



HAL
open science

Optimisation de l'ordonnancement des tâches de transport des robots intelligents et collaboratifs dans le contexte de l'industrie 5.0

Kader Sanogo

► **To cite this version:**

Kader Sanogo. Optimisation de l'ordonnancement des tâches de transport des robots intelligents et collaboratifs dans le contexte de l'industrie 5.0. Informatique [cs]. Arts et Métiers, 2024. Français. NNT : 2024ENAME056 . tel-04927759

HAL Id: tel-04927759

<https://hal.science/tel-04927759v1>

Submitted on 3 Feb 2025

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

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

ÉCOLE DOCTORALE SCIENCES ET MÉTIERS DE L'INGÉNIEUR
CESI LINEACT - Campus d'Angoulême

THÈSE

présentée par : **Kader SANOGO**
soutenue le : **05 décembre 2024**

pour obtenir le diplôme de **Docteur** délivré par :

L'École Nationale Supérieure d'Arts et Métiers

Discipline : Sections CNU 27 & 61

Spécialité : Informatique

**Optimisation de l'ordonnancement des tâches de
transport des robots intelligents et collaboratifs dans le
contexte de l'Industrie 5.0**

THÈSE dirigée par :
M'hammed SAHNOUN, CESI LINEACT

et co-encadrée par :
Abdelkader MEKHALEF BENHAFSSA, CESI LINEACT

Jury

Mme Leïla Kloul	Professeure, DAVID, Université Paris-Saclay	Présidente
M. Mamadou K. Traoré	Professeur, IMS, Université de Bordeaux	Rapporteur
M. Farouk Belkadi	MCF HDR, LS2N, École Centrale de Nantes	Rapporteur
Mme Olga Battaïa	Directrice de recherche, KEDGE Bordeaux	Examinatrice
M. M'hammed Sahnoun	Directeur de recherche, CESI Rouen	Examinateur
M. Abdelkader Mekhalef	EC, CESI Angoulême	Examinateur

« Formez-vous, armez-vous de sciences jusqu'aux dents (...) et
arrachez votre patrimoine culturel. »

Prof. Cheikh Anta DIOP

Remerciements

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux.

Louage à Allah, Seigneur de l'Univers, à Qui j'exprime ma profonde gratitude, pour Sa grâce infinie, Sa protection et Ses bénédictions qui m'ont accompagné tout au long de ma vie.

Je tiens à remercier chaleureusement ma famille, particulièrement mon père qui n'a ménagé aucun effort pour que je fasse de bonnes études. À mon épouse bien-aimée, Fatmath Zahara, pour son amour, son soutien indéfectible et sa compréhension sans faille.

Je souhaite également exprimer ma reconnaissance à mon équipe d'encadrement. À mon directeur de thèse, M'hammed SAHNOUN, pour ses conseils précieux, son expertise et sa bienveillance. À mon encadrant, Abdelkader MEKHALEF, mon grand frère, qui n'a jamais douté de mes capacités à réaliser ce projet de thèse. Mes remerciements vont également à l'endroit de mes collègues de travail, pour leur convivialité et leur sympathie à mon égard. À Johann CUENIN, avec qui j'ai partagé de nombreuses heures de réflexion et de discussion sur mon sujet de thèse entre autres. À Diana CARTHY-MCHUGH, dont l'assistance en anglais a été utile pour nos publications scientifiques. Je n'oublie pas Elodie, Wendy, Sama, Karim, Lilia, Alexandre, Célestin... aux cotés de qui j'ai pris plaisir à travailler.

Enfin, je souhaite exprimer ma sincère reconnaissance envers mon pays, la Côte d'Ivoire, qui m'a offert les opportunités et les ressources nécessaires pour mener à bien mon cursus universitaire. C'est avec une grande fierté que je dédie ce travail à ma patrie, en espérant que mes recherches contribueront modestement à son développement scientifique et industriel.

À toutes et à tous, mes remerciements infinis !

REMERCIEMENTS

Résumé

L'optimisation de l'ordonnancement des tâches dans des environnements industriels complexes, où cohabitent humains et robots, soulève de nouveaux défis, notamment en matière d'évitement des collisions, de gestion des blocages (deadlocks), et d'optimisation du temps de production (makespan). À l'ère de l'Industrie 5.0, caractérisée par une collaboration accrue entre travailleurs humains et robots, ce travail de recherche vise à proposer des solutions innovantes pour améliorer les performances des systèmes de production dans des environnements collaboratifs et inclusifs.

Ainsi, cette thèse retrace l'évolution des systèmes industriels et met en lumière les limites des méthodes traditionnelles d'ordonnancement face à ce nouveau paradigme collaboratif. Elle développe une approche basée sur la simulation, dans laquelle la modélisation des interactions complexes entre les différents acteurs de l'atelier est représentée par un système multi-agents. De plus, les comportements des travailleurs humains sont modélisés à l'aide de chaînes de Markov et directement intégrés dans la simulation, afin d'avoir une représentation complète de l'atelier. Le simulateur ainsi obtenu permet d'évaluer les performances des robots mobiles en tenant compte des contraintes opérationnelles imposées par la présence humaine. La thèse examine également l'impact des véhicules autonomes intelligents (AIV), qui offrent une plus grande flexibilité que les véhicules autonomes guidés (AGV), sur l'efficacité des processus de production. Enfin, un algorithme d'ordonnancement dynamique, basé sur la théorie des jeux, est développé pour optimiser en temps réel la gestion des tâches dans l'atelier, en intégrant la présence des opérateurs humains. Cette approche permet de générer des ordonnancements réactifs et efficaces, garantissant une production résiliente face aux fluctuations de l'environnement industriel.

Mots-clés : Industrie 5.0, JSSPT, ordonnancement dynamique, simulation, système multi-agents, chaîne de Markov, théorie des jeux, AIV, AGV, humains.

RÉSUMÉ

Abstract

Task scheduling optimization in complex industrial environments, where humans and robots work seamlessly together, raises new challenges, particularly in terms of collision avoidance, deadlock management, and production time optimization (makespan). In the era of Industry 5.0, characterized by increased human-robot collaboration, this research aims to propose innovative solutions to enhance the performance of production systems in collaborative and inclusive environments.

Consequently, this thesis reviews the evolution of industrial revolutions and highlights the limitations of traditional scheduling methods in light of this new collaborative paradigm. It develops a simulation-based approach, where the complex interactions between various workshop actors are modeled using a multi-agent system. Moreover, human worker behaviors are modeled using Markov chains and directly integrated into the simulation, providing a comprehensive representation of the workshop. The resulting simulator allows the evaluation of mobile robots' performance, taking into account the operational constraints imposed by human presence. The thesis also examines the impact of Autonomous Intelligent Vehicles (AIV), which offer greater flexibility than Automated Guided Vehicles (AGV), on the efficiency of production processes. Finally, a dynamic scheduling algorithm based on game theory is developed to optimize task management in real-time, integrating the presence of human operators. This approach generates responsive and efficient schedules, ensuring resilient production in the face of fluctuations in the industrial environment.

Keywords : Industry 5.0, JSSPT, dynamic scheduling, simulation, multi-agent systems, Markov chain, game theory, AIV, AGV, humans.

ABSTRACT

Table des matières

Remerciements	5
Résumé	7
Abstract	9
Liste des tableaux	20
Liste des figures	24
Introduction	25
1 Contexte et problématique de la thèse	29
1.1 Introduction	30
1.2 Évolution des systèmes de production	32
1.2.1 Systèmes Manufacturiers Dédiés (SMD)	33
1.2.2 Systèmes Manufacturiers Flexibles (SMF)	34
1.2.3 Systèmes Manufacturiers Cellulaires (SMC)	34
1.2.4 Systèmes Manufacturiers Reconfigurables (SMR)	35
1.2.5 Systèmes Manufacturiers Intelligents (SMI)	36
1.3 L'avènement de l'Industrie 4.0 (I4.0)	38
1.3.1 L'origine du concept	38

TABLE DES MATIÈRES

1.3.2	Les composantes de l'I4.0	39
1.3.3	Quelques technologies majeures de l'I4.0	39
1.3.3.1	Les capteurs intelligents	40
1.3.3.2	Les robots	40
1.3.3.3	La simulation	40
1.3.3.4	Les données	41
1.4	Les limites de l'I4.0	41
1.5	Transition de l'industrie 4.0 vers l'industrie 5.0	43
1.6	Industrie 5.0 (I5.0)	44
1.6.1	Quelques définitions de l'I5.0	45
1.6.1.1	Définition n°1 (Michael Rada, 2018)	45
1.6.1.2	Définition n°2 (Saeid Nahavandi, 2019)	45
1.6.1.3	Définition n°3 (Francesco Longo <i>et al.</i> , 2020)	45
1.6.1.4	Définition n°4 (Yoong K. L. <i>et al.</i> , 2021)	45
1.6.1.5	Définition n°5 (Praveen Kumar <i>et al.</i> , 2022)	46
1.6.1.6	Définition n°6 (Commission Européenne, 2021)	46
1.6.2	L'impact de l'I5.0 sur les systèmes de production	46
1.7	Ordonnancement des tâches d'un système de production	48
1.7.1	Flow shop	49
1.7.2	Job shop	50
1.7.3	Open shop	50
1.7.4	Les tâches de transport dans les systèmes de production	51
1.7.4.1	Robotisation des tâches de transport	51
1.7.4.2	Importance des tâches de transport dans l'ordonnancement	53
1.8	Problématique de la thèse	53

TABLE DES MATIÈRES

1.8.1	Contexte scientifique	54
1.8.2	Problématique de recherche	54
1.8.3	Verrous scientifiques et techniques	55
1.8.4	Analyse de l'état de l'art et positionnement scientifique	56
1.8.5	Les contributions de la thèse	72
1.9	Conclusion	73
2	Modélisation et simulation multi-agents de l'atelier de production	75
2.1	Introduction	75
2.2	État de l'art	76
2.2.1	Simulation numérique	77
2.2.2	Simulation à événements discrets	78
2.2.3	Simulation basée sur les agents	79
2.2.4	Utilisation de la simulation dans la résolution des problèmes industriels	80
2.3	Modélisation de l'atelier de production	81
2.3.1	Système multi-agents	81
2.3.2	Modélisation des activités de l'atelier	84
2.3.3	Modélisation de l'atelier	85
2.4	Développement du simulateur	87
2.4.1	Description du framework	87
2.4.2	Implémentation du modèle de simulation	90
2.4.2.1	Définition des <i>breeds</i> (types d'agent)	91
2.4.2.2	Définition des variables	92
2.4.2.3	Initialisation du modèle	93
2.4.2.3.1	setup-patch :	94
2.4.2.3.2	setup-globals :	96

TABLE DES MATIÈRES

2.4.2.3.3	setup-machines :	96
2.4.2.3.4	setup-stocks :	96
2.4.2.3.5	setup-transporters :	96
2.4.2.3.6	setup-jobs :	96
2.4.3	Interface du simulateur	97
2.4.4	Exécution de la simulation	99
2.4.4.1	Choix du type de transporteurs	100
2.4.4.2	Activation de la contrainte d'évitement de collisions	100
2.4.4.3	Choix du type d'ordonnancement	103
2.4.4.3.1	Simulation d'ordonnancement prévisionnel	104
2.4.4.3.2	Simulation d'ordonnancement dynamique	105
2.4.4.4	Procédure <i>go</i> du simulateur	105
2.4.5	Intérêt du simulateur	107
2.5	Conclusion	108
3	Simulation et optimisation de l'ordonnancement des tâches de l'atelier de production	109
3.1	Introduction	109
3.2	État de l'art	110
3.2.1	Principales méthodes de résolution	111
3.2.1.1	Résolution par méthodes exactes	111
3.2.1.2	Résolution par méthodes approchées	112
3.2.1.3	Résolution par simulation-optimisation	113
3.2.2	Problème d'évitement des collisions et deadlocks	114
3.3	Modélisation du problème	116
3.3.1	JSSPT	116

TABLE DES MATIÈRES

3.3.2	Description de l'évitement de collisions et des deadlocks	120
3.3.3	Résolution des deadlocks	120
3.4	Expérimentations	122
3.4.1	Description de l'algorithme VNS	122
3.4.2	Expérience n°1	123
3.4.2.1	Démarche expérimentale	123
3.4.2.2	Résultats	125
3.4.2.3	Analyse des résultats	126
3.4.3	Expérience n°2	131
3.4.3.1	Démarche expérimentale	131
3.4.3.2	Résultats	131
3.4.3.3	Analyse des résultats	132
3.5	Conclusion	134
4	Transition des AGV vers les AIV dans le contexte de l'industrie 5.0	139
4.1	Introduction	139
4.2	État de l'art	141
4.2.1	Modélisation du comportement des humains	141
4.2.1.1	Méthodes mathématiques	142
4.2.1.2	Méthodes basées sur la connaissance	143
4.2.1.3	Méthodes basées sur l'apprentissage automatique	144
4.2.1.4	Méthodes déterministes	144
4.2.2	L'utilisation des AIV dans les systèmes de production	145
4.3	Description du problème	147
4.3.1	Transition des AGV vers les AIV	147
4.3.2	Intégration des humains dans l'atelier	148

TABLE DES MATIÈRES

4.4	Modélisation du comportement des opérateurs humains	149
4.5	Mise à jour du modèle de simulation	153
4.6	Expérimentations	155
4.6.1	Expérience n°1	155
4.6.1.1	Démarche expérimentale	155
4.6.1.2	Résultats	156
4.6.1.3	Analyse des résultats	156
4.6.2	Expérience n°2	161
4.6.2.1	Démarche expérimentale	162
4.6.2.2	Résultats	163
4.6.2.3	Analyse des résultats	163
4.7	Conclusion	168
5	Optimisation dynamique de l'ordonnancement dans le contexte de l'industrie 5.0	171
5.1	Introduction	171
5.2	État de l'art	173
5.2.1	Ordonnancement dynamique	173
5.2.1.1	Approches d'ordonnancement dynamique	175
5.2.1.1.1	Ordonnancement entièrement réactif	175
5.2.1.1.2	Ordonnancement prédictif-réactif	175
5.2.1.1.3	Ordonnancement pro-actif robuste	175
5.2.1.2	Techniques d'ordonnancement dynamique	176
5.2.1.2.1	Les règles de priorité	176
5.2.1.2.2	Les heuristiques	176
5.2.1.2.3	Les métaheuristiques	176
5.2.1.2.4	Les techniques d'intelligence artificielle (IA)	176

TABLE DES MATIÈRES

5.2.1.2.5	Les techniques basées sur les agents	177
5.2.2	Allocation des tâches dans un système multi-robots (MRTA)	177
5.2.2.1	Comparaison des différentes techniques de résolution du MRTA	180
5.3	Ordonnancement dynamique par la théorie des jeux	181
5.3.1	Rappel des concepts théoriques	182
5.3.1.1	Définition	183
5.3.1.2	Équilibre de Nash	183
5.3.1.3	Jeux de potentiel	183
5.3.1.4	Problème d'ordonnancement des tâches de l'atelier (JSSPT)	184
5.3.2	Modélisation de l'ordonnancement	185
5.3.3	Définition des fonctions d'utilités	187
5.3.4	Optimisation de l'ordonnancement	190
5.3.5	Caractéristiques de la fonction d'utilité globale	191
5.4	Implémentation de l'algorithme d'ordonnancement	194
5.5	Expérimentations	195
5.5.1	Description de l'algorithme d'ordonnancement dynamique par entropie	197
5.5.2	Expérience n°1	198
5.5.2.1	Démarche expérimentale	199
5.5.2.2	Résultats	200
5.5.2.3	Analyse des résultats	200
5.5.3	Expérience n°2	206
5.5.3.1	Démarche expérimentale	206
5.5.3.2	Résultats	207
5.5.3.3	Analyse des résultats	207
5.6	Conclusion	211

TABLE DES MATIÈRES

Conclusion et perspectives	213
Bibliographie	217
Liste des annexes	248
A Les différents ensemble de jobs de l'atelier	249
B Les algorithmes du programme principal de la simulation	251

Liste des tableaux

1.1	État de l'art de la littérature (Notation : POT : Problème d'ordonnancement des tâches; RpO : Résolution par optimisation; RpS : Résolution par simulation; CdTT : Considération des tâches de transport; TdR : Type de robots; CdH : Considération des humains.)	71
2.1	Description des interactions entre les agents du MAS	83
2.2	Temps de trajet tels que proposés par Ü. Bilge et G. Ulusoy [1]	88
3.1	Notations	117
3.2	Résultats de l'expérience n°1	136
3.3	Résultats de l'expérience n°2	137
4.1	Les temps de trajets minimums requis pour les AIV	149
4.2	Description des interactions entre les agents du MAS	154
4.3	Résultats de l'expérience n°1 : comparaison AGV vs AIV	157
4.4	Résultats de l'expérience n°1 : intégration des humains dans l'atelier	164
5.1	Comparaison des différentes techniques de résolution du MRTA	180
5.2	Liste des symboles	182
5.3	Résultats de l'expérience n°1 : simulation dans un environnement déterministe	204
5.4	Résultats de l'expérience n°2 : simulation dans un environnement dynamique	205

LISTE DES TABLEAUX

A.1 Les ensembles de job proposés par Ü. Bilge et G. Ulusoy [1] 250

Table des figures

1.1	Illustration de l'évolution industrielle (Praveen K. M. R. <i>et al.</i> , 2022) [2]	31
1.2	Décomposition d'un système de production (Chalfoun, 2014) [3]	32
1.3	Illustration d'un SMD (Chalfoun, 2014) [3]	34
1.4	Illustration d'un SMF (Chalfoun, 2014) [3]	35
1.5	Illustration d'un SMC (S. Ariaifar et N. Ismail, 2009) [4]	35
1.6	Illustration d'un SMR (Chalfoun, 2014) [3]	36
1.7	Illustration d'un SMI (David Balderas <i>et al.</i> , 2021) [5]	37
1.8	Les neuf (09) piliers de l'I4.0 selon Gomez C. <i>et al.</i> [6]	38
1.9	Les principales technologies de l'I4.0 (Shyam M. <i>et al.</i> , 2021) [7]	39
1.10	Les fondements de l'I5.0 (Momenta, 2024) [8]	44
1.11	Les différents types d'interaction homme-robot (Roohollah Jahanmahin <i>et al.</i> , 2022) [9]	47
1.12	Schéma illustratif d'un flow shop (Framinan Jose <i>et al.</i> , 2014) [10]	50
1.13	Schéma illustratif d'un job shop (Framinan Jose <i>et al.</i> , 2014) [10]	50
1.14	Schéma illustratif d'un open shop (Framinan Jose <i>et al.</i> , 2014) [10]	51
1.15	Illustration de véhicules autonomes terrestres (Alstef Group, 2024) [11]	52
2.1	Le MAS (adaptation de Xu Yiyi <i>et al.</i> [12] (2017))	82
2.2	Processus de traitement d'un job avec trois opérations	85
2.3	Modélisation de la simulation	86
2.4	Les 4 différents layouts proposés par Ü. Bilge et G. Ulusoy [1]	87

TABLE DES FIGURES

2.5	Le logo de Netlogo [13]	88
2.6	Logigramme du programme principal de la simulation	92
2.7	Les principales variables du programme de simulation	93
2.8	Onglet <i>Interface</i> du simulateur	97
2.9	Sélecteur du type de transporteurs	100
2.10	Commutateur d'activation de la contrainte d'évitement de collisions	100
2.11	Évitement de collisions entre deux AGV	102
2.12	Évitement de collisions entre deux AIV	102
2.13	Système de réservation de portions de chemin	103
2.14	Mécanisme d'évitement de collisions aux intersections	103
2.15	Le choix du mode de simulation	104
2.16	Logigramme illustratif de la procédure <i>go</i>	106
3.1	Représentation de la méthode VNS proposée par Abderrahim <i>et al.</i> [14]	124
3.2	Logigramme de la démarche expérimentale de l'expérience n°1	125
3.3	Comparaison des résultats des simulations sur chaque layout (Kader Sanogo <i>et al.</i> , 2023) [15]	127
3.4	Les diagrammes de Gantt combinés des simulations des cas 1 et 2 de EX12 (Kader Sanogo <i>et al.</i> , 2023) [15]	128
3.5	Les diagrammes de Gantt combinés des simulations des cas 1 et cas 2 de EX32 (Kader Sanogo <i>et al.</i> , 2023) [15]	128
3.6	Un aperçu détaillé de l'ordonnancement des AGV sur l'instance EX32 (Kader Sanogo <i>et al.</i> , 2023) [15]	129
3.7	Les diagrammes de Gantt combinés des simulations des cas 1 et cas 2 de EX82 (Kader Sanogo <i>et al.</i> , 2023) [15]	130
3.8	Logigramme de la démarche expérimentale de l'expérience n°2	132

TABLE DES FIGURES

3.9	Les diagrammes de Gantt combinés des simulations de l’ordonnancement initial de EX62 (Kader Sanogo <i>et al.</i> , 2023) [15]	133
3.10	Les diagrammes de Gantt combinés des simulations de l’ordonnancement alternatif de EX62 (Kader Sanogo <i>et al.</i> , 2023) [15]	133
4.1	Illustration d’un AIV (produit par MG Tech [16])	146
4.2	Illustration d’une interaction humain-robot	150
4.3	La chaîne de Markov des techniciens	152
4.4	La chaîne de Markov des superviseurs	152
4.5	La chaîne de Markov du personnel administratif	153
4.6	Mise à jour du modèle de simulation	154
4.7	Évolution du makespan moyen en fonction du nombre de robots.	156
4.8	Évolution du gain moyen sur le makespan après augmentation du nombre de robot	158
4.9	Évolution du gain moyen sur le makespan par layout.	160
4.10	Diagrammes de Gantt combinés de EX82 (Kader Sanogo <i>et al.</i> , 2024) [17]	161
4.11	Diagrammes de Gantt combinés de EX51 dans les cas AIV-SH et S-AIV avec 12 humains (Kader Sanogo <i>et al.</i> , 2024) [17]	165
4.12	Évolution de l’augmentation moyenne du makespan en fonction du nombre d’humains.	166
4.13	Comparaison du nombre d’humains rencontrés sur chaque layout	167
4.14	Évolution du nombre de rencontres par type de travailleurs	170
5.1	Logigramme de l’algorithme d’ordonnancement	195
5.2	Logigramme de l’algorithme d’ordonnancement dynamique par entropie (Xu Yiyi <i>et al.</i>) [18]	198
5.3	Représentation du gap moyen entre l’algorithme du VNS [14] et celui de la TDJ.	201
5.4	Représentation du gap moyen entre l’algorithme basé sur l’entropie [18] et celui de la TDJ.	202

TABLE DES FIGURES

5.5	Évolution du makespan moyen en fonction de l’algorithme d’ordonnancement et du nombre d’AIV : en l’absence d’humains.	203
5.6	Comparaison des makespans moyens avec ou sans présence humaine dans l’atelier . . .	208
5.7	Représentation du gap moyen entre notre algorithme et celui du VNS [14].	209
5.8	Représentation du gap moyen entre notre algorithme et celui basé sur l’entropie [18]. .	210
5.9	Représentation du gain moyen d’efficacité après augmentation du nombre de robot . .	211
5.10	Évolution du nombre moyen d’interactions humain-robot en fonction de l’algorithme d’ordonnancement et du nombre d’AIV.	212

Introduction

L'évolution de l'industrie, rythmée par des révolutions successives, a profondément transformé les systèmes socio-techniques au cours des siècles. Aujourd'hui, à l'ère de la cinquième révolution industrielle, nous nous trouvons à un nouveau tournant, où la collaboration homme-robot se place au centre des préoccupations. Ce paradigme émergent redéfinit les contours de la production industrielle, en intégrant l'humain dans des systèmes hautement automatisés et en proposant une nouvelle vision de l'industrie centrée sur l'humain.

Les révolutions industrielles précédentes, de la mécanisation à l'automatisation grâce à la convergence des nouvelles technologies (robotique, intelligence artificielle, big data, systèmes cyber-physiques, etc.), ont été marquées par des avancées technologiques qui ont transformé les modes de production. Aujourd'hui, avec l'Industrie 5.0 (I5.0), il s'agit non seulement d'intégrer des technologies avancées telles que l'intelligence artificielle et les systèmes cyber-physiques, mais aussi de revaloriser le rôle de l'humain dans le processus de production. Cependant, cette transition vers une industrie plus inclusive pose des défis considérables, notamment en matière d'ordonnancement des tâches dans des environnements où cohabitent humains et robots.

L'optimisation de l'ordonnancement dans un contexte industriel mixte, combinant travailleurs humains et robots mobiles de transport, se révèle être un enjeu majeur. Les méthodes classiques d'ordonnancement montrent leurs limites face à la complexité et à la dynamique introduites par les interactions homme-robot. L'intégration des véhicules autonomes intelligents (AIV), plus flexibles et adaptatifs que les véhicules autonomes guidés (AGV), apporte une nouvelle dimension à ces problématiques, nécessitant des approches plus sophistiquées pour garantir une production fluide et efficace.

Dans cette thèse, nous nous intéressons à la problématique de l'ordonnancement des tâches de transport et de production dans des ateliers où interagissent humains et robots. Nous proposons d'ex-

plorer l'impact de la présence humaine sur les performances des systèmes de transport robotisés, en mettant en lumière les défis liés à la gestion de la flotte de robots, tels que l'évitement des collisions et la résolution des blocages (deadlocks), et à l'optimisation du temps de production (makespan). Pour ce faire, nous développerons des modèles de simulation qui intègrent les interactions complexes entre humains et robots, en s'appuyant sur des techniques avancées telles que les systèmes multi-agents, les chaînes de Markov et la théorie des jeux. L'objectif ultime de ce travail est de proposer des solutions d'ordonnancement dynamiques, capables de s'adapter en temps réel aux variations de l'environnement de production, tout en maximisant l'efficacité des processus industriels dans le contexte de l'I5.0. Cette recherche se situe à l'intersection de plusieurs disciplines, notamment l'informatique, la recherche opérationnelle, la robotique et le génie industriel, et contribue à la conception de systèmes de production plus résilients et collaboratifs.

Notre démarche repose sur une approche de simulation-optimisation, dont l'intérêt réside dans sa capacité à combiner les avantages de la simulation, qui permet de reproduire de manière réaliste les dynamiques complexes des systèmes industriels, avec ceux de l'optimisation, qui vise à identifier les solutions les plus efficaces dans un espace de recherche souvent vaste et complexe. En simulant les scénarios potentiels d'interaction entre les robots et les travailleurs, cette méthode offre la possibilité de tester et d'améliorer les ordonnancements dans un environnement contrôlé, avant de les déployer dans la réalité. Cela permet non seulement de minimiser les risques d'erreurs et d'inefficacités, mais aussi d'anticiper les problèmes potentiels et de les résoudre proactivement. De plus, l'intégration d'une composante d'optimisation dynamique permet d'ajuster en temps réel les plans de production en fonction des aléas et des imprévus, garantissant ainsi une meilleure réactivité et adaptabilité des systèmes de production.

Ce manuscrit de thèse est organisée en cinq (05) chapitres, chacun apportant une contribution spécifique à la résolution de la problématique de recherche :

- le premier chapitre (*cf.* chapitre 1) pose les bases théoriques et présente l'évolution des systèmes de production robotisés, tout en exposant la problématique centrale de la thèse ;
- le deuxième chapitre (*cf.* chapitre 2) se consacre à la simulation, en mettant en lumière son utilité dans la modélisation des systèmes industriels complexes. Il détaille également la réalisation d'un simulateur spécifique, conçu pour l'étude du problème d'ordonnancement des tâches de transport et de production, dans un atelier de type « job shop » ;

- le troisième chapitre (*cf.* chapitre 3) explore le problème d’ordonnancement avec la prise en compte des contraintes opérationnelles, telles que l’évitement des collisions entre robots de transport ;
- le quatrième chapitre (*cf.* chapitre 4) s’intéresse à l’intégration des AIV et analyse l’impact de la présence humaine sur l’ordonnancement des tâches ;
- enfin, le cinquième chapitre (*cf.* chapitre 5) propose le développement d’un algorithme d’ordonnancement dynamique basé sur la théorie des jeux, conçu pour intégrer les interactions humain-robot, et évalue ses performances à travers diverses simulations et études de cas.

