

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA



Faculté de Technologie
Département de Génie Electrique
Laboratoire des Technologies Industrielles et de l'information

THÈSE
EN VUE DE L'OBTENTION DU DIPLOME DE
DOCTORAT
Domaine : Sciences et Technologies. Filière : Electronique
Spécialité : Electronique

Présentée par
DJENADI Ali

Thème

Méthodes métaheuristiques pour le contrôle des systèmes multi-robot

Soutenue le : 27 juillet 2021

Devant le Jury composé de :

Nom et Prénom

Grade

Mr. MOKRANI Karim

Professeur

Univ. de Bejaia

Président

Mr. MENDIL Boubekour

Professeur

Univ. de Bejaia

Rapporteur

Mr. HASSAM Abdelouahab

Professeur

Univ. de Sétif 1

Examineur

Mr. ACHOUR Abdelyazid

MCA

Univ. de Bejaia

Examineur

Année Universitaire : 2020/2021

Remerciements

Je tiens à exprimer mes remerciements les plus sincères et ma vive reconnaissance au Professeur MENDIL Boubekour, mon directeur de thèse pour avoir accepté d'encadrer mon travail de recherche en manifestant dès le départ sa confiance en moi. Je le remercie amplement pour sa compréhension, ses conseils, son soutien, ses encouragements et sa patience.

Nous remercions tous les membres du jury de nous avoir honorés par leur présence et d'avoir accepté d'examiner ce travail.

Nous tenons également à exprimer notre reconnaissance et nos remerciements au Professeur REKIOUA Toufik, le directeur du laboratoire des Technologies Industrielles et de l'information (LTII) et tous les membres des équipes de recherche et du staff technique.

Nous souhaiterions remercier tous les enseignants de la faculté de Technologie, en particulier les enseignants du département de Génie Electrique qui m'ont accompagné, soutenu et encouragé durant tout mon cursus universitaire.

J'adresse également mes remerciements à toute ma famille, mes collègues doctorants, mes amis et toutes autres personnes qui ont contribué de loin ou de près à la réalisation de ce travail.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	i
LISTE DES FIGURES.....	iii
LISTE DES TABLEAUX.....	iv
ABRÉVIATIONS	v
INTRODUCTION GENERALE.....	1
1. Contexte.....	1
2. Problématique.....	1
3. Les contributions de cette thèse.....	2
4. Organisation de la thèse.....	3
CHAPITRE 1. Etat de l'art et revue de la littérature.....	6
1. Introduction	6
2. Les systèmes multi robots.....	6
3. Problème du transport : point de vue d'un système multi-robot.....	7
4. Allocation des tâches des systèmes multi robots.....	11
5. Gestion de l'énergie d'un système multi-robot.....	16
6. Conclusion	19
CHAPITRE 2. Stratégie D-ILS pour la résolution du MRTA à base de la gestion de l'énergie.....	21
1. Introduction	21
2. Système de transport multi-robot : Description du problème.....	21
3. Formulation du MRTA.....	22
4. Fonction d'utilité de gestion de l'énergie proposée.....	23
5. Allocation des tâches distribuée: Protocole de la stratégie D-ILS	24
6. Mécanisme de réallocation des tâches de la stratégie D-ILS	26
7. Conclusion	29
CHAPITRE 3. Implémentation algorithmique du MRTA à base de la stratégie D-ILS.....	30
1. Introduction	30
2. Présentation de l'ILS.....	30
3. Génération de la solution initiale.....	32
4. Recherche locale.....	33
5. Perturbation	34
6. Critères d'acceptation de la solution.....	35
7. Conditions d'arrêts	36
8. Conclusion.....	36

CHAPITRE 4. Expérimentations	37
1. Introduction	37
2. Objectifs.....	37
3. Environnement de simulation.....	37
4. Méthodologie d'évaluation	39
4.1. Critères d'évaluation.....	39
4.2. Scénarios de simulation.....	40
4.3. Stratégies de comparaison.....	41
5. Résultats et discussions	42
5.1. Scénario 1 : évaluation de la consommation d'énergie	42
5.2. Scénario 2: Productivité et PPEC.....	44
5.3. Scénario 3 : D-ILS et les tâches à priorité équivalentes.....	46
5.4. Scénario 4 et 5 : Scénarios dynamiques	48
6. Conclusion.....	50
CONCLUSION GENERALE	51
REFERENCES.....	53

LISTE DES FIGURES

Figure 1.1 – Exemple d'application de SMR.....	7
Figure 1.2 – Application des SMR dans le transport industriel et les centres hospitalisés	7
Figure 1.3 – Facteurs de design d'un système de transport multi-robot industriel.....	8
Figure 1.4 – Taxonomie du MRTA.....	12
Figure 1.5 – Classification des Méthodes d'optimisation classiques.....	14
Figure 2.1 – Organigramme de la stratégie D-ILS	28
Figure 4.1 – Environnement de simulation, avec 5 stations de chargement (noir), 5 stations de déchargement (gris), et 5 robots.....	38
Figure 4.2 – Moyenne d'énergie consommée par les robots dans le scénario 1	43
Figure 4.3 – Distribution dans le temps de l'exécution des tâches de la D-ILS, pour un SMR de 2 robots, dans le scénario 2	43
Figure 4.4 – Distribution dans le temps de l'exécution des tâches de NTF et STDF, pour un SMR de 2 robots, dans le scénario 2	44
Figure 4.5 – Moyenne d'énergie consommée par les robots dans le scénario 2.....	45
Figure 4.6 – Productivité des robots dans le scénario 2	46
Figure 4.7 – PPEC des robots dans le scénario 2	46
Figure 4.8 – Pourcentage de tâches complétées par les robots dans le scénario 3 comparé aux résultats obtenue pour un groupe de cinq robots dans le scénario 2 avec des tâches associées à des priorités de valeurs $0 \leq Q \leq 5$	47
Figure 4.9 – Moyenne de consommation d'énergie des robots dans le scénario 3 comparé aux résultats obtenue pour un groupe de cinq robots dans le scénario 2 avec des tâches associées à des priorités de valeurs $0 \leq Q \leq 5$	48
Figure 4.10 – Comparaison des moyennes de consommation entre les résultats du scénario 4 (cas dynamique) et les résultats du scénario 1 (sans dynamique), pour un SMR de 5 robots.....	49
Figure 4.11 – Comparaison des productivités entre les résultats du scénario 5 (cas dynamique) et les résultats du scénario 2 (sans dynamique), pour un SMR de 5 robots.....	50

LISTE DES TABLEAUX

Tableau 1.1 – Comparatif des approches de résolution du MRTA.....	15
Tableau 3.1 – Représentation de la solution de l'ILS.....	32
Tableau 3.2 – Exemple de fonctionnement de l'ILS	35
Tableau 4.1 – Paramètre des scénarios simulés	40
Tableau 4.2 – Paramètre des scénarios dynamiques simulés.....	41

ABRÉVIATIONS

AGV	Automated Guided Vehicle
SMR	Système Multi-Robot
MRTA	Multi-Robot Task Allocation
ILS	Iterative Local Search
D-ILS	Distributed-Iterative Local Search
VRP	Vehicle Routing Problem
HPTF	High Priority Task-First
NTF	Nearest Task-First
STDF	Shortest Traveling Distance-First

INTRODUCTION GENERALE

INTRODUCTION GENERALE

1. Contexte

Les systèmes Multi-robot (SMR) sont devenus une partie importante de l'industrie moderne. En effet, le besoin grandissant des industries en flexibilité et d'installations reconfigurables, qui répondent à la dynamique des marchés, a poussé vers l'utilisation de systèmes robotiques collaboratifs, multitâches et capables de s'adapter à différents environnements et processus de fabrication. Avec les progrès des plateformes robotiques et des systèmes de télécommunication et surtout, avec l'avènement de l'ère de l'industrie 4.0 et du Smart Manufacturing, les systèmes robotiques collaboratifs tels que les SMR, sont devenus très attractifs en raison de leurs potentiels pratiques dans diverses applications (*e.g.*, industrielles, militaires, domestiques). Toutefois, le transport de biens par des SMR constitue l'implémentation la plus réussie. Ces systèmes de transport multi-robot ont été adaptés pour différents environnements structurés tels que: les industries, la gestion des terminaux dans les ports, les entrepôts et les centres de distribution.

Bien que les SMR sont une technologie prometteuse qui apporte un impact majeur à la vie socioéconomique, leurs designs et exploitations restent très complexes. En effet, plusieurs éléments sont à considérer dans la conception d'un SMR incluant, l'architecture du groupe (*e.g.*, centralisée, décentralisée, hybride), résolution de conflits sur le partage de ressources, allocation des tâches, l'apprentissage, et les problèmes géométriques (*e.g.*, coordination, évitement d'obstacles, planification de chemin, mouvement en formation). Cependant, du point de vue d'un système de transport multi-robot dans un environnement industriel, il est nécessaire de tenir compte d'autres facteurs, tels que la logistique et l'architecture de l'environnement (*e.g.*, le nombre de station de chargement et de déchargement), les caractéristiques des robots (le nombre nécessaire, la capacité, le type) et la gestion des systèmes (*e.g.*, gestion de l'énergie, gestion des défauts).

2. Problématique

Dans cette thèse nous nous focalisons sur deux éléments de conception d'un système de transport multi-robot, à savoir, l'allocation des tâches et la gestion de l'énergie du système. En particulier, il s'agit de traiter la problématique du design d'une stratégie d'allocation des tâches qui maximise la productivité et optimise la gestion de la consommation de l'énergie d'un

système de transport multi-robot. Cette stratégie d'allocation des tâches utilise la métaheuristique de la recherche locale itérative (ILS pour Iterative local search) *de manière distribuée*. En outre, elle est basée sur une *nouvelle fonction d'utilité* modélisée sur des heuristiques élémentaires. Cette fonction exprime l'utilité individuelle de chaque robot du système en considérant son état, l'environnement et les caractéristiques des tâches disponibles.

Dans la stratégie proposée, chaque robot individuel du SMR apporte une solution pour l'allocation des tâches du groupe, en utilisant un algorithme basé sur l'ILS spécifiquement conçu pour ce but. Par conséquent, le SMR produit une population de solutions. L'idée de base de cette stratégie est d'utiliser l'aspect multi-agent des SMR afin de générer une population de solutions, au lieu d'une seule solution à base de l'ILS standard. La stratégie proposée se comporte comme une métaheuristique à population de points dont les individus sont les solutions apportées par chaque robot. Ainsi, elle optimise son exploration de l'espace de recherche à l'image d'une méthode à population de solutions. D'autre part, étant basée sur une méthode de recherche à un seul point (*i.e.*, l'ILS), cette stratégie permet une meilleure exploitation des régions locales de cet espace de recherche.

3. Les contributions de cette thèse

L'évidente stratégie pour maximiser la productivité d'un système de transport multi-robot mobile dans un environnement d'intérieur, en considérant la gestion de la consommation de l'énergie des robots, est d'affecter les robots aux tâches avec la plus courte distance de déplacement nécessaire pour les exécuter (*i.e.*, shortest travelling distance). Cependant, les tâches dans les applications réelles sont souvent associées à des priorités. Par conséquent, pour être productive, le système doit privilégier les tâches à priorités élevées. Toutefois dans certains cas, ces tâches à priorité élevée nécessitent une importante consommation d'énergie comparées aux tâches à faible priorité. Ainsi, maximiser la productivité et minimiser la consommation de l'énergie peuvent être des objectifs contradictoires, d'où la nécessité d'une stratégie d'allocation des tâches qui considère la consommation de l'énergie des robots, afin de trouver un équilibre entre la productivité et la consommation de l'énergie. Dans cette optique, notre travail de thèse apporte les contributions suivantes:

- Nous proposons une nouvelle stratégie d'allocation des tâches, dénommée D-ILS (*i.e.*, Distributed-Iterative Local Search), pour un système de transport multi-robot mobile. Cette stratégie a pour objectif de maximiser la productivité et d'optimiser la gestion de

la consommation d'énergie du système. La D-ILS considère la consommation d'énergie explicitement dans la phase de décision des allocations des tâches. Par ailleurs, le niveau d'énergie ainsi que son épuisement sont considérés explicitement dans l'estimation de l'utilité du robot et implicitement lors de l'allocation des tâches.

- Nous introduisons une nouvelle fonction d'utilité qui reflète la capacité des robots à exécuter les tâches disponibles. Cette utilité se base sur l'essentiel des informations que le robot peut obtenir de son état et de son environnement.
- Nous proposons une variante pour l'algorithme de la métaheuristique de la recherche locale itérative (ILS) pour la résolution de l'allocation des tâches dans les systèmes multi-robot.
- Nous présentons le protocole de résolution de l'allocation des tâches de la stratégie D-ILS. L'originalité de ce protocole est de générer une population de solutions au problème d'allocation des tâches en utilisant l'aspect multi-agent des système multi-robot. Ainsi, il permet à la stratégie D-ILS d'améliorer ses capacités d'exploration et de convergence.
- Nous menons des expériences sur la plateforme de simulation Player/Stage. Les programmes de contrôle sont implémentés en langage C++, ce qui permet de simuler le comportement multi-agent et celui distribué de la stratégie D-ILS en utilisant le concept du *multi-threading* de la bibliothèque du C++.
- Nous étudions les performances de la stratégie D-ILS proposée en termes d'énergie consommée, la productivité effective du système, et la productivité par unité d'énergie consommée.
- Nous étudions également les performances de la stratégie D-ILS dans des scénarios dynamiques avec le nombre de tâches disponibles qui augmente en fonction du temps.

4. Organisation de la thèse

Ce manuscrit est organisé en quatre chapitres. La présente partie constitue une introduction générale incluant le contexte d'étude, la problématique de la thèse, nos contributions et l'organisation de la thèse.

A travers *le premier chapitre*, nous présentons les SMR et leurs applications. Par la suite, nous introduisons le problème du transport d'une manière générale et du point de vue d'un SMR et nous explorons les facteurs considérés dans le design d'un système de transport multi-robot. Dans la quatrième section, nous revoyons la littérature du problème d'allocation

des tâches dans les SMR. Enfin, la cinquième et dernière parties du chapitre se focalise sur l'état de l'art de la gestion des systèmes énergétiques des SMR.

Le deuxième chapitre consiste à présenter la stratégie D-ILS proposée. Dans la première partie, du chapitre, nous décrivons le système de transport multi-robot étudié, puis, nous formulons le problème du transport traité dans ce travail de thèse, en définissant les paramètres des tâches et des robots utilisés. Dans la seconde partie, nous donnons la formulation du modèle de MRTA résolue. Ce modèle nous permet d'introduire dans la troisième partie du chapitre la nouvelle fonction d'utilité de la gestion d'énergie que nous proposons afin d'estimer l'utilité de chaque robot du SMR à accomplir une tâche donnée. Dans la quatrième partie, du chapitre nous introduisons le protocole d'allocation des tâches de la stratégie D-ILS proposée dans ce travail de thèse. Enfin, la dernière partie du chapitre introduit le mécanisme de *réallocation* des tâches implémenté dans la stratégie D-ILS.

Le troisième chapitre se focalise sur la présentation de l'algorithme de la métaheuristique de l'ILS. Dans la première partie, nous présentons le cadre général de l'ILS et le principe d'exploration de l'espace de recherche de cet algorithme. Dans la seconde partie du chapitre, nous adressons la méthode de génération des solutions initiales de l'ILS. Dans la troisième partie, nous introduisons l'heuristique de la recherche locale que nous proposons pour permettre à l'ILS d'exploiter les zones locales de l'espace de recherche. Nous adressons ensuite dans la quatrième partie, la tendance de la recherche locale à se bloquer dans un minimum local. Pour résoudre ce point, nous appliquons une perturbation à la solution générée par la recherche locale afin de trouver un nouveau point de départ, afin de permettre une meilleure exploration de l'espace de recherche. Dans la cinquième partie, nous traitons le critère d'acceptation de la solution. Ce dernier permet à l'ILS de comparer entre la solution actuelle et la solution générée par la recherche locale après l'application de la perturbation, puis de décider si la nouvelle solution est acceptée en tant que nouvelle solution. Enfin dans la dernière partie, nous proposons les conditions d'arrêts utilisées pour l'algorithme de l'ILS.

Dans *Le chapitre quatre*, nous présentons la méthodologie d'évaluation et les différentes expérimentations menées dans ce travail de thèse. Nous débuterons la première partie par exposer les objectifs fixés afin de valider la stratégie proposée. Ensuite, nous présentons l'environnement de simulation utilisé qui est basée sur une plateforme de simulation orientée multi-robot. Puis dans la partie suivante, nous exposons la méthodologie d'évaluation de la stratégie proposée incluant les critères d'évaluations, les scénarios de

simulation et les stratégies de comparaisons. Enfin, Dans la dernière partie, nous présentons les résultats des différent scénarios simulés, puis nous les discutons.

Enfin, dans la conclusion générale, nous dressons une synthèse des différentes contributions réalisées dans le cadre des travaux de cette thèse et nous donnons un ensemble de perspectives de recherche à court et long terme découlant de ces travaux.

CHAPITRE 1. Etat de l'art et revue de la littérature

CHAPITRE 1. Etat de l'art et revue de la littérature

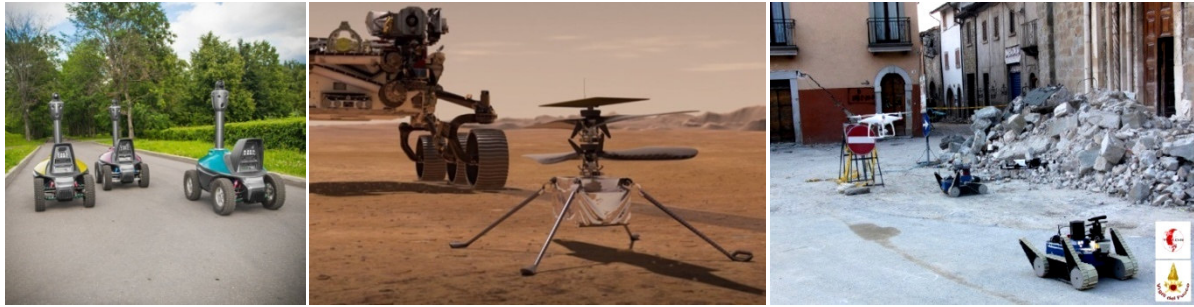
1. Introduction

Dans ce chapitre, nous présentons les systèmes multi-robot. Par la suite, nous donnons un aperçu sur le problème du transport d'une manière générale, ainsi que du point de vue d'un SMR dans un environnement industriel. Nous revoyons également l'état de l'art et la littérature de l'allocation des tâches dans les systèmes multi-robot ainsi que de la gestion des systèmes énergétiques des SMR dans l'industrie.

2. Les systèmes multi robots

Les systèmes de véhicules automatisés, à l'exemple des systèmes AGV (Automated Guided Vehicles), sont devenus une partie importante de l'industrie moderne (Grand View Research; MarketsandMarkets). La première implémentation pratique d'un système AGV a été introduite en 1954 aux USA par Barrett-Cravens of Northbrook (actuellement Savant Automation Inc. Michigan, USA.). les premiers prototypes d'AGV sont équipés d'un simple système de guidage et de capteurs tactiles (*i.e.*, bumpers) (Ullrich 2015). L'idée de base était de remplacer les conducteurs des tracteurs de remorques par un robot autonome afin de transporter des biens sur de longues distances dans l'enceinte d'une industrie. Cependant, avec l'évolution des plateformes robotiques et des technologies de télécommunication et de localisation, les systèmes robotiques collaboratifs tels que les SMR mobiles, sont devenus très attractifs en raison de leurs potentiels pratiques dans divers applications (*e.g.*, industrielles, militaires, domestiques).

L'intérêt de la recherche académique pour les SMR date des années 1980. Depuis, le domaine a évolué vers une discipline de recherche à part entière, qui exploite un large spectre de théories de divers domaines de recherches tels que l'intelligence artificielle, les systèmes distribués, les théories de jeux et d'économie, la biologie et l'éthologie (Cao et al. 1997; Parker 2008). Les SMR offrent des avantages en termes de niveau d'autonomie, flexibilité et multi-tasking, permettant une large utilisation dans plusieurs applications telles que l'exploration et la cartographie (Puig et al. 2011; Luperto et al. 2020), la surveillance et la patrouille (Huang et al. 2019; Scherer and Rinner 2020), la recherche et le sauvetage des personnes dans les zones accidentées et dangereuses (Alotaibi et al. 2019; Peake et al. 2020), le nettoyage et décontamination (Le et al. 2020; Hu et al. 2020) (**cf. Figure 1.1**).



(a) SMR de patrouille de SMP Robotics (b) Exploration - Perseverance rover et ingenuity aircraft (c) Mission de secoure en Italie - Robot TRADAR

Figure 1.1. Exemple d'application de SMR.

Les SMR ont été parfaitement adaptés aux transport de biens dans les environnements *structurés* tels que: les industries (Lamballais et al. 2017; De Ryck et al. 2020a), la gestion des terminaux dans les ports (Carlo et al. 2014; Chen et al. 2020), les entrepôts et les centres de distribution (Boysen et al. 2019). Toutefois, ces dernières années, les SMR ont été utilisés comme système de transport dans des environnements *non-structurés* tels que les centres hospitaliers (Jeon et al. 2017; Søråa and Fostervold 2021). Ces exemples indiquent clairement l'impact socioéconomique que peut avoir les SMR. En effet, l'exemple du stress engendré par la pandémie du COVID-19 sur les centres hospitaliers et les centres de distribution de produits médicaux, causé en grande partie par le manque de personnels ou d'équipements de protection, a conduit à l'utilisation de systèmes robotiques (Tavakoli et al. 2020; Bačík et al. 2020; Alsamhi and Lee 2021; Kriegel et al. 2021; Fang et al. 2021) (**cf. Figure 1.2**).



