021). User Access Control in Open Radio Access Networks : A Federated Deep Reinforcement Learning Approach. *IEEE Transactions on Wireless Communications*.

Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N. & Fox, D. (2019). Closing the sim-to-real loop : Adapting simulation randomization with real world experience. *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8973–8979.

Chen, D., Qi, Q., Zhuang, Z., Wang, J., Liao, J. & Han, Z. (2020). Mean field deep reinforcement learning for fair and efficient UAV control. *IEEE Internet of Things Journal*, 8(2), 813–828.

Chen, W., Yaguchi, Y., Naruse, K., Watanobe, Y., Nakamura, K. & Ogawa, J. (2018). A study of robotic cooperation in cloud robotics : Architecture and challenges. *IEEE Access*, 6, 36662–36682.

Chen, X., Li, Y. & Liu, L. (2019). A Coordinated Path Planning Algorithm for Multi-Robot in Intelligent Warehouse. *IEEE International Conference Robotics Biomimetics (ROBIO)*, pp. 2945–2950.

Choi, H.-B., Kim, J.-B., Han, Y.-H., Oh, S.-W. & Kim, K. (2022). MARL-based cooperative multi-AGV control in warehouse systems. *IEEE Access*, 10, 100478–100488.

da Costa Barros, Í. R. & Nascimento, T. P. (2021). Robotic Mobile Fulfillment Systems : A survey on recent developments and research opportunities. *Robotics and Autonomous Systems*, 137, 103729.

Dai, T., Arulkumaran, K., Gerbert, T., Tukra, S., Behbahani, F. & Bharath, A. A. (2022). Analysing deep reinforcement learning agents trained with domain randomisation. *Neurocomputing*, 493, 143–165.

Ekren, B. Y., Heragu, S. S., Krishnamurthy, A. & Malmborg, C. J. (2012). An approximate solution for semi-open queueing network model of an autonomous vehicle storage and retrieval system. *IEEE Transactions on Automation Science and Engineering*, 10(1), 205–215.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J. et al. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4), 219–354.

Fuzzylogx. (2020). AutoStore vs Ocado [Format]. Repéré à https://www.fuzzylogx.com.au/fuzzy-friday/autostore-vs-ocado/.

Ge, J., Liang, Y.-C., Joung, J. & Sun, S. (2020). Deep Reinforcement Learning for Distributed Dynamic MISO Downlink-Beamforming Coordination. *IEEE Transactions on Communications*, 68(10), 6070–6085.

Guidolin, F., Badia, L. & Zorzi, M. (2014). A distributed clustering algorithm for coordinated multipoint in LTE networks. *IEEE Wireless Communications Letters*, 3(5), 517–520.

Han, R., Chen, S., Wang, S., Zhang, Z., Gao, R., Hao, Q. & Pan, J. (2022a). Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards. *IEEE Robotics and Automation Letters*, 7(3), 5896–5903.

Han, S. D. & Yu, J. (2020). Ddm : Fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics. *IEEE Robotics and Automation Letters*, 5(2), 1350–1357.

Han, X., Liu, H., Sun, F. & Zhang, X. (2019). Active object detection with multistep action prediction using deep q-network. *IEEE Transactions on Industrial Informatics*, 15(6), 3723–3731.

Han, Y., Zhan, I. H., Zhao, W., Pan, J., Zhang, Z., Wang, Y. & Liu, Y.-J. (2022b). Deep reinforcement learning for robot collision avoidance with self-state-attention and sensor fusion. *IEEE Robotics and Automation Letters*, 7(3), 6886–6893.

Hanif, A. F., Tembine, H., Assaad, M. & Zeghlache, D. (2015). Mean-field games for resource sharing in cloud-based networks. *IEEE/ACM Transactions on Networking*, 24(1), 624–637.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D. & Meger, D. (2018). Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Ho, T. M., Tran, T. D., Nguyen, T. T., Kazmi, S., Le, L. B., Hong, C. S. & Hanzo, L. (2019). Next-generation wireless solutions for the smart factory, smart vehicles, the smart grid and smart cities. *arXiv preprint arXiv :1907.10102*.