La conclusion générale, qui passera en revue les contributions cette thèse ainsi que les perspectives des travaux futures, clôturera ce manuscrit.

Chapitre 1

Contexte et problématique de la thèse

Contenu

1.1	Introduction	30
1.2	Évolution des systèmes de production	32
1.2.1	Systèmes Manufacturiers Dédiés (SMD)	33
1.2.2	Systèmes Manufacturiers Flexibles (SMF)	34
1.2.3	Systèmes Manufacturiers Cellulaires (SMC)	34
1.2.4	Systèmes Manufacturiers Reconfigurables (SMR)	35
1.2.5	Systèmes Manufacturiers Intelligents (SMI)	36
1.3	L'avènement de l'Industrie 4.0 (I4.0)	38
1.3.1	L'origine du concept	38
1.3.2	Les composantes de l'I4.0	39
1.3.3	Quelques technologies majeures de l'I4.0	39
1.4	Les limites de l'I4.0	41
1.5	Transition de l'industrie 4.0 vers l'industrie 5.0	43
1.6	Industrie 5.0 (I5.0)	44
1.6.1	Quelques définitions de l'I5.0	45
1.6.2	L'impact de l'I5.0 sur les systèmes de production	46
1.7	Ordonnancement des tâches d'un système de production	48
1.7.1	Flow shop	49
1.7.2	Job shop	50
1.7.3	Open shop	50
1.7.4	Les tâches de transport dans les systèmes de production	51
1.8	Problématique de la thèse	53
1.8.1	Contexte scientifique	54
1.8.2	Problématique de recherche	54
1.8.3	Verrous scientifiques et techniques	55
1.8.4	Analyse de l'état de l'art et positionnement scientifique	56
1.8.5	Les contributions de la thèse	72
1.9	Conclusion	73

1.1 Introduction

L'histoire de l'humanité est rythmée par des périodes de profonds bouleversements, tant sociétaux qu'économiques, qui ont métamorphosés les systèmes socio-techniques [19] existants. Parmi ces transformations majeures, les révolutions industrielles occupent une place centrale, marquant des tournants décisifs dans notre évolution sociétal et technologique. De la fin du XVIII^e siècle jusqu'à aujourd'hui, ces révolutions successives n'ont pas manqué de façonner notre monde, conduisant ainsi à l'émergence de la société moderne. La figure 1.1 illustre cette évolution jusqu'à aujourd'hui.

Prenant son essor en Angleterre dans les années 1780, la première révolution industrielle a été caractérisée par l'utilisation de la machine à vapeur et l'adoption de la mécanisation dans les processus de production. Ce tournant a entraîné la naissance des usines et l'essor de l'industrie textile, bouleversant les modes de production traditionnels. Puis, s'en est suivi la deuxième révolution industrielle, autour des années 1870, marquée par la découverte de l'électricité et du pétrole comme nouvelles sources d'énergie, ouvrant la voie à la production de masse et à l'automatisation. L'émergence de l'électronique et de l'informatique dans les années 1950-1970, marque le début de la troisième révolution industrielle, introduisant une nouvelle ère de traitement et de diffusion de l'information. L'invention du transistor, des ordinateurs et de l'internet ouvrent la voie à l'automatisation des processus industriels et à l'essor de la robotique. Dès les années 2010, l'explosion des technologies de l'information et de la communication (TIC) entraîne le monde dans la quatrième révolution industrielle (aussi connue sous l'appellation de « Industrie 4.0 »), révolutionnant tous les aspects de la société, y compris l'industrie. L'intégration des technologies numériques, telles que l'internet des objets (IoT) et l'intelligence artificielle (IA), a révolutionné les chaînes de valeur industrielles, permettant une optimisation et une personnalisation accrues des produits et services. Les usines modernes intègrent largement des systèmes connectés, permettant aux machines et autres équipements de communiquer entre eux de manière fluide. Cette interconnexion donne naissance à une industrie autonome, capable de fonctionner sans supervision humaine directe. L'Industrie 4.0 (I4.0) s'appuie sur les systèmes cyber-physiques (CPS), qui fusionnent le monde physique et numérique pour piloter et optimiser les processus de production en toute autonomie.

L'intervention humaine devient de plus en plus superflue dans ces usines de production hautement sophistiquées, dotées de machines et de robots performants et précis [20]. La créativité humaine est

1.1. INTRODUCTION

écartée de la prise de décision quotidienne au profit de l'efficacité technologique, faisant ainsi courir le risque de perdre les compétences et le savoir-faire humain qui sont progressivement substitués par machines spécialisées automatisées. Face à ce constat, les pouvoirs publics et les chercheurs ont travaillé à l'élaboration d'un nouveau paradigme, une nouvelle révolution qui mettrait l'humain au coeur du système et favoriserait la collaboration humain-robot, permettant ainsi de tirer profit des qualités de chacun. C'est ainsi que, depuis 2020, la cinquième révolution industrielle ou l'« Industrie 5.0 » a vu le jour. Par ailleurs, même si l'humain est amené à jouer un rôle majeur dans l'industrie, il existe néanmoins des tâches qu'il est pertinent de robotiser car elles sont pénibles et répétitives, où l'humain n'apporte presque aucune valeur ajoutée. Il s'agit notamment des tâches de transport au sein de l'atelier qui obéissent à des ordonnancements qu'il convient d'optimiser pour garantir une meilleure productivité de l'industrie. Toutefois, une question subsiste : comment optimiser l'ordonnancement des tâches de transport réalisées par les robots en tenant compte de la présence humaine dans l'industrie ? Cette question centrale est au coeur des travaux de cette thèse.

Ainsi, ce chapitre se consacre à la présentation du contexte et à la description de la problématique des travaux de recherche. Il commence par rappeler l'évolution des systèmes de production robotisés tout en abordant les problèmes d'ordonnancement des tâches d'un atelier de production. Enfin, il expose la problématique de recherche et présente les contributions de la thèse.

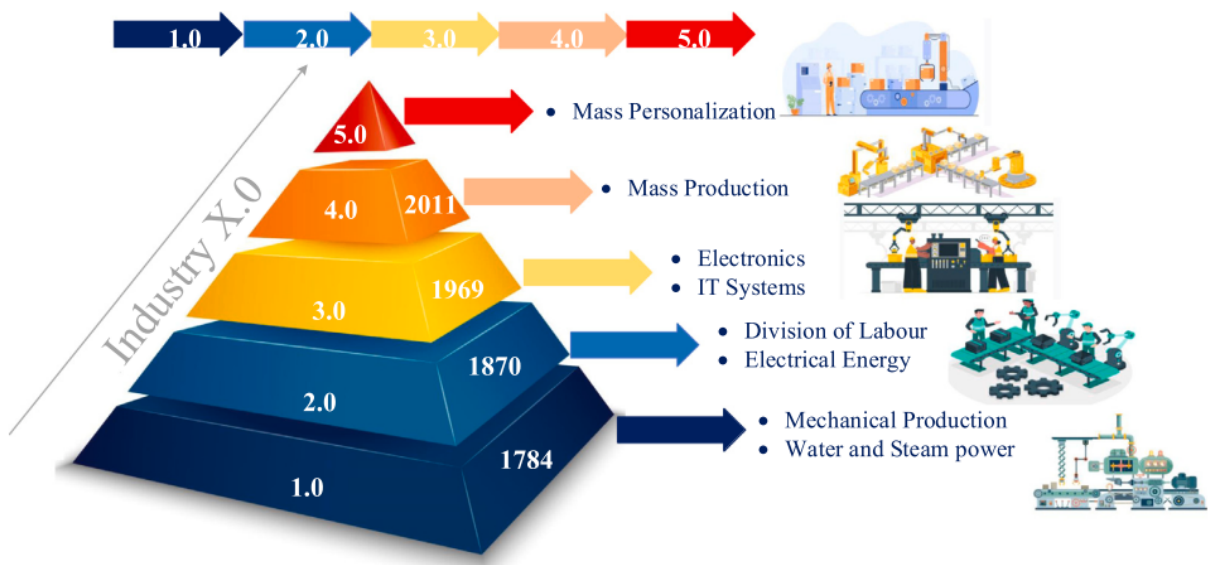


FIGURE 1.1 – Illustration de l'évolution industrielle (Praveen K. M. R. *et al.*, 2022) [2]

1.2 Évolution des systèmes de production

Un système de production désigne l'ensemble des acteurs (matériels et immatériels) et des ressources nécessaires à la transformation de matières premières en produits finis [21]. Il s'agit d'une structure complexe qui fait interagir le personnel, le matériel (stations de travail, machines et robots), les logiciels et les procédures pour répondre à un besoin [3]. Comme proposé par Imad Chalfoun [3], un système de production peut être décomposé en trois sous-systèmes comme le montre la figure 1.2 :

- le système d'information : c'est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information dans un but précis. Il assure le traitement et la diffusion de l'information aux autres sous-systèmes ;
- le système de décision : il prend en compte les informations transmises en amont pour commander le système physique ;
- le système physique : c'est l'ensemble des ressources matérielles et humaines nécessaires pour la transformation des matières premières en produits finis.

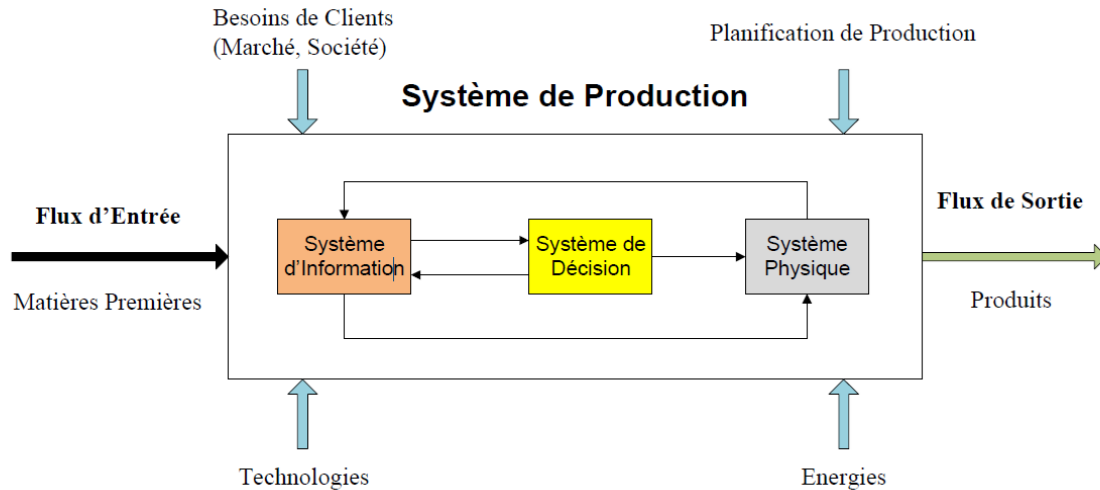


FIGURE 1.2 – Décomposition d'un système de production (Chalfoun, 2014) [3]

Ces trois (03) sous-systèmes doivent donc être efficacement et intelligemment intégrés afin d'améliorer les performances globales du système de production et d'accroître sa productivité. Aussi, dans un marché en perpétuelle mutation, le système de production doit être capable de réagir et de s'adapter aux exigences du marché. Si au début de l'ère industrielle l'offre était assurée par quelques acteurs industriels, la demande croissante des consommateurs et la multiplication des acteurs industriels ont

1.2. ÉVOLUTION DES SYSTÈMES DE PRODUCTION

conduit à une refonte du système de production afin de s'aligner sur les besoins du marché. Ainsi, à la fin du XVIII^e siècle, on avait des systèmes de production conçus pour la production en masse d'une seule référence de produit car l'offre industrielle était peu variée et les acteurs faisaient face à peu de concurrents [22]. Cette configuration du systèmes de production est connue sous l'appellation de *Système Manufacturiers Dédiés (SMD)* [23]. Puis, dès le XX^e siècle, avec l'évolution du marché, les *Systèmes Manufacturiers Flexibles (SMF)* ont vu le jour, permettant aux systèmes de production de produire désormais une variété de produits, dont le volume et le type peuvent être modifiés, sur les mêmes machines [24, 23]. Même si les SMF permettent la production de différents produits, grâce aux machines qui peuvent réaliser plusieurs tâches, il reste cependant difficile de modifier la configuration du système qui reste fixe. On peut difficilement rajouter de nouvelles machines ou changer la disposition des machines existantes [3]. Pour surmonter ces inconvénients, on a adopté des *Systèmes Manufacturiers Cellulaires (SMC)* qui consistent à décomposer le système de production en plusieurs cellules indépendantes qui sont dédiées à la fabrication de familles de produits ayant des exigences de traitement similaires [25, 22]. Cependant, les SMC montrent vite leurs limites face au passage d'une production de masse vers une personnalisation de masse [22], d'où l'émergence des *Systèmes Manufacturiers Reconfigurables (SMR)* dans les années 1990, permettant une agilité de restructuration du systèmes de production. Depuis quelques années, grâce notamment à la révolution numérique, les machines et outils de fabrication sont interconnectés, mais aussi et surtout connectés à internet, afin d'améliorer les performances de fabrication et d'optimiser la consommation de ressources. On entre ainsi dans l'ère de l'« usine intelligente » grâce aux *Systèmes Manufacturiers Intelligents (SMI)* [26].

La place et le rôle de l'humain ont eux aussi changé tout au long de l'évolution des systèmes de production. Si sa présence semblait triviale pour le bon fonctionnement des industries, l'avènement des SMI vient remettre en question sa place dans les systèmes de production[27, 28].

1.2.1 Systèmes Manufacturiers Dédiés (SMD)

Les SMD, également connues sous le nom de lignes de transfert, ont fait leur apparition depuis les années 1910 [29]. Ces lignes se composent d'un ensemble de machines relativement simples disposées en séquence le long d'une ligne comme sur la figure 1.3 [3]. Chaque machine est équipée d'un système d'automatisation fixe, conçu pour la production de produits spécifiques à haut volume. Chaque machine effectue une opération d'usinage spécifique, toujours au même endroit (perçage, fraisage, etc.).

1.2. ÉVOLUTION DES SYSTÈMES DE PRODUCTION

Cependant, leur principal inconvénient réside dans leur rigidité : elles ne peuvent pas être facilement modifiées pour produire de nouveaux produits [30]. Les humains ont joué un rôle majeur dans les SMD, en ce sens qu'ils étaient à la fois force de travail et preneurs de décision. Les machines qui n'étaient douées qu'à répéter les mêmes tâches indéfiniment avaient absolument besoin des humains pour assurer leur maintenance mais aussi pour garantir la transformation des pièces usinées en produits finis [31].

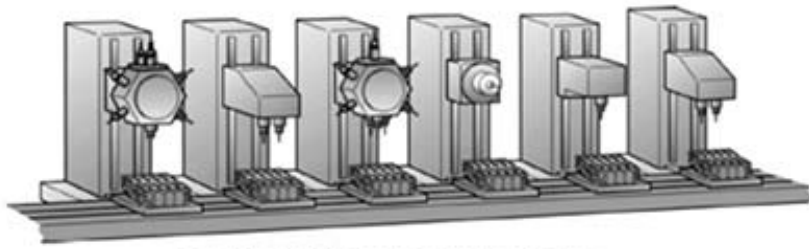


FIGURE 1.3 – Illustration d'un SMD (Chalfoun, 2014) [3]

1.2.2 Systèmes Manufacturiers Flexibles (SMF)

Les SMF, apparus dans les années 1970 [29], sont des systèmes de production capables de fabriquer une grande variété de produits, grâce notamment à des machines à commandes numériques programmables pouvant réaliser différentes tâches. Un SMF (voir figure 1.4) peut être défini comme « *un complexe intégré de dispositifs de manutention automatisés et de machines-outils à commande numérique (NC), contrôlé par ordinateur, qui peut traiter simultanément des volumes de taille moyenne de divers types de pièces* » [32]. Cette nouvelle avancée permet d'optimiser l'ensemble du processus de fabrication des produits sans assistance humaine [31]. Des robots sont désormais capables d'exécuter des séquences de traitement sans la moindre intervention humaine et plusieurs opérations traditionnellement effectuées par des humains, notamment la planification et le contrôle, sont réalisées par des moyens électroniques et informatiques [31].

1.2.3 Systèmes Manufacturiers Cellulaires (SMC)

Un SMC est une application de la technologie de groupe [33] qui consiste à décomposer le système de production en cellules (processus de fabrication) afin de réaliser la production d'un ensemble de produits nécessitant des traitements similaires (familles de produits) sur une cellule dédiée [25]. Cette subdivision en cellules, illustrée sur la figure 1.5, permet de rendre le système plus flexible tout en

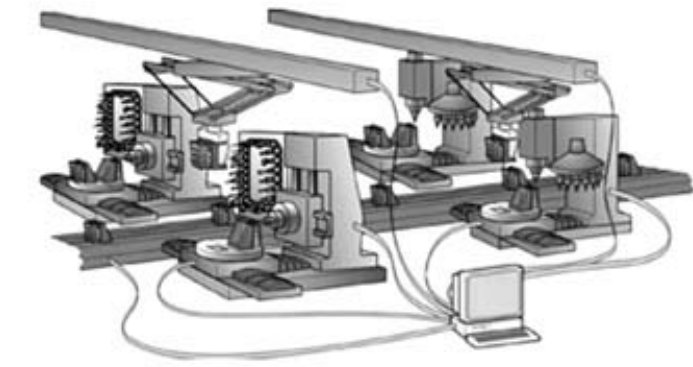


FIGURE 1.4 – Illustration d'un SMF (Chalfoun, 2014) [3]

augmentant les capacités de production [4]. Selon Wemmerlov et Johnson [34, 35] les SMC permettent d'améliorer la qualité tout en réduisant les coûts de manutention, la charge de travail, le temps de préparation, le délai d'exécution et le temps de production.

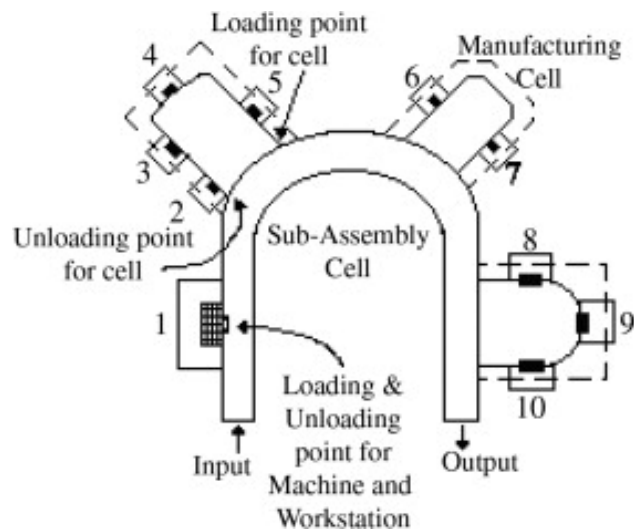


FIGURE 1.5 – Illustration d'un SMC (S. Ariaifar et N. Ismail, 2009) [4]

1.2.4 Systèmes Manufacturiers Reconfigurables (SMR)

Les SMR sont des « systèmes conçus dès le départ pour permettre un changement rapide de la structure, ainsi que des composants matériels et logiciels, afin d'adapter rapidement la capacité de production et les fonctionnalités au sein d'une famille de produits, en réponse aux changements brusques du marché ou à des exigences réglementaires » [23, 36]. Comme illustrée sur la figure 1.6 [3], Un SMR

1.2. ÉVOLUTION DES SYSTÈMES DE PRODUCTION

se distingue par sa capacité à ajuster rapidement la production en fonction des besoins du marché. Cela se traduit par la possibilité d'ajouter de nouvelles machines ou de modifier celles déjà en place dans les différentes étapes de production. Ces machines modulaires sont conçues pour une grande flexibilité et peuvent être facilement déplacées ou reconfigurées [22, 3]. Les SMR impliquent une évolution du rôle des opérateurs humains. Ils passent ainsi d'un rôle principalement axé sur l'exécution de tâches manuelles à un rôle plus centré sur la supervision et la programmation des machines intelligentes, la prise de décision stratégique et la résolution de problèmes [37, 38]. On peut donc résumer le rôle des humains par le délaissement des tâches physiques, qui sont automatisées et exécutées par des machines et robots intelligents, au profit des activités cognitives et de prise de décision.

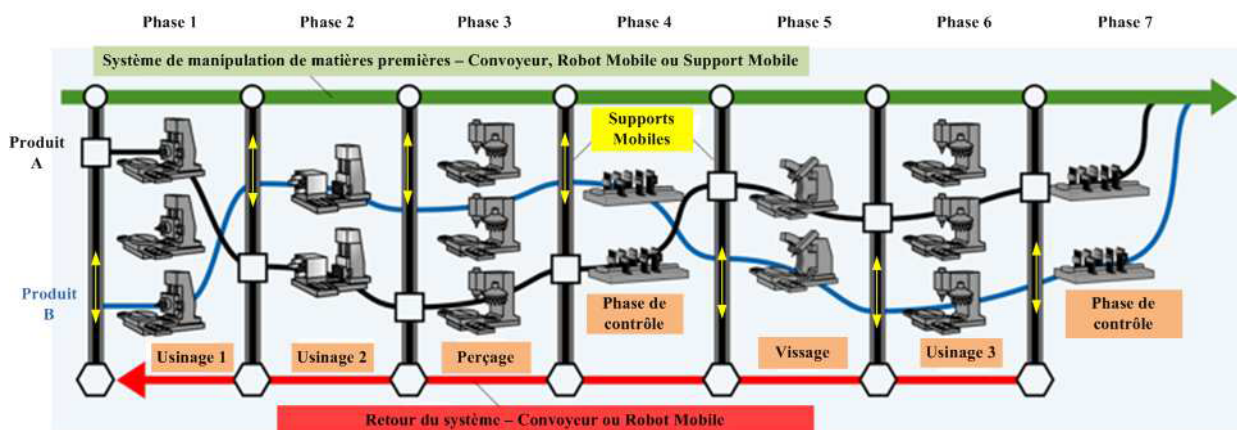


FIGURE 1.6 – Illustration d'un SMR (Chalfoun, 2014) [3]

1.2.5 Systèmes Manufacturiers Intelligents (SMI)

Un SMI, comme illustré sur la figure 1.7 [5], est un système de production intégré qui utilise des technologies avancées pour collecter et analyser des données en temps réel, optimiser les processus de production, prendre des décisions autonomes et s'adapter aux changements de manière flexible afin d'améliorer l'efficacité, la productivité, la qualité et la satisfaction des consommateurs [26, 39]. C'est un système de production essentiellement dominé par l'introduction des nouvelles technologies du numérique telles que l'IoT et les capteurs, le cloud computing, l'analyse de données et l'IA [39]. Ces technologies rendent le système plus intelligent et capable de relever les défis actuels, tels que les exigences de personnalisation croissantes, l'amélioration de la qualité et la réduction des délais de commercialisation [40]. Les capteurs sont utilisés en masse, notamment sur les machines-outils et les

1.2. ÉVOLUTION DES SYSTÈMES DE PRODUCTION

robots, pour permettre de se détecter entre eux, d'interagir et de communiquer [41]. Ces technologies permettent d'obtenir et de partager les données de production en temps réel afin de faciliter une prise de décision rapide et précise. La connexion à internet des équipements physiques de production, associée aux outils d'analyse de données massives (big data), a conduit à l'émergence d'un moyen de production révolutionnaire : les systèmes de production cyber-physiques (CPPS) [39]. Les SMI utilisent ainsi le concept général de CPS dans le cadre de la production manufacturière [39, 26], en permettant aux éléments du système de production de collecter des informations de leurs entourages et d'agir de manière autonome, de collaborer avec les autres éléments (y compris les humains) dans l'exécution des tâches et d'être réactifs face aux changements internes et externes au système [26, 42, 43]. De plus, si les SMR avaient restreints le rôle des humains aux activités cognitives et à la prise de décision, les SMI franchissent un nouveau cap en permettant au système de prendre des décisions de manière autonome, ce qui limite davantage les interventions humaines. En outre, selon plusieurs auteurs [44, 45, 46] les SMI sont à l'origine de l'I4.0.

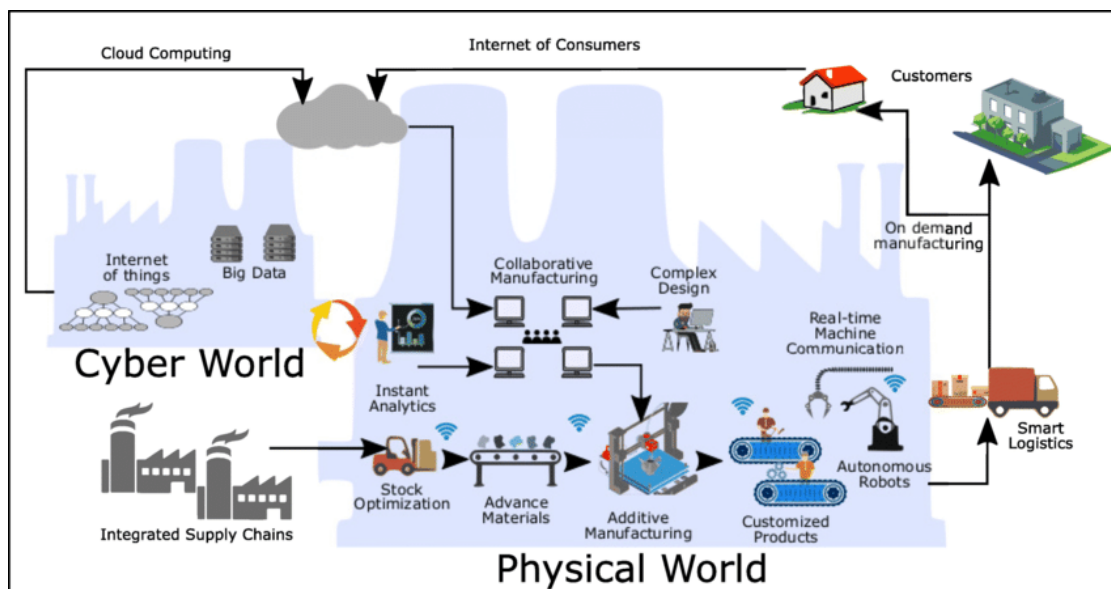


FIGURE 1.7 – Illustration d'un SMI (David Balderas *et al.*, 2021) [5]

1.3 L'avènement de l'Industrie 4.0 (I4.0)

1.3.1 L'origine du concept

L'I4.0 est un processus controversé par nature et par définition, étant donné les technologies porteuses qui permettent son existence ainsi que les opportunités qu'elle apporte [47]. L'expression « quatrième révolution industrielle » a été introduite pour la première fois en 1988 pour identifier les processus d'évolution des inventions en innovations grâce à la présence de scientifiques dans les équipes de production [48, 47]. Ce n'est qu'en 2011 que la quatrième révolution industrielle a été baptisé « Industrie 4.0 » d'après le plan industriel allemand du même nom [49], et évoqué pour la première fois à la foire de Hanovre [22]. L'I4.0 est aussi connu sous les termes de « industrie du futur », « l'usine intelligente » ou « l'usine connectée » [47]. Bien qu'aucune définition conceptuelle, opérationnelle ou universellement acceptée de l'I4.0 n'a été identifiée jusqu'à présent [47, 39], on peut la définir comme une industrie résolument tournée vers l'innovation technologique, la convergence numérique (IoT, IA, big data, etc.) et les CPPS pour assurer l'autonomisation, l'intelligence et l'efficacité du système de production. À cet effet, l'I4.0 fait intervenir plusieurs aspects conceptuels et technologiques qu'il convient de décrire pour une meilleure compréhension.