(a) Usine AUDI Hongrie (b) SKYPOD - Exotec-Solution (c) Robot TUG de Aethon dans un centre hospitalier

Figure 1.2. Application des SMR dans le transport industriel et dans les centres hospitaliers.

3. Problème du transport : point de vue d'un système multi-robot

Le problème du transport est un cas spécial du problème d'assignement (Srinivasan and

Thompson 1973; Schrijver 2005; Pentico 2007). La première formulation du problème de transport en problème d'optimisation combinatoire a été proposée par le mathématicien français Gaspard Monge en 1784. Ce dernier a proposé la minimisation de la distance parcourue nécessaire pour transporter de la terre entre deux surfaces de superficies égale (Schrijver 2005). En 1941, Hitchcock a formulé le problème du transport en termes d'usines approvisionnant des villes en marchandises. Ce dernier a proposé une méthodes de calcul pour minimiser le coût de la distribution (Hitchcock 1941; Flood et al. 1953).

Du fait de son importance dans le domaine de la logistique et dans les déplacements de personnes a travers le monde, le problème du transport a été largement étudié ces dernières années (Hall 1999; Singh et al. 2019). De ce point de vue, le principal objectif du problème de transport est d'optimiser la distribution de biens (ou de personnes), des points sources vers des points destinations. Cependant, du point de vue d'un SMR utilisé comme système de transport dans un environnement industriel (*e.g.*, le cas des systèmes AGV), il est nécessaire de tenir compte de plusieurs facteurs, tels que la planification de chemins, le problème de la tournée (*i.e.*, vehicle routing problem), la coordination, l'allocation des tâches, la logistique et l'architecture de l'environnement (*e.g.*, le nombre de station de chargement et de déchargement), les caractéristiques des robots (le nombre nécessaire, la capacité) et la gestion du système (*e.g.*, gestion de l'énergie, gestion des défauts) (Vis 2006a; Le-Anh and De Koster 2006; De Ryck et al. 2020b). **La Figure 1.3** donne un aperçu de ces facteurs.

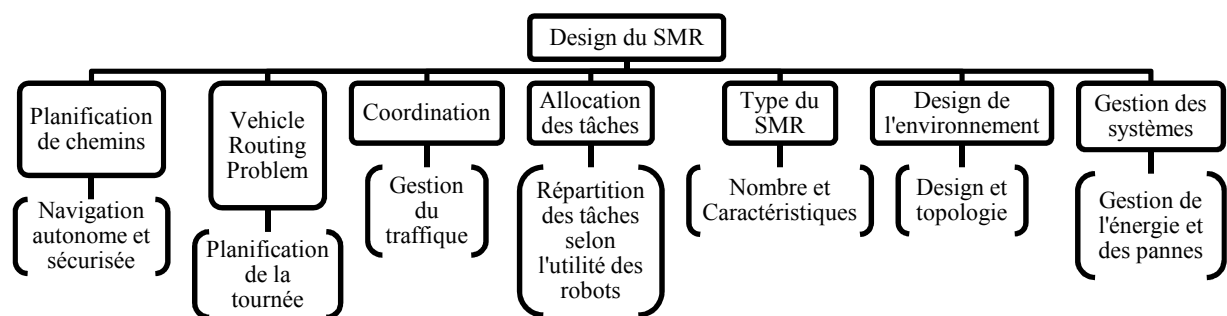


Figure 1.3. Facteurs de design d'un système de transport multi-robot industriel

Planification de chemins: Consiste à déterminer les chemins et les mouvements qui permettent aux robots de se déplacer d'un point initial à un point destination avec une navigation autonome et sécurisée (Madridano et al. 2021). Les méthodes de planification de

chemins classiques, initialement destinées pour les systèmes à un seul robot ont été largement adaptées pour les SMR telles que; l'algorithme A* (Fan Liu and Narayanan 2011; Fu et al. 2018), l'algorithme Dijkstra (Kim et al. 2019), champs de potentiel (Romero et al. 2012; Pashna et al. 2020), les heuristiques (Han and Yu 2019) et les techniques basées sur les graphes (Fransen et al. 2020). Néanmoins, de nouvelles techniques qui considèrent l'aspect multiple robot ont été proposées telles que; les algorithmes génétiques (Nazarahari et al. 2019), colonies de fourmis (Hao and Xu 2014), particle swarm optimization (Das and Jena 2020), mixed-integer linear programming (Asfora et al. 2020).

Le problème de tournée des robots: i.e., VRP pour vehicle routing problem, est un problème d'optimisation combinatoire. Il concerne le développement de stratégies de distribution de biens entre des points dépôts et des points sources par des véhicules en suivant un réseau de routes. La solution du VRP se résume dans l'optimisation d'une fonction objectif modélisant divers critères, tels que la distance totale du parcours et le coût de la tournée. La résolution de la version basic du VRP implique deux éléments : l'allocation de toutes les destinations dans un seul voyage (tour) et la séquence de visite des dépôts et destinations par les véhicules (Cordeau et al. 2007; Pollaris et al. 2015). Le VRP est d'une importance capitale dans le domaine du transport et de la logistique de distribution et de collecte. La littérature présente plusieurs variantes du VRP. La classification de ces variantes dépend de divers critères tels que; le type du réseau de distribution, le type de transport requis, les contraintes, la compositions de la flottes de véhicules et la fonction objectif (Irnich et al. 2014). Un exposé détaillé de l'état de l'art du VRP et des méthodes de résolution du problème est présenté dans (Toth and Vigo 2015).

Coordination: La coordination dans un SMR consiste dans la coopération entre les robots où chaque individu exécute des actions en considérant les actions des autres robots du groupe. L'objectif est d'avoir une cohérence dans les mouvements et les actions des robots. Cette coordination-coopération nécessite une forme de communication entre les robots, afin d'échanger des informations sur les états de chaque individu du groupe. La coordination efficace des actions et mouvements d'un SMR est une nécessité dans les applications, telles que la formation et agrégation (López-González et al. 2020; Beaver and Malikopoulos 2021), l'évitement d'obstacles (Heinemann et al. 2019; Scholz et al. 2019), la gestion du trafic (Herrero-Pérez and Martínez-Barberá 2011) et la résolution des conflits (Farinelli et al. 2017; Fanti et al. 2018).

Allocation des tâches: l'allocation des tâches consiste dans la répartition intelligente des tâches entre les robots dans le but d'optimiser les performances du SMR (Gerkey and Mataric 2003a). Dans les système AGV, on parle de règles de dispatching, qui sont des stratégies de sélection des véhicules pour l'exécution de tâches de transport (Vis 2006b; Le-Anh and De Koster 2006). Dépendant de l'application, les tâches peuvent être homogènes (*e.g.*, transport de biens) ou hétérogènes (*e.g.*, l'exploration et cartographie d'un environnement). En règles générales, le but est d'optimiser une fonction objectif afin de maximiser les profits et de minimiser les coûts. Dans le cas de l'allocation des tâches des SMR dans les industries (*e.g.*, les AGV et les règles de dispatching), les objectifs courants visés sont: minimiser le temps d'attente de chargement, maximiser le rendement, minimiser la longueur des files d'attente et minimiser le temps d'inactivité des robots (De Ryck et al. 2020b).

Type de SMR utilisé et design de l'environnement de travail: Ce facteur dépend largement du type de tâches exécutées. Le type de SMR utilisé joue un rôle important dans les performances du système. Plusieurs critères sont à considérer, tels que le nombre et le type de robots (Ji and Xia 2010; Valmiki et al. 2018; Fethi and Mehdi 2019). D'autre part, l'architecture de l'environnement de travail du SMR, notamment dans les environnements *structurés*, est un facteur essentiel dans l'utilisation d'un SMR. Le nombre et la localisation des stations de chargement et de déchargement influencent les performances opérationnelles du SMR, tels que la distance à parcourir entre les stations et le temps d'attente pour le chargement. Dans le cas des AGV, le design des bandes de guidage (*e.g.*, optique, induction, magnétique) prédétermine les stratégies de coordinations et de planification de chemins. Néanmoins, l'évolution de la robotique et des techniques de localisation (*e.g.*, laser, GPS, vision) apportent une flexibilité aux systèmes AGV où les changements dans les stratégies de coordination et de planification de chemins sont réalisés en ligne grâce aux softwares de design graphique (Martínez-Barberá and Herrero-Pérez 2010; Németh et al. 2019).

Gestion des défauts et diagnostic: comme tous système, les SMR sont sujets aux pannes et défauts qui peuvent survenir de manière imprévue. Ce qui peut causer des deadlocks dans le système ainsi que des retards dans l'exécution des tâches. Dans les systèmes de transport, les défauts sont définis par une violation non-permise d'en moins un temps de transport du système de sa valeur de référence nominale (Witczak et al. 2020). Avec l'utilisation d'important SMR en termes de nombre de robots, des stratégies ont été développées dans la

littérature pour assurer la robustesse des systèmes face aux défauts et pannes (Yan and Li 2012; Arrichiello et al. 2014; Yan et al. 2017; Witczak et al. 2018, 2019).

Gestion de l'énergie : ce facteur concerne la gestion du système énergétique du SMR incluant les batteries, la logistique de rechargements, et la consommation de l'énergie. La gestion de l'énergie peut influencer l'exécution des tâches ainsi que les performances du SMR. Cependant, la revue de la littérature des SMR révèle que la gestion de l'énergie n'est pas suffisamment traitée. En effet, les enquêtes menées par Vis (Vis 2006b) et Le-Anh *et al.* (Le-Anh and De Koster 2006) affirment que la gestion de l'énergie dans les systèmes AGV est peu adressée dans la littérature. De Ryck *et al.* soutient les mêmes conclusions dans leur étude de l'état de l'art des AGV (De Ryck et al. 2020b).

Dans cette optique, notre présente thèse explore deux éléments fondamentaux de design d'un système de transport multi-robot mobile dans un environnement d'intérieur (*e.g.*, entrepôt) à savoir l'allocation des tâches (*i.e.*, *Soit n robots et m tâches, quel robot doit être affecté à quelle tâche pour atteindre un objectif global?*), et la gestion de l'énergie du SMR (*i.e.*, *Gestion de la consommation d'énergie des robots afin d'optimiser la production*). Ces deux éléments sont revus en détail ci-après.

4. Allocation des tâches des systèmes multi robots

Aussi simple de design un SMR, affichant un comportement coopérative, doit être capable de déterminer quel robot du groupe doit exécuter quelle tâche (Cao et al. 1997b; Gerkey and Mataric 2003b). Les tâches devraient être réparties intelligemment entre les robots qui peuvent les accomplir efficacement pour des performances optimales. Cette répartition des tâches est connue dans la littérature comme le problème d'allocation des tâches des systèmes multi-robot, *e.i.*, MRTA pour Multi-Robot Task Allocation.

Le MRTA dépend du problème résolu par le SMR. Il est caractérisé par le type de robots, le type de tâches à accomplir ainsi que le type d'allocation. Suivant ces paramètres, Gerkey et Mataric ont proposé dans (Gerkey and Mataric 2004a) une taxonomie basée sur trois axes. **La figure 1.4** illustre cette taxonomie et les axes sont défini ci-après.

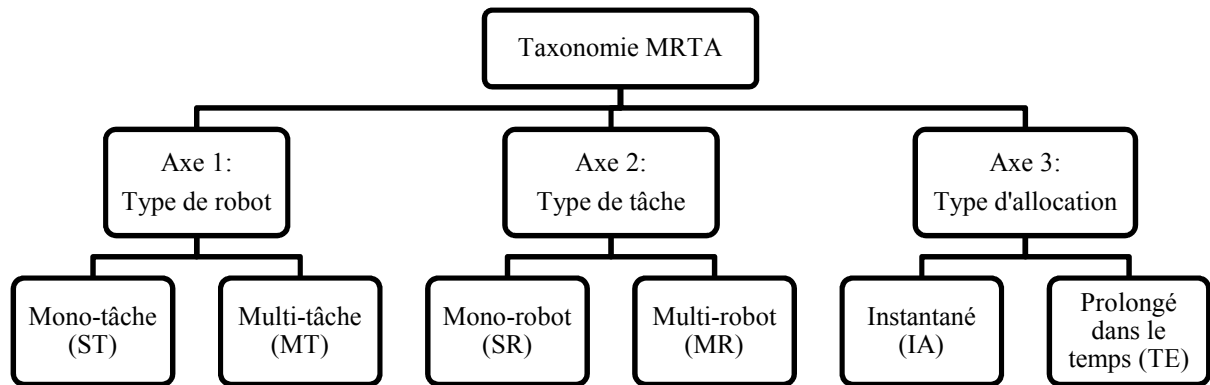


Figure 1.4. Taxonomie du MRTA (Gerkey and Mataric 2004a).

- *Robots mono-tâche (ST pour single-task) vs. robots multi-tâche (MT pour multi-task):* un SMR-ST est composé de robots qui peuvent exécuter au plus une tâche à la fois. Alors qu'un SMR-MT peut être composé de robots capables d'accomplir plusieurs tâches simultanément.
- *Tâches mono-robot (SR pour single-robot) vs. Tâches multi-robot (MR pour multi-robot):* Les tâches SR requièrent exactement un seul robot pour être exécuter. Alors qu'un ensemble de tâches MR peut contenir des tâches nécessitant l'implication de plusieurs robots en même temps pour les accomplir.
- *Allocation instantanée (IA, pour instantaneous assignment) vs. allocation prolongée dans le temps (TA, pour time-extended assignment) :* Dans une allocation de type IA, les informations concernant les robots, les tâches et l'environnement ne permettent que des allocations instantanées sans la considération des future allocations possibles (*i.e.*, sans planning). D'autre part, une allocation de type TA permet une planification de l'allocation des tâches aux robots, grâce aux informations supplémentaires disponibles sur l'ensemble des tâches existantes, ou bien grâce à un modèle de prévision de la disponibilité dans le temps de ces tâches.

La taxonomie proposée par Gerkey et Mataric est largement acceptée par la littérature académique et permet la classification de plusieurs instances du MRTA. Cependant, il convient de mentionner deux récents travaux qui ajoutent des niveaux supplémentaires à la taxonomie de Gerkey et Mataric. Dans le premier travail, Korsah *et al.* ont proposé une taxonomie, iTax, qui considère explicitement l'interdépendance des utilités et des contraintes (Korsah et al. 2013). Le deuxième travail est une extension de la classe TA, qui inclue les contraintes temporelles et d'ordonnancement est proposée dans (Nunes et al. 2017).

La littérature propose plusieurs méthodes pour résoudre le problème du MRTA. Ces dernières peuvent être classées en trois approches: approche comportementales, approche basée sur les marchés, et les techniques d'optimisation (Gerkey and Mataric 2003b; Dias and Stentz 2003; Korsah et al. 2013; Khamis et al. 2015).

Approches comportementales (behaviour-based): Ces approches permettent à chaque robot une planification et exécution des tâches indépendamment des autres robots du groupe. En général, les robots n'utilisent pas un échange explicite d'informations sur leurs tâches individuelles. Toutefois, chaque robot exploite les informations de l'état actuel des tâches, de ses propres actions, ainsi que des aptitudes estimées des autres membres du SMR. L'une des méthodes comportementales notoire dans la littérature est l'architecture ALLIANCE (Parker 1998). Dans cette architecture, les robots considèrent à chaque itération toutes les tâches non achevées pour une allocation. Chaque robot estime son utilité relative pour exécuter une tâche et les utilités relatives des autres robots, grâce à l'utilisation de degrés variables de comportements de motivation tels que l'impatience et le consentement. A titre d'exemple, l'impatience des robots sur une tâche augmente avec le retard dans l'exécution de celle-ci par un robot du groupe, jusqu'à atteindre un seuil limite. Par conséquent, la tâche est considérée non-accomplie et allouée à un autre robot.

Approches basées sur les marchés (Market-based): elles sont inspirées des théories économiques de marché. Leur principe repose sur l'exécution de tours de négociations (*i.e.*, auctions) entre les robots afin d'attribuer les tâches entre eux. En général, lors du processus de négociation, un robot joue le rôle de trader (*i.e.*, auctioneer) et assume la responsabilité de gérer la négociation entre les robots participant qui jouent le rôle d'acheteurs (*i.e.*, bidders). L'objectif de la négociation est d'optimiser une fonction objectif qui modélise l'utilité des robots à performer une tâche. Ces approches utilisent généralement des algorithmes de type glouton (*i.e.*, greedy) pour allouer les tâches aux robots. D'autre part, le groupe de robots a besoin d'une communication explicite pour réaliser ces négociations. Une introduction aux approches basées sur les marchés est proposée dans (Dias et al. 2006). Plusieurs méthodes basées sur ces approches sont proposées dans la littérature telle que; le contract net protocol (CNP) qui est appliqué aux systèmes multi-agents (Davis and Smith 1983), M+ (Botelho and Alami 1999), MURDOCK (Gerkey and Mataric 2002), TraderBots (Dias 2004), et le sequential single-item auction (Koenig et al. 2006).

Techniques d'optimisations: Le MRTA peut être traité comme un problème d'optimisation combinatoire. La résolution de cette classe de problèmes consiste dans le choix d'une meilleure solution dans un ensemble fini de solutions possibles. La littérature relève deux types de méthodes: les méthodes exactes et les méthodes approximatives. Les méthodes exactes donnent des solutions optimales et garantissent leurs optimalités. D'autre part, les méthodes approximatives génèrent des solutions de qualités dans un temps acceptable. Cependant, elles ne garantissent pas des solutions optimales globales (Talbi 2009). **La figure 1.5** illustre les différentes méthodes d'optimisations répertoriées dans la littérature. Plusieurs de ces méthodes ont été adopter pour résoudre le MRTA, telle que le mixed-integer linear programming (Atay and Bayazit 2006; Asfora et al. 2020), l'algorithme hongrois (Krause 2007) et les méthodes graphiques (Farinelli et al. 2017). Dans la classe des méthodes approximatives, on peut citer; les méthodes basées sur des heuristiques (Liu et al. 2013; Sarkar et al. 2018; Han and Yu 2019), les algorithmes évolutionnistes (Arif 2017; Soleimanpour-Moghadam and Nezamabadi-Pour 2020), le particle swarm optimization (Kong et al. 2019; Nedjah et al. 2021), les algorithmes de colonies de fourmis (Li et al. 2018), le recuit simulé (simulated annealing) (Mosteo 2006; Yu et al. 2021), la recherche tabou (Cortés et al. 2017).

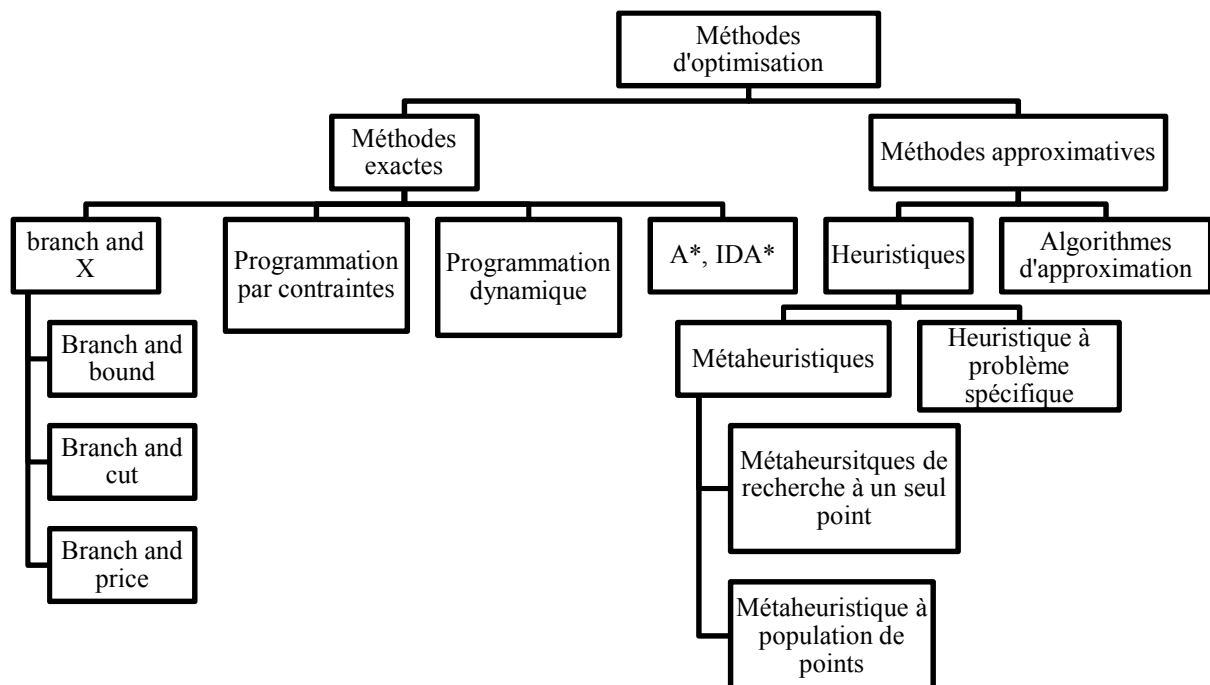


Figure 1.5. Classification des Méthodes d'optimisation classiques (Talbi 2009).