Ho, T. M., Nguyen, K.-K. & Cheriet, M. (2021a). UAV Control for Wireless Service Provisioning in Critical Demand Areas : A Deep Reinforcement Learning Approach. *IEEE Transactions on Vehicular Technology*, 70(7), 7138-7152. doi : 10.1109/TVT.2021.3088129.

Ho, T. M., Nguyen, T. T., Nguyen, K.-K. & Cheriet, M. (2021b). Deep reinforcement learning for URLLC in 5G mission-critical cloud robotic application. *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

Hönig, W., Kiesel, S., Tinka, A., Durham, J. W. & Ayanian, N. (2019). Persistent and robust execution of MAPF schedules in warehouses. *IEEE Robotics and Automation Letters*, 4(2), 1125–1131.

Huang, L., Qu, H., Fu, M. & Deng, W. (2018). Reinforcement learning for mobile robot obstacle avoidance under dynamic environments. *Pacific Rim International Conferences on Artificial Intelligence*, pp. 441–453.

Ishikawa, S. (1974). Fixed points by a new iteration method. *Proceedings of the American Mathematical Society*, 44(1), 147–150.

Jakobi, N., Husbands, P. & Harvey, I. (1995). Noise and the reality gap : The use of simulation in evolutionary robotics. *European Conference on Artificial Life*, pp. 704–720.

Janner, M., Fu, J., Zhang, M. & Levine, S. (2019). When to Trust Your Model : Model-Based Policy Optimization. *Advances in Neural Information Processing Systems*.

Jayaweera, N., Marasinghe, D., Rajatheva, N. & Latva-Aho, M. (2020). Factory Automation : Resource Allocation of an Elevated LiDAR System with URLLC Requirements. *6G Wireless Summit (6G SUMMIT)*, pp. 1–5.

Karabegović, I., Karabegović, E., Mahmić, M. & Husak, E. (2022). Application of Service Robots for Logistics During the COVID-19 Pandemic Accelerates the Implementation of Industry 4.0. *International Conference "New Technologies, Development and Applications"*, pp. 3–17.

Karimi, M. & Ahmadi, M. (2021). A Reinforcement Learning Approach in Assignment of Task Priorities in Kinematic Control of Redundant Robots. *IEEE Robotics and Automation Letters*, 7(2), 850–857.

Khan, J. & Jacob, L. (2020a). Availability Maximization Framework for CoMP Enabled URLLC With Short Packets. *IEEE Networking Letters*, 2(1), 1–4.

Khan, J. & Jacob, L. (2020b). Resource Allocation for CoMP Enabled URLLC in 5G C-RAN Architecture. *IEEE Systems Journal*.

Khoshnevisan, M., Joseph, V., Gupta, P., Meshkati, F., Prakash, R. & Tinnakornsrisuphap, P. (2019). 5G industrial networks with CoMP for URLLC and time sensitive network architecture. *IEEE Journal on Selected Areas in Communications*, 37(4), 947–959.

Kim, H.-J., Pais, C. & Shen, Z.-J. M. (2020). Item Assignment Problem in a Robotic Mobile Fulfillment System. *IEEE Transactions on Automation Science and Engineering*, 17(4), 1854–1867.

Kivnon. (2023). Everything about AGV and its different types. [Format]. Repéré à https://www.kivnon.com/.

Koetsier, J. (2022). Keeping Up With Amazon : How Warehouse Robots Are Revolutionizing The On-Demand Economy. Repéré le 2022-04-04 à https://www.forbes.com/sites/johnkoetsier/2022/04/04/keeping-up-with-amazon-how-warehouse-robotics-is-revolutionizing-the-on-demand-economy/?sh=d2e4d0862f16.

Lee, A. & Longton, P. (1959). Queueing processes associated with airline passenger check-in. *Journal of the Operational Research Society*, 10(1), 56–71.