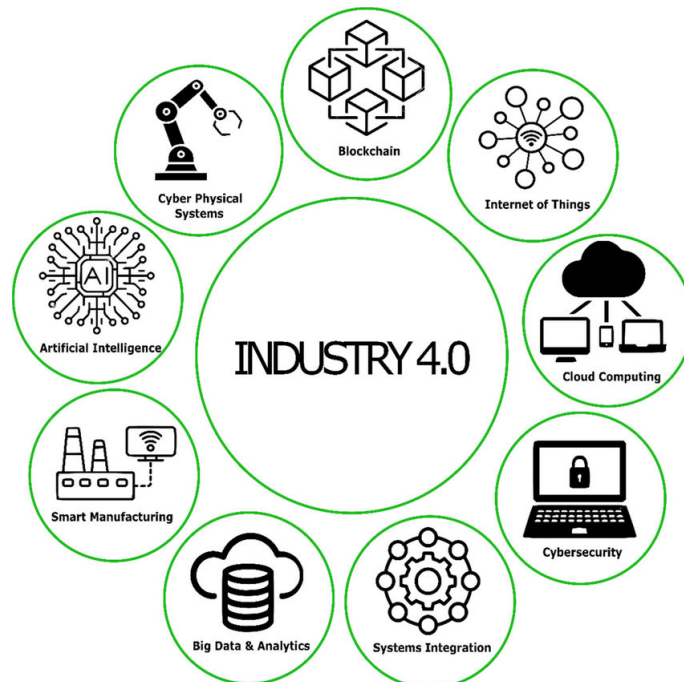


FIGURE 1.8 – Les neuf (09) piliers de l'I4.0 selon Gomez C. *et al.* [6]

1.3.2 Les composantes de l'I4.0

La réalisation effective de l'I4.0 dépend de deux facteurs fondamentaux qui sont *l'intégration* et *l'interopérabilité* [50, 51]. L'intégration des systèmes d'automatisation industrielle, à travers les CPPS [52, 53] par exemple, se traduit par des caractéristiques plus importantes et plus innovantes grâce à la mise en réseau des parties prenantes, à la fois horizontalement et verticalement. Elle contribue également à créer des liens entre les mondes cybernétique et physique. En outre, l'interopérabilité, grâce notamment à la notion de « fabrication intelligente » [22, 39, 26], facilite les processus de production, même sans continuité, à l'intérieur et au-delà des frontières d'une entreprise, afin d'interconnecter les systèmes et d'échanger les connaissances et les compétences [47].

1.3.3 Quelques technologies majeures de l'I4.0

D'après Shyam M. *et al.* [7], l'I4.0 compte huit (08) technologies majeures telles que représentées sur la figure 1.9. Cependant, nous n'en présenterons que celles ayant un lien direct avec le travail effectué.

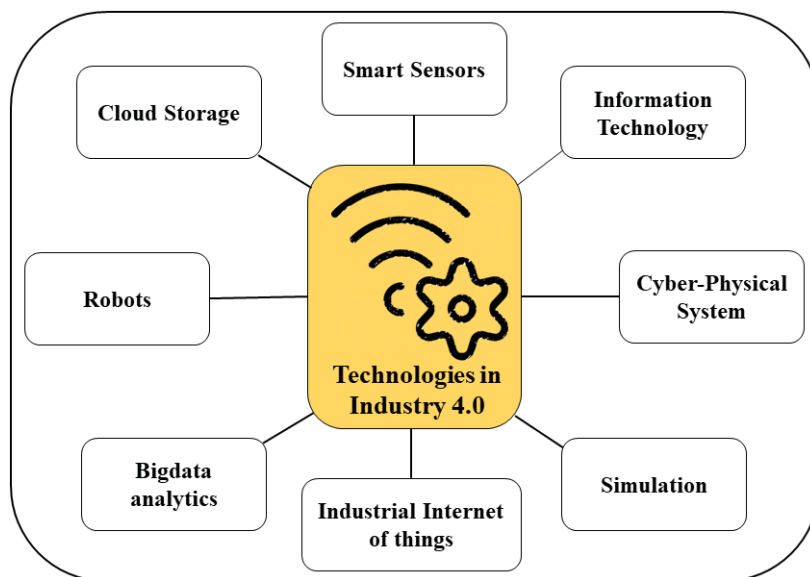


FIGURE 1.9 – Les principales technologies de l'I4.0 (Shyam M. *et al.*, 2021) [7]

1.3.3.1 Les capteurs intelligents

Un capteur intelligent, aussi connu sous le nom de capteur connecté, est un dispositif électronique qui permet la collecte de données physiques, leur traitement, ainsi que leur transmission afin d'aider à la prise de décision. C'est une source d'information utilisée pour mesurer des qualités physiques et techniques particulières qui sont converties en un signal pouvant être transmis à distance et traité ultérieurement [7]. Ils permettent de fournir des informations sur le mouvement, la position, la vitesse, la distance, les vibrations, les gaz, la force, la lumière, etc. Les capteurs jouent un rôle très important dans les usines intelligentes, car ils permettent notamment la communication de machine à machine (M2M) et le contrôle à distance des processus industriels.

1.3.3.2 Les robots

L'automatisation des méthodes de fabrication avec la création des robots industriels est une réalité depuis la troisième révolution industrielle. Le numérique joue un rôle crucial dans cette automatisation en intégrant les robots industriels aux opérations et aux systèmes de traitement de l'information [54, 55]. On distingue deux (02) catégories de robots industriels dans l'I4.0 [7] :

- Les robots d'assistance : ils facilitent le fonctionnement des machines et des processus industriels en réduisant le besoin de main-d'œuvre humaine, tout en étant précis et efficaces.
- Les robots apprenants : ils sont capables d'apprendre des humains et de leurs erreurs passées pour exécuter et maintenir leur processus [56].

Les robots sont utilisés pour la manipulation d'objets, le traitement d'opérations ou encore le transports d'objets [57] dans les usines intelligentes. Ils peuvent exécuter toutes ces différentes tâches de manière autonome et intelligente, sans nécessiter la présence humaine [56].

1.3.3.3 La simulation

Comme défini par J. Banks *et al.* [58], « *La simulation est l'imitation du fonctionnement d'un processus ou d'un système réel dans le temps. Elle implique la création d'un historique artificiel du système et l'observation de cet historique artificiel pour tirer des conclusions sur les caractéristiques de fonctionnement du système réel représenté.* » La simulation est un outil flexible et ajustable qui permet de tester différentes configurations et de solutionner des problèmes industriels sans toucher au système physique. Cette caractéristique permet d'éviter les dysfonctionnements dûs aux mauvais

choix et de limiter les gaspillages de ressources [7]. La simulation joue un rôle important dans l'I4.0, en aidant à optimiser les processus, à tester de nouvelles technologies et anticiper les défis dans un environnement virtuel avant leur mise en œuvre réelle [59, 7]. Elle contribue ainsi à l'amélioration de la prise de décision et à la stimulation de l'innovation.

1.3.3.4 Les données

L'I4.0 est caractérisée par l'omniprésence des données [60], et leur exploitation intelligente les place au cœur de la transformation industrielle [61]. Elles deviennent un atout stratégique surtout pour la prise de décisions, mais aussi pour optimiser les processus et pousser l'innovation encore plus loin. Les capteurs installés sur les machines, les robots, les produits et les infrastructures collectent des données en temps réel sur les processus industriels, la performance et l'environnement de production [62, 63]. De même, les données issues des systèmes de production, de la chaîne d'approvisionnement (*supply chain*, en anglais), des clients et des partenaires externes sont intégrées pour obtenir une vue globale des opérations [63]. Les techniques avancées d'analyse de données massives sont utilisées pour traiter et analyser de grands volumes de données hétérogènes afin d'en extraire des informations exploitables. Cette analyse peut être également automatisée et approfondie grâce à l'IA afin de détecter des anomalies, de faire des prédictions et de prendre des décisions intelligentes [63, 61].

1.4 Les limites de l'I4.0

Bien que prometteuse en termes d'innovation, de flexibilité et d'optimisation, l'I4.0 présente quelques limites [64, 65, 66] qu'il convient de considérer pour une transition industrielle responsable et inclusive.

D'abord, l'I4.0 n'a pas de définition consensuelle ou universellement acceptée, ce qui pose un problème au niveau de la standardisation [67, 68] et de la normalisation [66, 67] internationale, ainsi que de la comparabilité des travaux scientifiques [69]. En effet, selon Giovanna Culot *et al.* [69] de nombreux chercheurs font face à l'absence d'une définition commune de l'I4.0, ce qui limite considérablement l'élaboration de théories et la comparabilité des recherches. De son avènement en 2011 jusqu'à aujourd'hui, le paysage technologique et la compréhension de l'I4.0 ont évolué de manière significative, ce qui a conduit à plusieurs ambiguïtés.

Ensuite, l'I4.0 a un impact considérable sur la main d'oeuvre humaine, ce qui en fait son principal

talon d'Achille [20, 70]. En effet, selon Gábor S. *et al.* [70], on peut affirmer sans le moindre risque que l'I4.0 entraînera une perturbation sur le marché de l'emploi, étant donné les tendances récentes et les enseignements tirés des précédentes révolutions industrielles. Les solutions d'automatisation sont devenues plus abordables et plus performantes grâce aux récentes avancées technologiques, ce qui a pour conséquence de priver la main-d'œuvre humaine d'un nombre croissant d'emplois [71, 72]. De plus, même si certains emplois subsisteront, cela conduira à une concentration des compétences vers du personnel hautement qualifié [73] et spécialisé en informatique, en robotique, en analyse de données et en IA [74], exacerbant davantage les inégalités et la fracture numérique entre les travailleurs qualifiés et non qualifiés [75, 76]. Toutefois, l'intervention des pouvoirs publics et les changements de politique peuvent contribuer à réduire l'impact du développement technologique [70]. À l'évidence, plusieurs métiers seront impactés [77]. Il faudra donc engager de vastes programmes de formation [77] et de reconversion professionnelles [78] des personnes [79] touchées par la transition numérique engendrée par l'I4.0 [20], ce qui nécessitera un coût financier [80] énorme que les gouvernements, les industries et les individus devront supporter.

Enfin, L'I4.0 reste encore freinée par des questions éthiques, techniques et sociales. En effet, la collecte et le traitement massifs de données personnelles soulèvent des préoccupations en matière de confidentialité et de sécurité [81], nécessitant des cadres juridiques et des pratiques éthiques robustes [82, 83]. De plus, selon Mirela C. T. *et al.* [66], l'utilisation d'algorithmes dans l'évaluation des performances et la prise de décision peut entraîner une surveillance accrue des travailleurs, se traduisant par moins d'intimité [84], et une négligence de leur bien-être. Cela a pour conséquence de créer une dépendance aux robots et de rendre les travailleurs plus apathiques, plus introvertis, et moins heureux au travail [85]. Par ailleurs, Klaus Schwab [86] soutient que les gouvernements et les décideurs politiques doivent s'adapter et réagir rapidement à l'évolution rapide du paysage de l'I4.0 en fournissant l'environnement favorable, les garanties et les politiques qui peuvent orienter l'avenir vers un développement économique et social durable et qui exploitent les promesses que les technologies issues de l'I4.0 recèlent pour les personnes et les sociétés [81]. Ce point de vue est également partagé par Brynjolfsson et McAfee [87] qui soulignent qu'il est important de reconnaître l'impact de l'I4.0 sur toute la société afin de pouvoir tirer profit des opportunités qu'elle offre. Ainsi, ils suggèrent de considérer l'innovation sociale en même temps que la révolution technologique. En d'autres termes, il faudra aligner le développement technique et les nouvelles pratiques requises pour relever les défis

sociaux [81].

1.5 Transition de l'industrie 4.0 vers l'industrie 5.0

Bien que l'I4.0 soit une révolution industrielle récente, affectant à la fois l'industrie et la société dans son ensemble, d'autres initiatives plus ambitieuses ont été proposées depuis sa création [88]. En 2015, l'Organisation des Nations Unies (ONU) a adopté un plan dénommé « Agenda 2030 pour le développement durable », qui définit 17 objectifs de développement durable (ODD) [89] afin de lutter contre la faim, les inégalités sociales et réchauffement climatique entre autres [88]. En l'occurrence, les objectifs 8, 9, 11 et 12 s'intéressent au développement d'un nouveau modèle socio-économique plus durable, inclusif et résilient. L'objectif 9 en particulier vise à bâtir une industrie inclusive et durable qui favorise l'innovation.

Partant de là, en janvier 2016, le gouvernement du Japon a dévoilé le concept de la « société 5.0 » [90, 91] lors de son 5^e *plan de base pour la science et la technologie*. Se basant sur les ODD, la Société 5.0 représente une vision ambitieuse du futur où l'harmonie entre l'homme et la technologie est au cœur du progrès. Elle succède aux sociétés précédentes caractérisées par la chasse et la cueillette (Société 1.0), l'agriculture (Société 2.0), l'industrialisation (Société 3.0) et l'information (Société 4.0) [90]. Cette société définit l'humain comme l'acteur central qui s'épanouit et exploite ses potentiels grâce aux technologies [92, 88].

Par ailleurs, la pandémie mondiale de COVID-19 a causé des bouleversements majeurs dans nos sociétés, y compris pour les industries, conduisant ainsi à une réflexion sur la révision des chaînes de valeur et des pratiques de consommation d'énergie existantes pour rendre les industries plus résistantes aux chocs extérieurs [93]. En effet, les industries doivent mener les transitions numérique et écologique si elles veulent continuer à créer de la richesse. ce qui implique que leurs objectifs doivent aller au-delà de la recherche de l'efficacité et de la productivité pour renforcer leur rôle et leur contribution à la société [93, 94]. De plus, elles doivent placer le bien-être du travailleur au centre du processus de production et utiliser les nouvelles technologies pour assurer la prospérité, tout en limitant leurs impacts sur l'environnement [93]. Cette réflexion a conduit la Commission Européenne à réunir, lors de deux ateliers virtuels tenus les 2 et 9 juillet 2020, des chercheurs, des experts de l'industrie, de la technologie et de la finance pour discuter d'un nouveau paradigme. C'est ainsi que le concept

1.6. INDUSTRIE 5.0 (I5.0)

de l' « Industrie 5.0 » [94] a vu le jour. Ce nouveau paradigme vient compléter l'I4.0 en mettant spécifiquement la recherche et l'innovation au service de la transition vers une industrie durable, centrée sur l'humain et résiliente [93].

1.6 Industrie 5.0 (I5.0)

Née à la suite de la crise liée au COVID-19 qui a mis coup d'arrêt à l'économie mondiale et qui a également suscité de nombreux questionnements dans nos sociétés, l'I5.0 est considérée par plusieurs chercheurs [95, 96, 28] et organisations internationales comme la cinquième révolution industrielle. À la différence de l'I4.0 qui est révolution disruptive, l'I5.0 est une révolution incrémentale [97] qui s'appuie sur les acquis et les pratiques de l'I4.0 [98]. Impulsée par la Commission Européenne [93], l'I5.0 prône une industrie résiliente, durable et centrée sur l'humain, comme illustré par la figure 1.10. Bien qu'il existe plusieurs définitions de l'I5.0, elles convergent toutes vers ces fondements. C'est donc une industrie dans laquelle les humains et les robots (plus généralement, la technologie) sont des collaborateurs, plutôt que des antagonistes.



FIGURE 1.10 – Les fondements de l'I5.0 (Momenta, 2024) [8]

1.6.1 Quelques définitions de l'I5.0

L'I5.0 étant encore un concept assez récent, plusieurs définitions ont été proposées par les chercheurs, les industriels et les pouvoirs publics. Certaines de ces définitions sont présentées ici.

1.6.1.1 Définition n°1 (Michael Rada, 2018)

L'industrie 5.0 est une première évolution industrielle menée par l'homme et basée sur les principes 6R (Reconnaître, Reconsidérer, Réaliser, Réduire, Réutiliser et Recycler) de revalorisation industrielle, une technique systématique de prévention des déchets et une conception de l'efficacité logistique pour valoriser les normes de vie, les créations innovantes et produire des produits personnalisés de haute qualité [99].

1.6.1.2 Définition n°2 (Saeid Nahavandi, 2019)

L'industrie 5.0 ramène la main-d'œuvre humaine dans l'usine, où l'homme et la machine sont associés pour accroître l'efficacité du processus en utilisant la matière grise et la créativité de l'homme grâce à l'intégration des flux de travail avec des systèmes intelligents [28].

1.6.1.3 Définition n°3 (Francesco Longo *et al.*, 2020)

L'industrie 5.0 intègre les forces changeantes des systèmes de production cyber-physiques (CPPS) et l'intelligence humaine pour créer des usines synergiques. Elle vient combler la perte de main-d'œuvre humaine causée par l'industrie 4.0 grâce à une conception innovante, éthique et centrée sur l'homme [100].

1.6.1.4 Définition n°4 (Yoong K. L. *et al.*, 2021)

L'industrie 5.0, une innovation symétrique et la prochaine génération de gouvernance mondiale, est une avancée progressive de l'industrie 4.0 (innovation asymétrique). Elle vise à concevoir des sorties sûres orthogonales en séparant les systèmes d'automatisation hyperconnectés pour la fabrication et la production [101].

1.6.1.5 Définition n°5 (Praveen Kumar *et al.*, 2022)

L'industrie 5.0 est une solution de conception centrée sur l'humain où les cobots (robots collaboratifs) collaborent avec les ressources humaines pour permettre une fabrication autonome personnalisable par le biais des réseaux sociaux d'entreprise. Cela permet à l'homme et à la machine de travailler main dans la main. Les cobots ne sont pas des machines programmables, mais ils peuvent sentir et comprendre la présence humaine. Dans ce contexte, les cobots seront utilisés pour les tâches répétitives et les travaux à forte intensité de main-d'œuvre, tandis que les humains se chargeront de la personnalisation et de la réflexion critique [2].

1.6.1.6 Définition n°6 (Commission Européenne, 2021)

« L'industrie 5.0 reconnaît la capacité de l'industrie à atteindre des objectifs sociétaux allant au-delà de l'emploi et de la croissance, pour qu'elle devienne un pourvoyeur de prospérité résilient, en veillant à ce que la production respecte les limites de notre planète et en plaçant le bien-être du travailleur industriel au centre du processus de production. » [94]

1.6.2 L'impact de l'I5.0 sur les systèmes de production

Héritant de la révolution technologique introduite par l'I4.0 [102], l'I5.0 se veut être un catalyseur majeur de la personnalisation de masse [2], ce qui permet aux clients de privilégier des produits personnalisés et adaptés à leurs goûts et à leurs besoins. En outre, La collaboration entre les humains et les machines vise à augmenter la production à un rythme rapide [103, 2]. L'I5.0 peut améliorer la qualité de la production en confiant les tâches répétitives, monotones et pénibles aux robots/machines, et réservant les tâches nécessitant la créativité et la réflexion critique aux humains [2, 104]. Enfin, L'I5.0 favorise la création d'emplois plus qualifiés que l'I4.0, car les professionnels intellectuels travaillent au quotidien avec les nouvelles technologies et aux cotés des robots [20, 2]. À ce sujet, le passage de l'I4.0 vers l'I5.0 va nécessairement induire, selon les degrés de proximité, une interaction entre les opérateurs humains et les multiples robots présents dans les ateliers de production. Cette interaction a été étudiée par de nombreux chercheurs [9] qui l'ont définie et catégorisée.

En effet, selon J. Lindblom et B. Alenljung [105], l'interaction homme-robot (*Human-Robot Interaction - HRI*, en anglais) est la science qui étudie le comportement et les attitudes des personnes

à l'égard des robots en relation avec les caractéristiques physiques, technologiques et interactives des robots, dans le but de développer des robots qui facilitent l'émergence d'interactions homme-robot qui sont à la fois efficaces et acceptables pour les personnes, et qui répondent aux besoins sociaux et émotionnels de leurs utilisateurs individuels tout en respectant les valeurs humaines. Il existe trois (03) types d'interactions homme-robot selon le degré de proximité entre les hommes et les robots [106, 9] :

- **la coexistence** : La coexistence est un scénario dans lequel l'homme et le robot ne partagent pas un espace de travail commun, comme c'est le cas dans l'automatisation conventionnelle. Compte tenu du fait qu'aucune séparation physique n'empêche les contacts accidentels entre l'homme et le robot, le robot est équipé d'un mécanisme de sécurité qui lui permet d'arrêter de bouger dès qu'il détecte un humain dans son espace de travail [107]. Le contact physique n'est alors possible qu'à l'arrêt des opérations [108]. Cette interaction est donc limitée à l'évitement des collisions.
- **la coopération** : ce scénario permet aux humains et aux robots de partager un espace de travail commun. De plus, la coopération se caractérise par des étapes de travail séquentielles et ne nécessite pas nécessairement un contact physique pour accomplir le travail.
- **la collaboration** : Le degré d'interaction le plus élevé est atteint dans les scénarios de collaboration, dans lesquels les humains et les robots travaillent simultanément sur le même objet dans le même espace de travail tout en effectuant une activité de travail commune [108].

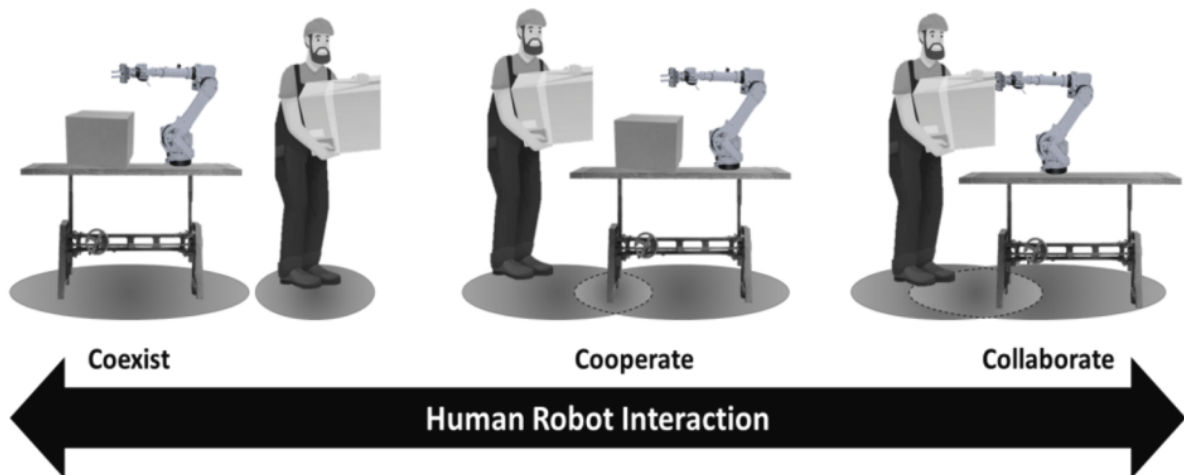


FIGURE 1.11 – Les différents types d'interaction homme-robot (Roohollah Jahanmahin *et al.*, 2022) [9]

Si l'interaction homme-robot était plutôt restreinte au stade de la coexistence dans le cadre de l'I4.0, elle évolue aujourd'hui vers la coopération, voire même la collaboration, grâce à l'I5.0 [9]. Les humains

et les robots seront donc amenés à travailler côte à côte, comme des collègues, chacun réalisant une tâche particulière dans le processus de fabrication des produits. Par ailleurs, la grande robotisation et l'hyperconnectivité des industries induites par l'I4.0 nécessite une meilleure organisation méthodique des systèmes de production et une meilleure répartition du travail entre les différentes ressources (humains, machines, robots) du système pour garantir l'efficacité de l'industrie. Cette organisation est plus connue sous le terme de « ordonnancement des tâches ».

1.7 Ordonnancement des tâches d'un système de production

L'ordonnancement peut être défini comme l'allocation de ressources partagées limitées dans le temps [109]. De façon générale, l'objectif derrière un problème d'ordonnancement est l'optimisation du temps de traitement de toutes les tâches [110]. Au fil des ans, les problèmes d'ordonnancement ont suscité l'intérêt de nombreuses recherches [111, 112, 113] et ont fait l'objet d'une littérature importante dans les domaines de la recherche opérationnelle, des mathématiques et de l'informatique. L'accent a été mis sur l'étude des problèmes d'ordonnancement des machines, où les tâches représentent des activités de traitement et les machines sont les ressources disponibles pour effectuer ces traitements. Ce type de problème apparaît dans une multitude de domaines réels divers, comme par exemple l'ordonnancement des tâches dans une chaîne de montage, l'affectation des tâches aux employés, l'ordonnancement des tâches dans l'informatique distribuée, etc. [114]

On distingue plusieurs types de problèmes d'ordonnancement, classés selon différents critères [22]. En l'occurrence, dans leur analyse de la littérature, Hamid Abedinnia *et al.* [115] ont identifié sept (07) principaux critères de classification des problèmes d'ordonnancement qui sont :

- **Le type de problème** : ce critère décrit les caractéristiques fondamentales du problème d'ordonnancement, à savoir si le problème est déterministe ou stochastique, statique ou dynamique, hors ligne ou en ligne, etc.
- **Le schéma des flux** : ce critère décrit la manière dont les tâches sont acheminées dans le système de production.
- **Les objectifs** : ce critère se réfère à l'objectif (ou aux objectifs) du problème d'ordonnancement. Il s'agit généralement de l'optimisation d'un ou d'un ensemble d'objectifs qui sont basés sur les coûts, les pénalités, les temps de traitement, etc.
- **Les caractéristiques des tâches et des machines** : ce critère traite des hypothèses liées aux

tâches et aux machines. On peut citer, entre autres, la préemption des tâches, les contraintes de précedence ou les pannes de machines.

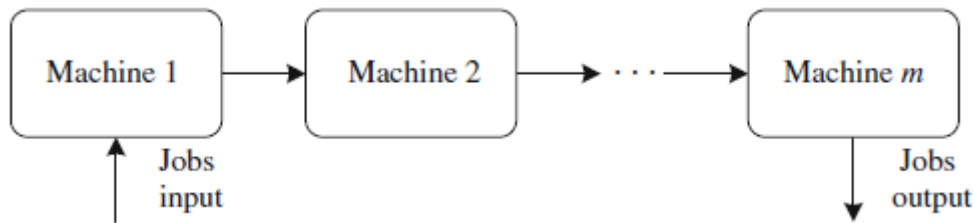
- **Les méthodes de résolution** : ce critère est relatif à la méthode de résolution utilisée pour traiter le problème d'ordonnancement. Les méthodes exactes, les heuristiques simples et les approches métaheuristiques sont des exemples de méthodes de résolution fréquemment utilisées.
- **L'applicabilité de l'ordonnancement** : ce critère s'intéresse à l'application des modèles d'ordonnancement sur des problèmes réels.
- **La modélisation théorique** : ce dernier critère fait référence aux aspects théoriques et mathématiques des problèmes d'ordonnancement. On retrouve par exemple le concept de la complexité informatique des problèmes.

Chaque déclinaison de problème présente des caractéristiques et une complexité propres. Par ailleurs, Framinan Jose *et al.* [10] proposent une classification des problèmes d'ordonnancement en se basant sur l'organisation du système de production. Ils distinguent ainsi trois catégories principales : les problèmes d'ordonnancement de type « flow shop », « job shop » et « open shop ».

Notre travail dans cette thèse s'est essentiellement focalisé sur les problèmes de type job shop, étant donné la large utilisation de cette organisation depuis les SMF jusqu'aux SMI. En effet, comme le mentionnent bien Framinan Jose *et al.* [10], les job shops sont très répandus car dans la pratique, les « jobs » (terme qui désigne le processus de traitement d'un produit) sont différents et nécessitent donc des traitements particuliers. En outre, l'organisation de type job shop s'inscrit dans le changement de paradigme de la production de masse vers la personnalisation de masse, offrant ainsi plus de flexibilité dans les processus de manufacture.

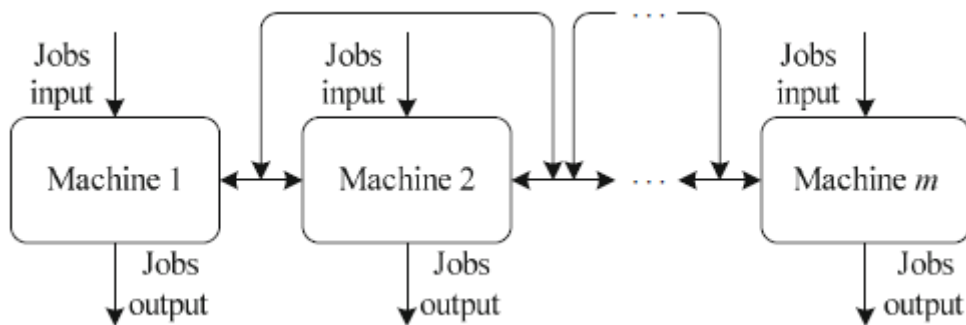
1.7.1 Flow shop

Un flow shop (ou atelier à flux continu) représente un type d'organisation de la production caractérisé par un flux linéaire et séquentiel des tâches à travers un ensemble de machines spécialisées. Il est composé de m machines rangées en série, les unes après les autres, comme sur la figure 1.12. À chaque machine est assignée une activité particulière et tous les jobs, doivent visiter toutes les machines dans le même ordre.