Le **tableau 1.1** présente une comparaison entre les trois approches de résolution du MRTA suivant différents critères incluant; l'architecture du SMR, le type du SMR, la taille de la flotte en terme de nombre de robots, le temps de calcul, la qualité de la solution.

Tableau 1.1. Comparatif des approches de résolution du MRTA.

Critères / Approche	comportementale	Basée sur les marchés	Techniques d'optimisation	
			Exactes	Approximatives
Architecture du SMR	Décentralisée	Décentralisée / Centralisée	Centralisée	Centralisée/ décentralisée pour certaines méthodes
Type du SMR	Hétérogène / Homogène	Hétérogène / Homogène	Homogène	Homogène / hétérogène pour certaines méthodes
Taille de la flotte	Petite	Moyenne / Large	Petite	Moyenne / Large
Temps de calcul	Très réduit	dépendant de l'algorithme	Très Long	Dépendant de l'algorithme
Qualité de la solution	Hautement sous-optimale	Hautement sous-optimale	Exacte	Bonne approximation
Implémentation dans les SMR industriels	Non	Oui	Oui	Oui

Dans cette thèse, le MRTA est exprimé sous forme d'un problème d'assignement classique (Classical Assignment Problem, CAP) qui est un cas particulier du problème d'assignement généralisé (Generalized Assignment Problem, GAP). Le GAP est très répondu dans le domaine de l'optimisation combinatoire (Cattrysse and Van Wassenhove 1992; Schrijver 2005; Pentico 2007). Le GAP est un problème de type NP-Hard (Sahni and Gonzalez 1976). Cependant, certains cas tel que le CAP peuvent être résolus en un temps polynomiale avec des méthodes exactes utilisant des algorithmes tels que branch and boundry, Hungarian, simplex, et Lagrangian relaxation (Davis et al. 1992; Kundakcioglu and Alizamir 2009). Bien que ces méthodes peuvent garantir une solution optimale, elles nécessitent un temps de calcul important. De plus, la plupart des applications temps-réel exigent des stratégies qui assurent des allocations rapides et valides, plutôt que des solutions exactes (Jones et al. 2002; O'Leary 2019). A cet effet, les heuristiques, telles que l'algorithme glouton (greedy algorithm), peuvent produire rapidement une solution sous-optimale. Néanmoins, elles ont l'inconvénient de se bloquer dans un minimum local. Par conséquent, l'emploi de métaheuristiques est plus approprié.

Les métaheuristiques sont une classe d'algorithmes d'optimisation. Elles sont définies comme des stratégies de niveau supérieur qui guident des heuristiques spécifiques pour améliorer leurs performances, et cela, via une exploration efficace de l'espace de recherche (Blum and Roli 2003; Talbi 2009). Ces stratégies reposent sur la gestion dynamique d'un équilibre entre l'exploration de l'espace de recherche et l'exploitation des régions local de cet espace. Cet équilibre permet aux métaheuristiques d'identifier rapidement les régions de l'espace de recherche contenant des solutions de bonne qualité. D'autre part, il permet aux métaheuristiques de s'échapper de régions déjà explorées ou bien ne contenant pas de bonnes solutions (Blum and Roli 2003; Talbi 2009).

Les métaheuristiques peuvent être classifiées selon le nombre de solutions. La littérature identifie deux classes: recherche à un seul point vs. méthodes à population de points (single-point search vs. population based). Les méthodes basées sur la recherche à un seul point (i.e., méthodes basées sur un parcours), telles que la recherche tabou et la recherche locale itérative (Iterative Local Search, ILS), manipulent un seul point à la fois durant la recherche de solution dans l'espace de recherche. En revanche, Les méthodes à population de points (e.g., essaim de particules, algorithmes génétique) manipulent une population de solutions à la fois. Ces méthodes sont orientées vers l'exploration intensive de l'espace de recherche.

5. Gestion de l'énergie d'un système multi-robot

Le deuxième élément de design d'un SMR exploré par cette thèse concerne la gestion de l'énergie d'un SMR. En général, la gestion de l'énergie (e.g., gestion des batteries) d'un SMR, si elle n'est pas radicalement omise de l'architecture de contrôle, est considérée comme une simple sécurité pour éviter la défaillance des robots lors de l'exécution des tâches.

Dans un SMR, les robots complètent les tâches allouées à eux jusqu'à l'épuisement de leurs batteries. Puis, ils se dirigent vers les stations de rechargement afin de recharger leurs batteries. Durant la période de rechargement, les robots sont temporairement indisponibles. Ce qui réduit le nombre de robots fonctionnels et diminue la productivité du SMR. Par conséquent, la gestion des batteries a un impact sur les performances d'un SMR. L'importance de la gestion des batteries d'un SMR a été mis en évidence par McHaney (McHANEY 1995) et Ebben (Ebben 2001). En particulier, les simulations conduites par McHaney ont permis de conclure que la considération de l'état des batteries, comme contrainte dans l'architecture de

contrôle d'un système AGV, exige une augmentation du nombre de robots utilisés pour maintenir un rendement égal à une simulation sans contraintes de batterie.

Récemment, la littérature a montré un intérêt pour la gestion du système énergétique d'un SMR. Parmi les travaux liés à la gestion de l'énergie d'un SMR, on peut citer ceux de Brito *et al.* dans (Brito et al. 2019) sur la distribution et le nombre de stations de rechargement. Les auteurs ont proposé deux algorithmes pour positionner les stations de rechargement de type statiques et dynamiques, afin d'améliorer les performances du SMR.

Zou *et al.* ont évalué deux stratégies de rechargement pour un système RMFS (Robotic Mobile Fulfillment Systems), le rechargement automatique et la permutation de batteries (*i.e.*, battery swapping) (Zou et al. 2018). Dans la stratégie de permutation des batteries, les robots changent leurs batteries déchargées par d'autres pré-chargées. D'autre part, dans le rechargement automatique, les robots gardent leurs batteries et le rechargement s'effectue dans une station de rechargement.

Les auteurs dans (Kabir and Suzuki 2018) ont étudié l'impact de la durée de rechargement des batteries d'un AGV sur la flexibilité d'un système de fabrication. Le concept proposé par les auteurs repose sur le fait que les batteries de type plomb-acide sulfurique, utilisées généralement sur les AGV, reçoivent une quantité de charge plus importante durant la phase initial de rechargement, contrairement aux phases suivantes. Par conséquent, les auteurs affirment que l'augmentation de la productivité du système peut être obtenue en réduisant la durée de rechargement des batteries des AGV. Suivant le même aspect lié au rechargement des batteries, les auteurs dans (Kawakami and Takata 2012) soulignent l'importance du choix des périodes et d'intervalles de rechargements des batteries afin d'éviter la détérioration des batteries et de prolonger leurs durée de vie. Les auteurs ont développé une simulation de gestion des batteries des systèmes AGV pour évaluer les coûts liés aux batteries. Cette simulation permet la planification d'une stratégie de gestion des batteries d'un système AGV qui minimise les coûts liés aux batteries incluant, le coût des batteries et des chargeurs, le coût de rechargement des batteries, les pertes dues à l'arrêt des AGV et les pertes en temps de travail liées aux temps de rechargement des batteries.

Par ailleurs, Berenz *et al.* ont proposé dans (Berenz et al. 2012) une stratégie de gestion des rechargements de la batterie d'un robot mobile. Cette stratégie estime le paramètre risque/gain sur la décision de rediriger le robot vers une station de rechargement de batteries et permet de garantir un compromis entre le risque et le taux d'accomplissement de la tâche du robot. Les auteurs proposent une fonction de densité de probabilité qui évalue l'incertitude sur

la capacité effective de la batterie, le taux de déchargement de la batterie et l'énergie nécessaire pour que le robot atteigne une station de rechargement.

Parmi d'autres facteurs liés à la gestion du système énergétique d'un SMR traités dans la littérature, on peut citer la résolution de conflit sur les ressources. Les conflits entre les robots peuvent survenir dans les situations où une ressource (*e.g.*, station de rechargement de batteries) est requise par plus d'un robot en même temps. Les auteurs dans (Iftikhar and Khan 2016) ont proposé une stratégie inspirée des approches basées sur les marchés pour résoudre les conflits sur les stations de rechargement et éviter les situations de deadlocks entre les robots.

Selon les travaux cités précédemment liés à l'étude du système énergétique d'un SMR, la majeure partie propose d'optimiser la gestion de l'énergie d'un SMR en termes des méthodes de rechargement, logistique et hardware (*e.g.*, batteries, station de rechargement). Cependant, du fait du caractère dynamique de l'environnement d'évolution des SMR, ces approches ne garantissent pas nécessairement des performances élevées et une consommation d'énergie optimale pour un SMR. Par ailleurs, la consommation d'énergie est considérée durant la phase de contrôle des systèmes. Par conséquent, la marge possible d'optimisation d'un SMR est exclusivement limitée durant l'exécution des tâches et seulement après la phase de planification et de prise de décisions.

En effet, si nous examinons le cas de l'optimisation du nombre et de la distribution des stations de rechargement de batteries, le design du SMR peut certainement être amélioré. Cependant, cette optimisation est statique. De plus, l'évolution en temps réel du système ne permet pas une adaptation aux changements dynamiques. Étant donné que, pour optimiser ces facteurs (*e.g.*, ajouter une station de rechargement), il est nécessaire d'aménager l'architecture de l'environnement du SMR. Ce qui n'est pas toujours pratique et ne peut être réalisable du point de vue d'une industrie.

Ainsi, afin d'assurer plus de flexibilité dans le contrôle du système énergétique d'un SMR en général et, en particulier, d'optimiser la gestion de sa consommation d'énergie, il est plus astucieux de considérer la gestion de la consommation d'énergie d'un SMR dans la phase de décision (*i.e.*, durant l'allocation des tâches). En effet, cette approche permettra aux robots de considérer leurs propres ressources énergétiques et de répondre convenablement aux changements selon leurs états.

Cette approche a récemment attiré l'attention de la recherche académique. Dans (Palmer et al. 2018), les auteurs considèrent la gestion des conflits sur les ressources (*e.g.*, les

station de chargement des batteries) dans le processus de résolution du MRTA. Ils proposent une méthode analytique pour estimer le coût de l'allocation des tâches en modélisant explicitement l'incertitude sur le temps de déplacement et la durée des tâches du robot. Deux objectifs sont considérés : minimiser les coûts de l'exécution des tâches et minimiser les coûts générés par un possible dépassement du délai limite d'exécution de la tâche (tardiness cost minimisation). Cependant, l'approche développée n'inclue pas explicitement la consommation d'énergie comme facteur. Ce qui peut causer un impact négatif sur l'efficacité de la gestion des ressources proposée (*i.e.*, la consommation d'énergie).

En revanche, les auteurs dans (Chen et al. 2018) proposent une méthode de résolution du MRTA basée sur l'optimisation multi-objectifs qui considère explicitement la consommation du temps et d'énergie dans le processus de l'allocation des tâches. A cet effet, deux fonctions d'utilité sont construites pour estimer la consommation en temps et en énergie des robots. Dans leurs conclusions les auteurs rapportent que la consommation du temps et de l'énergie sont des indicateurs importants dans l'évaluation de l'efficacité de la solution du MRTA et affirment que leur effet a un impact direct sur cette solution. Cependant, l'étude ne propose pas une évaluation sur la productivité effective ainsi que la consommation en énergie des robots. Ce qui ne permet pas d'évaluer les performances pratiques du SMR.

Par ailleurs, parmi d'autres travaux suivant la même approche, une stratégie d'allocation des tâche pour les SMR hétérogènes est proposée par Sullivan *et al.* dans (Sullivan et al. 2019). La méthode proposée est basée sur le *Sequential Single-Item auction*. C'est une méthode qui est basée sur les enchères, (*i.e.*, théories des marchés) essentiellement dédiée pour les SMR homogènes. Néanmoins, les auteurs proposent de nouvelles règles d'enchères afin d'adapter la méthode aux SMR hétérogènes. Deux objectifs sont visés : la minimisation de la consommation d'énergie et la minimisation du temps d'achèvement des tâches. Les résultats de la méthode sont comparés à divers règles et protocoles de gestion des enchères. Cependant, la stratégie proposée ne considère pas l'épuisement des batteries et la limite de stockage d'énergie disponible qui sont essentiels à l'évolution en temps réel de l'énergie du SMR.

6. Conclusion

Selon la revue de la littérature exposée dans ce chapitre, les SMR ont été parfaitement adaptés au transport de biens dans les environnements *structurés* et *non-structurés*. Le design de ce type de systèmes de transport nécessite de tenir compte de plusieurs facteurs incluant la

planification de chemins, le problème de la tournée, la coordination, l'allocation des tâches, l'architecture et la logistique de l'environnement, les caractéristiques des robots et la gestion du système.

Dans la quatrième partie du chapitre, nous avons revu la littérature sur le problème d'allocation des tâches des SMR ainsi que des méthodes de résolution de ce problème. En outre, nous avons présenté une comparaison de ces approches de résolution de l'allocation des tâches suivant différents critères.

Enfin dans la cinquième et dernière partie du chapitre, nous avons présenté l'état de l'art de la gestion de l'énergie des SMR. A cet effet, La littérature révèle que la gestion de d'un SMR, si elle n'est pas radicalement omise de l'architecture de contrôle, elle est considérée comme une simple sécurité pour éviter la défaillance des robots lors de l'exécution des tâches. Par ailleurs, la majeure partie des études liées à la gestion de l'énergie des SMR proposent des approches d'optimisation en termes des méthodes de rechargement, logistique et hardware. De plus, la consommation d'énergie est considérée durant la phase de contrôle des systèmes, ce qui restreint l'optimisation d'un SMR durant l'exécution des tâches et seulement après la phase de planification et de prise de décisions.

Dans cette optique, l'objectif du chapitre suivant est de présenter la nouvelle stratégie d'allocation des tâches (D-ILS). Cette stratégie maximise la productivité et d'optimise la gestion de la consommation d'énergie du SMR, en considérant la consommation d'énergie explicitement dans la phase de décision des allocations des tâches. En outre, les facteurs de niveau d'énergie ainsi que son épuisement sont considérés explicitement dans l'estimation de l'utilité du robot et implicitement lors de l'allocation des tâches.

CHAPITRE 2. Stratégie D-ILS pour la résolution du MRTA à base de la gestion de l'énergie

CHAPITRE 2. Stratégie D-ILS pour la résolution du MRTA à base de la gestion de l'énergie

1. Introduction

Ce chapitre se focalise sur la présentation de la stratégie d'allocation des tâches D-ILS selon les travaux réalisés dans (Djenadi and Mendil 2021). A travers la première section, nous décrivons le problème de transport dans les SMR. Puis dans la section 2, nous donnons le modèle MRTA de base utilisé. Dans la section 3, nous formulons une nouvelle fonction d'utilité (*i.e.*, fonction objectif) associée au problème traité. Le protocole d'allocation des tâches de la stratégie D-ILS et le mécanisme de réallocation des tâches entre les robots implémentés dans celle-ci sont présentés dans les sections 4 et 5.

2. Système de transport multi-robot : Description du problème

Le transport dans les environnements industriels concerne le transport de biens entre différentes positions de la manufacture. Dans un environnement structuré, tel que les entrepôts et les centres de distribution. Le flux de circulation des biens peut être résumé comme suit : après le déchargement des biens reçus par le centre de distribution, les biens sont labellisés et transportés dans des zones de stockages et rangés suivant des règles de stockage précises. Lorsqu'une commande est formulée pour des biens stockés, ces derniers sont transportés des zones de stockages vers les stations d'emballage et d'expédition.

Ainsi que mentionné précédemment, du point de vue d'un SMR, plusieurs facteurs doivent être considérés incluant la navigation, la planification de chemins, la coordination et l'allocation des tâches. Cette thèse se focalise sur la résolution du problème de l'allocation des tâches entre les robots, le MRTA. En particulier, un groupe de robots mobiles homogènes est considéré pour des tâches de transport de biens dans un environnement d'intérieur (*e.g.*, entrepôt). Les tâches concernent le transport d'objets de stations de chargement vers des stations de déchargement. Chaque tâche est associée à une valeur de priorité. L'exécution d'une tâche n'est pas conditionnée par l'exécution des autres. En d'autres termes, les tâches ne sont pas interdépendantes.

Le problème traité dans ce présent travail peut être formulé comme suit: soit un ensemble m de tâches $T = \{t_1, \dots, t_m\}$ et un ensemble n de robots $R = \{r_1, \dots, r_n\}$. Chaque tâche t_j est définie par $t_j = \{tID_j, Q_j, load_j, unload_j\}$, avec tID_j est l'identifiant de t_j dans

l'ensemble T , Q_j est la valeur de la priorité associée à la tâche t_j , $load_j$ est la position de la station de chargement de la tâche t_j , et $unload_j$ est la position de la station de déchargement de la tâche t_j . Dans l'ensemble de robots R , chaque robot r_i est défini par $r_i = \{rID_i, Rlist_i, d_{ij}, E_i, Emax_i\}$, avec rID_i est l'identifiant du robot r_i dans l'ensemble R , $Rlist_i$ est la liste de robots disponibles pour l'allocation des tâches, d_{ij} est la distance cartésienne que le robot r_i doit parcourir pour accomplir la tâche t_j depuis sa position actuelle, E_i est l'actuelle énergie consommée par le robot r_i depuis le commencement de la mission, et $Emax_i$ est l'énergie maximale initiale disponible que le robot peut utiliser avant d'atteindre le seuil d'épuisement de la batterie.

Dans le cas général, la consommation d'énergie d'un SMR dépend des activités de chaque robot (*e.g.*, déplacement, soulever et abaisser un objet). L'objectif du travail de cette thèse est de tester et de valider la stratégie D-ILS proposée. A cet effet, les hypothèses suivantes sont considérées :

- Seule la consommation d'énergie de l'activité de déplacement est prise en considération, étant donné que cette activité consomme le plus d'énergie (McHANEY 1995).
- La consommation des robots durant le déplacement avec charge est égale à la consommation durant le déplacement sans charge.
- La consommation d'énergie d'un robot est une fonction linéaire de la distance parcourue.

3. Formulation du MRTA

Pour élaborer la stratégie D-ILS, nous avons utilisé la formulation du MRTA présentée par Gerkey *et al.* dans (Gerkey and Mataric 2003a). Soit n robots et m tâches associées à des valeurs de priorités, le problème à résoudre est défini comme suit : "*quel robot doit être affecté à quelle tâche pour maximiser un objectif global ?*".

Dans la stratégie D-ILS, il est supposé que chaque robot soit capable d'exécuter une seule tâche à la fois et chaque tâche nécessite un seul robot pour l'accomplir. La stratégie D-ILS repose sur la résolution en temps réel du problème MRTA où chaque robot impliqué dans le processus d'allocation des tâches utilise les informations de son état et de son environnement, ainsi que les caractéristiques des tâches disponibles.

En suivant la taxonomie présentée dans (Gerkey and Mataric 2004b), le modèle MRTA appliqué dans la stratégie D-ILS est du type mono-robot et mono-tâche avec allocation instantanée (single-robot, single-task et instantaneous assignment : SR-ST-IA). Par ailleurs, en supposant que chaque robot r_i est capable d'estimer la valeur de son utilité u_{ij} à exécuter chaque tâche t_j . Ainsi, une matrice d'utilité du SMR, $\mathbf{U} = \{u_{ij}\}, 1 \leq i \leq n, 1 \leq j \leq m$, est construite.

Suivant cette formulation, l'objectif est de trouver les paires robot-tâche $(i_1, j_1) \dots (i_k, j_k), 1 \leq k \leq \min(n, m)$ qui maximisent l'utilité globale du SMR formulée par **l'équation (2.1)**. Par conséquent, suivant le modèle SR-ST-IA du MRTA traité, les paires (i_k, j_k) doivent être uniques.

$$\max \sum_{m=1}^k u_{i_m j_m} \quad (2.1)$$

4. Fonction d'utilité de gestion de l'énergie

Les applications temps-réel ainsi que l'environnement d'évolution des SMR nécessitent des stratégies qui assurent des résultats d'allocation rapides et valides. Dans la stratégie D-ILS proposée, l'objectif est de réaliser une allocation des tâches en temps réel avec une solution raisonnable. Par conséquent, une fonction d'utilité est proposée en se basant sur l'essentiel des informations que le robot peut obtenir de son état et de son environnement. Suivant les objectifs fixés précédemment (*i.e.*, maximiser la productivité, optimiser la gestion de la consommation d'énergie), la fonction d'utilité est formulée pour refléter les règles basiques suivantes :

- Plus la valeur de d_{ij} augmente, plus la valeur de u_{ij} diminue.
- Plus la valeur de Q_j augmente, plus la valeur de u_{ij} augmente.
- Plus la valeur de E_i augmente, plus la valeur de u_{ij} diminue.
- Plus la valeur du rapport $\frac{\text{nombre de robots disponibles (nbrR)}}{\text{nombre de tâches (nbrT)}}$ est importante, les robots sont plus enclins à choisir des tâches avec des priorités importantes.
- Plus la valeur du rapport $\frac{\text{nombre de robots disponibles (nbrR)}}{\text{nombre de tâches (nbrT)}}$ est petite, les robots sont plus enclins à choisir des tâches avec des distances de parcours courtes.