Lee, H.-Y. & Murray, C. C. (2019). Robotics in order picking : evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal of Production Research*, 57(18), 5821–5841.

Lee, S. & Choi, D.-H. (2020). Federated reinforcement learning for energy management of multiple smart homes with distributed energy resources. *IEEE Transactions on Industrial Informatics*, 18(1), 488–497.

Li, J., Tinka, A., Kiesel, S., Durham, J. W., Kumar, T. & Koenig, S. (2020a). Lifelong multi-agent path finding in large-scale warehouses. *arXiv preprint arXiv :2005.07371*.

Li, Q., Lin, W., Liu, Z. & Prorok, A. (2020b). Message-Aware Graph Attention Networks for Large-Scale Multi-Robot Path Planning. *arXiv preprint arXiv :2011.13219*.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A. & Smith, V. (2018). Federated optimization in heterogeneous networks. *arXiv preprint arXiv :1812.06127*.

Li, Z., Barenji, A. V., Jiang, J., Zhong, R. Y. & Xu, G. (2020c). A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand. *Journal of Intelligent Manufacturing*, 31(2), 469–480.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv :1509.02971*.

Liu, L. & Yu, W. (2018). A D2D-based protocol for ultra-reliable wireless communications for industrial automation. *IEEE Transactions on Wireless Communications*, 17(8), 5045–5058.

Liu, S. & Sun, D. (2013). Minimizing energy consumption of wheeled mobile robots via optimal motion planning. *IEEE/ASME Transactions on Mechatronics*, 19(2), 401–411.

Liu, Z., Wang, H., Wei, H., Liu, M. & Liu, Y.-H. (2020). Prediction, Planning, and Coordination of Thousand-Warehousing-Robot Networks With Motion and Communication Uncertainties. *IEEE Transactions on Automation Science and Engineering*.

Luo, R., Ni, W., Tian, H. & Cheng, J. (2022). Federated deep reinforcement learning for RIS-assisted indoor multi-robot communication systems. *IEEE Transactions on Vehicular Technology*, 71(11), 12321–12326.

Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y.-C. & Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking : A survey. *IEEE Communications Surveys & Tutorials*, 21(4), 3133–3174.

Lv, L., Zhang, S., Ding, D. & Wang, Y. (2019). Path planning via an improved DQN-based learning policy. *IEEE Access*, 7, 67319–67330.

Ma, Z., Wu, G., Ji, B., Wang, L., Luo, Q. & Chen, X. (2022). A Novel Scattered Storage Policy Considering Commodity Classification and Correlation in Robotic Mobile Fulfillment Systems. *IEEE Transactions on Automation Science and Engineering*.

Marsch, P. & Fettweis, G. P. (2011). *Coordinated Multi-Point in Mobile Communications : from theory to practice*. Cambridge University Press.

McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics*, pp. 1273–1282.

Mei, Y., Lu, Y.-H., Hu, Y. C. & Lee, C. G. (2006). Deployment of mobile robots with energy and timing constraints. *IEEE Transactions Robotics*, 22(3), 507–522.

Mismar, F. B., Evans, B. L. & Alkhateeb, A. (2019). Deep reinforcement learning for 5G networks : Joint beamforming, power control, and interference coordination. *IEEE Transactions on Communications*, 68(3), 1581–1592.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529–533.

Mostofi, Y., Gonzalez-Ruiz, A., Gaffarkhah, A. & Li, D. (2009). Characterization and modeling of wireless channels for networked robotic and control systems-a comprehensive overview. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4849–4854.

Nasir, A. A., Tuan, H., Nguyen, H. H., Debbah, M. & Poor, H. V. (2020). Resource Allocation and Beamforming Design in the Short Blocklength Regime for URLLC. *IEEE Transactions on Wireless Communications.*, 20(2), 1321–1335.

Neely, M. J. (2010). Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1), 1–211.

Newscientist. (2016). Robo shop [Format]. Repéré à https://www.newscientist.com/article/mg23231010-600-robo-shop/.