FIGURE 1.12 – Schéma illustratif d'un flow shop (Framinan Jose *et al.*, 2014) [10]

1.7.2 Job shop

Le job shop représente un type d'organisation de la production caractérisé par la flexibilité et la personnalisation. Contrairement au flow-shop, où les jobs suivent un cheminement linéaire prédéfini, le job shop offre une grande liberté dans le processus de traitement des jobs et dans la gestion des flux de production. Dans les problèmes de type job shop, les jobs sont différents. Ainsi, comme illustré par la figure 1.13, le traitement et l'ordre de passage sur les machines sont aussi différents pour chaque job. Selon Tliba Khalil (2022) [22], il s'agit du problème d'ordonnancement le plus rencontré dans la réalité et le plus difficile à résoudre.

FIGURE 1.13 – Schéma illustratif d'un job shop (Framinan Jose *et al.*, 2014) [10]

1.7.3 Open shop

L'open shop est une organisation plus générale de l'atelier de production. À la différence des deux précédentes organisations, le processus de traitement des jobs dans l'open shop n'est pas fixé, mais plutôt défini en fonction des besoins internes et de la demande externe. En effet, l'ordonnancement est libre (comme sur la figure 1.14), c'est-à-dire que le traitement et l'ordre de passage sur les machines sont librement définis au sein de l'atelier. On peut donc considérer le flow shop et le job shop comme

des cas particuliers de l'open shop [10, 21]

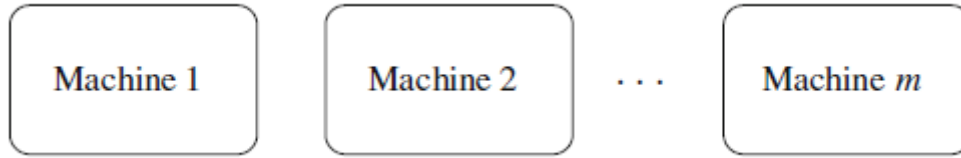


FIGURE 1.14 – Schéma illustratif d'un open shop (Framinan Jose *et al.*, 2014) [10]

1.7.4 Les tâches de transport dans les systèmes de production

Les tâches de transport jouent un rôle important dans les systèmes de production, assurant la liaison entre les différentes étapes du processus de manufacture et contribuant à son efficacité globale. Ces tâches englobent une large gamme d'activités, depuis le déplacement des jobs (matières premières et autres divers composants) jusqu'à l'entreposage des produits finis. En particulier, les activités de déplacement des jobs entre les différentes machines de traitement sont essentielles au bon fonctionnement du système de production car, comme le rappelle Moussa Abderrahim [21], les machines sont généralement distantes les unes des autres dans les ateliers.

1.7.4.1 Robotisation des tâches de transport

Traditionnellement, les tâches de transport étaient réalisées manuellement par des opérateurs humains munis de transpalettes, de chariot élévateurs ou de tout autre engin motorisé dédié au transport au sein de l'atelier. Cependant, bien que aidés par ces différents outils, les tâches de transport manuelles restent particulièrement pénibles physiquement pour les travailleurs. De plus, la répétitivité de ces tâches, associée aux dangers auxquels ils s'exposent parfois, rendent le travail harassant pour les humains. Ainsi, depuis quelques décennies maintenant, ces tâches sont confiées à des robots, comme sur la figure 1.15, connus sous l'appellation de véhicules terrestres autonomes (*Unmanned Ground Vehicles - UGV*, en anglais) [116], afin d'améliorer les conditions de travail, la productivité et la rentabilité. La robotisation des tâches de transport permet de soulager les opérateurs humains de ces tâches pénibles et de les rediriger vers des activités à plus forte valeur ajoutée, améliorant ainsi leur bien-être et leur sécurité au travail. En outre, les UGV sont programmables et peuvent s'adapter rapidement à des changements dans les processus de production ou les configurations d'entrepôt, ce qui offre une

1.7. ORDONNANCEMENT DES TÂCHES D'UN SYSTÈME DE PRODUCTION

flexibilité accrue pour répondre aux variations de la demande ou aux évolutions du marché.



FIGURE 1.15 – Illustration de véhicules autonomes terrestres (Alstef Group, 2024) [11]

Les UGV sont utilisés depuis plusieurs années dans les industries et font constamment l'objet d'amélioration en vue de les rendre encore plus performants [117]. On distingue essentiellement deux (02) grandes catégories d'UGV, à savoir : les véhicules autonomes guidés (*Autonomous Guided Vehicle - AGV*, en anglais) et les véhicules autonomes intelligents (*Autonomous Intelligent Vehicle - AIV*, en anglais). Les AGV ont prouvé leur efficacité dans les tâches répétitives de manutention depuis plusieurs décennies [118]. Cependant, en tant que robots guidés (grâce à des rails, des marquages au sol ou par guidage logiciel), ils nécessitent des zones de mouvement dédiées [119] dont la mise en place peut être coûteuse. De plus, ce sont des robots qui arrivent difficilement gérer la présence humaine sur leur chemin ainsi que les imprévus tels que les obstacles fixes et mobiles (suite à chute d'objets par exemple) [120]. Cette incapacité à faire face aux imprévus peut entraîner un arrêt complet des robots, ce qui peut entraver la bonne marche du processus de production. En revanche, Les AIV, aussi connus sous l'appellation de robots mobiles autonomes (*Autonomous Mobile Robot - AMR*, en anglais), peuvent facilement contourner les obstacles fixes et éviter les obstacles mobiles comme les humains. Contrairement aux AGV, ils n'ont pas besoin d'un espace de mouvement dédié, ce qui les rend plus adaptés aux usines modernes dans le contexte de l'I5.0, où les humains et les robots travaillent

ensemble en toute transparence [117, 121]. Ils sont également capables de prendre des décisions plus nombreuses et plus complexes, comme notamment la planification de leurs itinéraires. De ce fait, ils sont plus efficaces, plus adaptables et plus fiables que les anciennes méthodes de transport [119, 117].

1.7.4.2 Importance des tâches de transport dans l'ordonnancement

Bien que souvent ignorées, Les tâches de transport jouent un rôle primordial dans l'optimisation de l'ordonnancement des systèmes de production. Elles constituent un maillon essentiel reliant les différentes étapes de production et garantissant la continuité du processus de fabrication. En effet, sans un transport efficace des jobs entre les étapes de production, l'ensemble du système s'enlise, entraînant des retards, des gaspillages et une baisse de la productivité globale. Ainsi, comme le soulignent Kader Sanogo *et al.* [15], il est insuffisant de se concentrer uniquement sur l'optimisation des tâches de production en ignorant celles de transport. Pour une réelle amélioration des performances, il devient indispensable de considérer les deux aspects de manière simultanée. Cela implique l'intégration des tâches de transport dans l'ordonnancement globale du système de production, en tenant compte des interdépendances entre les tâches et des contraintes physiques dans l'atelier. Toutefois, la considération des tâches de transport dans l'optimisation de l'ordonnancement fait intervenir d'autres challenges [122, 123, 124], notamment : la gestion de la flotte de robots, la communication horizontale entre robots, la gestion des conflits sur les ressources partagées, l'évitement de collisions entre robots et avec les humains ou encore l'optimisation des tâches (transport et production) dans un environnement incertain et/ou dynamique. Il convient donc d'adresser toutes (ou presque) ces questions dans une approche d'optimisation résiliente et proche de la réalité industrielle.

1.8 Problématique de la thèse

Cette thèse se concentre sur l'optimisation de l'ordonnancement des tâches de transport et de production dans les ateliers de production flexibles, en prenant en compte les interactions imprévisibles avec les opérateurs humains. L'évolution des systèmes de production, depuis les SMF, a accru la nécessité de robotiser les tâches de transport en raison de leur répétitivité et leur pénibilité. Cependant, la transition de l'I4.0 vers l'I5.0 amène les humains à partager les premiers rôles avec les robots, où tous travaillent ensemble dans des environnements communs partagés. Ce changement de paradigme a pour conséquence d'intensifier et de multiplier les interactions humain-robot au sein des ateliers de

production. Ainsi, les robots de transport doivent naviguer dans des zones où les opérateurs humains sont présents, rendant les algorithmes d’ordonnancement actuels imprécis, voire inefficaces. La thèse vise à développer des algorithmes d’ordonnancement dynamique, intégrant des outils de modélisation stochastique, d’optimisation, de simulation et d’IA pour réagir face aux perturbations causées par les opérateurs humains et améliorer l’efficacité et la sécurité des processus de production.

1.8.1 Contexte scientifique

Ces dernières années, les systèmes de production ont subi des changements importants à tous les niveaux (disposition des machines, gestion de flux, outils de travail) afin de s’adapter aux exigences du marché et de la concurrence internationale. Les opérateurs humains ont la capacité de s’intégrer dans de nouveaux environnements, mais la technologie doit aussi prendre en compte le comportement évolutif et imprévisible de l’humain [125]. Le récent changement de paradigme, c’est-à-dire le concept d’I5.0, se concentre sur la durabilité du facteur humain au sein des technologies introduites par l’I4.0, notamment dans le cadre de l’ordonnancement des tâches de production, de transport et de maintenance dans les ateliers de production. En effet, l’utilisation de toute nouvelle technologie dans le contexte industriel est sujette aux phénomènes de résistance au changement et à l’adaptation de besoins propres des opérateurs. L’introduction des technologies est donc un processus incrémental qui se base sur le retour d’expérience des utilisateurs, leurs nouveaux besoins et les problèmes dérivés de la précédente utilisation [126].

Ainsi, Cette thèse s’intéresse à l’introduction des robots mobiles intelligents dans les ateliers de production. Ces robots sont en interaction permanente avec les opérateurs humains et doivent interagir aussi avec l’infrastructure de l’atelier de production. Cela va générer des temps de déplacement aléatoire et non maîtrisés à cause de leur dépendance au comportement des humains et à la configuration de l’atelier. L’objectif est donc d’étudier le comportement des opérateurs en interaction avec les robots mobiles et de développer une architecture de contrôle et d’ordonnancement dynamique pouvant intégrer le comportement imprévu et évolutif des opérateurs humains.

1.8.2 Problématique de recherche

La principale problématique de recherche de cette thèse, comme indiqué d’ailleurs dans le titre, porte sur l’optimisation de l’ordonnancement des tâches de transport des robots intelligents dans

le contexte de l'I5.0. Plus précisément, il s'agit de développer des algorithmes d'ordonnancement dynamique pour les tâches de transport et de production, prenant en compte les perturbations causées par les interactions imprévisibles et évolutives des opérateurs humains dans les ateliers de production flexibles et intelligents. Pour ce faire, on se doit de répondre aux questions suivantes :

- **Q1** : Comment modéliser et représenter l'ordonnancement joint des tâches de transport et de production ?
- **Q2** : Comment modéliser les comportements des opérateurs humains au sein de l'atelier de production ?
- **Q3** : Comment représenter, étudier et analyser les interactions robot-robot et humain-robot ?
- **Q4** : Quelles sont les effets de la présence d'opérateurs humains sur l'activité des robots ?
- **Q5** : Quelle(s) architecture(s) de contrôle et d'ordonnancement développer ?
- **Q6** : Comment optimiser l'ordonnancement joint des tâches de transport et de production tout en considérant les effets de la présence d'opérateurs humains ?

Dans la suite de ce document, les termes « ordonnancement des tâches » et « ordonnancement joint des tâches de transport et de production » seront utilisés de manière interchangeable. Lorsqu'il s'agira de parler d'une tâche en particulier (transport ou production), la précision sera faite.

1.8.3 Verrous scientifiques et techniques

Les verrous scientifiques techniques identifiés dans cette thèse comprennent plusieurs aspects complexes et multidisciplinaires qui nécessitent une approche complète combinant modélisation stochastique, optimisation multi-objectif, simulation et IA :

- **Modélisation et Simulation de l'Ordonnancement des tâches** : La résolution des problèmes d'ordonnancement des tâches de transport et de production fait intervenir des contraintes souvent difficiles à modéliser avec les méthodes analytiques classiques. Ainsi, la simulation se présente comme une alternative dans sa capacité à pouvoir reproduire le comportement physique d'un système et d'en permettre son étude.
- **Modélisation et Simulation des Comportements des Opérateurs Humains** : Il est nécessaire de développer des modèles de comportement des opérateurs humains basés sur la littérature et des expérimentations, ce qui inclut la collecte et l'analyse de données sur les interactions entre les humains et les robots dans des environnements de production flexibles.

- **Ordonnancement Prévisionnel et Dynamique** : La création d’algorithmes d’ordonnancement capables de prendre en compte les perturbations causées par les interactions avec les opérateurs humains est cruciale. Cela implique l’utilisation de diverses méthodes d’optimisation, incluant des méthodes exactes, métaheuristiques, et optimisation stochastique.
- **Intégration et Interaction des Robots Transporteurs** : La simulation des interactions entre les robots de transport (AIV et AGV) posent des défis importants qui sont (entre autres) : la gestion des flottes, l’évitement des collisions, la résolution des conflits sur les ressources partagées et la gestion des priorités dans les zones de déplacement.
- **Tests et Validation expérimentale** : Il est important de tester et de déployer les solutions développées sur des instances bien connues de littérature afin de pouvoir se positionner sur l’existant et d’identifier les contributions de ce travail de recherche.

1.8.4 Analyse de l’état de l’art et positionnement scientifique

L’optimisation de l’ordonnancement joint des tâches de transport et de production [1, 111, 127] ainsi que la considération des humains dans les problèmes d’ordonnancement [121, 128, 129] sont deux thématiques de recherche qui sont de plus en plus abordés dans la littérature. Cependant, malgré l’intérêt grandissant pour ces sujets, il existe encore un grand déficit de travaux de recherche faisant la jonction entre ces deux thématiques, c’est-à-dire intégrant la prise en compte des opérateurs humains dans l’optimisation de l’ordonnancement des tâches [130]. Comme peut le témoigner le tableau 1.1, la prise en compte des opérateurs humains dans l’optimisation de l’ordonnancement joint des tâches de transport et de production n’est presque pas abordée. Par ailleurs, Les AIV qui peuvent jouer un rôle important dans la transition vers l’I5.0, étant donné leur capacité à pouvoir partager un environnement commun avec les humains, interviennent peu dans les travaux consacrés à l’optimisation des tâches. L’intérêt y est plutôt porté vers les AGV. Ainsi dans cette thèse, nous essayons de combler le gap de la littérature en développant une approche décentralisée d’optimisation dynamique qui tient compte de la présence humaine dans l’atelier lors de la prise de décision des AIV.

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Luca Gualtieri <i>et al.</i> [131]	Proposer des lignes directrices pour la conception d'un assemblage collaboratif homme-robot sûr, en se concentrant sur les caractéristiques caractérisant l'ensemble du système.	Non	Non	Non	Étude de cas en laboratoire et jumeau numérique.	Non	Non spécifié (NS)	Oui
Javier G. Martin <i>et al.</i> [132]	Proposer un cadre de théorie des jeux coopératif pour résoudre les problèmes d'allocation de tâches multi-robots (MRTA).	Non	Non	Oui	Algorithme basé sur la théorie des jeux coopératifs	Non	NS	Non
Li Nie <i>et al.</i> [133]	Proposer une approche de théorie des jeux pour résoudre le FJSSP, où tous les jobs et le fabricant sont considérés comme des joueurs.	Oui	Oui	Non	Modèle de jeu non coopératif	Non	NS	Non
Peter Burggräf <i>et al.</i> [134]	Appliquer un Algorithme d'Apprentissage par Renforcement (RL) à gradient de politique pour résoudre le FJSSP.	Oui	Oui	Non	Algorithme RL	Non	NS	Non
Hao Hu <i>et al.</i> [118]	Proposer une approche adaptative de planification en temps réel des AGV basée sur l'Apprentissage par Renforcement Profond (DRL) avec des règles mixtes pour l'atelier flexible afin de minimiser la durée de fabrication et le retard.	Non	Oui	Non	Apprentissage par renforcement profond (DRL)	Oui	AGV	Non
Paolo Renna <i>et al.</i> [135]	Utiliser une approche de théorie des jeux pour Proposer une règle d'affectation des travailleurs dans les ateliers de production.	Non	Non	Oui	Modèle Shapley Gale	Non	NS	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Constantin Hofmann <i>et al.</i> [136]	Proposer une approche adaptée à l'ordonnement d'une production matricielle avec une flexibilité de transport et de configuration et de processus.	Oui	Oui	Oui	Recherche arborescente de Monte Carlo	Oui	NS	Non
Jin Wang <i>et al.</i> [137]	Présenter une approche de planification de production en temps réel basée sur une planification de production multi-période (MPPRS) pour l'atelier de travail flexible à faible émission de carbone (LFJS) P	Oui	Oui	Oui	Jeu répété à l'infini	Non	NS	Non
Jiaojiao Liu <i>et al.</i> [138]	Présenter un modèle de programmation à deux niveaux pour la planification intégrée avec les machines et AMR, subdivisant la stratégie de répartition partagée distribuée (DSDS) et la stratégie de répartition suivante (FDS) pour les AMR	Oui	Oui	Non	Q-learning intégré dans un Algorithme génétique	Oui	AIV	Non
Hui Zou et Yan Xi [139]	Proposer une méthode d'allocation dynamique et décentralisée des tâches basée sur le langage de description de jeu (GDL) et la théorie des jeux	Non	Non	Non	Langage de description de jeu (GDL) et théorie des jeux	Non	NS	Non
Han Wu et Huiliang S. [140]	Proposer une méthode d'allocation dynamique de tâches dans un système multi-agents.	Non	Oui	Non	Jeu de potentiel	Non	NS	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Xiaofei Zhu <i>et al.</i> [141]	Étudier le problème de planification en temps réel d'un atelier de travail flexible à double ressource avec des robots.	Oui	Oui	Non	Apprentissage par renforcement multi-tâches multi-agents (MTMARL)	Non	NS	Non
M. S. Ashhab <i>et al.</i> [142]	Résoudre le problème de planification d'atelier flexible multi-objectifs (MO-FJSSP) qui prend en compte l'utilisation du temps de transport.	Oui	Oui	Non	programmation non linéaire en nombres entiers mixtes (MILP)	Oui	NS	Non
Yiyi Xu <i>et al.</i> [18]	Résoudre le FJSSP avec mission de transport.	Oui	Oui	Oui	Algorithme de planification dynamique basé sur le calcul de l'entropie	Oui	AGV	Non
Milica Petrović <i>et al.</i> [143]	Proposer une méthodologie multi-objectifs Grey Wolf Optimizer (MOGWO) pour planifier efficacement les systèmes de transport de matériaux basés sur un robot mobile unique intelligent.	Non	Oui	Non	Algorithme d'optimisation de Grey Wolf	Oui	AIV	Non
Jinfeng Y. et H. Xu [144]	Résoudre le problème de planification distribuée et floue des ateliers de travail flexibles avec le transfert, en tenant compte de l'incertitude dans l'environnement de traitement.	Oui	Oui	Non	Algorithme mémé-tique de voisinage variable de décomposition hybride (HDVMA)	Oui	NS	Non
Fayong Zhang <i>et al.</i> [145]	Présenter un FJSSP soucieux de l'énergie avec des ressources de transport limitées.	Oui	Oui	Non	Algorithme mémé-tique profond basé sur le réseau Q	Oui	AGV	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Lazher Zaidi <i>et al.</i> [146]	Présenter une simulation d'atelier multi-robots dans lequel les robots participent de manière indépendante mais collaborative à la chaîne d'assemblage de téléphones cellulaires.	Oui	Oui	Oui	Algorithme génétique (GA)	Oui	NS	Non
Bilel Marzouki <i>et al.</i> [147]	Introduire une nouvelle variante du problème d'ordonnement flexible et dynamique des ateliers avec les ressources de transport (DFJSSPT) où chaque opération peut être réalisée dans un ensemble d'usines géographiquement réparties.	Oui	Oui	Non	Optimisation des réactions chimiques (CRO)	Oui	NS	Non
Chen Peng <i>et al.</i> [148]	Établir un modèle mathématique d'optimisation des plannings communs multi-objectifs fabrication-transport.	Oui	Oui	Non	Algorithme mémé-tique (MA) et Algorithme génétique de tri non dominé-II (NSGA-II)	Oui	NS	Non
Dalila B.M.M. Fontes <i>et al.</i> [149]	Étendre le problème de planification d'atelier économe en énergie aux ressources de transport en considérant des ressources à vitesse réglable.	Oui	Oui	Non	MILP et Algorithme génétique à clé aléatoire biaisé multi-objectifs	Oui	AGV	Non
Jiang-Ping Huang <i>et al.</i> [150]	Résoudre le problème de planification distribuée des ateliers de travail (DJSSP) avec les arrivées de jobs.	Oui	Oui	Non	Apprentissage par renforcement profond multi-actions (MDRL)	Non	NS	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Chee H. Lim <i>et al.</i> [151]	Résoudre le FJSSP avec tâches de transport.	Oui	Oui	Non	programmation mathématique et heuristique	Oui	AGV§	Non
A. Sobhani [152]	Intégrer les facteurs humains dans l'optimisation des systèmes d'assemblage.	Non	Oui	Non	Modèles d'optimisation des performances	Non	NS	Oui
I. El Mouayni <i>et al.</i> [153]	Aborder la planification des tâches en tenant compte de la fatigue et de la fiabilité des travailleurs dans des environnements agiles.	Oui	Oui	Non	MILP et procédure de recherche adaptative randomisée gourmande (GRASP)	Non	NS	Oui
B. Vahedi-Nouri <i>et al.</i> [154]	Optimiser la planification de la production dans les contextes de collaboration homme-robot.	Oui	Oui	Non	MILP et programmation par contraintes (CP)	Non	NS	Oui
L. Cai <i>et al.</i> [155]	Optimiser la planification des ateliers dans un environnement collaboratif en tenant compte de la logistique	Oui	Oui	Oui	Simulation de planification en temps réel	Oui	AGV	Non
Candice Destouet <i>et al.</i> [156]	Optimiser conjointement le makespan, la consommation totale d'énergie et l'indice OCRA (Occupational Repetitive Actions) (une mesure du risque ergonomique) dans le cas d'un atelier de travail flexible.	Oui	Oui	Non	NSGA-II	Non	NS	Oui
Robert Bridger [157]	Offrir une vision complète et actualisée de l'atelier, structurée selon un modèle homme-machine.	Non	Non	Non	NS	Non	NS	Oui

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Minghai Yuan <i>et al.</i> [158]	Développer une méthode de planification en temps réel pour le problème de planification flexible des ateliers avec AGV (FJSSP-AGV).	Oui	Oui	Non	Double deep Q network (DDQN)	Oui	AGV	Non
Yan Wang <i>et al.</i> [159]	Explorer la planification dans laquelle les tâches principales sont interrompues par des tâches en attente, en tenant compte de facteurs humains tels que la gestion de la charge de travail et l'amélioration de l'efficacité.	Oui	Oui	Non	Modèle mathématique et heuristique	Non	NS	Oui
Ercan Şenyigit <i>et al.</i> [160]	Étudier les effets des paramètres d'actions répétitives professionnelles (OCRA), du taux d'apprentissage sur les temps de processus et de la planification des machines.	Oui	Non	Non	Analyse statistique (ANOVA)	Non	NS	Oui
Soriano Jhurnieve <i>et al.</i> [161]	Résoudre le problème intégré de planification des employés qui prend en compte divers problèmes réels.	Non	Oui	Non	Processus de hiérarchie analytique et MILP	Non	NS	Oui
Anisha Khaitan <i>et al.</i> [162]	Déterminer les différents facteurs de stress du conducteur et les résoudre à l'aide d'un problème de tournée de véhicules.	Non	Oui	Non	Algorithme génétique hybride	Non	NS	Oui
M. P. Doan <i>et al.</i> [163]	Étudier l'impact des contrats de travail flexibles et des agents polyvalents sur un problème de planification des effectifs multi-objectifs.	Non	Oui	Non	MILP	Non	NS	Oui

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Zhao Peng <i>et al.</i> [164]	Proposer un mécanisme de contrainte de double ressource homme-machine dans un modèle de problème de planification d'atelier flexible et multi-objectif, contraint par le temps de transport du travail et l'effet d'Apprentissage.	Oui	Oui	Non	Algorithme hybride discret de compétition impériale multi-objectifs (HDMICA)	Non	NS	Oui
Jian Li <i>et al.</i> [165]	Résoudre le problème de planification d'atelier flexible et multi-objectifs avec des contraintes de transport.	Oui	Oui	Non	Algorithme de loups multi-objectifs (MOWPA)	Oui	NS	Non
Yanning Fu <i>et al.</i> [166]	Résoudre la planification des flowshops hybrides (HFSP) prenant en compte le makespan, la pollution sonore et la poussière.	Non	Oui	Non	Algorithme d'optimisation multi-objectifs du pilote (MOROA)	Non	NS	Oui
Francesco P. et Beatrice L. [167]	Présenter une formulation du problème de démontage parallèle qui maximise le degré de parallélisme, le niveau d'ergonomie et l'équilibre de la charge de travail, tout en minimisant le temps de démontage et le nombre de rotations du produit.	Non	Oui	Non	Algorithme métrique tensoriel (TeMA)	Non	NS	Oui
Christoph H. Glock <i>et al.</i> [168]	Étudier les activités de préparation de commandes manuelles dans un entrepôt où les articles sont stockés sur des palettes sur deux rangées l'une au-dessus de l'autre.	Non	Oui	Non	Modèle mathématique	Non	NS	Oui

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Shoujing Zhang <i>et al.</i> [169]	Résoudre le problème de planification flexible des ateliers de travail à double contrainte de ressources (DRCFJSSP) avec des différences de temps de fonctionnement entre les opérateurs.	Oui	Oui	Non	Algorithme de génétique quantique (QGA)	Non	NS	Oui
Li H. et Wu X. [170]	Résoudre le problème de planification durable et flexible des ateliers en tenant compte des doubles ressources (SFJSSPCDR), c'est-à-dire les machines et les travailleurs.	Oui	Oui	Non	Algorithme NSGA-III guidé par la durée de survie (SDG-NSGA-III)	Non	NS	Oui
Dai Min <i>et al.</i> [171]	Formuler un modèle d'optimisation multi-objectifs dans le but de minimiser la consommation d'énergie et le makespan dans le cadre d'un problème de planification flexible des ateliers avec un contexte de contraintes de transport.	Oui	Oui	Non	Algorithme génétique amélioré (EGA)	Oui	AGV	Non
Wei Weng <i>et al.</i> [172]	Proposer quatre méthodes qui forment un système de planification et de contrôle en temps réel pour la production juste à temps (JIT).	Oui	Non	Oui	Simulation	Non	NS	Oui
Marta Rinaldi <i>et al.</i> [173]	Résoudre le problème d'ordonnement des effectifs et améliorer les performances du système en intégrant l'ergonomie et les compétences humaines.	Non	Oui	Non	MILP	Non	NS	Oui