En outre, la supposition suivante est imposée : $\frac{\text{nombre de tâches (nbrT)}}{\text{nombre de robots disponibles (nbrR)}} \geq 1$. La fonction d'utilité est exprimée alors comme suit

$$h_{ij} = \frac{\frac{(\text{nbrT} + \text{nbrR})}{\text{nbrR}}}{\exp(d_{ij})} + \left(\frac{\text{nbrT} + \text{nbrR}}{\text{nbrR}} \right) Q_j^2 - (d_{ij}^2 - 1) \left(\frac{E_i}{E_{\max_i}} + 0.1 \right) - \left(\frac{E_i^2}{E_{\max_i}^2} - 1 \right) (d_{ij} + 0.1) \quad (2.2)$$

Cependant, si le robot ne possède pas suffisamment d'énergie pour accomplir la tâche t_j , la valeur de son utilité devient nulle. Alors, la formulation complète de la fonction d'utilité devient:

$$u_{ij} = \begin{cases} h_{ij} & \text{si } E_{\max_i} - E_i \geq d_{ij} \\ 0 & \text{autres} \end{cases} \quad (2.3)$$

5. Allocation des tâches distribuée: Protocole de la stratégie D-ILS

L'architecture de contrôle d'un SMR est fondamentale pour la conception du système. Ce facteur a un impact direct sur la robustesse et la scalabilité du système (Parker 2008). L'architecture du SMR fournit l'infrastructure de l'implémentation et de l'évolution du groupe de robot (Cao et al. 1997b). En d'autres termes, elle décrit les mécanismes d'interaction entre les robots du groupe et définit les aptitudes et les limites du système. Ce facteur englobe des éléments tels que la communication et le type des robots (Homogènes vs. Hétérogènes). La littérature relève deux types d'architectures, centralisée et décentralisée.

L'architecture centralisée, repose sur un contrôle centralisé réalisé grâce à une entité centrale (e.g., robot, point de contrôle). Dans ce type d'architecture, les robots du groupe communiquent les informations nécessaires à l'agent de contrôle centralisé. Ce dernier traite les informations reçues et produit les commandes appropriées pour chaque robot du groupe. Puis, il communique ces commandes aux robots. Au finale, ces derniers exécutent directement les tâches. Cette architecture présente des avantages non négligeables dans les cas des systèmes avec un nombre de robots et de tâches réduit et ou l'environnement de travail est statique. Par ailleurs, cette architecture est largement traitée dans la littérature (Jose and

Pratihar 2016; Li et al. 2017; Rehman et al. 2019; Simões et al. 2020; Sayed et al. 2020). Cependant, elle présente un grand défaut pratique en termes de robustesse et de flexibilité, dû à la vulnérabilité du point de contrôle aux pannes et défauts pouvant paralyser tout le système.

L'architecture décentralisée, est basée sur une configuration de contrôle décentralisée entre les individus du SMR. Cette architecture permet à chaque robot de décider des actions et de les exécuter indépendamment des autres robots du groupe, en s'appuyant simplement sur les informations locales de sa situation. Toutefois, afin d'avoir une cohérence dans l'exécution de ces actions (*i.e.*, tâches), les robots peuvent communiquer entre eux des informations utiles pour atteindre un objectif global, sans perdre l'aspect distribué de la prise de décision (Herrero-Pérez and Martínez-Barberá 2011; Draganjac et al. 2016; Batinović et al. 2020). Cependant, cet état de cohésion globale entre les robots est difficilement réalisable. Par ailleurs, les solutions locales produites par chaque individu ne sont pas nécessairement une solution globale.

Malgré ces points négatifs, l'architecture décentralisée présente les caractéristiques recherchées dans le concept de l'industrie 4.0 des industries modernes en termes de flexibilité et de robustesse aux défauts et pannes. Étant donnée l'absence d'un contrôleur central, la défection d'un individu du SMR ne résulte pas dans l'arrêt total du système. Par ailleurs, la littérature admet les limites de l'architecture centralisée dans les manufactures modernes requérant plus de flexibilité (Hermann et al. 2016; Meissner et al. 2017; Kattepur et al. 2018; Ljasenko et al. 2019).

La stratégie D-ILS proposée dans cette thèse est structurée suivant une architecture décentralisée. Les robots impliqués dans le processus du MRTA participent d'une manière distribuée dans la prise de décision. Dans cette stratégie, chaque robot utilise un algorithme basé sur l'ILS afin de résoudre le MRTA. Cependant, le résultat global généré par la stratégie D-ILS s'apparente aux résultats d'une métaheuristique à population de solutions, où chaque solution produite par un robot avec l'ILS peut être considérée comme une solution-individu d'une population de solutions. Ainsi, l'architecture de la D-ILS permet une prise de décision décentralisée, grâce aux multiples solutions proposées par les robots du SMR. Par conséquent, la défection d'un membre du SMR ne causera pas une défaillance dans la décision finale du groupe de robots. Par ailleurs, du point de vue optimisation, la stratégie D-ILS améliore l'exploitation de l'espace de recherche grâce à l'ILS. D'autre part, elle augmente l'exploration de cet espace grâce à sa structure distribuée et à l'aspect multi-agent des SMR.

Le protocole de la stratégie D-ILS exécuté par chaque robot du groupe et illustré dans la figure 2.1 (cf. page 28). En premier lieu, chaque robot initialise la liste \mathbf{Rlist}_i des robots disponibles pour l'allocation des tâches et la liste des tâches \mathbf{T} disponibles pour être allouées aux robots disponibles. Puis chaque robot $r_i \in \mathbf{Rlist}_i$ estime la valeur de son utilité pour chaque tâche $t_i \in \mathbf{T}$, selon l'équation (2.3), et communique les résultats aux autres robots de la liste \mathbf{Rlist}_i . Par la suite, chaque robot $r_i \in \mathbf{Rlist}_i$ construit la matrice d'utilité grâce aux valeurs des utilités des robots de la liste \mathbf{Rlist}_i . Puis, il applique l'algorithme basé sur l'ILS proposé (cf. Chapitre 3) pour trouver la meilleure solution (*i.e.*, allocation des tâches) possible pour chaque robot de la liste \mathbf{Rlist}_i . Dans le cas particulier de la *réallocation* des tâches, défini ci-après, les robots $r_i \in \mathbf{Rlist}_i$ comparent et choisissent la meilleure solution entre le nouveau résultat d'allocation des tâches généré par l'ILS et l'actuelle allocation en vigueur entre les robots. Toutefois, à ce stade du protocole, les résultats des robots impliqués dans ce processus ne sont pas synchronisés, du fait du caractère aléatoire de certains paramètres de l'ILS proposé. Ainsi, pour avoir une cohérence dans les résultats, chaque robot $r_i \in \mathbf{Rlist}_i$ communique ses résultats aux autres robots. Enfin, le robot r_i compare tous les résultats (*i.e.*, les siens et ceux des autres robots) et choisit la meilleure solution d'allocation.

6. Mécanisme de réallocation des tâches de la stratégie D-ILS

Le MRTA est un problème décisionnel dynamique. Par conséquent, afin de garantir la robustesse et la flexibilité sur les décisions du SMR, il convient de traiter l'allocation des tâches de manière itérative dans le temps. A cet effet, la majeure partie de la littérature implémente des mécanismes de réallocation des tâches entre les robots. Cependant, ce type de mécanisme exige que le SMR utilisé ainsi que l'application traitée admettent un échange de tâches entre les robots et dans le cas extrême, ils permettent aux robots d'abandonner leurs tâches actuelles en faveur d'une autre plus adéquate.

Dans l'application de transport considérée dans cette thèse, les robots se déplacent de leurs positions initiales non chargés vers une station de chargement pour prendre un objet, puis ils se dirigent vers les stations de déchargement pour déposer les objets. Ainsi, si on suppose que l'environnement de l'application ne possède pas des stations intermédiaires et n'autorise pas un transfert direct d'objets entre les robots, alors, la *réallocation* des tâches est uniquement possible entre la position initiale du robot et la position de chargement de la tâche à exécuter. Par conséquent, la stratégie D-ILS admet la *réallocation* des tâches si les deux conditions de l'équation (2.4) sont vérifiées.

$$\left\{ \begin{array}{l} nbrT \geq nbrR \\ \text{et} \\ \|r_iCurrPos - r_iInitPos\| \leq \beta \|r_iInitPos - t_jLoadPos\| \end{array} \right. \quad (2.4)$$

Avec, $nbrT$ le nombre de tâches disponibles, $nbrR$ le nombre de robots disponibles, $r_iInitPos$ la position initial du robot r_i , $r_iCurrPos$ la position actuelle du robot r_i , $t_jLoadPos$ la position de la station de chargement de la tâche allouée au robot r_i , et β est un paramètre constant.

La première condition permet aux robot r_i d'uniquement participer à la *réallocation* si le nombre de robots disponibles (incluant le robot r_i) est inférieur ou égal au nombre de tâches disponibles (incluant la tâche actuelle du robot r_i). Ainsi, il est garanti pour les robots impliqués dans la *réallocation* d'être toujours attribués pour d'autres tâches. Par exemple, concéderons le cas inverse ($nbrT < nbrR$), soit l'ensemble de tâches à allouer $\mathbf{T} = \{t_1, t_2\}$ et l'ensemble de robots à attribuer $\mathbf{R} = \{r_1, r_2, r_3\}$. Supposons que les tâches t_1 et t_2 sont allouées aux robots r_1 et r_2 respectivement. Si la tâche t_2 est réallouée au robot r_3 , alors, le robot r_2 a consommé de l'énergie sans produire.

La deuxième condition permet, grâce au paramètre β , de varier le niveau d'engagement des robots à leurs tâches actuelles. En défini l'engagement du robot r_i à sa tâche actuelle t_j par son refus de participer à la *réallocation* des tâches. A titre d'exemple, supposons que $\beta = 1$, ainsi d'après la deuxième condition, le robot r_i est autorisé à participer à la *réallocation* depuis sa position initiale jusqu'à la position de la station de chargement de sa tâche actuelle t_j . Ce qui est supposée la période maximale dans laquelle le robot peut participer à la *réallocation* des tâches. D'autre part, si $\beta = 0.5$, le robot r_i est autorisé à participer à la *réallocation* uniquement durant la moitié du parcours entre sa position initiale et la position de la station de chargement de sa tâche actuelle t_j . Ce qui représente la moitié de la période maximale dans laquelle le robot peut participer à la *réallocation* des tâches. Ainsi, l'engagement du robot r_i est inversement proportionnelle à la valeur de β .

Il est raisonnable d'avancer que la *réallocation* permet un degré de flexibilité de robustesse du SMR. Cependant, une utilisation excessive peut causer un effet négatif sur les performances du système (e.g., consommation d'énergie). Par conséquent, le choix de la valeur du paramètre β est crucial pour le mécanisme de *réallocation* de la stratégie D-ILS. A cet effet, les travaux réalisés dans (Ali Djenadi and Boubekeur Mendil 2019), permettent de conclure que le mécanisme de *réallocation* avec la valeur de β adéquate, produit un effet

positif sur les performances du SMR. En outre, en considération des mêmes travaux, la stratégie D-ILS proposée dans cette thèse utilise la valeur de $\beta = 0.5$.

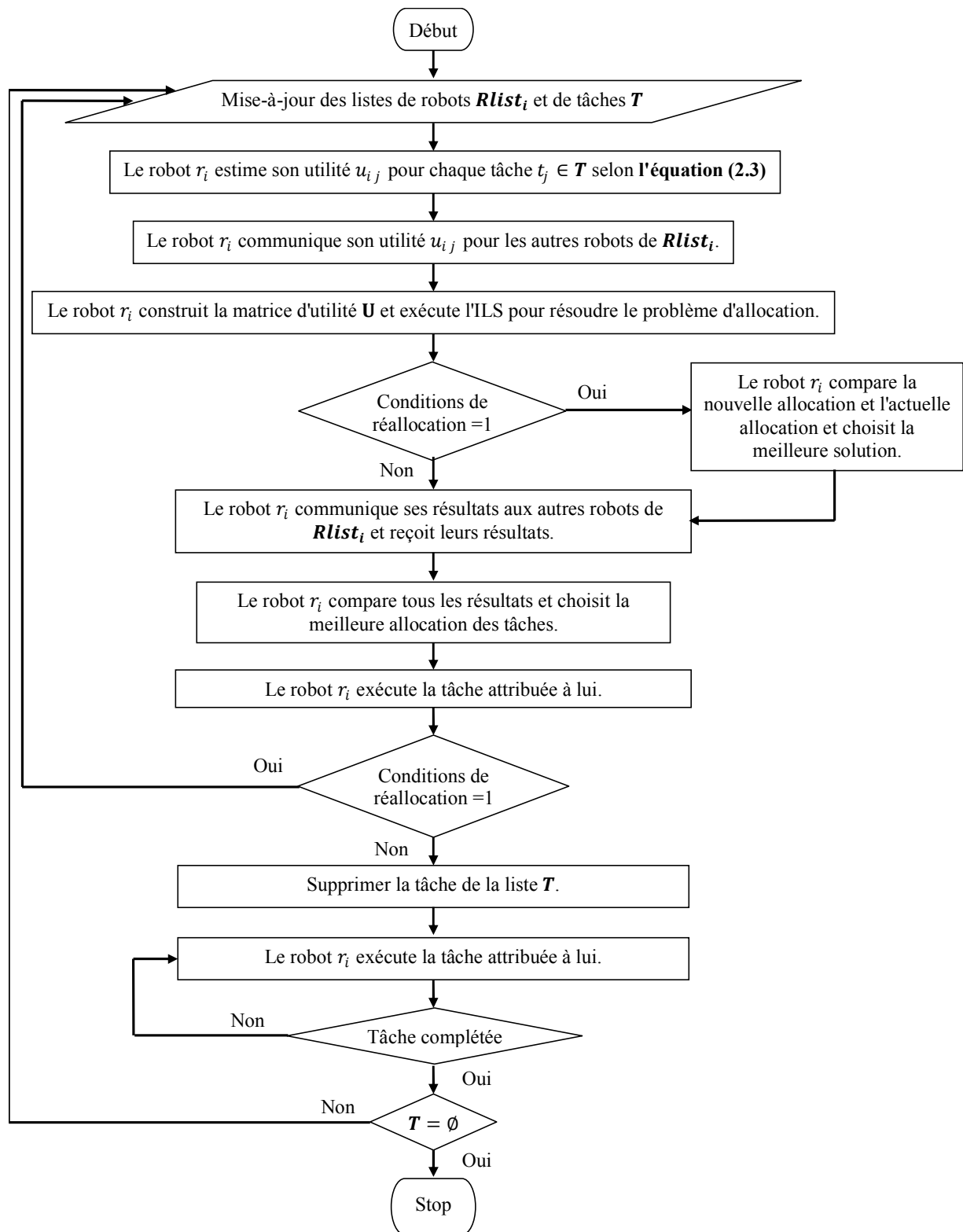


Figure 2.1. Organigramme de la stratégie D-ILS.

7. Conclusion

Ce chapitre s'est focalisé sur l'introduction de la nouvelle stratégie d'allocation des tâches D-ILS proposée dans ce travail de thèse. Cette stratégie a pour objectif de maximiser la productivité et d'optimiser la gestion de la consommation d'énergie d'un système de transport multi-robot. Ce dernier consiste dans un groupe de robots mobiles exécutant des tâches de transport de biens dans un environnement d'intérieur, de stations de chargement vers des stations de déchargement. A cette effet, nous avons proposé une fonction d'utilité basée sur des heuristiques élémentaires. Cette fonction permet aux robots d'estimer leurs utilité à exécuter chaque tâche de transport en se basant sur les informations de leurs états et des tâches disponibles. Enfin dans la dernière partie du chapitre, nous avons présenté le protocole d'allocation des tâches de la stratégie D-ILS; ainsi que son mécanisme de réallocation des tâches.

Le chapitre suivant est consacré l'implémentation algorithmique du MRTA de la stratégie D-ILS qui est basée sur la métaheuristique de la recherche locale itérative (ILS).

CHAPITRE 3. Implémentation algorithmique du MRTA à base de la stratégie D-ILS

CHAPITRE 3. Implémentation algorithmique du MRTA à base de la stratégie D-ILS

1. Introduction

L'objectif du modèle de MRTA utilisé dans ce travail est de trouver les paires robot-tâche qui maximisent l'utilité globale du SMR formulé dans l'équation (2.1). A cet effet, la stratégie D-ILS est basée sur la métaheuristique de la recherche locale itérative (ILS pour Itérative Local Search). Dans cette stratégie, chaque robot du groupe apporte une solution pour l'allocation des tâches du groupe en utilisant un algorithme basé sur l'ILS spécifiquement conçu pour ce but. Cette métaheuristique de recherche à un seul point permet à la stratégie D-ILS une meilleure exploitation des régions locales de l'espace de recherche.

Dans ce chapitre, nous présentons l'algorithme de l'ILS proposé et nous décrivons son principe d'optimisation des solutions. Par ailleurs, nous détaillerons les cinq composants considérés dans le design de cet algorithme.

2. Présentation de l'ILS

La recherche locale itérative (ILS) est une métaheuristique basée sur la recherche à un seul point (méthode basée sur un parcours). L'ILS construit une séquence de points-solutions résultants du guidage itératif d'une heuristique dans l'espace de recherche. Cette séquence de points-solutions représente un parcours menant d'une solution initiale vers une meilleure solution, en comparaison à la solution que peut générer une recherche aléatoire (*i.e.*, random search).

L'ILS est caractérisé par deux aspects, l'unicité du parcours construit par la séquence de points-solutions générées et le caractère réduit des voisinages des solutions améliorées dans l'espace de recherche. Ces solutions sont générées par une heuristique intégrée dans l'ILS. En général, la recherche locale est l'heuristique la plus implémentée dans les algorithmes de l'ILS.

La recherche locale est une fonction heuristique qui génère une amélioration de la solution. Cette amélioration est une optimisation permettant l'exploration de l'espace de recherche \mathcal{S} en se déplaçant d'une solution s vers une meilleure solution s^* (cf. équation (3.1)). A titre d'exemple, cette optimisation peut être définie par la première amélioration (*i.e.*, first improvement), ou encore par la meilleure amélioration (*i.e.*, best improvement).

$$s^* = LocalSearch(s), \quad s, s^* \in \mathcal{S} \quad (3.1)$$

Afin d'éclaircir le fonctionnement de la recherche locale ainsi que son implémentation dans l'ILS, considérons le cas suivant: Soit \mathcal{F} la fonction objectif à minimiser (où maximiser) d'un problème d'optimisation. La recherche locale génère pour une entrée $s \in \mathcal{S}$, une sortie $s^* \in \mathcal{S}^*$ avec $\mathcal{F}(s^*) < F(s)$; ou $\mathcal{F}(s^*) > F(s)$ dans le cas de la maximisation. Ainsi, la recherche locale réalise un déplacement d'un espace \mathcal{S} vers un espace réduit \mathcal{S}^* , où \mathcal{F} est minimisée (maximisée). Par conséquent, la définition de la structure des voisinage \mathcal{S} et \mathcal{S}^* est essentielle à l'exécution de la recherche locale.

Dans le but d'améliorer les résultats de cette recherche locale, il est clair que l'approche la plus directe est d'appliquer une seconde fois la recherche locale à la solution $s^* \in \mathcal{S}^*$, le résultat est la solution $s^{**} \in \mathcal{S}^{**}$, où \mathcal{S}^{**} est plus petite et inclut dans \mathcal{S}^* , avec $\mathcal{F}(s^{**}) < F(s^*)$; ou $(\mathcal{F}(s^{**}) > F(s^*))$ dans le cas de maximisation. Cependant cette approche exige de définir à chaque fois les voisinages de recherche ($\mathcal{S}, \mathcal{S}^*, \mathcal{S}^{**}$, etc...). Cela peut être difficile et couteux en termes de temps de calcul (Lourenço et al. 2003).

Le principe de l'ILS est d'explorer l'espace \mathcal{S}^* , en générant un parcours de points-solutions allant d'un point de départ s^* vers un point mitoyen sans définir explicitement les voisinages de recherche. A cet effet, l'ILS suit la procédure suivante : soit la solution actuelle s^* générée par une recherche locale. En premier lieu, l'ILS applique une perturbation sur s^* qui génère une solution intermédiaire $s' \in \mathcal{S}$. Puis, la recherche locale est appliquée sur la solution s' et donne la solution $s^{*'} \in \mathcal{S}^*$. Enfin, un critère d'acceptation est utilisé pour comparer la nouvelle solution $s^{*'}$ à l'actuelle solution s^* . Si la solution $s^{*'}$ réussit le test, elle devient la nouvelle point-solution dans le parcours réalisé par l'ILS. Dans le cas contraire, le parcours reste sur s^* .

Dans cette thèse, un algorithme basé sur l'ILS est spécifiquement construit pour le problème d'optimisation traité. Cet algorithme est basé sur la formulation de l'ILS proposée par Lourenço *et al.* dans (Lourenço et al. 2003) (cf. **Algorithme 3.1**). Les caractéristiques de l'espace de recherche sont définies par l'instance du problème résolue. En outre l'algorithme utilisé décrit la stratégie d'exploration de cet espace. Ainsi, cinq composants sont considérés dans la conception de l'algorithme incluant : La solution initiale, la recherche locale, la perturbation, le critère d'acceptation de la solution et la condition d'arrêt.