Nourbakhsh, V. & Turner, J. (2022). Dynamized routing policies for minimizing expected waiting time in a multi-class multi-server system. *Computers & Operations Research*, 137, 105545.

Pan, C., Ren, H., Deng, Y., Elkashlan, M. & Nallanathan, A. (2019). Joint blocklength and location optimization for URLLC-enabled UAV relay systems. *IEEE Communnications Letters*, 23(3), 498–501.

Polyanskiy, Y., Poor, H. V. & Verdú, S. (2010). Channel coding rate in the finite blocklength regime. *IEEE Transactions in Information and Theory*, 56(5), 2307–2359.

Qiu, C., Hu, Y., Chen, Y. & Zeng, B. (2019). Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. *IEEE Internet of Things Journal*, 6(5), 8577–8588.

Qualcomm. (2019). How will 5G transform Industrial IoT ? [Format]. Repéré à https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/how_will_5g_transform_industrial_iot_2.pdf?_hsenc=p2ANqtz-88Dv078ksg3VsNl1BdYy0m2fHs0b2BWphbx70kya0P4t8n-FEUg_tf7zkshugytDj8bz2u07Bx-fgVR4im72Zl1n1aCw.

Qualcomm. (2021). 5G : Bringing precise positioning to the connected intelligent edge. Repéré à https://www.qualcomm.com/media/documents/files/5g-positioning-for-the-connected-intelligent-edge.pdf.

Ramamonjison, R., Haghnegahdar, A. & Bhargava, V. K. (2014). Joint optimization of clustering and cooperative beamforming in green cognitive wireless networks. *IEEE Transactions on Wireless Communications.*, 13(2), 982–997.

Ren, H., Liu, N., Pan, C., Elkashlan, M., Nallanathan, A., You, X. & Hanzo, L. (2018). Power-and rate-adaptation improves the effective capacity of C-RAN for Nakagami-*m* fading channels. *IEEE Transactions on Vehicular Technology*, 67(11), 10841–10855.

Ren, H., Pan, C., Deng, Y., Elkashlan, M. & Nallanathan, A. (2019a). Joint power and blocklength optimization for URLLC in a factory automation scenario. *IEEE Transactions on Wireless Communications.*, 19(3), 1786–1801.

Ren, H., Pan, C., Deng, Y., Elkashlan, M. & Nallanathan, A. (2019b). Resource allocation for URLLC in 5G mission-critical IoT networks. *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6.

Ren, H., Pan, C., Deng, Y., Elkashlan, M. & Nallanathan, A. (2020a). Joint Pilot and Payload Power Allocation for Massive-MIMO-Enabled URLLC IIoT Networks. *IEEE Journal on Selected Areas on Communication*, 38(5), 816–830.

Ren, H., Pan, C., Deng, Y., Elkashlan, M. & Nallanathan, A. (2020b). Resource allocation for secure URLLC in mission-critical IoT scenarios. *IEEE Transactions on Communications*, 68(9), 5793–5807.

Rivière, B., Hönig, W., Yue, Y. & Chung, S.-J. (2020). GLAS : Global-to-Local Safe Autonomy Synthesis for Multi-Robot Motion Planning with End-to-End Learning. *IEEE Robotics and Automation Letters*, 5(3), 4249–4256.

Roy, D., Krishnamurthy, A., Heragu, S. S. & Malmborg, C. J. (2013). Blocking effects in warehouse systems with autonomous vehicles. *IEEE Transactions on Automation Science and Engineering*, 11(2), 439–451.

Roy, D., Nigam, S., de Koster, R., Adan, I. & Resing, J. (2019). Robot-storage zone assignment strategies in mobile fulfillment systems. *Transportation Research Part E : Logistics and Transportation Review*, 122, 119–142.

Rusu, A. A., Večerík, M., Rothörl, T., Heess, N., Pascanu, R. & Hadsell, R. (2017). Sim-to-real robot learning from pixels with progressive nets. *Conference on robot learning*, pp. 262–270.