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Yin Liu <i>et al.</i> [174]	Étudier le problème d'ordonnement flexible d'atelier multi-objectifs (MOFJSSP) considérant les affectations des travailleurs qui minimisent trois objectifs, à savoir le makespan, la charge de travail totale des machines et le coût de la main d'œuvre.	Oui	Oui	Non	Algorithme génétique de tri non dominé amélioré-II (INSGA-II)	Non	NS	Oui
Weihua Tan <i>et al.</i> [175]	Résoudre le DRCFJSSP.	Oui	Oui	Non	NSGA-II amélioré (ENSGA)	Oui	NS	Oui
Shichang Xiao <i>et al.</i> [176]	Résoudre les problèmes de planification des ateliers avec des temps de traitement stochastiques (SJSSP) sous des contraintes de double ressource machine-ouvrier.	Oui	Oui	Oui	Algorithme d'estimation hybride multi-objectif de la distribution (MO-HEDA)	Non	NS	Oui
A. Gnana-velbabu <i>et al.</i> [177]	Aborder un FJSSP stochastique multi-objectif avec flexibilité des travailleurs.	Oui	Oui	Oui	Simulation Monte-Carlo (MCS)	Non	NS	Oui
Guiliang Gong <i>et al.</i> [178]	Proposer un modèle mathématique d'un problème étendu de planification d'atelier flexible avec flexibilité des travailleurs (FJSSPW).	Oui	Oui	Non	Algorithme de colonie d'abeilles artificielles hybrides (HABCA)	Non	NS	Oui
David M. et Dominik K. [179]	Résoudre un problème de planification flexible des ateliers de travail qui prend en compte les contraintes de main-d'œuvre et vise à minimiser le makespan.	Oui	Oui	Non	Heuristique basée sur un filtre et un ventilateur (F&F)	Non	NS	Oui

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Farah Farjallah <i>et al.</i> [180]	Résoudre le DRCFJSSP en tenant compte des contraintes causées par la disponibilité des travailleurs.	Oui	Oui	Non	Recherche taboue à multi-démarrage basé sur des agents (MuSTAM)	Non	NS	Oui
Dominik Kress <i>et al.</i> [181]	Considérer une variante du FJSSP avec des temps de configuration dépendants de la séquence qui intègre des qualifications hétérogènes d'opérateurs de machines.	Oui	Oui	Non	Logique des inégalités et Algorithme heuristique	Non	NS	Oui
Paweł Stańczek <i>et al.</i> [182]	Tester l'environnement d'exploitation d'un robot mobile autonome (AMR) avec jumeau numérique.	Non	Non	Oui	Jumeau Numérique	Oui	AIV	Oui
Alejandro Vital-Soto <i>et al.</i> [183]	Résoudre une variante du FJSSP intégrant la flexibilité de séquençage des affectations des machines et des travailleurs (DR-FJSSPS).	Oui	Oui	Non	NSGA-II	Non	NS	Oui
Qiang Luo <i>et al.</i> [184]	Étudier le problème de planification distribuée des ateliers flexibles (DF-JSSP) avec l'arrangement des travailleurs.	Oui	Oui	Non	MILP et Algorithme mémétique amélioré (IMA)	Non	NS	Oui
Atefeh M. et Marco C. [185]	Proposer des Algorithmes pour résoudre la planification de trajectoire et l'évitement des collisions dans un système de transport autonome.	Non	Non	Oui	Algorithme de Dijkstra amélioré	Non	AGV	Non
Lucas B. <i>et al.</i> [186]	Résoudre le FJSSPT.	Oui	Oui	Non	Recherche taboue	Oui	NS	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Wenfeng Li <i>et al.</i> [187]	Étudier le problème de planification d'atelier économe en énergie avec un nombre limité de travailleurs.	Oui	Oui	Non	Mécanisme d'évaluation de la condition physique (FEM) basé sur l'entropie de corrélation floue (FCE)	Non	NS	Oui
Martin Čech <i>et al.</i> [188]	Créer une base de connaissances complète pour la conception, la sélection et la mise en œuvre de la technologie AMR sous la forme de supports de charge unitaires pour l'approvisionnement des chaînes de montage de l'industrie automobile.	Non	Non	Non	NS	Oui	AGV et AIV	Non
Soovadeep Bakshi <i>et al.</i> [189]	Introduire un Algorithme en deux étapes pour la planification rapide des AMR qui effectuent des tâches prioritaires impliquant le transport d'outils/matériaux depuis un lieu de ramassage jusqu'à un point de dépôt dans l'usine.	Non	Oui	Non	Algorithme heuristique	Oui	AIV	Non
Chaima Bensaci <i>et al.</i> [190]	Fournir une analyse des risques et une comparaison plus pratique entre les deux architectures (centralisée et décentralisée) de contrôle d'un système mobile multi-robot	Non	Non	Non	Processus de hiérarchie analytique (AHP)	Oui	AIV	Oui
You-Jie Yao <i>et al.</i> [191]	Étudier le JSSPT.	Oui	Oui	Non	MILP	Oui	AIV	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Khadija Bakdi <i>et al.</i> [192]	Étendre le problème de planification d'atelier de flux de permutation (PFSP) en intégrant une flotte d'AGV identiques pour le transport de travaux semi-finis entre les machines.	Oui	Oui	Non	MILP	Oui	AGV	Non
Chee Han. Lim <i>et al.</i> [151]	Proposer une heuristique en deux phases utilisant la programmation mathématique pour minimiser le makespan dans un problème d'atelier flexible avec le transport (FJSSPT).	Oui	Oui	Non	Algorithme heuristique basé sur la programmation mathématique itérative en deux phases	Oui	NS	Non
Zhengying Cai <i>et al.</i> [193]	Introduire un nouvel Algorithme de colonie artificielle Physarum polycephalum pour résoudre le problème de planification flexible des ateliers avec des contraintes de transport, en utilisant des diagrammes de transition d'état et une fonction multi-objectifs.	Oui	Oui	Non	Algorithme de colonie artificielle Physarum polycephalum	Oui	NS	Non
Sebastiano Gaiardelli <i>et al.</i> [194]	Trouver un ordonnancement commun presque optimal pour le traitement des jobs et le transfert via des tapis convoyeurs.	Oui	Oui	Non	Algorithme heuristique basé sur la randomisation	Oui	NS	Non
Qinhui Liu <i>et al.</i> [195]	Intégrer les ressources AGV dans la planification de la production.	Oui	Oui	Non	Algorithme génétique amélioré	Oui	AGV	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Qihao Liu <i>et al.</i> [196]	Étudier un problème de planification et d'ordonnement intégrés des processus (IPPS) en considérant la tâche de transport AGV.	Oui	Oui	Non	Algorithme génétique amélioré (IGA)	Oui	AGV	Non
Dalila B.M.M. Fontes <i>et al.</i> [127]	Étudier le JSSPT en considérant la minimisation de deux mesures de performances alternatives, à savoir : le makespan et le temps de sortie.	Oui	Oui	Non	MILP, optimisation d'essai de parties hybrides et Algorithme de recuit simulé	Oui	AGV	Non
S. Mahdi Homayouni <i>et al.</i> [197]	Résoudre le FJSSPT.	Oui	Oui	Non	Algorithme génétique à clé aléatoire biaisé (BRKGA)	Oui	AGV	Non
Qianfa Gao <i>et al.</i> [198]	Résoudre le problème de planification flexible des ateliers de travail avec des ressources de transport limitées (FJSSP-T) en cas de perturbations anormales.	Oui	Non	Oui	Architecture cloud	Oui	AGV	Non
Zi-Qi Zhang <i>et al.</i> [199]	Résoudre le problème de planification distribuée et flexible des ateliers avec le transport par grue (DFJSSPC) pour minimiser la somme pondérée du makespan et de la consommation totale d'énergie.	Oui	Oui	Non	Algorithme évolutif hyper-heuristique basé sur le Q-learning (QHHEA)	Oui	NS	Non
Youjie Yao <i>et al.</i> [200]	Résoudre le problème de planification des ateliers en intégrant des ressources de transport limitées (JSSPIFTR).	Oui	Oui	Non	Recherche taboue	Oui	AGV	Non

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Qiang Luo <i>et al.</i> [184]	Étudier le DFJSP avec organisation des travailleurs (DFJSPW), où les machines, les opérations et les travailleurs sont pris en compte simultanément.	Oui	Oui	Non	Algorithme métrique amélioré (IMA)	Non	NS	Oui
Jelena Peronjivic <i>et al.</i> [129]	Intégrer les marges temporelles et les facteurs humains en situation d'incertitude dans le JSSP.	Oui	Oui	Oui	Simulation-optimisation basée sur un système multi-agents	Non	NS	Oui
Dalila B.M.M. Fontes <i>et al.</i> [127]	Résoudre le JSPT en adressant simultanément les tâches de transport et de production.	Oui	Oui	Non	MILP et Algorithme hybride d'optimisation par essais de particules et de recuit simulé (PSOSA)	Oui	AGV	Non
Guohui Zhang <i>et al.</i> [201]	Aborder la planification flexible de l'atelier de travail en tenant compte des temps de transport et de préparation.	Oui	Oui	Non	RL	Oui	AGV	Non
Yujie Yao <i>et al.</i> [202]	Résoudre le problème de planification flexible des ateliers avec des AGV limités (FJSSPLA).	Oui	Oui	Non	MILP	Oui	AGV	Non
A. Ahmadi-Javid <i>et al.</i> [130]	Traiter une variante du JSSPT intégrant l'emploi du temps des employés et un nombre fini de transporteurs hétérogènes.	Oui	Oui	Non	MILP et Optimisation de la société anarchique (ASO)	Oui	AGV	Oui

Table 1.1 (suite du tableau)

Référence	Objectif du travail réalisé	POT	RpO	RpS	Technique adoptée	CdTT	TdR	CdH
Hangyu Lou <i>et al.</i> [203]	Étudier un FJSP multi-objectif prenant en compte les facteurs humains (MO-FJSPHF) afin de minimiser simultanément le makespan et la charge de travail totale de la machine.	Oui	Oui	Non	Algorithme mémé-tique multi-objectif (MOMA-LD) et Algorithme évolutionnaire multi-objectif (MOEA/D)	Oui	NS	Oui
Jiaojiao Liu <i>et al.</i> [204]	Résoudre le DFJSSPT dans lequel l'insertion de nouveaux jobs est traitée comme une perturbation dynamique.	Oui	Oui	Non	MILP et Algorithme de recherche adaptative multi-objectifs à large voisinage (MOALNS)	Oui	AGV	Non
Youjie Yao <i>et al.</i> [205]	Explorer le problème de planification des ateliers économes en énergie avec un nombre limité d'AGV.	Oui	Oui	Non	Algorithme mémé-tique multi-objectif avec réseau Q profond (DQNNMA)	Oui	AGV	Non
Kader Sanogo <i>et al.</i> [15]	Intégrer les contraintes d'évitement des collisions dans les problèmes de planification des ateliers avec les tâches de transport.	Oui	Non	Oui	simulation basée sur un système multi-agents	Oui	AGV	Non
Cette thèse	Développer une approche de simulation pour la résolution du JSSPT en considérant la présence humaine dans l'atelier	Oui	Oui	Oui	Théorie des jeux et la simulation basée sur un système multi-agents	Oui	AGV et AIV	Oui

TABLE 1.1 – État de l'art de la littérature

(Notation : **POT** : Problème d'ordonnement des tâches ; **RpO** : Résolution par optimisation ; **RpS** : Résolution par simulation ; **CdTT** : Considération des tâches de transport ; **TdR** : Type de robots ; **CdH** : Considération des humains.)

1.8.5 Les contributions de la thèse

La prise en compte de la présence humaine dans l'optimisation des tâches demeure un défi complexe, nécessitant l'exploration de plusieurs problématiques. Cette thèse se propose d'y répondre en apportant des contributions originales à travers le développement d'une approche basée sur la simulation pour résoudre les problèmes d'ordonnancement de type job shop. La thèse devra considérer les aspects suivants :

- **Développement d'un simulateur** : la première contribution de cette thèse consistera à développer un simulateur, basé sur un système multi-agents, capable de modéliser, simuler et optimiser les tâches d'un atelier de production. Il supporte à la fois les ordonnancements prévisionnels, où l'ensemble des tâches et leur séquence d'exécution sont définis en amont, et les ordonnancements dynamiques, pour lesquels le simulateur adopte une approche de simulation-optimisation en intégrant un algorithme d'ordonnancement dynamique pour l'optimisation de l'ordonnancement des tâches.
- **Modélisation du comportement des humains dans l'atelier** : cette thèse s'intéresse aux ateliers collaboratifs, où humains et robots partagent le même espace. Ainsi, il faudra modéliser le comportement des travailleurs, en particulier leurs déplacements, étant donné qu'ils peuvent impacter les activités des robots mobiles.
- **Introduction des AIV** : en remplacement des AGV, les AIV offrent une bien meilleure flexibilité et sont capables de naviguer dans les environnements complexes, aux côtés des humains. Cette thèse les étudiera dans le cadre du JSSPT et évaluera leurs performances dans un environnement mixte avec les humains.
- **Développement d'un algorithme d'ordonnancement dynamique intégrant le facteur humain** : Cet algorithme décentralisé, basé sur la théorie des jeux, permettra d'optimiser en temps réel l'allocation des tâches aux robots mobiles, en tenant compte des fluctuations liées aux déplacements des humains. Il devra améliorer la résilience et l'adaptabilité du système de production, garantissant une meilleure efficacité dans des environnements industriels complexes et changeants.

1.9 Conclusion

Somme toute, ce chapitre a permis d'explorer le contexte et la problématique qui fondent cette thèse. Il aborde notamment l'évolution des systèmes de production jusqu'à l'avènement de l'I5.0, une industrie centrée sur l'humain, où les problèmes d'ordonnancement doivent tenir compte du facteur humain. Ainsi dans cette thèse, il est essentiellement question d'optimiser l'ordonnancement joint des tâches de transport et de production, tout en considérant la présence des travailleurs humains dans l'atelier.

Pour répondre à la problématique principale de cette thèse, les travaux seront menés en deux phases : la première se concentrera sur le développement d'un simulateur pour simuler et optimiser l'ordonnancement des tâches dans un atelier de production, et la deuxième phase consistera au développement d'une approche décentralisée pour l'optimisation des ordonnancements dynamiques, ce qui impliquera la conception et l'intégration d'un algorithme d'ordonnancement dynamique basé sur la théorie des jeux dans le simulateur.

1.9. CONCLUSION

Chapitre 2

Modélisation et simulation multi-agents de l'atelier de production

Contenu

2.1	Introduction	75
2.2	État de l'art	76
2.2.1	Simulation numérique	77
2.2.2	Simulation à événements discrets	78
2.2.3	Simulation basée sur les agents	79
2.2.4	Utilisation de la simulation dans la résolution des problèmes industriels	80
2.3	Modélisation de l'atelier de production	81
2.3.1	Système multi-agents	81
2.3.2	Modélisation des activités de l'atelier	84
2.3.3	Modélisation de l'atelier	85
2.4	Développement du simulateur	87
2.4.1	Description du framework	87
2.4.2	Implémentation du modèle de simulation	90
2.4.3	Interface du simulateur	97
2.4.4	Exécution de la simulation	99
2.4.5	Intérêt du simulateur	107
2.5	Conclusion	108

2.1 Introduction

Dans le paysage manufacturier compétitif d'aujourd'hui, les entreprises cherchent constamment à rationaliser leurs opérations et à éliminer les gaspillages. Cette quête incessante d'efficacité les oblige à découvrir des méthodes et des outils innovants pour améliorer la production et maintenir un niveau élevé de satisfaction de la clientèle. Cependant, la gestion des processus de fabrication peut s'avérer

complexe, avec de nombreux éléments interconnectés et des événements imprévus. Les entreprises doivent donc tester rigoureusement leurs idées avant de mettre en œuvre des changements importants.

Dans ce contexte, la simulation apparaît comme un outil puissant pour examiner et analyser des situations qui sont soit trop complexes pour être modélisées mathématiquement, soit trop coûteuses pour être testées dans le monde réel [206, 207]. En effet, la simulation permet d'étudier et de prévoir l'évolution de divers comportements (ou phénomènes) dans un système, et d'anticiper les problèmes potentiels qui pourraient surgir. Un simulateur bien conçu, avec des données et des contraintes clairement définies, peut produire des résultats très proches de la réalité. Ainsi, cette capacité à pouvoir approximer la réalité lui vaut d'être très souvent utilisée dans la modélisation et la résolution des problèmes industriels.

Dans ce chapitre, nous débuterons par un bref état de l'art sur la simulation et son utilisation dans la résolution de problèmes industriels. Puis, nous présenterons la modélisation de notre atelier de production, basée sur un Système Multi-Agents (SMA), ainsi que le simulateur développé pour résoudre le problème d'ordonnancement. L'interface du simulateur est paramétrable, permettant de mettre en œuvre différents types d'ordonnancement et de tester plusieurs scénarios. Nous détaillerons ensuite le développement du simulateur, en expliquant les divers algorithmes qui le composent, et enfin, le chapitre se conclura par une synthèse des points abordés.

2.2 État de l'art

La simulation est un outil qui s'est avéré être un atout précieux pour les chercheurs pendant des décennies, leur permettant d'observer et d'analyser le comportement de phénomènes complexes. C'est une technique qui consiste à construire un modèle représentant un système réel, qui peut ensuite être exécuté pour observer et analyser le comportement dudit système, sans l'impacter lui-même dans la réalité [206, 207]. Le principal défi de la simulation d'un système avec un ordinateur réside dans la capacité à le décrire d'une manière compréhensible pour la machine. Cela est réalisé en définissant un ensemble de variables qui capturent l'état du système. Chaque combinaison unique de ces valeurs de variables représente une condition spécifique, ou un « état », du système. En modifiant ces variables selon des règles prédéfinies, le système simulé passe essentiellement d'un état à un autre. Ce processus de manipulation des variables pour représenter le comportement dynamique du système constitue

l'essence même de la simulation [208, 207].

L'utilité et l'efficacité de la simulation ont conduit à son adoption dans un large éventail de domaines, comme le témoigne Cristina Ruiz-Martin [209], et elle est particulièrement appropriée pour représenter des systèmes difficiles à modéliser avec des méthodes traditionnelles [210, 211]. Par exemple, J. de Moij *et al.* [212] utilisent la simulation pour développer un cadre d'étude des comportements sociaux complexes et des interventions pouvant influencer les épidémies. Ce cadre a été utilisé pour modéliser la propagation de la COVID-19 en Virginie, démontrant sa capacité à gérer des simulations à grande échelle avec des agents complexes. De même, C. Adam *et al.* [213] développent un simulateur pour l'étude des épidémies, conçu à des fins éducatives, afin d'aider les utilisateurs à comprendre les mécanismes fondamentaux des épidémies. La simulation est également utilisée par Yasser I. et Rasha H. [214] pour étudier comment les médias influencent le comportement des foules, afin de comprendre les conditions pouvant favoriser ou empêcher la formation de révolutions populaires.

La simulation occupe une place singulière au sein du domaine scientifique, se situant à l'intersection de plusieurs disciplines fondamentales telles que les mathématiques, la physique et l'informatique [110]. Cette position unique lui confère une puissance et une flexibilité remarquables, lui permettant de s'adapter à une large gamme de problèmes et de domaines d'application. De plus, en tant que discipline à part entière, la simulation regroupe une multitude de méthodes et d'approches distinctes, chacune apportant ses propres forces et faiblesses. Ces méthodes peuvent être classées selon divers critères, tels que le type de modèle utilisé (déterministe, stochastique, hybride), le niveau de détail du modèle, la technique de résolution (analytique, numérique, approchée) et les outils informatiques employés. Toutefois, on peut subdiviser la simulation en trois (03) techniques, à savoir : la simulation numérique, la simulation à événement discrets et la simulation basée sur les agents.

Le choix de la technique de simulation la plus adaptée dépend de la nature du problème à étudier, des ressources disponibles et des objectifs poursuivis. La flexibilité de la simulation réside dans sa capacité à combiner différentes méthodes et à les adapter aux spécificités de chaque cas.

2.2.1 Simulation numérique

Un modèle numérique, selon Drogoul Alexis [215], constitue une représentation mathématique simplifiée de la réalité, décrite par un ensemble de paramètres numériques et d'équations mathématiques qui définissent les relations entre ces paramètres. Ces paramètres peuvent être classés en deux caté-

gories : les causes et les effets. Les causes, tels que les variables exogènes ou les comportements des agents économiques, influencent les effets, qui représentent les résultats ou les variables endogènes du système. La simulation numérique s'appuie donc sur ces modèles mathématiques pour générer des réalisations aléatoires du système étudié, permettant ainsi d'évaluer son comportement sous différentes hypothèses et conditions. Comme le souligne Merouane Mazar [110], cette approche nécessite cependant une simplification excessive des agents et des interactions, modifiant ainsi la complexité inhérente aux systèmes réels en ne prenant pas en compte, par exemple, la dimension temporelle des interactions [216].

Dans son livre paru en 2022, Gerardo Diaz [217] offre une introduction à la simulation numérique des plasmas, où l'accent est mis sur les formulations hors équilibre et l'analyse des techniques de modélisation et des exigences en puissance de calcul qui influencent considérablement la simulation de phénomènes physiques complexes. Bingrui Lv *et al.* [218] proposent une méthode de micro-fraisage 3D assistée par vibration afin d'améliorer l'efficacité de la fabrication et de réduire les coûts de création de surfaces structurées. La méthode proposée repose sur l'utilisation d'un modèle numérique précis pour prédire les formes des surfaces structurées générées par le micro-fraisage. Nicolas Roussel *et al.* [219], quant à eux, présentent une revue des différentes approches de simulation numérique utilisées pour étudier l'écoulement du béton dans le contexte de la fabrication additive par extrusion. Ces approches incluent la dynamique des fluides computationnelle, les méthodes particulières et les modèles hybrides.

2.2.2 Simulation à événements discrets

La simulation d'événements discrets (*discrete-event simulation - DES*, en anglais), telle que définie par Ouda *et al.* [220], constitue une méthodologie puissante pour modéliser des systèmes complexes caractérisés par des événements discrets, tels que des processus stochastiques à temps discret. C'est une technique de modélisation qui s'appuie sur un moteur de simulation pour faire évoluer un système selon une chronologie d'événements [221, 222]. Cette approche se distingue par le traitement des événements à des instants précis de la chronologie simulée, d'où l'appellation « événements discrets ». Contrairement aux simulations à temps continu, la DES ne requiert aucune synchronisation avec le temps réel, ce qui lui confère une grande flexibilité pour prédire des phénomènes futurs ou étudier des processus complexes qui se déroulent sur de courtes périodes [110]. Cette approche s'avère particulièrement adaptée à la modélisation de systèmes dynamiques sujets à des changements aléatoires, tels

que les systèmes de production industriels ou les réseaux de transport.

Par exemple, Allal *et al.*[223] démontrent l'efficacité de la DES couplée à une approche d'optimisation pour optimiser l'acheminement et la programmation de la maintenance dans les parcs éoliens en mer. Leur modèle intègre une approche d'optimisation au sein du processus de simulation, permettant de déterminer de manière optimale les stratégies de maintenance en tenant compte des contraintes opérationnelles et économiques. Merdan *et al.* [224] illustrent également l'utilité de la DES dans l'optimisation des systèmes de production. Ils proposent une approche basée sur la DES pour optimiser le rendement total d'un système en tenant compte des pannes inattendues de machines ou de convoyeurs. Leur modèle permet de déterminer l'ordonnancement optimal du flux de travail en réponse à des événements dynamiques, maximisant ainsi la productivité du système. La DES peut également s'avérer être un outil efficace pour la prédiction précise de l'occurrence d'événements discrets dans le domaine du dépistage de maladies complexes. W. Jian-Min *et al.* [225] Adoptent cette approche pour établir la fonction de distribution de probabilité (FDP) de l'instant d'apparition de ces événements, offrant une compréhension approfondie de la dynamique temporelle de la maladie.

2.2.3 Simulation basée sur les agents

Comme définie par George Datsoris *et al.* [226], la simulation basée sur les agents (*agent-based modeling - ABM*, en anglais) est une technique de simulation où des entités indépendantes, appelées « agents », suivent des règles programmées pour interagir entre elles et avec leur environnement. Cette méthode est particulièrement utile pour modéliser et simuler des systèmes complexes caractérisés par les interactions entre leurs composants. À ce sujet, C. M. Macal [227] affirme que la modélisation et la simulation basées sur les agents sont une approche prévalente utilisée dans divers travaux de recherche et dans un large éventail de domaines tels que l'économie, la santé, les systèmes sociaux et le transport.

Entre autres, Andrew J. C. et Erika F. [228] explorent la formation stratégique de groupes avec un modèle basé sur les agents, en se concentrant sur la modélisation des personnes en tant que membres de groupes, plutôt qu'en tant qu'individus. De leur côté, Gayane G. *et al.* [229] utilisent une approche hybride de modélisation basée sur les agents, appelée ABMSCORE, pour comparer l'efficacité des agents informatisés et des humains réels dans la recherche d'une coalition idéale dans les jeux de gants. De plus, l'ABM a également été utilisée pour aborder des questions liées à la mobilité et aux villes intelligentes. Par exemple, pour comprendre comment les changements de tarifs de bus influencent

les choix des passagers, Baozhen Y. *et al.* [230] ont employé un modèle de simulation basé sur les agents pour imiter le flux de passagers en simulant leurs processus de prise de décision. De même, García-Suárez A. *et al.* [231] ont adopté l'ABM, combiné aux automates cellulaires, pour analyser le déploiement des stations de recharge de véhicules électriques à travers des simulations microscopiques du trafic.

2.2.4 Utilisation de la simulation dans la résolution des problèmes industriels

L'utilisation de techniques de simulation s'est révélée efficace pour relever divers défis rencontrés dans les systèmes de production au cours de ces dernières années. En effet, la simulation permet de modéliser des comportements complexes au sein des systèmes qui pourraient être difficiles à exprimer dans des modèles mathématiques théoriques [232], et d'explorer différentes configurations et stratégies de production avant leur mise en œuvre réelle. Plusieurs études récentes illustrent l'application de la simulation pour résoudre des problèmes concrets. Par exemple, M. B. Abdelkader *et al.* [233] utilisent la simulation pour intégrer la consommation d'énergie dans l'optimisation de l'ordonnancement des ateliers de production. En simulant différents scénarios de production, les auteurs identifient les configurations et les stratégies qui minimisent la consommation d'énergie tout en respectant les contraintes de production. De même, M'hammed Sahnoun *et al.* [122] utilisent la simulation pour déterminer la taille optimale d'une flotte d'AGV dans un atelier de production. En simulant le trafic d'AGV, les auteurs identifient le nombre d'AGV nécessaire pour assurer un flux de production fluide et minimiser les temps de production.

Lazher Zaidi *et al.* [146] illustrent le potentiel de la simulation pour révéler des divergences inattendues entre un programme théorique et sa mise en œuvre simulée, même dans un cas apparemment simple. Leur analyse met en évidence des écarts significatifs entre les performances prédites par le modèle théorique et les résultats obtenus par la simulation. Ce constat est également observé dans l'étude de Kader Sanogo *et al.* [15] qui révèle que la prise en compte de contraintes réelles, notamment l'évitement de collisions entre les AGV, peut rendre des ordonnancements théoriques irréalisables. Ils démontrent que la contrainte d'évitement de collision est particulièrement complexe à modéliser mathématiquement car elle demande de prendre en compte les formes des robots, les espaces occupés et la prédiction des éventuelles rencontres entre AGV qui pourraient apparaître. N. Bouaziz *et al.* [234] utilisent également la simulation pour représenter et analyser la nature aléatoire des déplacements

des opérateurs humains au sein d'un atelier. Ce modèle est conçu pour évaluer l'impact des divers comportements humains sur les performances du système de production.

2.3 Modélisation de l'atelier de production

Notre modélisation de l'atelier repose sur un système multi-agents (*Multi-agent system - MAS*, en anglais), qui permet de représenter les interactions entre les entités individuelles qui le composent. Ce MAS est intégré à un outil de simulation basée sur les agents afin de reproduire fidèlement l'ordonnancement des tâches dans l'atelier. Dans cette section, nous décrivons en détail le MAS et la modélisation du simulateur.