Algorithme 3.1: Recherche locale itérative (ILS)

$s_0 = \text{GenerateInitialSolution}()$

$s^* = \text{LocalSearch}(s_0)$

Repeat

$s' = \text{Perturbation}(s^*)$

$s^{*'} = \text{LocalSearch}(s')$

$s^* = \text{AcceptanceCriterion}(s^*, s^{*'})$

Until (Termination Conditions met)

La formulation d'une solution $s \in \mathbf{S}$ est exprimée par l'équation (3.2). Elle est constituée de trois éléments : la matrice d'utilité $\mathbf{U} = \{u_{ij}\}$, la liste de robots disponibles pour l'allocation des tâches $\mathbf{Rlist} = \{r_i\}, 1 \leq i \leq p \leq n$ et la liste des tâches disponibles $\mathbf{T} = \{t_j\} 1 \leq j \leq m$ (cf. tableau 3.1).

$$s = \{\mathbf{U}, \mathbf{Rlist}, \mathbf{T}\} \quad (3.2)$$

Avec l'allocation des tâches qui correspond aux paires (r_i, t_j) avec $i = j = k, 1 \leq k \leq \min(p, m)$.

Tableau 3.1. Représentation de la solution de l'ILS.

	t_1	\dots	t_m
r_1	u_{11}	\dots	u_{1m}
\vdots	\vdots	\ddots	\vdots
r_p	u_{p1}	\dots	u_{pm}

3. Génération de la solution initiale

Le point de départ s_0^* du parcours de l'ILS dans l'espace de recherche \mathbf{S} est généré en appliquant la fonction de la recherche locale sur la solution initiale s_0 . Par conséquent, une bonne solution initiale permet d'atteindre rapidement un point de départ de qualité. En général, la solution initiale est construite suivant deux méthodes : le choix aléatoire ou un algorithme de type glouton (*i.e.*, greedy algorithm).

L'utilisation d'un algorithme glouton présente l'avantage de la qualité de la solution initiale générée et la réduction du temps nécessaire pour atteindre cette solution. Néanmoins dans ce présent travail, la solution initiale est construite suivant un choix aléatoire, du fait de la simplicité d'implémentation de la méthode. D'autre part, l'influence de la solution n'est pas persistante grâce à l'efficacité d'exploration des autres composants de l'ILS proposé (e.g., efficacité de la perturbation).

4. Recherche locale

Comme mentionné précédemment, la recherche locale est l'heuristique la plus implémentée dans les algorithmes de l'ILS. Toutefois, une simple heuristique d'optimisation, adaptée au problème traité et capable d'explorer les solutions de l'espace de recherche peut être acceptée.

Suivant la formulation de la solution présentée dans **l'équation (3.2)**, l'ILS proposé dans ce travail utilise une recherche locale basée sur une heuristique qui procède à des mouvements de permutation sur les colonnes de la matrice d'utilité et les éléments de la liste de tâches. Pour illustrer le processus de cette heuristique, nous considérons l'exemple suivant: Soit une liste initiale de tâches $\mathbf{T}_0 = \{t_1, t_2, t_3, t_4\}$ pour être allouées aux robots de la liste $\mathbf{Rlist}_0 = \{r_1, r_2, r_3\}$. Suivant la stratégie D-ILS, chaque robot de la liste \mathbf{Rlist}_0 forme une matrice d'utilité \mathbf{U}_0 (cf. **Tableau 3.2.a**). La solution initiale est représentée par

$$s_0 = \{\mathbf{U}_0, \mathbf{Rlist}_0, \mathbf{T}_0\} \quad (3.3)$$

Dans un premier temps, l'heuristique de la recherche locale choisit aléatoirement deux colonnes de la matrice d'utilité \mathbf{U}_0 , la *i*ème et la *j*ème, et deux éléments de la liste des tâches \mathbf{T}_0 de même position *i*ème et *j*ème, avec $(i, j) \leq \min(p, m)$. Par la suite, elle procède à un mouvement de permutation des positions *i*ème et *j*ème des colonnes de \mathbf{U}_0 et des éléments de \mathbf{T}_0 . Supposant que $i = 2$ et $j = 3$, le mouvement de permutation est réalisé entre la 2ème et la 3ème colonne de la matrice d'utilité et le 2ème et 3ème élément de la liste des tâches (**Tableau 3.2.b**). Après l'exécution du mouvement de permutation, une nouvelle solution s_0^* est formée. Cette dernière est composée d'une nouvelle matrice \mathbf{U}_0^* et d'un nouvel arrangement des éléments de la liste des tâches \mathbf{T}_0^* . La nouvelle solution est représentée par

$$s_0^*\{\mathbf{U}_0^*, \mathbf{Rlist}_0^*, \mathbf{T}_0^*\} = LocalSearch(s_0\{\mathbf{U}_0, \mathbf{Rlist}_0, \mathbf{T}_0\}) \quad (3.4)$$

Avec $\mathbf{Rlist}_0^* = \mathbf{Rlist}_0$.

5. Perturbation

Le principal point négatif de la recherche locale est sa tendance à se bloquer dans un minimum local (maximum local). Afin de se soustraire du bassin de ce minimum local, l'ILS applique une perturbation à la solution s_0^* générée par la recherche locale. L'objectif est de trouver un nouveau point de départ s_0' (*i.e.*, solution intermédiaire) pour la recherche locale.

Cependant, le bassin d'attraction de s_0' doit être différent de celui de s_0^* , sans pour autant être éloigné l'un de l'autre dans l'espace de recherche \mathbf{S} , afin d'éviter un comportement aléatoire à chaque nouveau point de départ. Cette dernière caractéristique de la perturbation est exprimée par la force de la perturbation. Cette force est défini par l'ampleur du changement exercé sur l'actuelle solution.

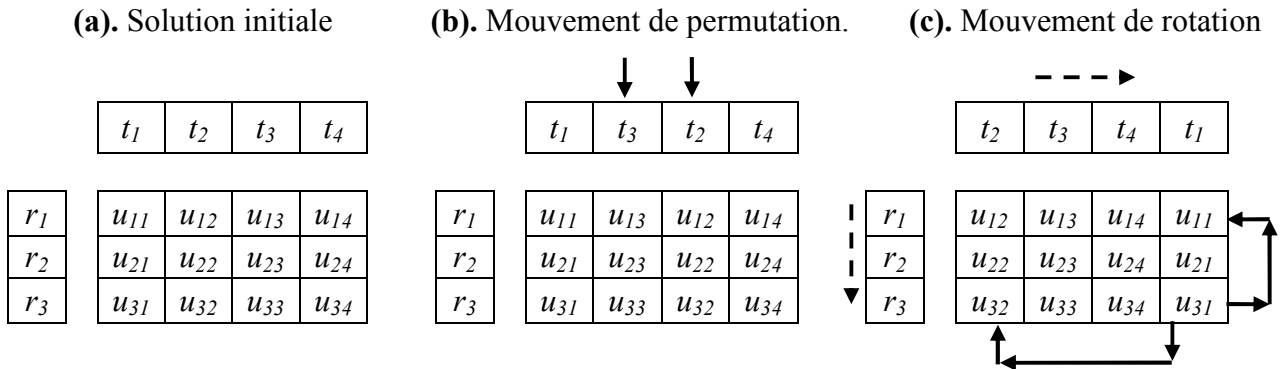
La force de perturbation a un effet important sur le comportement de l'ILS. En effet, si la perturbation est trop faible, l'ILS ne pourra pas s'échapper d'un minimum local. D'un autre côté, si la perturbation est trop forte, l'ILS peut réagir comme un algorithme basé sur un départ aléatoire (Blum and Roli 2003).

La perturbation peut être de valeur fixe, aléatoire, ou s'adapter aux besoins de la dualité Exploration vs. Exploitation de l'ILS. Dans l'ILS proposé dans ce travail, la perturbation est une heuristique exerçant une rotation sur les éléments de la liste des tâche \mathbf{T}_0 et les colonnes de la matrice d'utilité \mathbf{U}_0 , ainsi qu'une rotation sur les éléments de la liste de robots disponibles \mathbf{Rlist}_0 et les lignes de la matrice d'utilité \mathbf{U}_0 . Suivant l'exemple précédent, l'application de la perturbation proposée sur s_0^* produit la solution intermédiaire s_0' (cf. **tableau 3.2.c**). La formulation de la perturbation est donnée par

$$s_0'\{\mathbf{U}_0', \mathbf{Rlist}_0', \mathbf{T}_0'\} = Perturbation(s_0\{\mathbf{U}_0, \mathbf{Rlist}_0, \mathbf{T}_0\}) \quad (3.5)$$

Par ailleurs, la force de la perturbation proposée est définie par le nombre de rotations appliquées à la solution s_0^* . A cet effet, la force de la perturbation est supposée égale au $\min(p, m)$. Ainsi, dans le cas de l'exemple précédant, la force de perturbation est égale à trois rotations (cf. **tableau 3.2.c**).

Tableau 3.2. Exemple de fonctionnement de l'ILS



6. Critères d'acceptation de la solution

Le critère d'acceptation permet à l'ILS de comparer entre la solution actuelle et la solution générée par la recherche locale, après l'application de la perturbation, puis de décider si la nouvelle solution est acceptée en tant que nouveau point-solution du parcours construit par l'ILS.

La littérature propose deux approches pour le critère d'acceptation. La première accepte une nouvelle solution seulement si elle apporte une amélioration sur la solution actuelle. La deuxième approche accepte toujours la nouvelle solution comme nouveau point-solution. Ces deux approches extrêmes, reflètent grossièrement la dualité Exploitation (la première approche) vs. Exploration (la deuxième approche). Ainsi, semblable à la perturbation, le critère d'acceptation permet un certain degré de contrôle sur le comportement de l'ILS. Cependant, il convient de noter qu'une troisième approche intermédiaire est possible, telle que l'utilisation d'une liste de recuit (Glover 1995) où les solutions sont toutes acceptées avec une certaine probabilité. Une autre possibilité est d'utiliser les chaînes de Markov à grand-pas (Martin et al. 1992).

Dans l'ILS proposé dans ce travail, une nouvelle solution est acceptée si et seulement si elle présente une amélioration en comparaison à la solution actuelle. La qualité de la solution est évaluée avec la fonction.

$$G(s) = \sum_{i=j=1}^k u_{ij}, \quad 1 \leq k \leq \min(p, m) \quad (3.6)$$

Le critère d'acceptation de solutions est formulé comme suit

$$Better(s, s') = \begin{cases} s' & \text{if } G(s') > G(s) \\ s & \text{autre} \end{cases} \quad (3.7)$$

Avec s est la solution actuelle et s' est la nouvelle solution.

7. Conditions d'arrêts

Les conditions d'arrêts permettent de stopper l'algorithme. Plusieurs conditions peuvent être utilisées, telles que : le nombre d'itérations, la valeur de la solution générée (seuil d'acceptation), et le nombre d'itération sans amélioration de la solution générée. Dans ce travail, le nombre d'itérations est proposé comme condition d'arrêt de l'algorithme.

8. Conclusion

Le contenu de ce chapitre a exposé le design des différents composants du processus d'allocation des tâches à base de la métaheuristique de l'ILS, tels que : la génération des solutions initiales, la génération des nouvelles solutions, le mécanisme d'évitement du blocage dans les minimums à travers une heuristique de perturbation, le critère d'acceptation de la solution et, enfin, les conditions d'arrêts.

Le chapitre suivant se focalise sur la validation de la stratégie D-ILS et l'évaluation de ses performances dans un environnement de simulation orienté multi-robot.

CHAPITRE 4. Expérimentations

CHAPITRE 4. Expérimentations

1. Introduction

A travers ce chapitre, nous exposons les différentes étapes d'expérimentation et de validation de notre approche. Dans la section 1, nous citons les objectifs des expériences conduites. Dans la seconde section, nous décrivons l'environnement de simulation utilisé et la topologie adoptée pour permettre de valider l'approche proposée. Dans la section 3, nous présentons la méthodologie suivie afin d'analyser les performances de la stratégie D-ILS en termes de productivité et de gestion de la consommation d'énergie. Dans la quatrième et dernière section, nous illustrons les résultats obtenus et leur discussion.

2. Objectifs

Notre étude porte sur le contrôle d'un SMR utilisé dans une application de transport d'objets dans un environnement industriel d'intérieur. Plus précisément, nous traitons le problème d'allocation des tâches des systèmes multi-robot (MRTA). A cet effet, nous avons proposé une nouvelle stratégie d'allocation des tâches qui optimise les performances du système en termes de productivité et de gestion de la consommation d'énergie.

Afin d'évaluer et de valider cette stratégie, des expérimentations ont été menées sur la plateforme de simulation Player/Stage (Jones et al. 2002; Gerkey et al. 2003; Vaughan 2008). La plateforme Player/Stage est orientée SMR. Elle permet de considérer les comportements physiques de bas niveau des robots (*e.g.*, déplacements, trajectoires, évitement d'obstacles). Elle permet non seulement l'évaluation de la MRTA, mais elle offre la possibilité d'observer l'effet de la stratégie D-ILS sur le comportement des robots. En outre, les programmes de contrôle sont implémentés en langage C++. Ce qui permet de simuler le comportement multi-agent et celui distribué de la stratégie D-ILS proposée en utilisant le concept du *multi-threading* de la bibliothèque du C++.

3. Environnement de simulation

L'environnement de simulation est basé sur le simulateur 2D Stage. La topologie adoptée est inspirée d'un entrepôt de stockage de marchandises robotisés, utilisant des SMR pour transporter des biens d'un point à l'autre de l'entrepôt.

Le design est simplifié pour permettre de valider l'approche proposée dans ce travail de thèse. La structure de l'environnement simulé est présentée dans la **figure 4.1**. C'est un espace assimilé à un entrepôt de 35 mètres de longueur et de 27 mètres de largeur. Cette espace contient cinq stations de chargement et cinq stations de déchargement d'objets (e.g., marchandises)

La simulation utilise un modèle de robot de 0.5 mètre de diamètre basé sur la plateforme du robot Pioneer 2DX (Silverman et al. 2002; Zadarnowska and Tchoń 2015). Chaque robot est équipé de huit capteurs de type sonar pour la détection d'obstacles, avec une vitesse maximale de déplacement de 0.2 m/sec. D'autre part, les robots utilisent l'approche basée sur le *wavefront propagation* (Gerkey and Howard 2011) pour la planification de chemins et le *Vector Field Histogram* (VFH) (Ulrich and Borenstein) pour l'évitement d'obstacles.

Par ailleurs, étant donné que l'objectif de cette étude concerne l'allocation des tâches, les obstacles dans l'environnement de simulation ont été limités dans la mesure du possible, afin de minimiser l'impact des comportements des robots dû aux algorithmes d'évitement d'obstacles et de la planification de chemins sur les décisions de l'allocation des tâches. Toutefois, il convient de noter que la considération de ces deux facteurs dans l'approches proposée fera l'objet de futurs travaux de recherche.

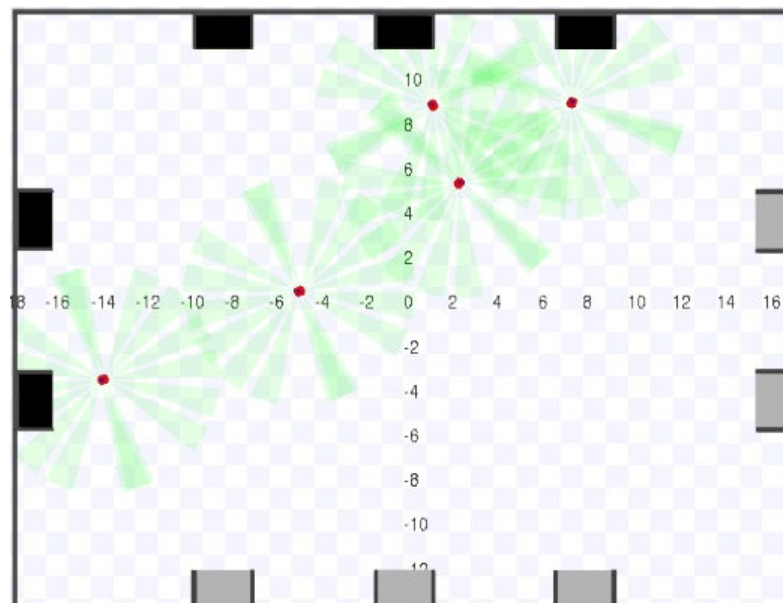


Figure 4.1. Environnement de simulation, avec 5 stations de chargement (noir), 5 stations de déchargement (gris), et 5 robots.

4. Méthodologie d'évaluation

4.1. Critères d'évaluation

Dans le but d'analyser les performances de la stratégie D-ILS, en termes de productivité et de gestion de la consommation d'énergie, elle est évaluée selon trois critères incluant; la moyenne de consommation d'énergie des robots, la productivité et la productivité par unité d'énergie consommée (PPEC). Ces critères sont définis ci-après :

- La moyenne de consommation d'énergie des robots à la fin de la simulation, exprimée par:

$$MoyE = \frac{\sum_{i=1}^n E_i}{n} \quad (4.1)$$

Avec, $MoyE$ la moyenne de consommation d'énergie du groupe de robots, E_i l'énergie consommée par le robot r_i à la fin de la simulation et n le nombre de robots utilisés.

- La productivité, qui est estimée par la moyenne arithmétique pondérée des tâches complétées, avec les priorités comme poids. Afin d'illustrer ce critère, considérons $nbrT_Q$ l'ensemble des nombres de tâches complétées suivant leurs priorités Q (équation (4.2)). La productivité est définie par l'équation (4.3). Ce critère permet d'indiquer la productivité réelle de la stratégie en considérant explicitement le nombre de tâches complétées selon la priorité.

$$nbrT_Q = \{nbrT_{Q=0} \quad nbrT_{Q=1} \quad nbrT_{Q=2} \quad nbrT_{Q=3} \quad nbrT_{Q=4} \quad nbrT_{Q=5}\} \quad (4.2)$$

$$productivité = \frac{\sum_{i=0}^5 i \cdot nbrT_{Q=i}}{\sum_{i=0}^5 i} \quad (4.3)$$

Avec, $nbrT_{Q=k}$ est le nombre complété de tâches avec une priorité k , avec $0 \leq k \leq 5$.

- Le PPEC, ce critère est défini par la productivité par unité d'énergie consommée. Il permet d'estimer la productivité du système en fonction de l'énergie consommée. Ainsi, on peut évaluer la gestion de l'énergie des robots par rapport à leur productivité. Le PPEC est défini par

$$PPEC = \frac{\text{productivité}}{\text{totale d'énergie consommée}} \quad (4.4)$$

4.2. Scénarios de simulation

Afin d'évaluer les performances de la stratégie D-ILS, trois scénarios de simulation sont considérés. Le **tableau 4.1** présente les paramètres de chaque scénario. Ces dernières sont défini par le nombre initial de robots, le nombre initial de tâches, la capacité initiale d'énergie disponible pour chaque robot.

Par ailleurs, afin d'évaluer les performances de la stratégies D-ILS dans le cas d'une application de transport dynamique, deux scénarios sont simulés. Dans ces derniers, les tâches sont ajoutées progressivement dans le temps. Ainsi, la disponibilité des tâches est dynamique. Ces scénarios peuvent être assimilés aux cas d'augmentation brusque des tâches à accomplir. A titre d'exemple, les cas des pics d'afflux de commandes, qui peuvent survenir dans les centres de distribution de marchandises. Le **tableau 4.2** présente les paramètres de chaque scénario dynamique.

Dans chaque scénario, les paramètres de chaque tâche (*i.e.*, priorité, station de chargement et de déchargement) sont générés aléatoirement. Aussi, chaque simulation est répétée pour 20 essais. Les résultats considérés sont la moyenne des 20 essais.

Tableau 4.1. Paramètre des scénarios simulés.

Paramètres	Scénarios		
	Scénario 1	Scénario 2	Scénario 3
Nombre de robot	2, 3, 5, et 7	2, 3, 5, et 7	5
Nombre de tâches	20	80	80
Valeurs des priorités Q_j	0 - 5	0 - 5	0
E_{max_i} (unité)	500	500	500

Tableau 4.2. Paramètre des scénarios dynamiques simulés.

Scénarios / Paramètres	Nombre initial de tâches	Nombre de tâches ajoutées	Période de temps entre chaque ajout	Nombre final de tâches
Scénario 4	5	1	60	20
Scénario 5	10	5	120	80

4.3. Stratégies de comparaison

Afin de valider la stratégie D-ILS et d'analyser ses performances, on propose d'évaluer cette stratégie selon chaque objectif d'optimisation recherché (*i.e.*, productivité, gestion de la consommation de l'énergie). A cet effet, la stratégie D-ILS est comparée à trois heuristiques inspirées des règles de dispatching des AGV. Pour rappeler, les règles de dispatching sont très utilisées dans la littérature (Chawla et al. 2018 ; Zhang and Roy 2019). Elles sont définies comme de simples heuristiques d'attribution de robots à des tâches (Vis 2006b; Đurasević and Jakobović 2018). En général, les règles de dispatching sont conçues pour optimiser un seul objectif de production. Ainsi dans ce travail, chaque heuristique de comparaison est structurée sous forme d'une stratégie d'allocation des tâches, sans réallocation et centrée sur l'optimisation d'un seul objectif. Ces stratégies sont définies ci-après.