Sartoretti, G., Kerr, J., Shi, Y., Wagner, G., Kumar, T. S., Koenig, S. & Choset, H. (2019). PRIMAL : Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotic Automation Letters*, 4(3), 2378–2385.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv :1707.06347.*

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1), 3–55.

Shi, D., Gao, H., Wang, L., Pan, M., Han, Z. & Poor, H. V. (2020). Mean field game guided deep reinforcement learning for task placement in cooperative multiaccess edge computing. *IEEE Internet of Things Journal*, 7(10), 9330–9340.

Simon, M. K. & Alouini, M.-S. (1998). A unified approach to the performance analysis of digital communication over generalized fading channels. *Proceedings of the IEEE*, 86(9), 1860–1877.

Smartstorage. (2022). Autonomous Mobile Robots (AMR) : How they work and how they can become part of your warehouse [Format]. Repéré à https://www.smart-storage.eu/autonomous-mobile-robots-amr-how-they-work-and-how-they-can-become-part-of-your-warehouse/.

Sun, C., She, C., Yang, C., Quek, T. Q., Li, Y. & Vucetic, B. (2018). Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications. *IEEE Transactions on Wireless Communications*, 18(1), 402–415.

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning : An introduction*. MIT press.

Sutton, R. S., McAllester, D. A., Singh, S. P. & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, pp. 1057–1063.

Švancara, J., Vlk, M., Stern, R., Atzmon, D. & Barták, R. (2019). Online multi-agent pathfinding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 7732–7739.

Tan, A. H., Bejarano, F. P., Zhu, Y., Ren, R. & Nejat, G. (2022). Deep Reinforcement Learning for Decentralized Multi-Robot Exploration with Macro Actions. *IEEE Robotics and Automation Letters*, 8(1), 272–279.

Tang, H., Wang, A., Xue, F., Yang, J. & Cao, Y. (2021). A novel hierarchical soft actor-critic algorithm for multi-logistics robots task allocation. *IEEE Access*, 9, 42568–42582.

Tao, F., Zhang, H., Liu, A. & Nee, A. Y. (2018). Digital twin in industry : State-of-the-art. *IEEE Transactions on industrial informatics*, 15(4), 2405–2415.

Thefabricator. (2021). Autonomous mobile robots could change metal fabrication [Format]. Repéré à https://www.thefabricator.com/thefabricator/article/shopmanagement/autonomous-mobile-robots-could-change-metal-fabrication.

Tse, D. & Viswanath, P. (2005). *Fundamentals of wireless communication*. Cambridge university press.

Tzeng, E., Devin, C., Hoffman, J., Finn, C., Abbeel, P., Levine, S., Saenko, K. & Darrell, T. (2020). Adapting deep visuomotor representations with weak pairwise constraints. Dans *Algorithmic Foundations of Robotics XII* (pp. 688–703). Springer.

Wang, B., Liu, Z., Li, Q. & Prorok, A. (2020a). Mobile Robot Path Planning in Dynamic Environments through Globally Guided Reinforcement Learning. *arXiv preprint arXiv :2005.05420*.

Wang, J., Lan, Z., Sum, C.-S., Pyo, C.-W., Gao, J., Baykas, T., Rahman, A., Funada, R., Kojima, F., Lakkis, I. et al. (2009). Beamforming codebook design and performance evaluation for 60GHz wideband WPANs. *2009 IEEE 70th Vehicular Technology Conference Fall*, pp. 1–6.

Wang, L., Peters, G., Liang, Y.-C. & Hanzo, L. (2020b). Intelligent User-Centric Networks : Learning-Based Downlink CoMP Region Breathing. *IEEE Transactions on Vehicular Technology*, 69(5), 5583–5597.

Wang, W., Wu, Y., Zheng, J. & Chi, C. (2020c). A comprehensive framework for the design of modular robotic mobile fulfillment systems. *IEEE Access*, 8, 13259–13269.