2.3.1 Système multi-agents

Le concept de système multi-agents (MAS) trouve ses racines dans le domaine de l'intelligence artificielle distribuée (DAI) [235]. Contrairement aux systèmes traditionnels contrôlés de manière centralisée, les MAS sont composés d'entités autonomes, les agents, qui interagissent les uns avec les autres et avec leur environnement pour atteindre des objectifs communs ou individuels. Chaque agent dans un MAS possède des caractéristiques essentielles qui lui permettent de fonctionner et d'interagir avec son environnement [213]. Ces caractéristiques sont :

- **l'autonomie** : Les agents ont la capacité d'agir et de prendre des décisions de manière indépendante, en se basant sur leurs propres perceptions et connaissances ;
- **la capacité de communication** : Les agents peuvent échanger des informations et coordonner leurs actions avec d'autres agents grâce à la communication et au partage de connaissances ;
- **la capacité d'apprentissage** : Les agents peuvent raisonner sur leur environnement et sur les actions possibles, et apprendre de leurs expériences pour améliorer leurs performances au fil du temps ;
- **les motivations et objectifs** : Les agents sont guidés par des motivations et des objectifs spécifiques qui déterminent leurs actions et leurs interactions avec les autres agents.

Les MAS ont été appliqués avec succès dans une large gamme de domaines [236, 237], notamment la robotique, la gestion de la production, la logistique, les réseaux de communication et les sciences sociales. Leur capacité à modéliser des systèmes complexes et dynamiques, ainsi que leur flexibilité et leur adaptabilité, font des MAS un outil intéressant pour la recherche et l'innovation dans divers

2.3. MODÉLISATION DE L'ATELIER DE PRODUCTION

domaines. La décentralisation qui leur est inhérente offre plusieurs avantages significatifs, notamment :

- **la flexibilité et l'adaptabilité** : Les MAS peuvent s'adapter à des environnements dynamiques et complexes en raison de l'autonomie et de la capacité d'apprentissage des agents ;
- **la robustesse et la tolérance aux pannes** : La défaillance d'un agent n'entraîne pas nécessairement l'effondrement du système entier, car les autres agents peuvent continuer à fonctionner et à reconfigurer le système si nécessaire ;
- **le parallélisme et l'efficacité** : Les agents peuvent exécuter des tâches simultanément, ce qui peut améliorer considérablement les performances du système, en particulier pour les problèmes à grande échelle ;
- **l'émergence de comportements complexes** : L'interaction entre des agents autonomes peut conduire à l'émergence de comportements globaux non prévus par la conception initiale du système, offrant de nouvelles perspectives pour la compréhension et la modélisation de phénomènes complexes.

Notre modélisation de l'atelier de production repose sur un système multi-agents qui permet de représenter de manière simple et efficace des systèmes complexes. Le MAS est composé de quatre (04) types d'agents principaux, comme illustré par la figure 2.1, et dont les interactions sont résumées dans le tableau 2.1.

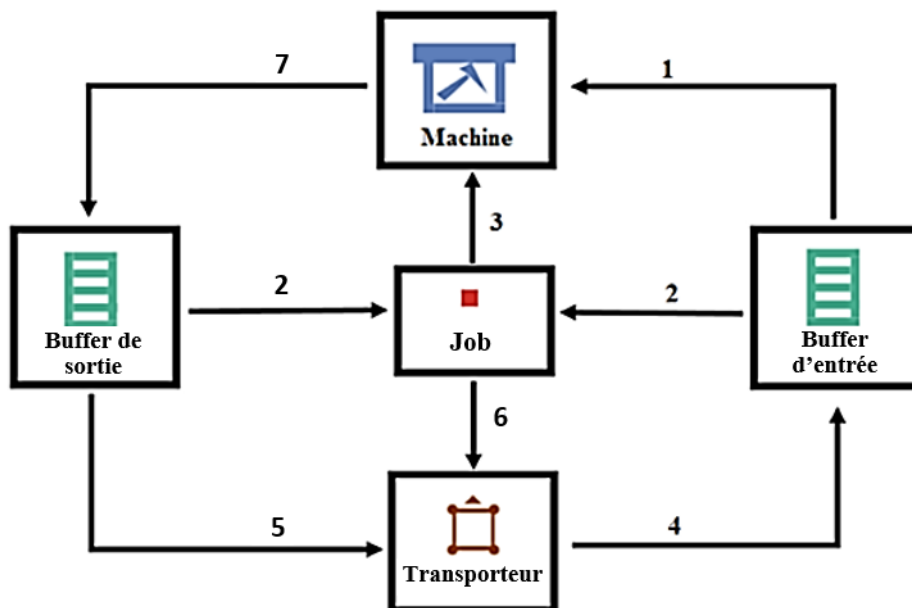


FIGURE 2.1 – Le MAS (adaptation de Xu Yiyi *et al.* [12] (2017))

2.3. MODÉLISATION DE L'ATELIER DE PRODUCTION

Interaction	Description
1	Les machines récupèrent les jobs dans les buffers d'entrée (stocks).
2	Les stocks assurent le stockage des jobs.
3	Les machines assurent le traitement des opérations des jobs.
4	Les transporteurs se chargent de livrer les jobs aux stocks.
5	Les transporteurs passent récupérer les jobs dans les stocks.
6	Les transporteurs assurent le transport des jobs dans l'atelier.
7	Les machines déposent les jobs traités dans les buffers de sortie (stocks).

TABLE 2.1 – Description des interactions entre les agents du MAS

Chaque type d'agent est décrit comme suit.

1. **Transporteurs (AGV ou AIV)** : Ces agents sont responsables du déplacement des jobs entre les différents stocks et les machines. Ils peuvent être guidés (par des rails, des marquages au sol, le filoguidage, etc.) ou se déplacer de manière autonome dans l'atelier.
2. **Stocks** : Les stocks représentent les emplacements de stockage des jobs à différents stades de production. On distingue trois types de stocks :
 - Stock d'entrée : Ce stock accueille les jobs au début du processus de production. Il peut être considéré comme le stock des matières premières.
 - Buffers : Chaque machine est équipée de deux buffers, un buffer d'entrée et un buffer de sortie. Le buffer d'entrée stocke les jobs en attente de traitement sur la machine, tandis que le buffer de sortie stocke les jobs terminés en attente d'être transférés vers le stock de sortie ou vers une autre machine.
 - Stock de sortie : Ce stock accueille les jobs finis, prêts à être expédiés ou livrés aux clients.
3. **Machines** : Les machines représentent les ressources de production de l'atelier. Elles sont responsables de la transformation des jobs en produits finis. Chaque machine possède des caractéristiques spécifiques, telles que sa capacité de traitement, son temps de traitement et sa disponibilité.
4. **Jobs** : Les jobs représentent les unités de production qui traversent l'atelier. Chaque job est caractérisé par un itinéraire de production, c'est-à-dire des opérations à subir, qui définit les machines qu'il doit visiter et l'ordre dans lequel il doit les visiter.

Cette modélisation permet de capturer les aspects essentiels du fonctionnement de l'atelier de production tout en restant flexible et adaptable. Aussi, on peut ajouter d'autres types d'agents au

2.3. MODÉLISATION DE L'ATELIER DE PRODUCTION

modèle initial sans mettre en péril son fonctionnement. Cette faculté la rend modulaire, permettant un développement progressif et incrémental. Elle peut ainsi être utilisée pour étudier différents scénarios de production, évaluer l'impact de modifications sur la performance de l'atelier et identifier des axes d'amélioration.

2.3.2 Modélisation des activités de l'atelier

L'interaction entre les agents du MAS permet de simuler le fonctionnement de l'atelier de production de manière réaliste. Les transporteurs déplacent les jobs entre les stocks et les machines, en suivant les instructions définies dans les itinéraires de production. Les machines traitent les jobs selon leurs capacités et leurs temps de traitement. Les stocks gèrent le flux des jobs et assurent la disponibilité des matières premières et des produits finis.

Les actions individuelles de chaque agent contribuent à la réalisation d'une opération concrète et essentielle au bon fonctionnement de l'atelier. Les actions collectives de tous les agents donnent lieu à un comportement émergent qui reflète le fonctionnement global de l'atelier. Ce comportement peut être résumé par la figure 2.2, qui décrit le parcours de traitement d'un job composé de trois opérations. Dans cet exemple, le job commence son parcours de traitement au stock d'entrée, où tous les jobs entrants sont stockés dans l'attente du début de leur parcours. Le traitement du job n'est pas complet tant qu'il n'a pas subi toutes ses trois opérations. Une fois cela terminé, il peut être transporté vers le stock final et considéré comme traité.

De façon générale, dans un atelier avec plusieurs jobs différents, chacun d'entre eux suivra son processus de traitement suivant l'exemple de la figure 2.2. Le processus comprend deux activités (ou tâches) principales, à savoir les activités de production (traitement sur les machines) et de transport. Ces deux activités sont indépendantes, ce qui permet aux transporteurs de déplacer certains jobs pendant que les machines en traitent d'autres. Ainsi, la simulation du modèle nécessite de déterminer l'affectation des jobs aux machines, l'affectation des jobs aux transporteurs et le séquençage de l'exécution des tâches. Ces informations ne sont fournies que par l'ordonnancement des tâches de l'atelier. Par conséquent, la simulation du modèle, c'est-à-dire les interactions entre les agents du MAS, permet d'exécuter l'ordonnancement des tâches de l'atelier. Il existe donc une interconnexion entre le MAS et le modèle de la simulation, comme illustré sur la figure 2.3

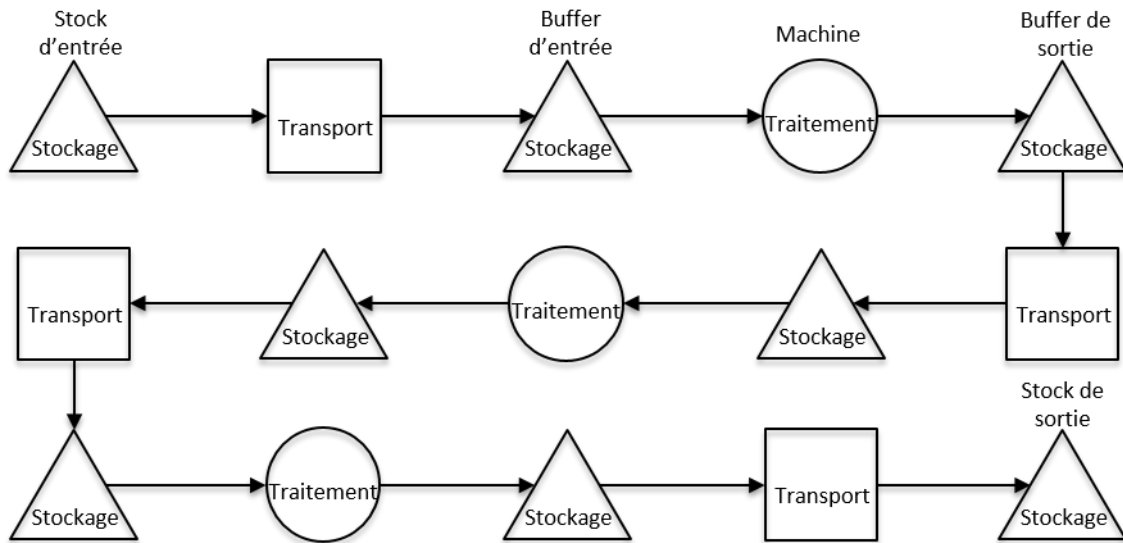


FIGURE 2.2 – Processus de traitement d'un job avec trois opérations

2.3.3 Modélisation de l'atelier

Notre représentation de l'atelier est basée sur les travaux de Ü. Bilge et G. Ulusoy [1] qui ont proposé des instances de benchmarking bien connues dans la littérature. Ces instances décrivent quatre (04) configurations distinctes d'un atelier de production, communément appelées « *layouts* », chacune composée d'une station de chargement et de déchargement (L/U) et de quatre machines, comme on peut le voir sur la figure 2.4. La station L/U sert de point de stockage pour les jobs brutes avant leur traitement et les jobs traités après avoir terminé leur processus de traitement. Les quatre machines, elles, effectuent les opérations de traitement des jobs.

Dans leur proposition, les tâches de transport sont assurées par deux AGV identiques qui empruntent les chemins les plus rapides pour se déplacer, tout en respectant les orientations imposées par les layouts (voir figure 2.4). Les temps de trajet dans les différents layouts sont renseignés dans le tableau 2.2. En plus des configurations d'atelier, les instances de référence incluent dix (10) ensembles de jobs distincts, chacun comprenant entre cinq (05) et huit (08) jobs. Comme on peut le voir dans le tableau A.1, chaque job se compose de plusieurs opérations à effectuer sur les machines correspondantes, chacune ayant un temps de traitement associé. Les instances de test sont identifiées par la notation « $EX\alpha\beta$ », où α représente l'identifiant de l'ensemble de job et β représente l'identifiant du layout. Par exemple $EX11$ désigne un test réalisé sur le layout n°1 avec l'ensemble de jobs n°1. La

2.3. MODÉLISATION DE L'ATELIER DE PRODUCTION

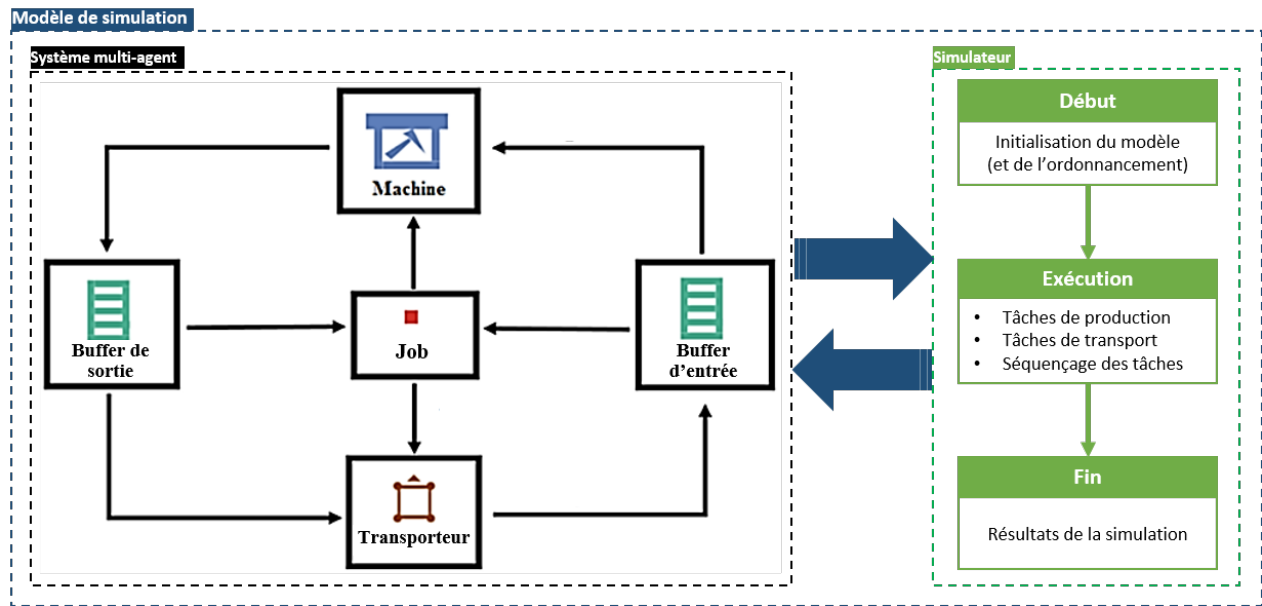


FIGURE 2.3 – Modélisation de la simulation

diversité des layouts et des ensembles de jobs permet une analyse de l'impact des différents paramètres sur les performances de l'ordonnancement. Par ailleurs, les expériences basées sur ces instances permettent d'évaluer les méthodes d'ordonnancement et de les comparer avec plusieurs autres travaux réalisés sur les mêmes instances.

Nous concernant, nous avons supposé que les temps de transport et de traitement sont exprimés en secondes (s). De plus, nous avons modélisé chaque layout sous la forme d'un graphe orienté connexe. Le graphe est orienté simple quand les tâches de transport sont réalisées par les AGV. Cependant, il devient orienté symétrique (ou non orienté) quand les tâches sont réalisées par les AIV. Dans cette modélisation, chaque point d'intérêt, tels que les machines, la station L/U, les coins et les intersections, est représenté par un nœud du graphe. Les arcs qui relient ces nœuds symbolisent quant à elles les couloirs de connexion qui permettent aux robots de se déplacer dans l'atelier. Lorsqu'une tâche est attribuée à un robot, celui-ci entame un processus de planification et de navigation pour accomplir sa mission. La stratégie adoptée dépend du type de robot utilisé : pour les AGV, l'itinéraire, défini comme le chemin spécifique dans le graphe reliant leur position initiale à leur destination finale, est préalablement établi. Ces robots suivent scrupuleusement les instructions préprogrammées, se déplaçant de manière déterministe le long du chemin prédéfini. En revanche, les AIV adoptent une approche plus flexible et adaptative. Face à une tâche assignée, ils déterminent leur itinéraire en temps

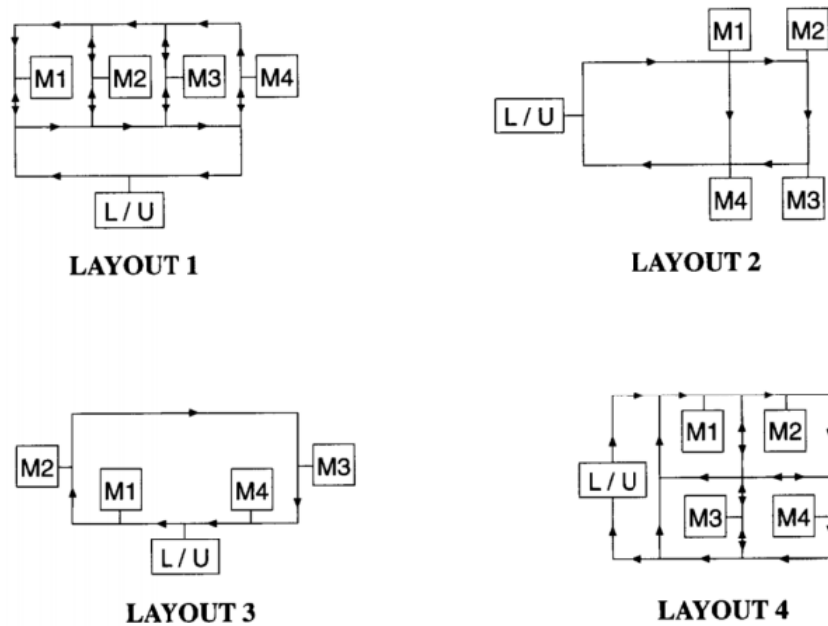


FIGURE 2.4 – Les 4 différents layouts proposés par Ü. Bilge et G. Ulusoy [1]

réel en utilisant un algorithme de navigation inspiré de l'algorithme de Dijkstra. Cet algorithme leur permet de calculer le chemin le plus rapide entre leur position actuelle et leur destination, en tenant compte de l'état dynamique de l'atelier.

2.4 Développement du simulateur

Cette section explore en détail le simulateur, commençant par présenter le cadre sur lequel il est construit, puis décrivant les différents éléments le composant.

2.4.1 Description du framework

Notre simulateur a été développé avec Netlogo [13], qui est à la fois un langage de programmation multi-agents et un environnement de modélisation permettant d'explorer des phénomènes naturels et sociaux complexes [238]. NetLogo est spécialement conçu pour modéliser des systèmes complexes composés de multiples agents. Ces agents, pouvant atteindre des centaines ou des milliers en nombre, interagissent de manière autonome et simultanée, permettant aux utilisateurs d'observer comment les actions individuelles façonnent des comportements émergents à grande échelle. Il ne se contente pas d'exécuter seulement des simulations, mais favorise aussi une exploration interactive. Les utilisateurs

2.4. DÉVELOPPEMENT DU SIMULATEUR

	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

(a) Layout 1

	L/U	M1	M2	M3	M4
L/U	0	2	4	10	12
M1	12	0	2	8	10
M2	10	12	0	6	8
M3	4	6	8	0	2
M4	2	4	6	12	0

(c) Layout 3

	L/U	M1	M2	M3	M4
L/U	0	4	6	8	6
M1	6	0	2	4	2
M2	8	12	0	2	4
M3	6	10	12	0	2
M4	4	8	10	12	0

(b) Layout 2

	L/U	M1	M2	M3	M4
L/U	0	4	8	10	14
M1	18	0	4	6	10
M2	20	14	0	8	6
M3	12	8	6	0	6
M4	14	14	12	6	0

(d) Layout 4

TABLE 2.2 – Temps de trajet tels que proposés par Ü. Bilge et G. Ulusoy [1]

peuvent « jouer » avec les simulations, ajustant les conditions initiales et les paramètres pour observer les réactions du système. Cette approche simple encourage la découverte et la compréhension approfondie des mécanismes sous-jacents.

Comme le souligne [239], NetLogo privilégie la simplicité d'utilisation, reflétant ses origines en tant qu'outil pédagogique. Son langage de programmation se caractérise par des structures et des primitives efficaces qui minimisent la complexité du codage, rendant l'apprentissage et l'utilisation accessibles à un large public. De plus, une documentation complète et facilement accessible accompagne les utilisateurs tout au long de leur exploration. Grâce à sa combinaison unique de puissance de modélisation, d'interface intuitive et de facilité d'utilisation, NetLogo s'impose comme un des meilleurs outils pour explorer et comprendre les systèmes complexes, qu'on soit novice, amateur ou expert.



FIGURE 2.5 – Le logo de Netlogo [13]

NetLogo offre une interface intuitive divisée en trois onglets principaux, chacun jouant un rôle dans le processus de création et d'exploration de modèles.

1. L'onglet « **Interface** » : cet onglet constitue le principal endroit d'interaction avec la simulation. C'est ici qu'on observe le déroulement de la simulation en temps réel, qu'on manipule les éléments du modèle et qu'on recueille des données pour mener une analyse des résultats. Cet espace permet de visualiser les comportements émergents du système modélisé.
2. L'onglet « **Info** » : cet onglet sert de dépôt d'informations complémentaires sur le programme de simulation. On peut y inclure une description détaillée du phénomène simulé, expliquer le fonctionnement du modèle ou fournir des instructions d'utilisation. Ces informations contextuelles enrichissent le programme et facilitent son utilisation.
3. L'onglet « **Code** » : cet onglet renferme le cœur de la simulation, c'est-à-dire le code du modèle. C'est ici qu'on écrit les instructions qui définissent le comportement des agents, les règles d'interaction et la dynamique globale du système.

NetLogo utilise des « *turtles* » comme agents individuels, ces agents étant regroupés en catégories distinctes appelées « *breeds* ». Cette organisation permet de gérer efficacement les interactions entre les agents et de définir des comportements spécifiques pour chaque groupe. Cela offre une grande flexibilité dans la définition des paramètres de simulation. On peut par exemple définir des paramètres globaux qui s'appliquent à l'ensemble du modèle, ou des paramètres spécifiques à chaque *breed*. Cette granularité permet d'affiner le modèle.

Le démarrage de la simulation se fait grâce au bouton « *setup* ». Il initialise ou réinitialise le modèle et son environnement, préparant les agents pour le déroulement de la simulation. Une fois que tout est en place, la simulation s'exécute en cliquant sur le bouton « *go* ». Ce bouton propose deux modes :

- « *one-time* » (une fois) : La simulation se déroule pendant une seule itération (action) puis s'arrête.
- « *forever* » (continue) : La simulation se poursuit indéfiniment jusqu'à ce qu'une condition d'arrêt soit programmée dans le code ou qu'on appuie à nouveau sur le bouton *go*. Ce mode est utile pour explorer des phénomènes à long terme et étudier l'évolution du système sur plusieurs itérations.

Le temps de simulation est mesuré en « *ticks* ». Chaque *tick* représente un pas de simulation, une unité de temps discrète pendant laquelle les agents interagissent et que le modèle évolue. En ajustant

la vitesse de simulation et en observant le nombre de *ticks*, on peut analyser le comportement du système à différentes échelles temporelles.

L'environnement de simulation, appelé « *world* », se matérialise sous la forme d'une grille 2D, offrant une fenêtre sur le déroulement du programme en temps réel. Cette grille est composée de carrés élémentaires, appelés « *patches* », sur lesquels les agents du modèle bougent et interagissent. Par ailleurs, il faut préciser que les *patches* sont des agents déjà présent dans les programmes NetLogo. Chaque *patch* possède un emplacement unique, identifié par un système de coordonnées à deux dimensions : x et y (respectivement « *pxcor* » et « *pycor* » dans le langage NetLogo). Ce système de coordonnées permet de déterminer quels agents se trouvent sur des *patches* spécifiques, et inversement. En d'autres termes, on peut savoir quel *turtle* occupe actuellement un *patch* donné, et vice versa. La grille 2D joue un rôle crucial dans l'exploration intuitive de la simulation. Elle permet de visualiser en temps réel les positions des agents, leurs interactions, et l'évolution globale du système. En outre, en observant la grille, on peut identifier les comportements émergents, analyser les comportements individuels et collectifs, et déceler les mécanismes sous-jacents qui régissent le modèle.

2.4.2 Implémentation du modèle de simulation

Afin de reproduire au mieux le modèle de l'atelier présenté dans la section 2.3.3, nous avons considéré les hypothèses suivantes :

- Les transporteurs, les machines, les stocks et les jobs représentent les principales *breeds* de notre modèle. Chaque *breed* (ou type d'agent) est constituée de plusieurs agents qui sont indépendants les uns des autres.
- Nous avons établi une correspondance entre le temps en seconde de notre modèle et le temps en ticks de la simulation. Pour cela, nous supposons que 20 *ticks* équivalent à 1 seconde. Cela signifie que chaque *tick* représente 0,05 seconde, permettant une simulation précise des durées d'opérations et de déplacements.
- De même, Nous avons établi une correspondance pour les distances. On considère que 20 *patches* correspondent à 1 mètre. Cette conversion permet de visualiser les distances parcourues par les transporteurs.

Le programme principale de la simulation est représenté par le logigramme de la figure 2.6. Ce diagramme illustre les étapes principales du programme. La première étape consiste à définir les

différents *breeds* du modèle, c'est-à-dire les différents types d'agents. Ensuite, il est nécessaire de définir les variables globales qui s'appliquent à l'ensemble du modèle simulé. Elles peuvent représenter des paramètres de contrôle ou d'autres éléments qui influencent le comportement de tous les *breeds*. En plus des variables globales, il est possible de définir les variables propres à chaque *breed*. Ces variables représentent les caractéristiques individuelles des entités au sein d'une *breed*. Elles peuvent inclure des paramètres physiques ou d'autres attributs pertinents pour la simulation. L'étape d'après consiste à initialiser le modèle. Cela implique de définir les valeurs initiales des variables globales et des variables propres à chaque *breed*. Cette étape permet de mettre en place l'état initial du modèle simulé avant de le lancer. Une fois que le modèle a été initialisé, on peut maintenant l'exécuter. Cette étape permet de simuler à proprement dit le comportement du système. À chaque itération de la simulation, il est nécessaire de vérifier si la simulation est terminée. La fin de la simulation peut être déterminée par une condition spécifique, par exemple l'atteinte d'un certain nombre d'itérations ou la satisfaction d'un critère particulier. Si la simulation n'est pas terminée, le processus d'exécution se poursuit. Sinon, le processus s'arrête et les résultats de la simulation peuvent être recueillis et analysés.

2.4.2.1 Définition des *breeds* (types d'agent)

Les *breeds* sont un concept fondamental dans NetLogo qui permet de regrouper des agents partageant des caractéristiques et des comportements communs. La définition d'un *breed* se fait au tout début du programme, à l'aide du mot-clé `breed`. Il faut absolument définir tous les *breeds* avant la définition d'une autre quelconque procédure. `breed` prend deux arguments :

- **Le nom du *breed*** : Il s'agit du nom qui sera utilisé pour identifier le *breed* dans le code.
- **Le nom d'un agent du *breed*** : Il s'agit du nom qui sera utilisé pour désigner un agent individuel du *breed*.