High Priority Task-First (HPTF): est une stratégie d'allocation qui alloue aux robots du groupe les tâches selon leurs priorités. Cette heuristique réorganise la liste des tâches disponible selon leurs priorités, de la plus élevée à la plus faible. Puis, les tâches sont distribuées sur les robots disponibles, où chaque robot est affecté à une seule tâche à la fois. Par la suite, chaque robot exécute sa tâche. Le cycle recommence lorsque des robots sont disponibles. La simulation s'arrête si les tâches sont toutes accomplies ou bien, tous les robots sont à court d'énergie. L'objectif de cette stratégie HPTF est d'optimiser la productivité du SMR en exécutant les tâches selon leurs priorités, sans considérer l'énergie consommée par les robots. Les performances de cette stratégie seront le point de repère pour évaluer la stratégie D-ILS en termes de productivité. Cette stratégie ne permet pas la réallocation des tâches.

Nearest Task-First (NTF): est une stratégie d'allocation qui alloue aux robots disponibles les tâches proches de leurs positions. Dans cette stratégie chaque robot estime la distance du parcours entre sa position et la position de la station de chargement de toutes les tâches disponibles. Cela permet de trouver la tâche la plus proche de sa position. Les tâches sont

distribuées aux robots disponibles en ordre, du premier de la liste au dernier. L'objectif de cette stratégie est de minimiser la consommation d'énergie en minimisant la distance parcourue, avec l'allocation de tâches proches des robots, sans considérer la priorité. Cette stratégie ne permet pas la réallocation des tâches.

Shortest Travel Distance-First (STDF): est une stratégie qui minimise la consommation d'énergie en minimisant la distance parcourue par les robots. Cependant, contrairement à la stratégie NTF, la STDF considère la totalité de la distance du parcours nécessaire pour accomplir une tâche par un robot, de sa position vers la station de chargement de la tâche, puis vers la station de déchargement de cette tâche. Cette stratégie procède à l'allocation des tâches sans réallocation et sans considérer les priorités de ces tâches.

5. Résultats et discussions

5.1. Scénario 1 : évaluation de la consommation d'énergie

L'objectif de ce scénario est d'évaluer les performances de la stratégie D-ILS en termes de consommation d'énergie. Dans ce scénario, les simulations sont effectuées pour des SMR de tailles différentes en termes de nombre de robots (*i.e.*, 2, 3, 5, 7 robots). Ces robots disposent initialement de 500 unités d'énergie et doivent accomplir 20 tâches générées au début de la simulation. Dans ce scénario, les robots possèdent suffisamment d'énergie initiale (*i.e.*, 500 unités d'énergie) pour accomplir toutes les 20 tâches disponibles.

La **figure 4.2** illustre la consommation moyenne d'énergie du groupe. Les résultats montrent que la stratégie D-ILS consomme moins d'énergie comparée à l'heuristique HPTF et consomme une quantité d'énergie proche, néanmoins légèrement inférieure à la consommation des heuristique NTF et STDF. Ces résultats sont une conséquence implicite du design de la fonction d'utilité proposée dans **chapitre 2**. En effet, cette fonction considère la distance du parcours des tâches et le rapport entre le niveau d'énergie consommée par les robots et l'énergie initiale disponible. Par conséquent, les robots ont tendance à minimiser la consommation d'énergie en choisissant des tâches avec une distance de parcours réduite.

Toutefois, la stratégie D-ILS n'ignore pas la priorité des tâches. En effet, la D-ILS considère la priorité des tâches dans l'estimation de l'utilité des robots. En outre, étant donné que le rapport entre le nombre de robots et le nombre de tâches disponibles est proche de sa valeur maximale qui égale à $\frac{nbrR}{nbrT} \cong 1$ et avec de faibles niveaux de consommation d'énergie

des robots, la stratégie D-ILS privilégie les tâches ayant des priorités élevées. Cela est illustré dans **la figure 4.3** par la distribution dans le temps de l'exécution des tâches pour un SMR de 2 robots, qui montre que la stratégie D-ILS privilégie les tâches avec les priorités les plus élevées en comparaison aux stratégies NTF et STDF. Ces dernières présentent une distribution, pour un SMR de 2 robots, qui paraît pseudo-aléatoire, comme montré dans **la figure 4.4**. Ces résultats sont prévisibles, étant donné que NTF et STDF ne considèrent pas la priorité des tâches. Ainsi, dans un scénario avec des robots disposant de suffisamment d'énergie pour accomplir les tâches disponibles, la D-ILS prend les bonnes décisions afin de minimiser la consommation d'énergie tout en considérant la priorité des tâches.

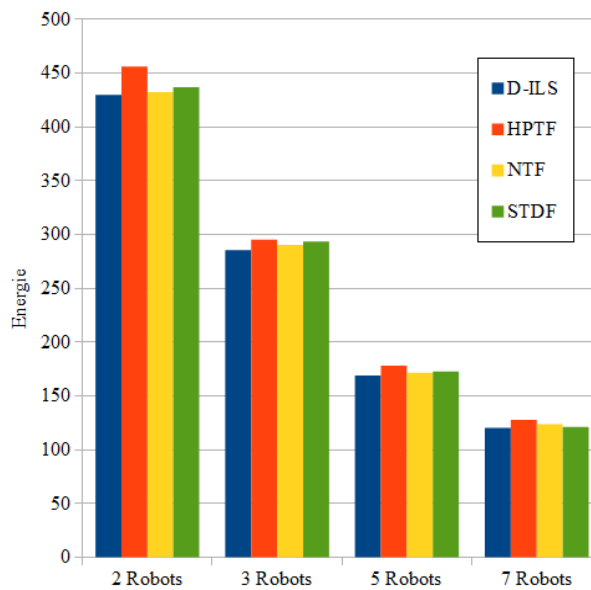


Figure 4.2. Moyenne d'énergie consommée par les robots dans le scénario 1.

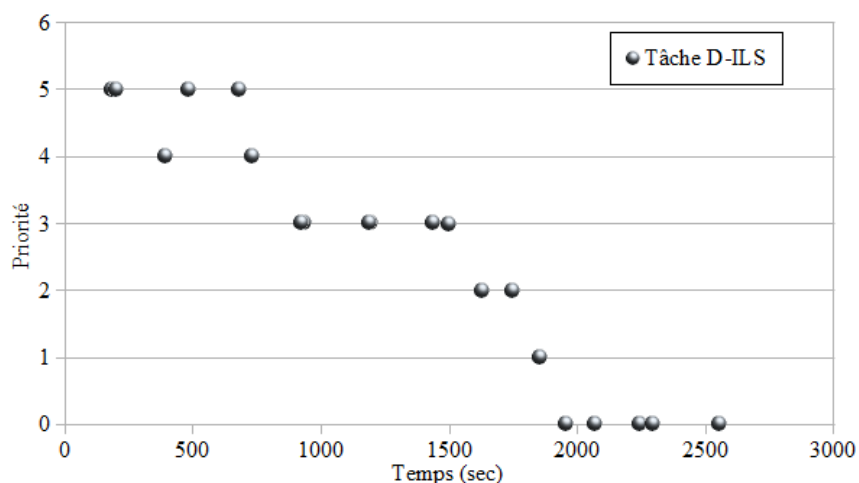


Figure 4.3. Distribution dans le temps de l'exécution des tâches de la D-ILS, pour un SMR de 2 robots, dans le scénario 2.

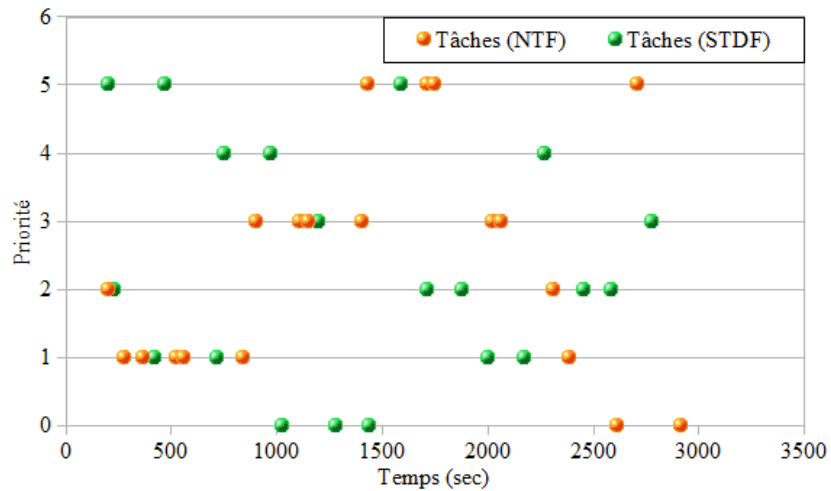


Figure 4.4. Distribution dans le temps de l'exécution des tâches de NTF et STDF, pour un SMR de 2 robots, dans le scénario 2.

Par ailleurs, la réallocation des tâches est un autre facteur qui agit explicitement sur l'allocation des tâches de la stratégie D-ILS et permet une redistribution active des tâches. Ainsi, l'optimisation des décisions du SMR dans chaque situation s'effectue en temps-réel. Ce qui permet de s'adapter à la disponibilité des robots ou à la découverte d'une meilleure solution grâce à la stratégie D-ILS et au travail de résolution collective du problème du MRTA par les robots.

5.2. Scénario 2: Productivité et PPEC

Dans ce scénario, différentes tailles de SMR sont simulées (i.e., 2, 3, 5, et 7)). Ces robots disposent initialement de 500 unités d'énergie et doivent accomplir 80 tâches. Dans ce scénario, les robots ne disposent pas suffisamment d'énergie pour accomplir les 80 tâches disponibles. Par conséquent, le SMR est mis dans une situation de pénurie d'énergie. Ainsi, le but du scénario est de déterminer la capacité de la stratégie D-ILS à gérer l'énergie disponible afin d'être le plus productive possible. La productivité du SMR est estimée suivant **l'équation (4.3)**.

Les figures 4.5, 4.6, et 4.7 présentent respectivement la moyenne d'énergie consommée par les robots, la productivité et la productivité par unité d'énergie consommée du SMR. Les résultats de la figure 4.5 indiquent que la stratégie D-ILS consomme moins d'énergie que la stratégie HPTF. Cependant, à l'inverse des résultats du scénario 1, la stratégie D-ILS consomme plus d'énergie que les stratégies NTF et STDF. Ces résultats peuvent être expliqués par le fait que la stratégie D-ILS considère la priorité des tâches, ce qui peut contraindre les robots à consommer de grandes quantités d'énergie afin de compléter des

tâches à priorités élevées. En revanche, les stratégies NTF et STDF, basées sur une approche qui minimise les distances parcourues, sans considérer la priorité des tâches, permet de minimiser la consommation d'énergie au fil du temps. Cependant, il faut rappeler que dans le **scénario 2**, les robots sont confrontés à un déficit d'énergie. Par conséquent, ils ne peuvent pas compléter la totalité des tâches disponibles. Ainsi, la stratégie D-ILS proposée tend à maximiser la productivité du système, en accomplissant plus de tâches à priorité élevée avec l'énergie disponible.

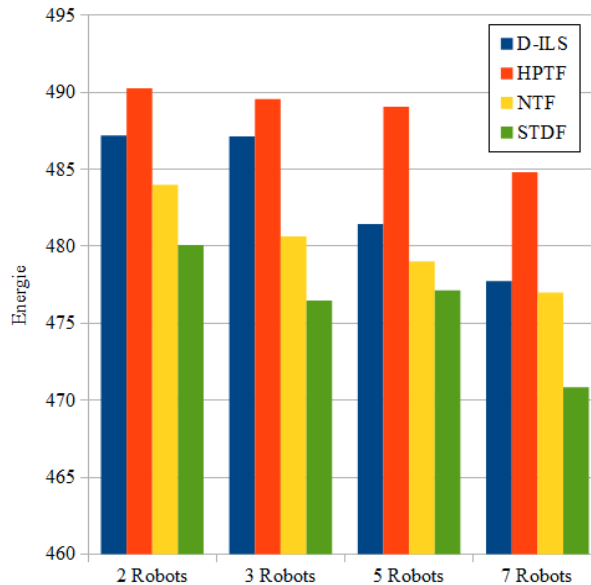


Figure 4.5. Moyenne d'énergie consommée par les robots dans le scénario 2.

Les résultats de la productivité de **la figure 4.6** accèdent la dernière conclusion. En effet, ces résultats indiquent que les performances de la stratégie D-ILS, en termes de productivité, surpasse les autres stratégies. Plus encore, les résultats de la productivité par unité d'énergie consommée, illustrés dans **la figure 4.7**, indiquent que la stratégie D-ILS possède une production plus élevée par unité d'énergie consommée. Cela souligne la capacité de la stratégie D-ILS à gérer la consommation de l'énergie disponible afin de maximiser la productivité du système.

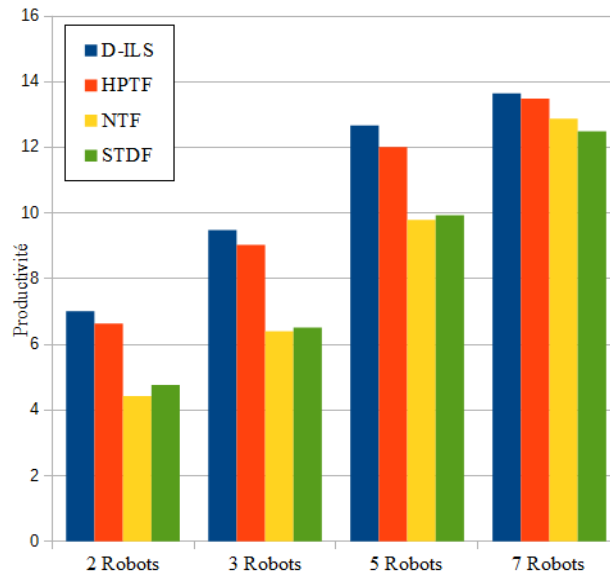


Figure 4.6. Productivité des robots dans le scénario 2.

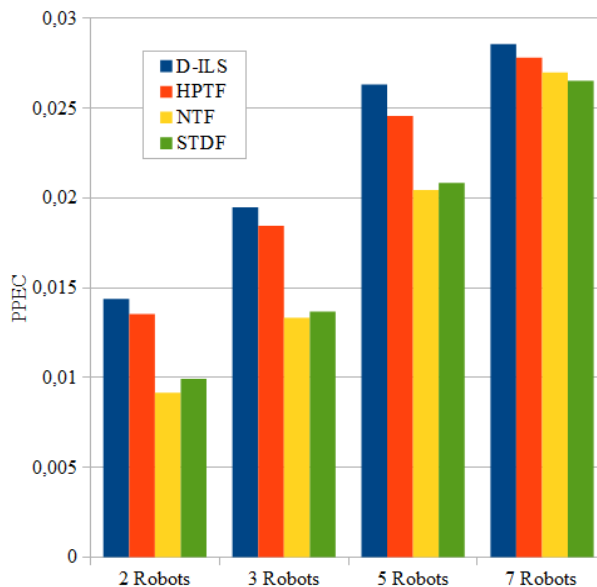


Figure 4.7. PPEC des robots dans le scénario 2.

5.3. Scénario 3 : D-ILS et les tâches à priorité équivalentes

Dans ce scénario, un SMR de cinq robots est simulé. Ces robots disposent initialement de 500 unités d'énergie et doivent accomplir 80 tâches générées au début de la simulation. Toutefois, ces 80 tâches sont associées à des priorités équivalentes, égales à $Q = 0$. L'objectif de ce scénario est d'évaluer les performances de la stratégie D-ILS, sans l'impact des priorités des tâches sur la stratégie d'allocation.

Les **figure 4.8** et **4.9** illustrent le pourcentage de tâches complétées et la moyenne de consommation d'énergie des robots dans le **scénario 3**, comparés aux résultats obtenus pour

un groupe de cinq robots dans le **scénario 2** avec des tâches associées à des priorités de valeurs $0 \leq Q \leq 5$. Les résultats de la **figure 4.8** indiquent que la stratégie D-ILS dans le cas de tâches avec priorités $Q = 0$, complète moins de tâches que la stratégie STDF. Cependant, elle complète plus de tâches que les stratégies NTF et HPTF, ainsi que la stratégie D-ILS dans le cas de tâches avec priorités $0 \leq Q \leq 5$. Par ailleurs, les résultats de la **figure 4.9** montrent que la stratégie D-ILS dans le cas de tâches avec priorités $Q = 0$, a une moyenne de consommation d'énergie réduite comparée aux stratégies STDF, NTF, HPTF, ainsi que la stratégie D-ILS dans le cas de tâches avec priorités $0 \leq Q \leq 5$.

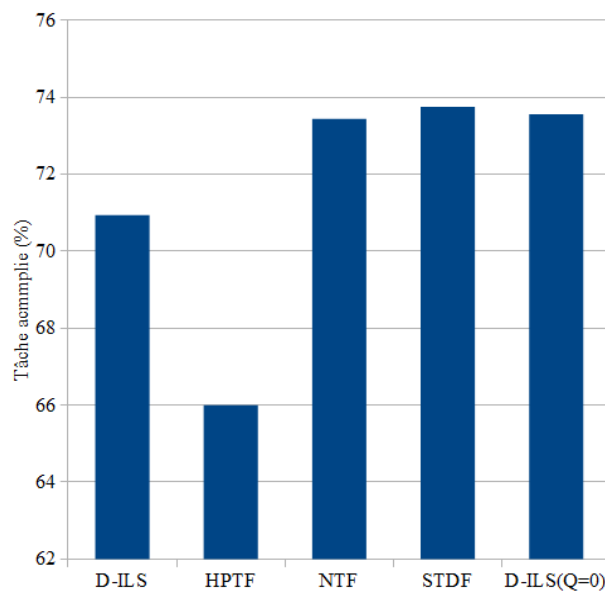


Figure 4.8. Pourcentage de tâches complétées par les robots dans le scénario 3 comparé aux résultats obtenus pour un groupe de cinq robots dans le scénario 2 avec des tâches associées à des priorités de valeurs $0 \leq Q \leq 5$

Les bonnes performances affichées par la stratégie D-ILS dans le cas de tâches avec priorité $Q = 0$ sont une conséquence directe de l'effet implicite de la fonction d'utilité proposée dans **chapitre 2**. Effectivement, en considérant dans le processus de l'allocation des tâches les variables incluant la distance nécessaire pour compléter une tâche, le niveau d'énergie des robots et l'épuisement de cette énergie, permet une bonne gestion de la consommation d'énergie, mais aussi d'avoir une meilleure distribution des tâches sur les robots. On note aussi que le facteur de *réallocation* des tâches de la stratégie D-ILS permet une allocation dynamique des tâches, contrairement, à l'allocation statique des stratégies HPTF, NTF, et STDF. Par ailleurs, les résultats du **scénario 3** soulignent que les stratégies basées sur une approche qui minimise la consommation d'énergie, en minimisant la distance

parcourue, telles que les stratégies NTF et STDF, n'est pas suffisante pour produire de bonnes solutions.

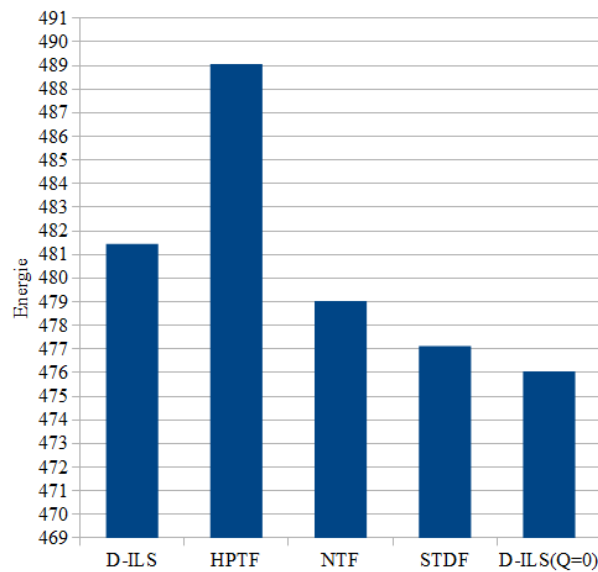


Figure 4.9. Moyenne de consommation d'énergie des robots dans le scénario 3 comparé aux résultats obtenus pour un groupe de cinq robots dans le scénario 2 avec des tâches associées à des priorités de valeurs $0 \leq Q \leq 5$

5.4. Scénario 4 et 5 : Scénarios dynamiques

Afin d'évaluer les performances de la stratégie D-ILS face à une application dynamique en termes de nombre de tâches disponibles, **les scénarios 4 et 5** sont envisagés (cf. **tableau 4.2**). Dans chaque scénario, la simulation débute par un nombre de tâches fixe, puis à chaque période de temps des tâches sont ajoutées jusqu'à atteindre un nombre final de tâches générées. La simulation s'arrête si toutes les tâches sont achevées, ou bien si les robots ne disposent pas de suffisamment d'énergie pour continuer l'exécution des tâches restantes.

Dans **le scénario 4**, un SMR de cinq robots est simulé. Ces robots disposent initialement de 500 unités d'énergie. La simulation débute avec un nombre initial de 5 tâches, puis une tâche est générée toutes les 60 seconds pour atteindre un nombre de final de 20 tâches. D'autre part, dans **le scénario 5**, un SMR de cinq robots est simulé. Ces robots disposent initialement de 500 unités d'énergie. La simulation débute avec un nombre initial de 10 tâches, puis 5 tâches sont ajoutées toutes les 120 seconds pour atteindre un nombre final de 80 tâches.