Wang, X., Li, R., Wang, C., Li, X., Taleb, T. & Leung, V. C. (2020d). Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching. *IEEE Journal on Selected Areas in Communications*, 39(1), 154–169.

Wang, X., Wang, C., Li, X., Leung, V. C. & Taleb, T. (2020e). Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching. *IEEE Internet of Things Journal*, 7(10), 9441–9455.

Wen, J., He, L. & Zhu, F. (2018). Swarm robotics control and communications : imminent challenges for next generation smart logistics. *IEEE Communication Magazine*, 56(7), 102–107.

Willms, A. R. & Yang, S. X. (2006). An efficient dynamic system for real-time robot-path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(4), 755–766.

Yang, H., Xiong, Z., Zhao, J., Niyato, D., Yuen, C. & Deng, R. (2020a). Deep reinforcement learning based massive access management for ultra-reliable low-latency communications. *IEEE Transactions on Wireless Communications*, 20(5), 2977–2990.

Yang, P., Xi, X., Quek, T. Q., Chen, J., Cao, X. & Wu, D. (2020b). How should I orchestrate resources of my slices for bursty URLLC service provision ? *IEEE Transactions on Communications*.

Yang, P., Xi, X., Fu, Y., Quek, T. Q., Cao, X. & Wu, D. (2021a). Multicast eMBB and bursty URLLC service multiplexing in a CoMP-enabled RAN. *IEEE Transactions on Wireless Communications*.

Yang, P., Xi, X., Quek, T. Q., Chen, J., Cao, X. & Wu, D. (2021b). RAN slicing for massive IoT and bursty URLLC service multiplexing : Analysis and optimization. *IEEE Internet of Things Journal*.

Yang, S., Zhang, Y., Ma, L., Song, Y., Zhou, P., Shi, G. & Chen, H. (2021c). A Novel Maximin-Based Multi-Objective Evolutionary Algorithm Using One-by-One Update Scheme for Multi-Robot Scheduling Optimization. *IEEE Access*, 9, 121316–121328.

Yoshitake, H., Kamoshida, R. & Nagashima, Y. (2019). New automated guided vehicle system using real-time holonic scheduling for warehouse picking. *IEEE Robotics and Automation Letters*, 4(2), 1045–1052.

Yuan, Z. & Gong, Y. Y. (2017). Bot-in-time delivery for robotic mobile fulfillment systems. *IEEE Transactions on Engineering Management*, 64(1), 83–93.

Zhang, C., Patras, P. & Haddadi, H. (2019a). Deep learning in mobile and wireless networking : A survey. *IEEE Communications surveys & tutorials*, 21(3), 2224–2287.

Zhang, H., Luo, H., Wang, Z., Liu, Y. & Liu, Y. (2019b). Multi-robot cooperative task allocation with definite Path-Conflict-Free handling. *IEEE Access*, 7, 138495–138511.

Zhang, L., Sun, Y., Barth, A. & Ma, O. (2020). Decentralized control of multi-robot system in cooperative object transportation using deep reinforcement learning. *IEEE Access*, 8, 184109–184119.

Zhang, P., Wang, C., Jiang, C. & Han, Z. (2021). Deep reinforcement learning assisted federated learning algorithm for data management of IIoT. *IEEE Transactions on Industrial Informatics*, 17(12), 8475–8484.

Zhou, L., Shi, Y., Wang, J. & Yang, P. (2014). A balanced heuristic mechanism for multirobot task allocation of intelligent warehouses. *Mathematical Problems in Engineering*, 2014.

Zhu, K., Li, B., Zhe, W. & Zhang, T. (2022). Collision Avoidance Among Dense Heterogeneous Agents Using Deep Reinforcement Learning. *IEEE Robotics and Automation Letters*, 8(1), 57–64.

Zou, B., Gong, Y., Xu, X. & Yuan, Z. (2017). Assignment rules in robotic mobile fulfilment systems for online retailers. *International Journal of Production Research*, 55(20), 6175–6192.