Dans notre programme, nous avons définis les *breeds* comme suit :

```
breed [machines machine]
breed [stocks stock]
breed [transporters transporter]
breed [jobs job]
```

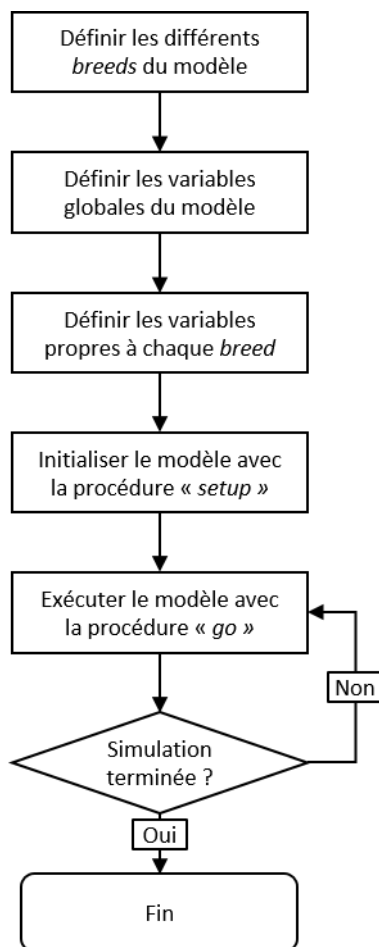


FIGURE 2.6 – Logigramme du programme principal de la simulation

2.4.2.2 Définition des variables

NetLogo permet de définir trois (03) types de variables : les variables globales qui sont accessibles partout et par n'importe qui, les variables propres aux *breeds* qui ne sont accessibles que par les agents de la *breed*, et enfin les variables locales qui sont accessibles uniquement à la procédure (ou fonction) dans laquelle elles sont définies. La définition des variables globales et des variables des *breeds* doit se faire au début du programme, avant l'écriture de toute procédure (ou fonction), tandis que les variables locales sont définies directement dans la procédure (ou fonction) qui les utilisent.

La définition des variables globales se fait à l'aide du mot-clé `globals`, suivi d'une liste de noms de variables en argument `globals[var1 var2 ... varN]`. En revanche, la définition des variables d'une *breed* se fait grâce à la construction d'un mot-clé composé du nom de la *breed*, auquel on ajoute

2.4. DÉVELOPPEMENT DU SIMULATEUR

le suffixe `-own`. Par exemple, pour définir les variables de la breed `jobs`, on procède comme suit : `jobs-own[var1 var2 ... varN]`. Les variables locales sont quant à elles définies grâce au mot-clé `let` suivi du nom de la variable et de sa valeur d'initialisation. Par exemple, « `let i 0` » permet de créer la variable `i`, initialisée à 0. La figure 2.7 montre les principales variables (globales et propres aux *breeds*) qui ont été définies dans notre programme.

globals	transporters-own	stocks-own
<ul style="list-style-type: none">• stock-limits• job-lists• job-number• schedule• tasks-completed• instance• layout• robot• transporter-number• simulation-type• number-of-simulation	<ul style="list-style-type: none">• id• working-status• orders-list• current-order• path• current-location• mileage• moving-loaded?	<ul style="list-style-type: none">• id• machine-number• jobs-in• waiting-time• jobs-on-the-way
patches-own	jobs-own	machines-own
<ul style="list-style-type: none">• id• nature	<ul style="list-style-type: none">• id• type• operations-list• status• location• queue-time	<ul style="list-style-type: none">• id• input-stock• output-stock• working-status• occupied-by?• working-time• idle-time• operations• operation-times• processing-list

FIGURE 2.7 – Les principales variables du programme de simulation

2.4.2.3 Initialisation du modèle

Comme évoqué précédemment, l'initialisation du modèle se fait grâce au bouton `setup` qui déclenche la procédure du même nom qui se trouve dans le code. La syntaxe de base pour déclarer une procédure dans NetLogo est la suivante :

```
to nom-procedure [arguments]
  corps-de-la-procedure
end
```

- `to` : Ce mot-clé indique le début de la déclaration de la procédure.
- `nom-procedure` : Il s'agit du nom de la procédure. Ce nom doit être unique dans le modèle.
- `[arguments]` : Cette partie est facultative. Elle permet de définir des arguments pour la procédure. Les arguments sont des valeurs qui peuvent être passées à la procédure lors de son appel.
- `corps-de-la-procedure` : Il s'agit du contenu de la procédure. C'est le code qui définit ce que la procédure doit faire.
- `end` : Ce mot-clé marque la fin de la déclaration de la procédure.

Dans notre cas, la procédure `setup` comprend plusieurs autres procédures qui permettent chacune d'initialiser une partie du programme. `file-close-all`, `clear-all`, `reset-ticks`, et `reset-timer` sont des primitives intégrées :

- `file-close-all` : permet de fermer tous les fichiers ouverts (ou pas) qui sont utilisés par le programme.
- `clear-all` : permet de remettre à zéro tout le programme, y compris l'environnement de simulation et les variables.
- `reset-ticks` : remet à zéro le compteur de *ticks*.
- `reset-timer` : remet à zéro le compteur du temps de calcul (temps CPU). Il s'exprime en seconde (s).

```
to setup
  file-close-all
  clear-all
  setup-patch
  setup-globals
  setup-machines
  setup-stocks
  setup-transporters
  setup-jobs
  reset-ticks
  reset-timer
end
```

2.4.2.3.1 setup-patch : Cette procédure permet d'implémenter la configuration de l'atelier, en fonction du choix de layout. Nous avons adopté la démarche suivante pour le codage du layout :

1. Créer un fichier Excel dont chaque cellule représentera un *patch* de l'environnement de simulation.

Les cellules sont remplies comme suit :

- les cellules qui serviront d'emplacement pour la station L/U ont été remplies avec le chiffre 3;
- celles qui serviront d'emplacement pour les machines ont été remplies avec le chiffre 2;
- les cellules représentant les couloirs de déplacement des transporteurs ont été remplies avec le chiffre 1;
- toutes les autres cellules ont été remplies avec le chiffre 0.

L'ensemble des cellules forme en tout une matrice de taille : 124×105 pour le layout 1, 109×47 pour le layout 2, 107×45 pour le layout 3, et 165×120 pour le layout 4.

2. Après le remplissage du fichier Excel, il a été exporté en un fichier texte (.txt) et c'est ce dernier qui été importé dans le programme NetLogo.
3. Un algorithme permet d'implémenter le layout. Il est représenté par le pseudo-code de l'algorithme 1

Algorithme 1 Pseudo-code de la création d'un layout

Requiert : fichier texte valide

```
1: procédure SETUP-PATCH :
2:   ouvrir "fichier.txt"
3:   tmp ← lire "fichier.txt"
4:   Pour x allant de 0 à longueur(tmp), faire :
5:     Pour y allant de 0 à longueur(tmp[x]), faire :
6:       créer patch(x, y)
7:       patch(x, y).nature ← tmp[x][y]
8:       Si nature ← 0 alors
9:         colorer patch(x, y) en gris
10:      Sinon, si nature ← 1
11:        colorer patch(x, y) en blanc
12:      Sinon, si nature ← 2
13:        colorer patch(x, y) en bleu
14:      Sinon
15:        colorer patch(x, y) en rouge
16:      Fin si
17:    Fin pour
18:  Fin pour
19:  fermer "fichier.txt"
20: Fin procédure
```

2.4.2.3.2 setup-globals : Cette procédure de définir les valeurs initiales de toutes les variables globales du modèle. C'est donc ici qu'on choisit, entre autres, le nombre de transporteurs qui vont assurer les tâches de transport, ou encore la capacité de la station L/U et des buffers des machines. NetLogo offre également un avantage considérable en matière de configuration des paramètres grâce à l'interface graphique. Avec des boutons, des listes déroulantes, des commutateurs et autres éléments interactifs, on peut directement modifier les valeurs des paramètres dans l'onglet *Interface* (voir section 2.4.3). Cette approche intuitive permet d'explorer plus facilement différentes configurations.

2.4.2.3.3 setup-machines : Cette procédure crée quatre (04) machines (de machine 0 à machine 3), définit leur apparence de base et leur état initial, puis assigne des opérations et des temps de traitement spécifiques à chaque machine sur la base de son identifiant unique et de l'ensemble de jobs (la variable globale *instance*) définit dans *setup-globals*. Elle définit également la position de chaque machine en fonction du layout choisit pour la simulation.

2.4.2.3.4 setup-stocks : Tout comme la procédure précédente, celle-ci crée tous les stocks qui serviront dans la simulation. Elle en crée 10 au total : le stock d'entrée, le stock de sortie et deux (02) buffers pour chaque machine. Les stocks d'entrée et de sortie représentent la station L/U. À ce titre, ils sont différents des buffers en taille, en capacité et en fonction. Par exemple, un buffer appartient nécessairement à une machine, ce qui n'est pas le cas pour les stocks de la station L/U. Leur emplacement dépend également du layout de la simulation.

2.4.2.3.5 setup-transporters : Cette procédure crée autant de transporteurs que définit dans *setup-globals*. Ces transporteurs sont tous identiques mais avec des *id* différents. La procédure initialise leurs variables et les tient prêts pour le début de la simulation. Les transporteurs sont tous initialement situés près de la station L/U, lieu à partir duquel commencent toutes les premières tâches de transport.

2.4.2.3.6 setup-jobs : Cette procédure commence par créer le nombre de jobs définit par l'instance de simulation. Chaque job étant différent des autres, la procédure initialise les opérations et le parcours de traitement en fonction du type de chacun. Ils sont tous initialement rangés dans le stock d'entrée de la station L/U et restent en attente de leur transport afin de subir la première opération.

2.4.3 Interface du simulateur

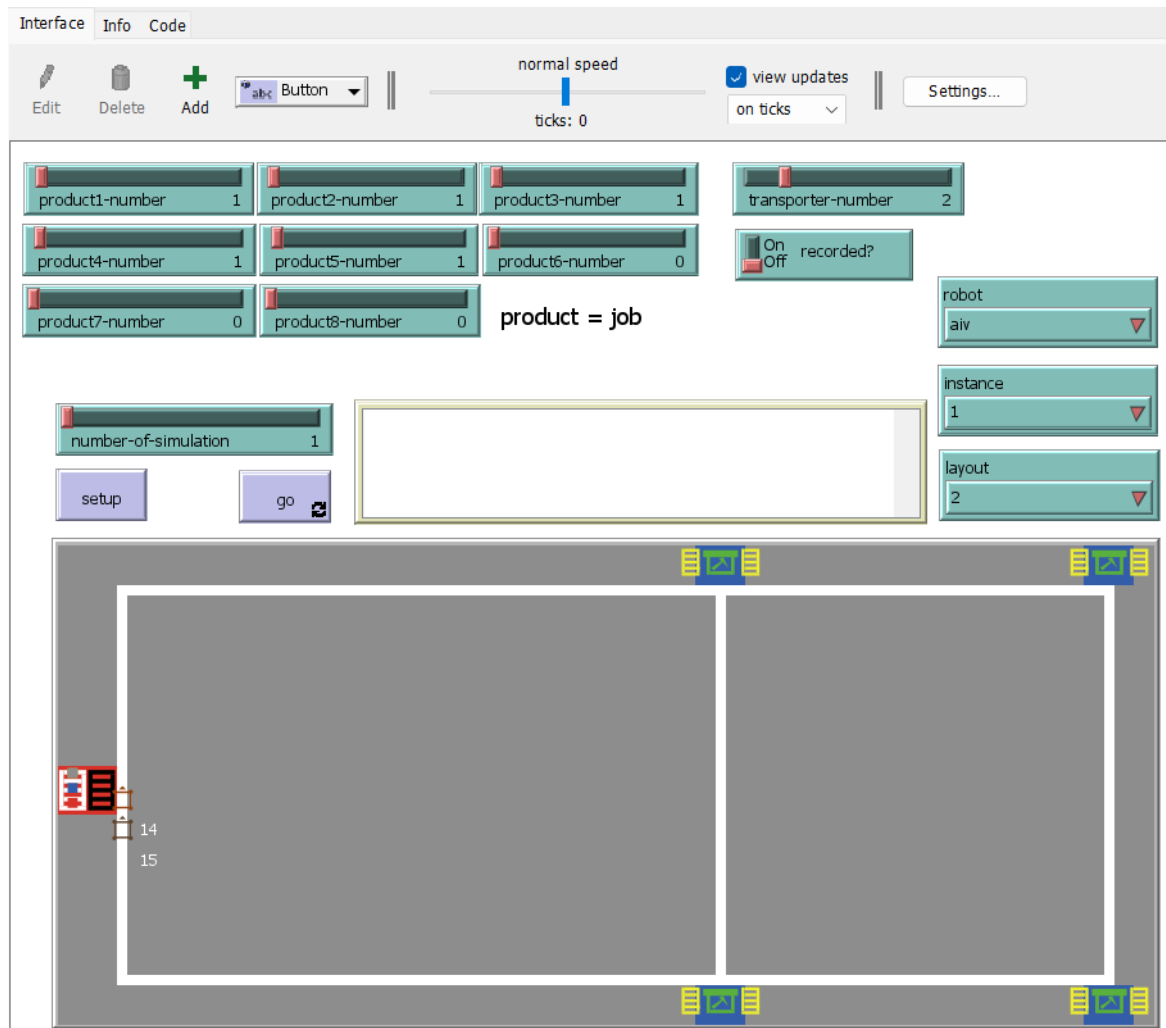


FIGURE 2.8 – Onglet *Interface* du simulateur

Comme l'illustre la Figure 2.8, l'interface du simulateur est composée de plusieurs éléments. Chacun de ces éléments joue un rôle spécifique en facilitant l'interaction avec la simulation :

- **Les boutons** : l'interface offre un ensemble de boutons qui permettent de contrôler le déroulement de la simulation, d'interagir avec elle et de la rendre beaucoup plus intuitive. Dans notre cas, l'interface comprend deux boutons qui sont *setup* et *go*. Le bouton *setup* permet de déclencher la procédure du même nom (voir section 2.4.2.3) qui conduit à une (ré-)initialisation complète de l'environnement. Cela implique la remise à zéro des variables, le repositionnement des agents et la création des conditions initiales nécessaires pour le bon déroulement de la

simulation. Cette action est comparable à la réinitialisation d'un jeu vidéo, permettant de recommencer une nouvelle partie avec un environnement vierge. Le bouton *go*, quant à lui, agit comme une commande de lecture/pause pour la simulation. En appuyant une première fois sur ce bouton, on lance l'exécution du modèle en déclenchant la procédure *go* qui se trouve dans le code. Les agents commencent à interagir, les événements se déroulent et la simulation prend vie. Un deuxième clic sur le bouton *go* met la simulation en pause, permettant d'observer l'état actuel du système et d'analyser les résultats intermédiaires. Un troisième clic reprend l'exécution de la simulation depuis le point où elle a été mise en pause.

- **Les curseurs** : ils offrent le moyen de modifier les valeurs numériques de certaines variables sans avoir à le faire dans le code. En manipulant ces curseurs, on peut facilement explorer différentes configurations sans toucher au code. Ce qui rend le programme beaucoup plus intuitif et plus simple d'utilisation. Dans notre contexte, les curseurs sont utiles dans la configuration de la simulation et le suivi de son déroulement. Ils permettent de définir des valeurs numériques pour des paramètres clés tels que le nombre de jobs à traiter ou le nombre de transporteurs disponibles. Un curseur particulier (*number-of-simulation*) sert de compteur pour suivre le nombre de fois que la simulation a été exécutée.
- **Les sélecteurs** : ils offrent un moyen pratique de sélectionner une valeur spécifique pour une variable parmi un ensemble prédéfini. Cette fonctionnalité intuitive permet de changer les configurations du modèle en toute simplicité et d'explorer différentes options sans avoir à saisir manuellement les valeurs. Ils prennent en charge une large gamme de types de données, permettant de choisir parmi des valeurs numériques, des valeurs booléennes (vrai/faux) ou des chaînes de caractères. Cette flexibilité s'adapte à de nombreux cas d'utilisation et permet de configurer le modèle selon le besoin. Dans notre cas, les sélecteurs permettent de choisir le layout de la simulation, définissant les disposition des machines et de la station L/U, l'instance de jobs à traiter et le type de transporteurs (AGV ou AIV).
- **Les commutateurs** : Les commutateurs de NetLogo permettent de contrôler l'état d'une variable en la basculant entre deux options prédéfinies. Ces options peuvent être binaires (vrai/faux, activé/désactivé) ou représenter des choix plus spécifiques (gauche/droite, haut/bas, etc.). En manipulant ces commutateurs, on peut activer ou désactiver des fonctionnalités spécifiques du programme, permettant d'explorer différents scénarios plus simplement. Le commutateur

de notre simulateur (*recorded ?*) permet d'activer l'enregistrement des détails de la simulation.

- **Le moniteur** (espace d'affichage) : Le moniteur permet d'afficher des informations en temps réel sur le déroulement de la simulation. Ces informations donnent un aperçu du comportement du modèle et permettent d'évaluer ses performances. Dans le cas de ce simulateur, le moniteur joue un rôle important dans le suivi de l'efficacité du système. Il affiche le temps de réalisation (en ticks) de chaque job ainsi que le temps de réalisation de tous les jobs (appelé *makespan* ou C_{max}), permettant de mesurer la durée des opérations et les performances de l'ordonnancement.
- **L'environnement de simulation** : c'est une fenêtre immersive sur l'exécution du programme. Cette interface graphique permet d'observer en temps réel le comportement des agents et de voir les mécanismes du système. Dans notre simulation, l'environnement de simulation prend la forme d'un atelier de production. Il montre l'agencement complet de l'atelier, incluant les différents agents qui y évoluent. Pour faciliter la visualisation et la compréhension de l'environnement, le sol de l'atelier est représenté par des *patches* colorés différemment : les *patches* blancs représentent les couloirs de déplacement empruntés par les transporteurs, tandis que les *patches* gris représentent des zones occupées où il est impossible pour les transporteurs de se déplacer.

2.4.4 Exécution de la simulation

Le simulateur a été conçu pour offrir une flexibilité dans l'exploration des performances de l'atelier de production. Il prend en charge deux types de transporteurs : les AGV et les AIV, permettant ainsi de comparer leurs performances. Il propose également deux modes d'ordonnancement distincts pour gérer les tâches de production : les ordonnancements prévisionnels où toutes les tâches et leur séquence d'exécution sont connues à l'avance, et les ordonnancements dynamiques dans lesquels l'exécution des tâches est déterminée en temps réel en fonction de l'état du système. Pour une simulation encore plus réaliste, le simulateur offre la possibilité d'activer la contrainte d'évitement des collisions. En activant cette option, les transporteurs sont obligés de prendre en compte la présence des autres agents et d'adapter leurs trajectoires pour éviter les collisions.

2.4.4.1 Choix du type de transporteurs

Le simulateur permet de spécifier le type de transporteurs à utiliser pour les tâches de transport. Cette sélection s'effectue via le sélecteur *robot* comme le montre la Figure 2.9 :

- si *robot = agv* : les tâches de transport sont effectuées par des AGV. Dans ce cas, les robots se déplacent le long des couloirs en suivant des chemins prédéfinis et imposés par le layout de l'atelier.
- si *robot = aiv* : les tâches de transport sont effectuées par des AIV. Contrairement aux AGV, les AIV ne sont pas contraints de suivre des chemins prédéterminés. Ils ont l'intelligence nécessaire pour naviguer par eux-mêmes et trouver l'itinéraire le plus approprié. Cette autonomie se traduit par une plus grande flexibilité et agilité, leur permettant de manœuvrer autour d'obstacles fixes et dynamiques.

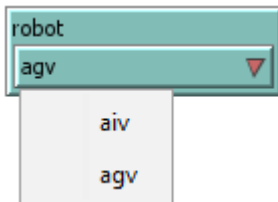


FIGURE 2.9 – Sélecteur du type de transporteurs

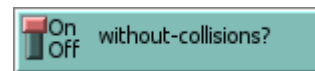


FIGURE 2.10 – Commutateur d'activation de la contrainte d'évitement de collisions

Dans les deux (02) cas, les transporteurs doivent uniquement se déplacer le long des couloirs, c'est-à-dire sur les *patches* blancs.

2.4.4.2 Activation de la contrainte d'évitement de collisions

La sécurité et l'efficacité d'un système de production reposent en grande partie sur la capacité des robots à naviguer en toute fluidité sans se heurter les uns aux autres. Pour simuler cet aspect essentiel, le simulateur met à disposition un commutateur nommé "*without-collisions?*". Comme illustré sur la figure 2.10, l'activation de ce commutateur introduit la contrainte d'évitement de collisions dans la simulation.

Deux méthodes principales permettent de gérer les collisions dans un système multi-robots, comme décrit par De Ryck *et al.* [240] :

1. La planification centralisée des mouvements : Cette approche traditionnelle, largement utilisée

dans les systèmes industriels, consiste à définir une trajectoire sans collision ni blocage pour l'ensemble des robots à l'aide d'un contrôleur central. Ce contrôleur omniscient orchestre les déplacements de tous les robots, garantissant une navigation fluide et coordonnée.

2. La planification décentralisée des mouvements : Cette approche plus flexible permet aux robots de prendre des décisions individuelles pour éviter les collisions en fonction de leur connaissance de l'environnement local. Les robots communiquent entre eux pour partager des informations sur leurs positions et leurs intentions, leur permettant d'adapter leurs trajectoires en temps réel et d'éviter les obstacles dynamiques. On peut regrouper les techniques de cette approche décrites dans [240] en deux variantes :

- l'approche basée sur les champs potentiels : Les robots calculent des champs potentiels attractifs vers leurs destinations et répulsifs vis-à-vis des autres robots et des obstacles. Ils adaptent ensuite leurs mouvements en suivant les gradients de ces champs potentiels ;
- l'approche basée sur les règles de navigation : Les robots suivent un ensemble de règles prédéfinies pour gérer les interactions et éviter les collisions. Ces règles peuvent inclure des règles de priorité, des distances de sécurité et des stratégies de dépassement.

Pour notre simulateur, nous avons opté pour une approche décentralisée car elle est plus appropriée pour les systèmes multi-agents et offre l'avantage d'être plus robuste et moins gourmande en calcul par rapport à l'approche centralisée. Par ailleurs, parmi les méthodes décentralisées d'évitement de collisions décrites dans [240], l'une des plus simples et efficaces consiste à doter chaque robot d'un capteur de sécurité embarqué. Ce capteur surveille en permanence l'environnement devant le robot et détecte la présence d'obstacles ou de personnes. Le robot définit une zone de sécurité virtuelle autour de lui, délimitant la distance minimale autorisée entre le robot et les objets environnants. Lorsqu'un obstacle ou une personne pénètre dans cette zone de sécurité, le robot réduit sa vitesse progressivement jusqu'à un potentiel arrêt total. Une fois le chemin libéré, le robot reprend son mouvement normal en maintenant une vitesse et une trajectoire appropriées.

Nous avons repris ce même principe en définissant deux rayons de sécurité pour tous les transporteurs comme sur les figures 2.11 et 2.12 :

- **un rayon de détection** : il s'agit d'un grand rayon de cinq (05) *patches* autour du robot qui lui permet de détecter tous les obstacles qui se trouvent à cette portée. Lorsqu'il détecte un obstacle, le robot réduit progressivement sa vitesse jusqu'à atteindre la moitié de sa vitesse de

croisière.

- **un rayon critique** : ce rayon, plus petit que le précédent, peut être essentiellement résumé en un champ de vision de 180° à trois (03) *patches* devant le robot. Une fois qu'un obstacle rentre dans ce rayon, le robot réduit davantage sa vitesse jusqu'à s'arrêter complètement.

Toutefois, le mécanisme d'évitement de collisions connaît des particularités en fonction du type de robots utilisé pour le transport.

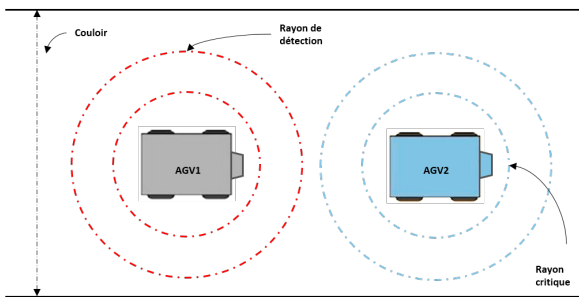


FIGURE 2.11 – Évitement de collisions entre deux AGV

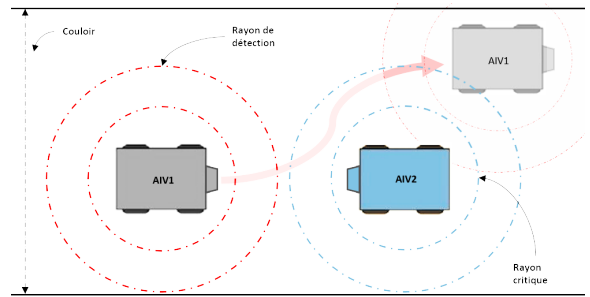


FIGURE 2.12 – Évitement de collisions entre deux AIV

Lorsqu'il s'agit des AGV, le mécanisme fonctionne comme décrit précédemment, avec deux rayons de sécurité et des actions de réduction de vitesse ou d'arrêt en cas d'obstacles détectés. Ce fonctionnement est illustré sur la figure 2.11. Néanmoins, pour éviter les collisions aux intersections, le mécanisme intègre un système de réservation de portions de chemin. Le premier AGV à atteindre une intersection réserve la portion de chemin nécessaire pour traverser en toute sécurité. Les AGV suivants doivent patienter à des *check points* (points de contrôle) jusqu'à ce que le chemin soit libre. Un check point est comme une tour de contrôle précédant la prochaine intersection, où l'AGV vérifie l'état de la portion suivante du chemin qui sera utilisée pour son déplacement. Les figures 2.13 et 2.14 illustrent ce mécanisme.

En revanche, quand le transport est assuré par des AIV, le mécanisme intègre plutôt un système de gestion des priorités. En effet, contrairement aux AGV qui se déplacent dans un sens imposé par le layout, les AIV peuvent emprunter les couloirs dans les deux sens. En cas de rencontre frontale, le système de gestion des priorités détermine quel AIV doit céder le passage. Le critère de priorité peut être la distance à la destination, le niveau de batterie, l'importance du job transporté, etc. Lorsqu'ils sont dans le rayon de détection, les deux AIV réduisent leur vitesse jusqu'à atteindre la moitié de leur vitesse de croisière. Puis, une fois dans le rayon critique, l'un des AIV s'arrête, tandis que celui qui est

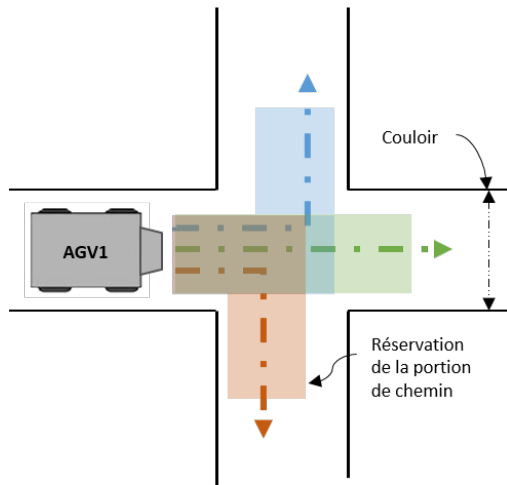


FIGURE 2.13 – Système de réservation de portions de chemin

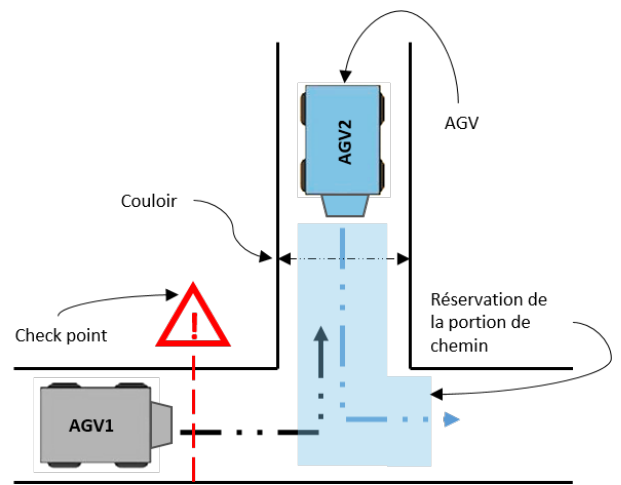


FIGURE 2.14 – Mécanisme d'évitement de collisions aux intersections

prioritaire entame une manœuvre de contournement, tout en réduisant davantage sa vitesse. Quand la voie est libre, les deux robots accélèrent progressivement pour retrouver leur vitesse de croisière (voir figure 2.12). Dans notre cas, on estime la vitesse de croisière des transporteurs (AGV et AIV) à un mètre par seconde (1m/s).