La figure 4.10 illustre une comparaison des moyennes de consommation d'énergie entre les résultats du **scénario 4** (cas dynamique) et les résultats obtenus dans le **scénario 1** (sans dynamique). Ces résultats indiquent que les performances des quatre stratégies incluant,

D-ILS, NTF, STDF, et HPTF, ont diminué sous l'effet de la dynamique de la disponibilité des tâches. Ce qui pouvait être prévisible. Par ailleurs, la moyenne de consommation d'énergie de la stratégie D-ILS dans le **scénario 4** (cas dynamique) reste inférieure à la moyenne de consommation d'énergie des stratégies STDF et HPTF et elle est proche, néanmoins très légèrement inférieure, à la moyenne de consommation d'énergie de la stratégie NTF.

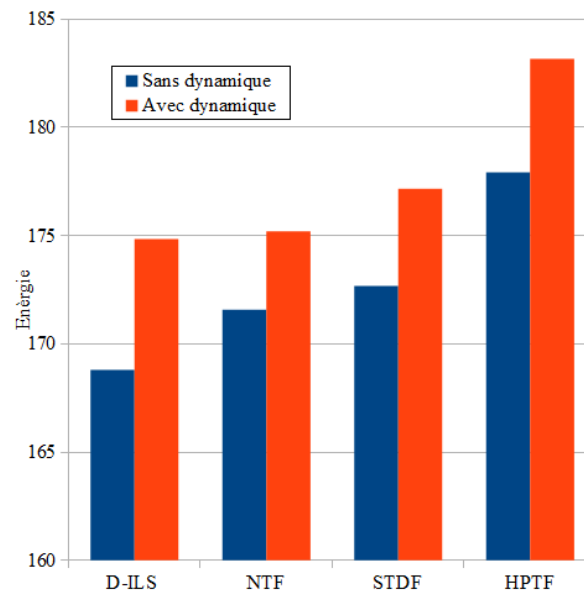


Figure 4.10. Comparaison des moyennes de consommation entre les résultats du scénario 4 (cas dynamique) et les résultats du scénario 1 (sans dynamique), pour un SMR de 5 robots.

La **figure 4.11** illustre une comparaison des productivités entre les résultats du scénario 5 (cas dynamique) et les résultats obtenus dans le **scénario 2** (sans dynamique). Ces résultats indiquent une réduction de la productivité dans le cas dynamique pour les stratégies incluant, D-ILS, NTF, STDF et HPTF, par rapport aux résultats du **scénario 2** (sans dynamique). Comme mentionné précédemment, cette diminution des performances est prévisible, à cause de l'effet de la dynamique des tâches. Toutefois, la réduction des performances de la stratégie HPTF sont très légères comparé aux autres stratégies. Cela indique que HPTF n'est pas très influencée par la dynamique du scénario. Malgré cela, la stratégie D-ILS est toujours plus productive que les autres stratégies. Ces résultats indiquent les bonnes performances de la stratégie D-ILS en termes de productivité ainsi que sa capacité à gérer les augmentations dynamiques du nombre de tâches disponibles.

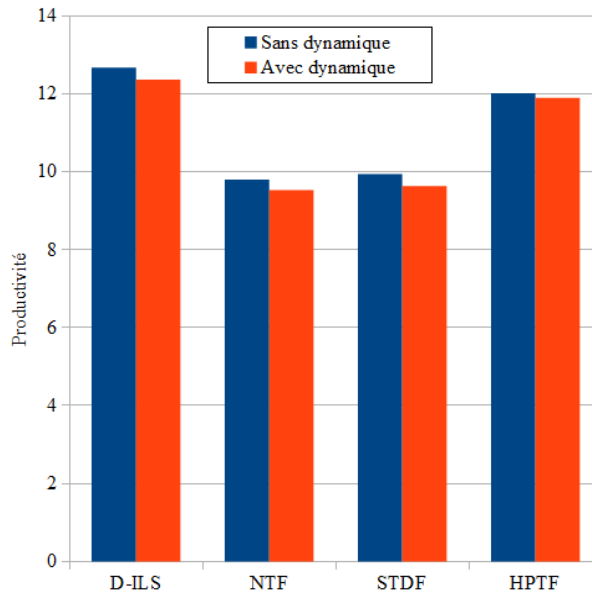


Figure 4.11. Comparaison des productivités entre les résultats du scénario 5 (cas dynamique) et les résultats du scénario 2 (sans dynamique), pour un SMR de 5 robots.

6. Conclusion

Ce chapitre s'est focalisé sur la validation et l'évaluation de la stratégie D-ILS. Plusieurs expérimentations ont été menées sous forme de scénarions de simulation en utilisant une plateforme de simulation orientée multi-robot.

La stratégie D-ILS a été évaluée selon différents critères incluant la moyenne d'énergie consommée, la productivité et de la productivité par unité d'énergie consommée. Par ailleurs, les performances de la stratégie D-ILS ont été étudiées dans un environnement dynamique. Afin de valider les performances de la stratégie D-ILS, elle a été comparée à trois heuristiques inspirées des règles de dispatching des AGV. Chaque heuristique de comparaison est structurée sous forme d'une stratégie d'allocation de tâches, sans réallocation et centrée sur l'optimisation d'un seul objectif.

Dans la dernière partie de ce présent chapitre, les résultats obtenus des différentes simulations ont été présentés. Ces derniers indiquent que la stratégie donne de bonnes performances en termes de productivité et de gestion de la consommation de l'énergie du SMR.

CONCLUSION GENERALE

CONCLUSION GENERALE

Les travaux de recherche menés dans le cadre de cette thèse ont exploré deux éléments fondamentaux du design d'un système de transport multi-robot mobile dans un environnement d'intérieur à savoir : le problème d'allocation des tâches (MRTA) et la gestion de l'énergie d'un SMR.

A cet effet, une nouvelle stratégie d'allocation des tâches, D-ILS, pour un système de transport multi-robot mobile, a été proposée. Elle a pour objectif de maximiser la productivité et d'optimiser la gestion de la consommation d'énergie du SMR. Cette stratégie apporte comme originalité, la considération de la consommation d'énergie explicitement dans la phase de décision des allocations des tâches. D'autre part, elle inclue explicitement le niveau d'énergie des robots ainsi que son épuisement dans l'estimation de l'utilité du robot et implicitement lors de l'allocation des tâches.

La stratégie D-ILS proposée est basée sur la métaheuristique de l'ILS et sur une nouvelle fonction d'utilité. Le protocole de résolution du MRTA de la D-ILS imite les métaheuristicues à population de points en se basant sur l'aspect multi-agent des SMR. Chaque robot du groupe apporte sa solution du problème, générée par l'algorithme ILS à travers un processus d'amélioration d'un seul point de recherche. Par conséquent, le SMR produit une population de solutions à défaut d'une seule solution que peut produire un ILS standard. Ainsi, ce protocole permet à la stratégie D-ILS d'associer les avantages d'exploration des métaheuristicues à population de points ainsi que les avantages en exploitation des métaheuristicues de recherche à un seul point.

Sur la base des travaux réalisés, nous avons implémenté la stratégie D-ILS sur la plateforme de simulation Player/Stage. Les programmes ont été réalisés en langage C++. Ce qui a permis de simuler le comportement multi-agent et distribué de la stratégie D-ILS en utilisant le concept du *multi-threading* de la bibliothèque du C++.

Les expérimentations menées dans ce travail de thèse ont permis d'évaluer les performances de la stratégie D-ILS proposée en termes d'énergie consommée, de la productivité effective du système et de la productivité par unité d'énergie consommée. Les performances de la stratégie D-ILS ont été étudiées dans des scénarios dynamiques avec le nombre de tâches disponibles qui augmente en fonction du temps. La D-ILS a été comparée à trois heuristiques inspirées des règles de dispatching des AGV.

Les résultats obtenus ont montré l'intérêt de nos contributions. Effectivement, La stratégie D-ILS permet une gestion efficace de la consommation d'énergie d'un SMR afin de

maximiser la productivité du système. Par ailleurs, ces résultats indiquent les bonnes performances de la stratégie D-ILS en termes de productivité ainsi que sa capacité à gérer les augmentations dynamiques dans le nombre de tâches disponibles

Sur la base du travail réalisé, nous pouvons dresser plusieurs perspectives de recherche à court et à long terme. À court terme, nous travaillons sur une stratégie d'allocation des tâches basée sur l'optimisation lexicographique ; qui est une méthode d'optimisation multi-objectifs. Cette dernière résout une série de problèmes mono-objectifs de manière séquentielle suivant un ordre de priorité. Par ailleurs, l'optimisation lexicographique est considérée dans les cas de problèmes multi-objectifs contradictoires de nature, ce qui est le cas pour les deux objectifs visés. Dans notre cas, les objectifs sont de maximiser la productivité et de minimiser la consommation d'énergie.

Les perspectives à long terme visent à reconsidérer le profil énergétique non-linéaire des batteries et d'envisager l'implémentation dans un environnement industriel réel avec la considération des contraintes de la planification de chemins et l'évitement d'obstacles. Enfin, nous envisageons l'implémenter la stratégie D-ILS proposée sur d'autre plateformes robotiques notamment les systèmes multi-drone autonomes, en effet, la gestion de l'énergie dans ces systèmes est très importante pour l'optimisation de leurs temps de vol.

REFERENCES

REFERENCES

- Alotaibi ET, Alqefari SS, Koubaa A (2019) LSAR: Multi-UAV Collaboration for Search and Rescue Missions. *IEEE Access* 7:55817–55832.
<https://doi.org/10.1109/ACCESS.2019.2912306>
- Alsamhi SH, Lee B (2021) Blockchain-Empowered Multi-Robot Collaboration to Fight COVID-19 and Future Pandemics. *IEEE Access* 9:44173–44197.
<https://doi.org/10.1109/ACCESS.2020.3032450>
- Arif MU (2017) An Evolutionary Algorithm Based Framework for Task Allocation in Multi-Robot Teams
- Arrichiello F, Marino A, Pierri F (2014) A decentralized fault tolerant control strategy for multi-robot systems. *IFAC Proceedings Volumes* 47:6642–6647.
<https://doi.org/10.3182/20140824-6-ZA-1003.01875>
- Asfora BA, Banfi J, Campbell M (2020) Mixed-Integer Linear Programming Models for Multi-Robot Non-Adversarial Search. *IEEE Robotics and Automation Letters* 5:6805–6812. <https://doi.org/10.1109/LRA.2020.3017473>
- Atay N, Bayazit B (2006) Mixed-Integer Linear Programming Solution to Multi-Robot Task Allocation Problem
- Bačik J, Tkáč P, Hric L, et al (2020) Phollower—The Universal Autonomous Mobile Robot for Industry and Civil Environments with COVID-19 Germicide Addon Meeting Safety Requirements. *Applied Sciences* 10:7682. <https://doi.org/10.3390/app10217682>
- Batinović A, Oršulić J, Petrović T, Bogdan S (2020) Decentralized Strategy for Cooperative Multi-Robot Exploration and Mapping. *IFAC-PapersOnLine* 53:9682–9687.
<https://doi.org/10.1016/j.ifacol.2020.12.2618>
- Beaver LE, Malikopoulos AA (2021) An overview on optimal flocking. *Annual Reviews in Control*. <https://doi.org/10.1016/j.arcontrol.2021.03.004>
- Berenz V, Tanaka F, Suzuki K (2012) Autonomous battery management for mobile robots based on risk and gain assessment. *Artificial Intelligence Review* 37:217–237.
<https://doi.org/10.1007/s10462-011-9227-9>
- Blum C, Roli A (2003) Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Comput Surv* 35:268–308.
<https://doi.org/10.1145/937503.937505>
- Botelho SC, Alami R (1999) M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. pp 1234–1239 vol.2
- Boysen N, de Koster R, Weidinger F (2019) Warehousing in the e-commerce era: A survey. *European Journal of Operational Research* 277:396–411.
<https://doi.org/10.1016/j.ejor.2018.08.023>

- Brito RC, Saque N, Gonçalves DA, et al (2019) Optimized Division of Exploration Areas in Multi-robot Systems Considering Static and Dynamic Charging Stations. In: 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). pp 888–891
- Cao YU, Fukunaga AS, Kahng A (1997a) Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots* 4:7–27. <https://doi.org/10.1023/A:1008855018923>
- Cao YU, Fukunaga AS, Kahng A (1997b) Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots* 4:7–27. <https://doi.org/10.1023/A:1008855018923>
- Carlo HJ, Vis IFA, Roodbergen KJ (2014) Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research* 236:1–13. <https://doi.org/10.1016/j.ejor.2013.11.023>
- Cattrysse DG, Van Wassenhove LN (1992) A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research* 60:260–272. [https://doi.org/10.1016/0377-2217\(92\)90077-M](https://doi.org/10.1016/0377-2217(92)90077-M)
- Chawla VK, Chanda AK, Angra S, Rani S (2018) Simultaneous Dispatching and Scheduling of Multi-Load AGVs in FMS-A Simulation Study. *Materials Today: Proceedings* 5:25358–25367. <https://doi.org/10.1016/j.matpr.2018.10.339>
- Chen J, Wang J, Xiao Q, Chen C (2018) A Multi-Robot Task Allocation Method Based on Multi-Objective Optimization. In: 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). Singapore, pp 1868–1873
- Chen X, He S, Zhang Y, et al (2020) Yard crane and AGV scheduling in automated container terminal: A multi-robot task allocation framework. *Transportation Research Part C: Emerging Technologies* 114:241–271. <https://doi.org/10.1016/j.trc.2020.02.012>
- Cordeau J-F, Laporte G, Savelsbergh MWP, Vigo D (2007) Chapter 6 Vehicle Routing. In: Barnhart C, Laporte G (eds) *Handbooks in Operations Research and Management Science*. Elsevier, pp 367–428
- Cortés P, Gómez-Montoya RA, Muñuzuri J, Correa-Espinal A (2017) A tabu search approach to solving the picking routing problem for large- and medium-size distribution centres considering the availability of inventory and K heterogeneous material handling equipment. *Applied Soft Computing* 53:61–73. <https://doi.org/10.1016/j.asoc.2016.12.026>
- Das PK, Jena PK (2020) Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Applied Soft Computing* 92:106312. <https://doi.org/10.1016/j.asoc.2020.106312>
- Davis NJ, Carpenter BA, Glover CW, Culioli J-C (1992) Parallel approaches to the solution of the assignment problem
- Davis R, Smith RG (1983) Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20:63–109. [https://doi.org/10.1016/0004-3702\(83\)90015-2](https://doi.org/10.1016/0004-3702(83)90015-2)

- De Ryck M, Versteyhe M, Debrouwere F (2020a) Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems* 54:152–173. <https://doi.org/10.1016/j.jmsy.2019.12.002>
- De Ryck M, Versteyhe M, Debrouwere F (2020b) Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems* 54:152–173. <https://doi.org/10.1016/j.jmsy.2019.12.002>
- Dias MB (2004) *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*. <https://doi.org/10.1184/R1/6561401.v1>
- Dias MB, Stentz A (2003) A comparative study between centralized, market-based, and behavioral multirobot coordination approaches. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (Cat. No.03CH37453). pp 2279–2284 vol.3
- Dias MB, Zlot R, Kalra N, Stentz A (2006) Market-Based Multirobot Coordination: A Survey and Analysis. *Proceedings of the IEEE* 94:1257–1270. <https://doi.org/10.1109/JPROC.2006.876939>
- Djenadi A, Mendil B (2021) Energy-aware task allocation strategy for multi robot system. *International Journal of Modelling and Simulation* 0:1–15. <https://doi.org/10.1080/02286203.2020.1855405>
- Djenadi A, Mendil B (2019) Effect of tasks reallocation on the performance of a multi robot system in goods transportation. In: *The Electrical Engineering International Conference EEIC'19*. Bejaia, Algeria
- Draganjac I, Miklić D, Kovačić Z, et al (2016) Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications. *IEEE Transactions on Automation Science and Engineering* 13:1433–1447. <https://doi.org/10.1109/TASE.2016.2603781>
- Đurasević M, Jakobović D (2018) A survey of dispatching rules for the dynamic unrelated machines environment. *Expert Systems with Applications* 113:555–569. <https://doi.org/10.1016/j.eswa.2018.06.053>
- Ebben M (2001) *Logistic control in automated transportation networks*. Twente Univ. Press, Enschede
- Fan Liu, Narayanan A (2011) Real time replanning based on A* for collision avoidance in multi-robot systems. In: *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. pp 473–479
- Fang B, Mei G, Yuan X, et al (2021) Visual SLAM for robot navigation in healthcare facility. *Pattern Recognition* 113:107822. <https://doi.org/10.1016/j.patcog.2021.107822>
- Fanti MP, Mangini AM, Pedroncelli G, Ukovich W (2018) A decentralized control strategy for the coordination of AGV systems. *Control Engineering Practice* 70:86–97. <https://doi.org/10.1016/j.conengprac.2017.10.001>

- Farinelli A, Boscolo N, Zanotto E, Pagello E (2017) Advanced Approaches for Multi-robot Coordination in Logistic Scenarios. *Robot Auton Syst* 90:34–44. <https://doi.org/10.1016/j.robot.2016.08.010>
- Fethi A, Mehdi S (2019) The effect of AGVs number on a flexible manufacturing system. In: 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID). pp 1–5
- Flood MM, Koopmans TC, Dantzig [b B, et al (1953) On The Hitchcock Distribution Problem. *Pacific Journal of Mathematics* 369–386
- Fransen KJC, van Eekelen JAWM, Pogromsky A, et al (2020) A dynamic path planning approach for dense, large, grid-based automated guided vehicle systems. *Computers & Operations Research* 123:105046. <https://doi.org/10.1016/j.cor.2020.105046>
- Fu B, Chen L, Zhou Y, et al (2018) An improved A* algorithm for the industrial robot path planning with high success rate and short length. *Robotics and Autonomous Systems* 106:26–37. <https://doi.org/10.1016/j.robot.2018.04.007>
- Gerkey B, Howard A (2011) Player: wavefront. http://playerstage.sourceforge.net/doc/Player-cvs/player/group__driver__wavefront.html. Accessed 21 Feb 2019
- Gerkey BP, Mataric MJ (2002) Sold!: auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation* 18:758–768. <https://doi.org/10.1109/TRA.2002.803462>
- Gerkey BP, Mataric MJ (2003a) Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422). IEEE, Taipei, Taiwan, pp 3862–3868
- Gerkey BP, Mataric MJ (2003b) Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422). IEEE, Taipei, Taiwan, pp 3862–3868
- Gerkey BP, Matarić MJ (2004a) A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research* 23:939–954. <https://doi.org/10.1177/0278364904045564>
- Gerkey BP, Matarić MJ (2004b) A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research* 23:939–954. <https://doi.org/10.1177/0278364904045564>
- Gerkey BP, Vaughan RT, Howard A (2003) The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: In Proceedings of the 11th International Conference on Advanced Robotics. pp 317–323
- Glover F (1995) Tabu Thresholding: Improved Search by Nonmonotonic Trajectories. *ORSA Journal on Computing* 7:426–442. <https://doi.org/10.1287/ijoc.7.4.426>
- Grand View Research Automated Guided Vehicle Market | AGV Industry Report, 2027. In: www.grandviewresearch.com/. <https://www.grandviewresearch.com/industry-analysis/automated-guided-vehicle-agv-market>. Accessed 5 Apr 2021

- Hall R (ed) (1999) Handbook of Transportation Science. Springer US
- Han SD, Yu J (2019) Effective Heuristics for Multi-Robot Path Planning in Warehouse Environments. In: 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). pp 10–12
- Hao W, Xu X (2014) Immune ant colony optimization network algorithm for multi-robot path planning. In: 2014 IEEE 5th International Conference on Software Engineering and Service Science. pp 1118–1121
- Heinemann T, Riedel O, Lechler A (2019) Generating Smooth Trajectories in Local Path Planning for Automated Guided Vehicles in Production. *Procedia Manufacturing* 39:98–105. <https://doi.org/10.1016/j.promfg.2020.01.233>
- Hermann M, Pentek T, Otto B (2016) Design Principles for Industrie 4.0 Scenarios. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). pp 3928–3937
- Herrero-Pérez D, Martínez-Barberá H (2011) Decentralized Traffic Control for Non-Holonomic Flexible Automated Guided Vehicles in Industrial Environments. *Advanced Robotics* 25:739–763. <https://doi.org/10.1163/016918611X563283>
- Hitchcock FL (1941) The Distribution of a Product from Several Sources to Numerous Localities. *Journal of Mathematics and Physics* 20:224–230. <https://doi.org/10.1002/sapm1941201224>
- Hu D, Zhong H, Li S, et al (2020) Segmenting areas of potential contamination for adaptive robotic disinfection in built environments. *Building and Environment* 184:107226. <https://doi.org/10.1016/j.buildenv.2020.107226>
- Huang L, Zhou M, Hao K, Hou E (2019) A survey of multi-robot regular and adversarial patrolling. *IEEE/CAA Journal of Automatica Sinica* 6:894–903. <https://doi.org/10.1109/JAS.2019.1911537>
- Iftikhar K, Khan MT (2016) Resource sharing and deadlock avoidance in multi robot systems using Market Based Approach. In: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). Vancouver, BC, pp 1–7
- Irnich S, Toth P, Vigo D (2014) Chapter 1: The Family of Vehicle Routing Problems. pp 1–33
- Jeon S, Lee J, Kim J (2017) Multi-robot task allocation for real-time hospital logistics. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp 2465–2470
- Ji M, Xia J (2010) Analysis of vehicle requirements in a general automated guided vehicle system based transportation system. *Computers & Industrial Engineering* 59:544–551. <https://doi.org/10.1016/j.cie.2010.06.013>
- Jones PB, Blake MA, Archibald JK (2002) A real-time algorithm for task allocation. In: Proceedings of the IEEE International Symposium on Intelligent Control. Vancouver, BC, Canada, pp 672–677

- Jose K, Pratihari DK (2016) Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems* 80:34–42. <https://doi.org/10.1016/j.robot.2016.02.003>
- Kabir QS, Suzuki Y (2018) Increasing manufacturing flexibility through battery management of automated guided vehicles. *Computers & Industrial Engineering* 117:225–236. <https://doi.org/10.1016/j.cie.2018.01.026>
- Kattepur A, Rath HK, Simha A, Mukherjee A (2018) Distributed optimization in multi-agent robotics for industry 4.0 warehouses. In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. Association for Computing Machinery, New York, NY, USA, pp 808–815
- Kawakami T, Takata S (2012) Battery Life Cycle Management for Automatic Guided Vehicle Systems. In: Matsumoto M, Umeda Y, Masui K, Fukushige S (eds) *Design for Innovative Value Towards a Sustainable Society*. Springer Netherlands, Dordrecht, pp 403–408
- Khamis A, Hussein A, Elmogy A (2015) Multi-robot Task Allocation: A Review of the State-of-the-Art. In: Koubâa A, Martínez-de Dios JR (eds) *Cooperative Robots and Sensor Networks 2015*. Springer International Publishing, Cham, pp 31–51
- Kim S, Jin H, Seo M, Har D (2019) Optimal Path Planning of Automated Guided Vehicle using Dijkstra Algorithm under Dynamic Conditions. In: *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*. pp 231–236
- Koenig S, Tovey C, Lagoudakis M, et al (2006) The power of sequential single-item auctions for agent coordination. In: *proceedings of the 21st national conference on Artificial intelligence - Volume 2*. AAAI Press, Boston, Massachusetts, pp 1625–1629
- Kong X, Gao Y, Wang T, et al (2019) Multi-robot Task Allocation Strategy based on Particle Swarm Optimization and Greedy Algorithm. In: *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. pp 1643–1646
- Korsah GA, Stentz A, Dias MB (2013) A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research* 32:1495–1512. <https://doi.org/10.1177/0278364913496484>
- Krause A (2007) *The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs*. In: *The Robotics Institute Carnegie Mellon University*. <https://www.ri.cmu.edu/publications/the-dynamic-hungarian-algorithm-for-the-assignment-problem-with-changing-costs/>. Accessed 7 Apr 2021
- Kriegel J, Rissbacher C, Reckwitz L, Tuttle-Weidinger L (2021) The requirements and applications of autonomous mobile robotics (AMR) in hospitals from the perspective of nursing officers. *International Journal of Healthcare Management* 0:1–7. <https://doi.org/10.1080/20479700.2020.1870353>

- Kundakcioglu OE, Alizamir S (2009) Generalized assignment problem. In: Floudas CA, Pardalos PM (eds) *Encyclopedia of Optimization*. Springer US, Boston, MA, pp 1153–1162
- Lamballais T, Roy D, De Koster MBM (2017) Estimating performance in a Robotic Mobile Fulfillment System. *European Journal of Operational Research* 256:976–990. <https://doi.org/10.1016/j.ejor.2016.06.063>
- Le CP, Pham AQ, La HM, Feil-Seifer D (2020) A Multi-Robotic System for Environmental Dirt Cleaning. In: 2020 IEEE/SICE International Symposium on System Integration (SII). pp 1294–1299
- Le-Anh T, De Koster MBM (2006) A review of design and control of automated guided vehicle systems. *European Journal of Operational Research* 171:1–23. <https://doi.org/10.1016/j.ejor.2005.01.036>
- Li B, Liu H, Xiao D, et al (2017) Centralized and optimal motion planning for large-scale AGV systems: A generic approach. *Advances in Engineering Software* 106:33–46. <https://doi.org/10.1016/j.advengsoft.2017.01.002>
- Li X, Liu Z, Zhang Y (2018) A Novel Improved Ant Colony Algorithm for Multi-Robot Task Allocation. In: 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). pp 1629–1633
- Liu R, Xie X, Augusto V, Rodriguez C (2013) Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research* 230:475–486. <https://doi.org/10.1016/j.ejor.2013.04.044>
- Ljasenko S, Ferreira P, Justham L, Lohse N (2019) Decentralised vs partially centralised self-organisation model for mobile robots in large structure assembly. *Computers in Industry* 104:141–154. <https://doi.org/10.1016/j.compind.2018.09.002>
- López-González A, Meda Campaña JA, Hernández Martínez EG, Contro PP (2020) Multi robot distance based formation using Parallel Genetic Algorithm. *Applied Soft Computing* 86:105929. <https://doi.org/10.1016/j.asoc.2019.105929>
- Lourenço HR, Martin OC, Stützle T (2003) Iterated Local Search. In: Glover F, Kochenberger GA (eds) *Handbook of Metaheuristics*. Springer US, Boston, MA, pp 320–353
- Luperto M, Antonazzi M, Amigoni F, Borghese NA (2020) Robot exploration of indoor environments using incomplete and inaccurate prior knowledge. *Robotics and Autonomous Systems* 133:103622. <https://doi.org/10.1016/j.robot.2020.103622>
- Madridano Á, Al-Kaff A, Martín D, de la Escalera A (2021) Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications* 173:114660. <https://doi.org/10.1016/j.eswa.2021.114660>
- MarketsandMarkets Automated Guided Vehicle Market by Type, Navigation Technology, Industry | COVID-19 Impact Analysis | MarketsandMarkets™.

- <https://www.marketsandmarkets.com/Market-Reports/automated-guided-vehicle-market-27462395.html>. Accessed 5 Apr 2021
- Martin O, Otto SW, Felten EW (1992) Large-step markov chains for the TSP incorporating local search heuristics. *Operations Research Letters* 11:219–224. [https://doi.org/10.1016/0167-6377\(92\)90028-2](https://doi.org/10.1016/0167-6377(92)90028-2)
- Martínez-Barberá H, Herrero-Pérez D (2010) Autonomous navigation of an automated guided vehicle in industrial environments. *Robotics and Computer-Integrated Manufacturing* 26:296–311. <https://doi.org/10.1016/j.rcim.2009.10.003>
- McHANEY R (1995) Modelling battery constraints in discrete event automated guided vehicle simulations. *International Journal of Production Research* 33:3023–3040. <https://doi.org/10.1080/00207549508904859>
- Meissner H, Ilse R, Aurich JC (2017) Analysis of Control Architectures in the Context of Industry 4.0. *Procedia CIRP* 62:165–169. <https://doi.org/10.1016/j.procir.2016.06.113>
- Mosteo AR (2006) Simulated annealing for multi-robot hierarchical task allocation with flexible constraints and objective functions
- Nazarahari M, Khanmirza E, Doostie S (2019) Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications* 115:106–120. <https://doi.org/10.1016/j.eswa.2018.08.008>
- Nedjah N, Ribeiro LM, de Macedo Mourelle L (2021) Communication optimization for efficient dynamic task allocation in swarm robotics. *Applied Soft Computing* 105:107297. <https://doi.org/10.1016/j.asoc.2021.107297>
- Németh I, Püspöki J, Viharos AB, et al (2019) Layout configuration, maintenance planning and simulation of AGV based robotic assembly systems. *IFAC-PapersOnLine* 52:1626–1631. <https://doi.org/10.1016/j.ifacol.2019.11.433>
- Nunes E, Manner M, Mitiche H, Gini M (2017) A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems* 90:55–70. <https://doi.org/10.1016/j.robot.2016.10.008>
- O’Leary B (2019) Don’t be Greedy, be Neighborly, a new assignment algorithm. In: 2019 IEEE Aerospace Conference. Big Sky, MT, USA, pp 1–8
- Palmer AW, Hill AJ, Scheduling SJ (2018) Modelling Resource Contention in Multi-Robot Task Allocation Problems with Uncertain Timing. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane, QLD, pp 3693–3700
- Parker LE (1998) ALLIANCE: an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation* 14:220–240. <https://doi.org/10.1109/70.681242>
- Parker LE (2008) Multiple Mobile Robot Systems. In: Siciliano B, Khatib O (eds) *Springer Handbook of Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 921–941

- Pashna M, Yusof R, Ismail ZH, et al (2020) Autonomous multi-robot tracking system for oil spills on sea surface based on hybrid fuzzy distribution and potential field approach. *Ocean Engineering* 207:107238. <https://doi.org/10.1016/j.oceaneng.2020.107238>
- Peake A, McCalmon J, Zhang Y, et al (2020) Wilderness Search and Rescue Missions using Deep Reinforcement Learning. In: 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). pp 102–107
- Pentico DW (2007) Assignment problems: A golden anniversary survey. *European Journal of Operational Research* 176:774–793. <https://doi.org/10.1016/j.ejor.2005.09.014>
- Pollaris H, Braekers K, Caris A, et al (2015) Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum* 37:297–330. <https://doi.org/10.1007/s00291-014-0386-3>
- Puig D, Garcia MA, Wu L (2011) A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation. *Robotics and Autonomous Systems* 59:635–653. <https://doi.org/10.1016/j.robot.2011.05.004>
- Rehman FU, Thomas G, Anderlini E (2019) Centralized Control System Design for Underwater Transportation using two Hovering Autonomous Underwater Vehicles (HAUVs). *IFAC-PapersOnLine* 52:13–18. <https://doi.org/10.1016/j.ifacol.2019.09.111>
- Romero RAF, Prestes E, Idiart MAP, Faria G (2012) Locally oriented potential field for controlling multi-robots. *Communications in Nonlinear Science and Numerical Simulation* 17:4664–4671. <https://doi.org/10.1016/j.cnsns.2011.10.027>
- Sahni S, Gonzalez T (1976) P-Complete Approximation Problems. *J ACM* 23:555–565. <https://doi.org/10.1145/321958.321975>
- Sarkar C, Paul HS, Pal A (2018) A Scalable Multi-Robot Task Allocation Algorithm. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp 5022–5027
- Sayed AS, Ammar HH, Shalaby R (2020) Centralized Multi-agent Mobile Robots SLAM and Navigation for COVID-19 Field Hospitals. In: 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES). pp 444–449
- Scherer J, Rinner B (2020) Multi-Robot Persistent Surveillance With Connectivity Constraints. *IEEE Access* 8:15093–15109. <https://doi.org/10.1109/ACCESS.2020.2967650>
- Scholz M, Lieret M, Bigott M, Franke J (2019) Distributed Software Architecture for AGVs for Separation of Dynamic and Static Obstacles. *Procedia Manufacturing* 38:367–374. <https://doi.org/10.1016/j.promfg.2020.01.047>
- Schrijver A (2005) On the History of Combinatorial Optimization (Till 1960). In: Aardal K, Nemhauser GL, Weismantel R (eds) *Handbooks in Operations Research and Management Science*. Elsevier, pp 1–68

- Silverman MC, Nies D, Jung B, Sukhatme GS (2002) Staying alive: a docking station for autonomous robot recharging. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292). pp 1050–1055 vol.1
- Simões D, Lau N, Paulo Reis L (2020) Multi-agent actor centralized-critic with communication. *Neurocomputing* 390:40–56. <https://doi.org/10.1016/j.neucom.2020.01.079>
- Singh S, Martinetti A, Majumdar A, Dongen L van (eds) (2019) *Transportation Systems: Managing Performance through Advanced Maintenance Engineering*. Springer Singapore
- Soleimanpour-Moghadam M, Nezamabadi-Pour H (2020) Discrete Genetic Algorithm for Solving Task Allocation of Multi-robot Systems. In: 2020 4th Conference on Swarm Intelligence and Evolutionary Computation (CSIEC). pp 006–009
- Søraa DrRA, Fostervold DrME (2021) Social domestication of service robots: The secret lives of Automated Guided Vehicles (AGVs) at a Norwegian hospital. *International Journal of Human-Computer Studies* 102627. <https://doi.org/10.1016/j.ijhcs.2021.102627>
- Srinivasan V, Thompson GL (1973) An Algorithm for Assigning Uses to Sources in a Special Class of Transportation Problems. *Operations Research*. <https://doi.org/10.1287/opre.21.1.284>
- Sullivan N, Grainger S, Cazzolato B (2019) Sequential single-item auction improvements for heterogeneous multi-robot routing. *Robotics and Autonomous Systems* 115:130–142. <https://doi.org/10.1016/j.robot.2019.02.016>
- Talbi E-G (2009) *Metaheuristics: From Design to Implementation*. John Wiley & Sons
- Tavakoli M, Carriere J, Torabi A (2020) Robotics, Smart Wearable Technologies, and Autonomous Intelligent Systems for Healthcare During the COVID-19 Pandemic: An Analysis of the State of the Art and Future Vision. *Advanced Intelligent Systems* 2:2000071. <https://doi.org/10.1002/aisy.202000071>
- Toth P, Vigo D (2015) *Vehicle Routing: Problems, Methods, and Applications*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104)
- Ullrich G (2015) The History of Automated Guided Vehicle Systems. In: Ullrich G (ed) *Automated Guided Vehicle Systems: A Primer with Practical Applications*. Springer, Berlin, Heidelberg, pp 1–14
- Ulrich I, Borenstein J VFH*: Local Obstacle Avoidance with Look-Ahead Verification. 7
- Valmiki P, Simha Reddy A, Panchakarla G, et al (2018) A Study on Simulation Methods for AGV Fleet Size Estimation in a Flexible Manufacturing System. *Materials Today: Proceedings* 5:3994–3999. <https://doi.org/10.1016/j.matpr.2017.11.658>
- Vaughan R (2008) Massively multi-robot simulation in stage. *Swarm Intelligence* 2:189–208. <https://doi.org/10.1007/s11721-008-0014-4>

- Vis IFA (2006a) Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research* 170:677–709. <https://doi.org/10.1016/j.ejor.2004.09.020>
- Vis IFA (2006b) Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research* 170:677–709. <https://doi.org/10.1016/j.ejor.2004.09.020>
- Witczak M, Majdzik P, Stetter R, Lipiec B (2020) A fault-tolerant control strategy for multiple automated guided vehicles. *Journal of Manufacturing Systems* 55:56–68. <https://doi.org/10.1016/j.jmsy.2020.02.009>
- Witczak M, Majdzik P, Stetter R, Lipiec B (2019) Multiple AGV fault-tolerant within an agile manufacturing warehouse □□The work was supported by the National Science Centre, Poland under grant: UMO-2017/27/B/ST7/00620. *IFAC-PapersOnLine* 52:1914–1919. <https://doi.org/10.1016/j.ifacol.2019.11.482>
- Witczak M, Stetter R, Buciakowski M, et al (2018) Design of diagnostic estimators for an automated guided vehicle. *IFAC-PapersOnLine* 51:1004–1009. <https://doi.org/10.1016/j.ifacol.2018.09.710>
- Yan J, Li L (2012) Fault-tolerant controller design for automated guided vehicle systems based on Petri nets. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems. pp 1531–1536
- Yan R, Jackson LM, Dunnett SJ (2017) Automated guided vehicle mission reliability modelling using a combined fault tree and Petri net approach. *Int J Adv Manuf Technol* 92:1825–1837. <https://doi.org/10.1007/s00170-017-0175-7>
- Yu VF, Jewpanya P, Redi AANP, Tsao Y-C (2021) Adaptive neighborhood simulated annealing for the heterogeneous fleet vehicle routing problem with multiple cross-docks. *Computers & Operations Research* 129:105205. <https://doi.org/10.1016/j.cor.2020.105205>
- Zadarnowska K, Tchoń K (2015) Modeling and motion planning of wheeled mobile robots subject to slipping. In: 2015 10th International Workshop on Robot Motion and Control (RoMoCo). pp 78–83
- Zhang H, Roy U (2019) A semantics-based dispatching rule selection approach for job shop scheduling. *J Intell Manuf* 30:2759–2779. <https://doi.org/10.1007/s10845-018-1421-z>
- Zou B, Xu X, Gong Y (Yale), De Koster R (2018) Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system. *European Journal of Operational Research* 267:733–753. <https://doi.org/10.1016/j.ejor.2017.12.008>

Résumé

Les systèmes de transport multi-robot dans les industries, constitue l'implémentation la plus réussie pour les systèmes multi robot mobile (SMR). L'objectif de ce système de transport est de maximiser la productivité en maximisant le nombre d'objets transportés. Toutefois, l'avènement de l'industrie durable à inciter à une gestion efficace de l'énergie des systèmes.

Dans cette thèse, nous proposons une nouvelle stratégie d'allocation des tâches, intitulée D-ILS, pour un système de transport multi-robot, qui maximise la production et minimise la consommation de l'énergie du SMR. la D-ILS est basée sur la métaheuristique de la recherche locale itérative (ILS) et sur une nouvelle fonction d'utilité modélisée sur des heuristiques élémentaires. Le protocole de la D-ILS utilise l'aspect multi-agent des SMR, pour imiter les métaheuristiques à population de points afin de résoudre l'allocation des tâches entre les robots de manière parallèle et distribuée. Chaque robot propose une solution pour l'allocation des tâches du groupe en utilisant l'ILS. Ainsi, la D-ILS associe les avantages d'exploration des métaheuristiques à population de points et les avantages d'exploitation des métaheuristiques de recherche à un seul point.

La D-ILS est implémentée sur une plateforme de simulation. Les performances sont évaluées en termes d'énergie consommée, de la productivité, de la productivité par d'énergie consommée, et de robustesse. La D-ILS est comparée à trois heuristiques inspirées des règles de dispatching des véhicules à guidage automatique. Les résultats indique que la D-ILS optimise la productivité, gère efficacement la consommation d'énergie du SMR et donne de bonnes performances en termes de productivité dans un scénario dynamique.

Mots-clés: Système multi-robot; Allocation des tâches; Gestion d'énergie; Recherche locale itérative.

Abstract

The industrial transportation multi-robot systems are the most successful implementation of the multi-robot systems (MRS). The aim in such transport systems is to maximize the productivity by transporting the maximum of goods. However, the emergence of the sustainable manufacturing stressed the need for an efficient energy management of the systems.

In this thesis, we propose a novel task allocation strategy (D-ILS) for a transportation multi-robot system. This strategy aim to maximize the productivity and minimize the energy consumption of the MRS. The D-ILS is based on the iterative local search metaheuristic (ILS) and a novel heuristic-based utility function. The D-ILS protocol emulate the population based metaheuristics to solve the task allocation, in parallel and distributed manner, by using the multi-agent aspect of the MRS. Each robot solves the task allocation for the group using the ILS. Thus, D-ILS combine the exploration advantage of the population-base metaheuristics and the exploitation advantage of the single-point-search metaheuristics.

The D-ILS is implemented in a simulation platform. The performance are evaluated in terms of, the energy consumed, the productivity, the productivity per energy consumed, and the robustness. The D-ILS is compared to three heuristics inspired from the dispatching rules of the automated guided vehicles. The results indicate that D-ILS optimises the productivity, efficiently manages the energy consumption of the MRS, and denote the good performance in terms of the productivity in dynamic scenarios.

Keywords: Multi-robot system; Multi-robot task allocation; Energy management; Iterative local search.

ملخص:

تعد أنظمة النقل متعددة الروبوتات في المصانع، التطبيق الأكثر نجاحًا للأنظمة متعددة الروبوتات (SMR). الهدف من نظام النقل هذا هو زيادة الإنتاجية من خلال زيادة المنقولات إلى حد أقصى. ومع ظهور الصناعة المستدامة تم التحفيز على الإدارة الفعالة لطاقة هذه الأنظمة.

في هذه الأطروحة نقترح إستراتيجية جديدة لتخصيص المهام تسمى (D-ILS) لنظام نقل متعدد الروبوتات، ميزتها الزيادة من الإنتاج والتقليل من استهلاك الطاقة في (SMR). تعتمد (D-ILS) على خوارزمية الإرشاد الذاتي للبحث المحلي المتكرر (ILS) وعلى دالة جديدة مبنية على خوارزمية إرشاد أساسية وظيفتها تقدير نفعية الروبوتات. يستخدم بروتوكول الإستراتيجية (D-ILS) صفة متعددة الوكلاء لنظام (SMR)، لتقليد خوارزمية الإرشاد الذاتي متعددة الحلول من تخصيص المهام بين الروبوتات بطريقة موازية وموزعة حيث يقترح كل روبوت حلاً لتخصيص مهام المجموعة باستخدام (ILS) ومنه فإن (D-ILS) يجمع بين مزايا استكشاف خوارزمية الإرشاد الذاتي متعددة الحلول مع مزايا استغلال خوارزمية الإرشاد الذاتي أحادي الحلول يتم تنفيذ إستراتيجية (D-ILS) على منصة محاكاة رقمية و يتم تقييم نجاعة (D-ILS) من حيث الطاقة المستهلكة، الإنتاجية والإنتاجية لكل وحدة من الطاقة المستهلكة وكذا من حيث المتانة. تتم مقارنة (D-ILS) بثلاث خوارزميات الإرشاد مستوحات من قواعد الإرسال للمركبات الموجهة أوتوماتيكياً.

تظهر النتائج المتحصل عليها أن (D-ILS) تمكن من تحسين الإنتاجية، وتسيير استهلاك الطاقة في نظام (SMR) كما تشير إلى متانة الإستراتيجية من حيث الإنتاجية في مواجهة السيناريوهات الديناميكية.

الكلمات المفتاحية: نظام متعدد الروبوتات – تخصيص المهام – تسيير الطاقة - البحث المحلي المتكرر.