2.4.4.3 Choix du type d'ordonnancement

Afin de répondre aux besoins d'analyse et d'optimisation des systèmes de production, le simulateur a été conçu pour exécuter deux types d'ordonnancement :

- **Ordonnancement prévisionnel** : dans ce mode, le simulateur récupère en entrée les résultats d'un ordonnancement préétabli, définissant l'affectation des tâches aux différents transporteurs et machines. Il agit alors comme un outil d'exécution, reproduisant fidèlement le déroulement de la production selon ce plan prédéterminé. Ce mode permet d'évaluer les performances d'un ordonnancement spécifique et d'identifier d'éventuels inefficacités.
- **Ordonnancement dynamique** : ce mode s'appuie sur une approche de simulation-optimisation (sim-optim) pour générer l'ordonnancement de manière dynamique. Intégré au simulateur, un algorithme d'ordonnancement dynamique analyse en temps réel l'état du système et ajuste ensuite les affectations des tâches de manière proactive pour optimiser les flux de production et minimiser les temps d'attente. Ce mode permet d'explorer un large éventail de stratégies d'ordonnancement et de découvrir des solutions optimales adaptées aux conditions dynamiques

de l'environnement de production.

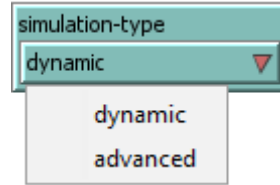


FIGURE 2.15 – Le choix du mode de simulation

Le choix du mode de simulation s'effectue via le sélecteur *simulation-type*. Par exemple, dans la Figure 2.15, le simulateur est configuré pour la simulation d'ordonnements dynamiques. Toutefois, quel que soit le mode de simulation, le simulateur reproduit fidèlement le processus de transport des jobs, depuis leur localisation initiale jusqu'à leur destination finale. Lorsqu'un transporteur reçoit une tâche (une commande), il commence d'abord par localiser l'emplacement exact du job à transporter avant de se déplacer. Une fois le job repéré, il planifie ensuite son itinéraire, puis se met en route. Après avoir récupéré le job, il planifie à nouveau son itinéraire vers la destination finale et se met en mouvement. Enfin, une fois arrivé à destination, il dépose le job et se met en attente de la prochaine tâche.

2.4.4.3.1 Simulation d'ordonnement prévisionnel : Lorsque le simulateur est configuré en mode d'ordonnement prévisionnel (`simulation-type = "advanced"`), il s'appuie sur une matrice d'ordonnement préétablie comme feuille de route pour la simulation. Cette matrice centralise toutes les informations importantes relatives aux tâches de transport et de production à effectuer. Elle recense l'ensemble des tâches de transport et de production à réaliser, précise les affectations et définit l'ordre dans lequel elles doivent être exécutées.

$$M = \begin{pmatrix} J_1 & J_1 & J_2 & J_1 & J_2 & J_2 & J_1 & J_2 \\ M_1 & M_2 & M_1 & M_4 & M_3 & M_2 & L/U & L/U \\ T_2 & T_1 & T_1 & T_2 & T_1 & T_2 & T_1 & T_2 \\ O_{11} & O_{12} & O_{21} & O_{33} & O_{22} & O_{23} & R & R \end{pmatrix} \quad (2.1)$$

Pour illustrer cela, on peut examiner la matrice M 2.1, qui détaille un ordonnancement prévisionnel impliquant deux (02) jobs, nécessitant chacun trois (03) opérations. Chaque ligne de la matrice M recèle des informations utiles pour comprendre la planification des tâches de transport et de production. La première ligne présente l'identifiant unique de chaque job (noté $J_i, i \in \{1, 2\}$). La deuxième ligne indique la destination finale de chaque job. Cette destination peut être une machine (désignée par

2.4. DÉVELOPPEMENT DU SIMULATEUR

$M_k, k \in \{1, 2, 3, 4\}$) ou la station L/U. La troisième ligne identifie le transporteur (noté $T_p, p \in \{1, 2\}$) chargé de déplacer le job vers sa destination indiquée sur la ligne 2. La quatrième ligne détaille les opérations spécifiques (notées O_{ij} , où i est le numéro du job et j l'ordre de l'opération) que chaque job doit subir. Cette information précise les étapes nécessaires au traitement complet de chaque job. La lettre « R » indique le retour du job à la station L/U après sa dernière opération de traitement, marquant la fin de son cycle de production. En analysant chaque colonne de la matrice M , on peut reconstituer le déroulement précis de l'ordonnancement. Par exemple, la première colonne indique que le transporteur 1 doit déplacer le job 1 vers la machine 1 pour sa première opération. En suivant cette logique, on peut identifier toutes les tâches de transport pour chaque transporteur et créer leurs listes de commandes individuelles. De cette même manière, on peut extraire toutes les tâches de production pour chaque machine.

2.4.4.3.2 Simulation d'ordonnancement dynamique : Lorsqu'il est configuré en mode d'ordonnancement dynamique (`simulation-type = "dynamic"`), le simulateur adopte une approche de sim-optim, où la simulation et l'optimisation s'entremêlent de manière directe. À chaque fois qu'un transporteur est disponible, c'est-à-dire en attente d'une tâche, le simulateur déclenche l'algorithme d'ordonnancement dynamique. Cet algorithme analyse l'état actuel de l'atelier, prenant en compte la position des jobs, l'état des machines et la disponibilité des transporteurs. Sur la base de cette analyse approfondie, l'algorithme détermine la tâche la plus optimale à confier au transporteur disponible, garantissant ainsi une optimisation du processus de manufacture. Les tâches de transport sont représentées par des triplets structurés (J_i, d, O_{ij}) , où J_i représente l'identité du job à transporter, d la destination du job et O_{ij} (ou « R ») l'opération à réaliser. Parallèlement à l'optimisation des tâches de transport, le simulateur gère le traitement des jobs par les machines en mode FIFO (first in, first out). Cela signifie que les machines traitent les jobs dans l'ordre exact de leur arrivée dans leurs buffers d'entrée.

2.4.4.4 Procédure *go* du simulateur

La procédure *go* représente la partie centrale du programme de simulation, assurant notamment les interactions entre les jobs, les machines, les stocks et les transporteurs. Elle comprend deux procédures qui sont *transporter-jobs* et *traiter-jobs*. La première procédure implique le déplacement des jobs par

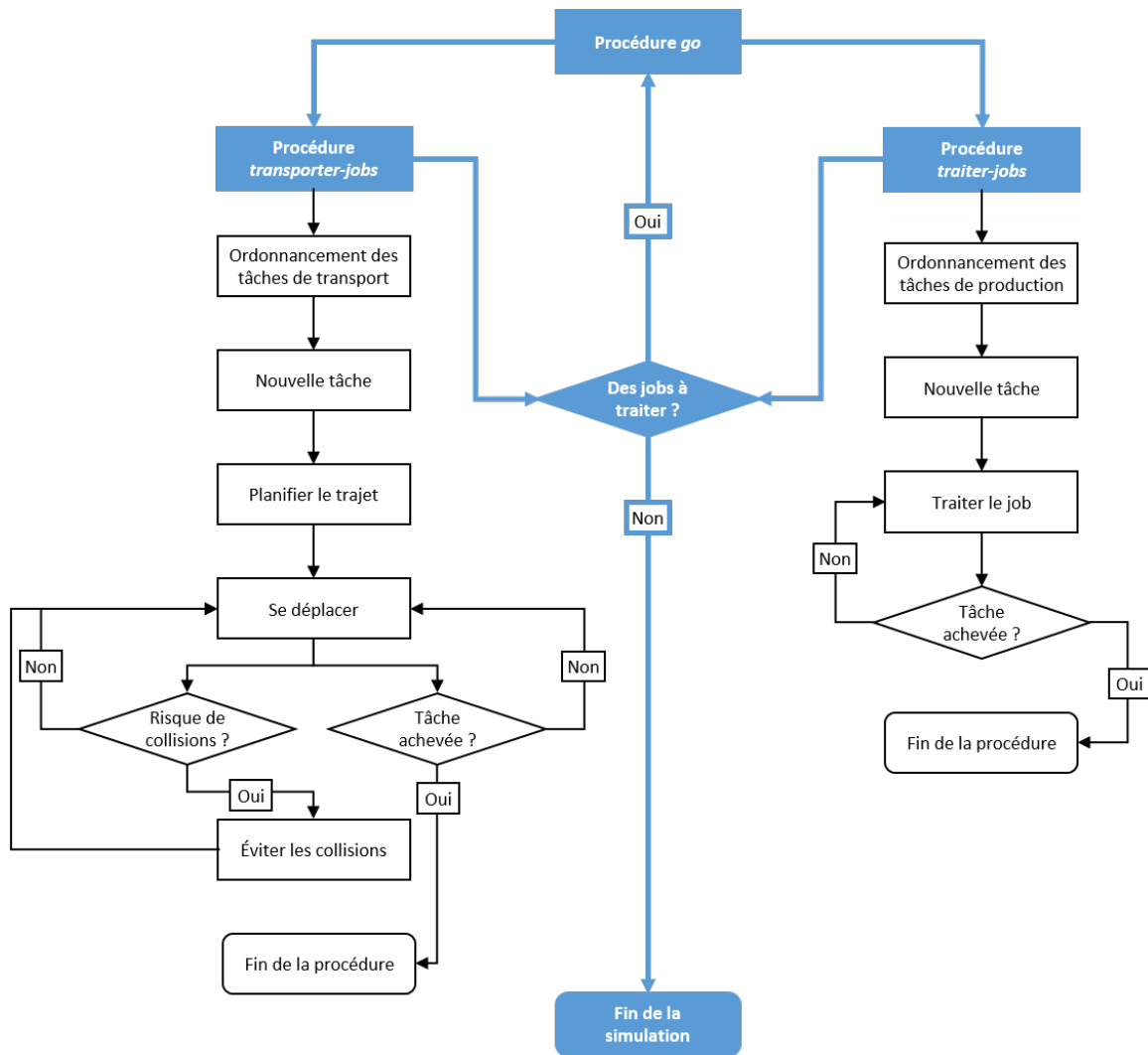


FIGURE 2.16 – Logigramme illustratif de la procédure *go*

les transporteurs vers les destinations (machines où station L/U) appropriées. C’est notamment durant les tâches de transport que les transporteurs doivent veiller à éviter les collisions. La dernière procédure garantit le traitement des jobs sur les machines. Cela inclut la gestion des temps de traitement et la mise à jour des stocks.

Les pseudo-codes des algorithmes 4-12 (situés en annexe B) fournissent un aperçu sur la structure de base du simulateur. Ils décrivent les principales étapes de la simulation, y compris le transport des jobs, leur traitement et la vérification de la fin de la simulation. Cette description est également schématisée par le logigramme de la figure 2.16

2.4.5 Intérêt du simulateur

L'atout majeur de notre approche basée sur la simulation réside dans sa flexibilité et sa capacité à représenter des environnements de production complexes et réalistes. En effet, contrairement aux méthodes classiques qui se basent sur des modèles mathématiques (plutôt simplifiés), notre approche par simulation permet d'intégrer aisément des contraintes complexes qui sont souvent difficiles, voire impossibles, à modéliser formellement. Parmi ces contraintes, on peut citer les contraintes d'espace et de forme, comme celles évoquées dans [15, 122].

En effet, les contraintes de forme et d'espace sont importantes dans la conception et la mise en œuvre de simulations, particulièrement des systèmes physiques où il y a des interactions entre des entités dans un environnement spatial. Elles peuvent se manifester de différentes manières et influencer divers aspects de la simulation. Il s'agit entre autres :

- des contraintes géométriques qui définissent la forme, la taille et la position des objets dans l'espace simulé. Elles peuvent inclure des restrictions sur les dimensions, les angles, les distances et les superpositions entre les objets ;
- des contraintes topologiques qui définissent la relation spatiale entre les objets, comme l'adjacence, la connectivité et l'inclusion. Elles peuvent être utilisées pour modéliser des structures spatiales complexes, telles que des réseaux, des graphes ou des hiérarchies spatiales ;
- et des contraintes de mouvement qui définissent les règles de mouvement des objets dans l'espace simulé. Elles peuvent inclure des restrictions sur la vitesse, l'accélération, la trajectoire et les interactions entre les objets en mouvement.

En tenant compte de ces contraintes, notre simulateur peut reproduire fidèlement les conditions réelles de production, en intégrant des éléments tels que la taille et la forme des différents agents, la configuration des machines et de l'atelier, ainsi que les restrictions de mouvement des transporteurs.

Par ailleurs, comme le précisent Kader Sanogo *et al.* [15] et M. Peter *et al.* [241], les modèles mathématiques sont difficiles à utiliser dans des situations complexes comportant de nombreuses variables et plusieurs incertitudes [15, 241]. La simulation se présente donc comme une meilleure alternative pour pallier ces insuffisances.

2.5 Conclusion

Le simulateur présenté dans ce chapitre offre un outil utile et polyvalent pour l'étude et l'analyse des systèmes de production complexes. Grâce à sa capacité à reproduire le comportement des machines, des transporteurs et des jobs, le simulateur permet de tester différents scénarios d'ordonnancement, d'identifier les insuffisances et d'évaluer l'impact de diverses modifications sur les performances globales du système. En plus de ses fonctionnalités actuelles, le simulateur peut être enrichi par l'intégration de nouvelles fonctionnalités, telles que la prise en compte des aléas, la modélisation de comportements humains et l'optimisation de l'ordonnancement dans un environnement dynamique.

Néanmoins, pour garantir son efficacité et sa fiabilité, le simulateur doit faire l'objet de tests rigoureux afin de s'assurer qu'il reproduit fidèlement les ordonnancements fournis et qu'il restitue avec précision le comportement des systèmes de production. Par ailleurs, l'intégration des contraintes de formes et d'espaces dans le simulateur constitue un atout majeur pour la résolution de problèmes d'ordonnancement plus réalistes. Ces aspects seront donc explorés en détail dans le chapitre suivant, qui se concentrera sur l'utilisation du simulateur pour résoudre le problème d'ordonnancement des tâches d'un atelier de production (JSSPT) en intégrant la contrainte d'évitement de collisions entre les transporteurs.

Chapitre 3

Simulation et optimisation de l'ordonnancement des tâches de l'atelier de production

Contenu

3.1	Introduction	109
3.2	État de l'art	110
3.2.1	Principales méthodes de résolution	111
3.2.2	Problème d'évitement des collisions et deadlocks	114
3.3	Modélisation du problème	116
3.3.1	JSSPT	116
3.3.2	Description de l'évitement de collisions et des deadlocks	120
3.3.3	Résolution des deadlocks	120
3.4	Expérimentations	122
3.4.1	Description de l'algorithme VNS	122
3.4.2	Expérience n°1	123
3.4.3	Expérience n°2	131
3.5	Conclusion	134

3.1 Introduction

La compétitivité accrue des entreprises industrielles impose une optimisation constante des processus de production. L'ordonnancement des tâches, qui consiste à déterminer l'ordre d'exécution des opérations, joue un rôle crucial dans l'atteinte de cet objectif. Parmi les problématiques d'ordonnancement les plus complexes, on trouve le *Job Shop Scheduling Problem with Transportation (JSSPT)*, qui intègre les tâches de transport des produits (ou plus généralement des jobs) entre les machines.

Le JSSPT vise à optimiser l'ordonnancement joint des tâches de production et de transport dans un système de manufacture flexible (SMF). Bien que les recherches antérieures aient apporté des contributions significatives, elles n'ont pas toujours pris en compte l'ensemble des complexités liées aux environnements de production réels, notamment la nécessité de considérer les interactions sur les chemins partagés et l'évitement des collisions entre les véhicules de transport par exemple. Cela peut conduire à des ordonnancements théoriquement optimaux mais inopérants en pratique, générant des situations de blocage et/ou des impasses, communément appelées « deadlocks », et des inefficacités.

Dans ce chapitre, nous adoptons une approche par simulation pour évaluer l'efficacité des ordonnancements issus de la littérature, en mettant en évidence les collisions et deadlocks potentiels. À cet effet, un algorithme est développé pour résoudre les situations de blocage causées par les contraintes spatiales et géométriques inhérentes à l'évitement de collisions. Les simulations sont réalisées sur les instances de référence de Bilge et Ulusoy [1], en utilisant des ordonnancements issus de l'algorithme d'optimisation basé sur la Recherche à Voisinage Variable (VNS), développé par M. Abderrahim *et al.* [14].

Le reste du chapitre est structuré comme suit : d'abord, il commence par présenter un état de l'art relatif au JSSPT et aux problèmes d'évitement des collisions et des deadlocks. Ensuite, il s'intéresse à la description du problème étudié, comprenant le modèle du JSSPT considéré et l'intégration du problème d'évitement des collisions. Il présente également l'algorithme proposé pour résoudre les situations de blocage qui surviennent suite à l'intégration de la contrainte d'évitement de collisions. Enfin, les expériences réalisées, l'analyse de leurs résultats et la conclusion clôturent ce chapitre.

3.2 État de l'art

L'ordonnancement est un problème d'optimisation combinatoire NP-difficile [242], c'est-à-dire que tout autre problème appartenant à la classe NP peut être réduit à ce problème en un temps polynomial, qui vise à déterminer la séquence optimale d'exécution des tâches sur un ensemble de ressources limitées, sous contraintes diverses (temps, ressources, précédences, etc.) [243]. Les approches utilisées pour résoudre ce type de problème sont généralement basées sur une formulation mathématique, en employant des techniques de résolution variées.

3.2.1 Principales méthodes de résolution

Les méthodes de résolution du problème d'ordonnancement se divisent principalement en deux catégories : les méthodes exactes et les méthodes approchées. Les premières garantissent de trouver la solution optimale mais peuvent s'avérer très coûteuses en temps de calcul pour des instances de grande taille [244, 21]. Les secondes, comme les heuristiques et les métaheuristiques, offrent un compromis entre la qualité de la solution et le temps de calcul [110, 21]. À côté de ces deux catégories, la simulation constitue une approche complémentaire qui permet d'évaluer la performance de différentes stratégies d'ordonnancement dans un environnement avec plus de contraintes et d'incertitudes.

3.2.1.1 Résolution par méthodes exactes

Les méthodes exactes, telles que définies par Zobolas *et al.* [244], assurent la détermination d'une solution optimale à un problème d'optimisation, en un temps limité. Ces méthodes procèdent par exploration exhaustive de l'espace de toutes les solutions possibles, ce qui peut conduire à des temps de calcul prohibitifs, notamment pour les problèmes NP-difficiles comme le JSSPT, caractérisés par un très grand nombre de solutions potentielles. En raison de cette complexité, les méthodes exactes sont souvent privilégiées pour des instances de petite taille ou pour fournir des bornes inférieures solides, mais elles sont moins adaptées à la résolution de problèmes de grande envergure.

Dans ses travaux, Andy Ham [245] met en évidence le caractère contraignant des machines et des AGV dans le JSSPT. Il propose une formulation de ce problème en utilisant la programmation par contraintes, et valide sa méthode sur des instances de référence. Fontes et Homayouni [246] adoptent une approche intégrée pour traiter simultanément les problèmes de planification de la production et du transport. Ils formulent ce problème complexe sous forme d'un modèle mathématique MILP et démontrent, à travers des expérimentations numériques menées avec Gurobi, l'efficacité de leur approche pour déterminer des solutions optimales sur des instances de référence. De même, You-Jie Yao *et al.* [191] proposent une nouvelle formulation MILP du JSSPT qui optimise le makespan. Leur modèle, basé sur un graphe disjonctif modifié, offre une représentation plus concise et précise des interactions entre les tâches de production et de transport. Dans une autre étude [202], les mêmes auteurs étendent cette approche à des environnements d'atelier flexibles. Ils décomposent le problème en quatre sous-problèmes distincts, chacun étant formulé mathématiquement de manière spécifique. Cette formulation est particulièrement intéressante car elle réduit significativement le nombre de variables

et de contraintes par rapport aux modèles précédents, simplifiant ainsi la résolution.

3.2.1.2 Résolution par méthodes approchées

Face à la complexité croissante des problèmes d'optimisation, notamment dans le contexte d'instances de grande taille, les méthodes exactes, bien qu'optimales en théorie, se révèlent souvent trop coûteuses en temps de calcul. Pour pallier cette limitation, les chercheurs [247, 248, 249] ont développé des méthodes approchées capables de fournir des solutions de qualité en un temps raisonnable. Parmi ces méthodes, les heuristiques et les métaheuristiques occupent une place centrale. Les heuristiques sont des algorithmes spécifiques à un problème donné, conçus pour explorer l'espace de recherche de manière efficace et trouver une solution réalisable, mais sans garantie d'optimalité. Leur conception repose souvent sur une connaissance approfondie du problème et sur des intuitions spécifiques. Les métaheuristiques, quant à elles, offrent un cadre plus général pour la résolution de problèmes d'optimisation. Elles s'inspirent de phénomènes naturels (recuit simulé, algorithmes génétiques, colonies de fourmis, etc.) et proposent des mécanismes de recherche adaptatifs, permettant d'explorer de vastes espaces de solutions et d'échapper aux optima locaux. Les métaheuristiques sont souvent préférées aux heuristiques car elles offrent une plus grande flexibilité et sont généralement plus robustes face à des problèmes complexes et mal posés.

Entre autres travaux, Bilel Marzouki *et al.* [147] abordent une nouvelle variante du problème d'ordonnancement d'atelier flexible distribué (DFJSPT), en introduisant les tâches de transport entre les traitement des machines. Leur problème combine trois problèmes NP-difficiles, à savoir : l'affectation des tâches aux machines, la distribution des tâches entre les usines, et le routage des robots. Pour résoudre le DFJSPT avec un seul robot par usine, ils proposent une métaheuristique améliorée d'optimisation par réaction chimique (CRO) visant à minimiser le makespan. De leur côté, Lucas Berterottière *et al.* [186] développent une métaheuristique utilisant une fonction de voisinage qui permet d'explorer un grand nombre de mouvements, afin de résoudre le JSSPT dans un environnement flexible (FJSSPT). Ils étendent le modèle classique de graphe disjonctif aux opérations de transport et exploitent le graphe dans une approche intégrée pour résoudre le problème. En outre, afin de résoudre le JSSPT en considérant la consommation énergétique de l'atelier, Chengfeng Peng *et al.* [148] ont d'abord établi un modèle mathématique d'optimisation conjointe multi-objectifs production-transport, intégrant la consommation d'énergie et le makespan comme objectifs d'optimisation. Ensuite, en se

basant sur l'idée de l'algorithme mémétique (MA), ils développent un nouvel algorithme inspiré de l'algorithme génétique à tri non dominé II (NSGA-II). De même, Dalila B.M.M. Fontes *et al.* [149] étendent le problème de planification d'atelier économe en énergie aux ressources de transport en considérant des ressources à vitesse réglable de deux types, à savoir : les machines sur lesquelles les jobs sont traités et les véhicules qui transportent les jobs dans l'atelier. En raison de la complexité du problème, ils proposent un algorithme génétique à clés aléatoires biaisées multi-objectifs qui fait évoluer simultanément plusieurs populations. Dans une autre étude [127], les mêmes auteurs abordent le JSSPT dans lequel les tâches de transport sont réalisées par un nombre limité d'AGV. Pour résoudre ce problème sur de grandes instances, ils proposent un algorithme hybride d'optimisation par essaims de particules et de recuit simulé (PSOSA). Youjie Yao *et al.* [200] abordent le même problème en présentant de nouvelles structures de voisinage spécialement conçues pour adresser les propriétés distinctives du problème. Ces structures sont combinées avec la recherche taboue et comparées à d'autres méthodes sur les instances de benchmark de Bilge et Ulusoy [1].

3.2.1.3 Résolution par simulation-optimisation

Xu Yiyi *et al.* [18] sont parmi les premiers à proposer une approche de simulation-optimisation pour résoudre le FJSSPT. Leur algorithme génétique multi-objectif, couplé à un simulateur multi-agents, vise à optimiser simultanément plusieurs critères, tels que le temps de traitement, la distance de déplacement des robots, le temps d'attente, etc. Ce simulateur interactif fournit un retour en temps réel sur l'état du système, facilitant ainsi l'évaluation des solutions proposées. Similairement, Erica P. et Arianna A. [250] traitent un problème de FJSSPT dans lequel les ressources de transport sont limitées et peuvent circuler librement entre les machines. L'objectif est de déterminer l'ordre de production des pièces sur chaque machine ainsi que l'itinéraire des ressources de transport. En raison de la complexité du problème, les auteurs utilisent la simulation à événements discrets pour comparer différentes politiques de planification et d'allocation des ressources de transport à travers l'analyse de scénarios. De même, Rakesh K. Phanden *et al.* [251] abordent le problème de planification d'atelier flexible en utilisant l'algorithme de recherche coucou (CSO). Plutôt que d'utiliser des fonctions mathématiques, les auteurs évaluent la qualité des solutions en simulant le fonctionnement de l'atelier avec le logiciel PROMODEL. Leurs résultats montrent que cette méthode basée sur la simulation permet d'optimiser efficacement la planification en minimisant le makespan.

Lazher Zaidi *et al.* [146] présentent une simulation d'un atelier multi-robots participant de manière indépendante mais collaborative à une chaîne d'assemblage de téléphones portables, où les tâches sont allouées aux robots grâce à un algorithme génétique multi-objectifs. Leur simulation démontre l'efficacité de cet algorithme avec des entités intelligentes capables de décider de manière autonome de leur trajectoire. Lei Cai *et al.* [155] développent également une approche basée sur la simulation, pour la planification en temps réel des ateliers de production complexes et dynamiques incluant la logistique. Cette méthode utilise l'information locale disponible rapidement pour générer des plannings adaptables aux changements constants de l'environnement. En intégrant une prévision de l'information future, elle améliore la robustesse des plannings générés.

3.2.2 Problème d'évitement des collisions et deadlocks

La sécurité des opérateurs et la fluidité des opérations dans un atelier automatisé reposent en grande partie sur l'évitement des collisions et des blocages entre AGV. L'évitement de collisions, comme son nom l'indique, consiste à éviter tout contact physique entre les véhicules et leur environnement [240]. C'est un domaine de recherche actif [252, 253], surtout dans les systèmes multi-robots. Les AGV sont aujourd'hui largement déployés dans les entrepôts et les usines pour optimiser les flux de production. Afin donc de garantir la sécurité des opérations et d'éviter les interruptions coûteuses, la prévention des collisions est d'un enjeu majeur.

De nombreux travaux de recherche se sont penchés sur cette problématique, proposant diverses méthodes de résolution. Par exemple, Hyansu Bae *et al.* [254] présentent un algorithme de planification de trajectoire pour plusieurs robots combinant l'apprentissage profond et les réseaux de neurones convolutifs (CNN). Cette approche permet aux robots de naviguer indépendamment tout en collaborant efficacement. De même, Y. F. Chen *et al.* [255] développent un algorithme décentralisé de planification de trajectoire pour éviter les collisions entre plusieurs agents (les robots) dans un environnement où ils ne peuvent pas communiquer entre eux. Cet algorithme utilise l'apprentissage par renforcement profond pour prédire les interactions entre les agents en amont, puis un réseau de neurones est utilisé pour estimer le temps nécessaire pour atteindre l'objectif en fonction de la position et de la vitesse des agents voisins. En revanche, A. Salimi Lafmejani et S Berman [256] adoptent une méthode centralisée de contrôle prédictif en ligne non linéaire pour la navigation sans collision ni blocage de plusieurs robots mobiles autonomes. Cette méthode résout à chaque pas de temps un problème d'optimisation

non linéaire contraint pour calculer une séquence de commandes optimales permettant aux robots d'atteindre leurs positions cibles en suivant des trajectoires sans collision. La méthode utilise un modèle cinématique de type unicycle pour prédire les états futurs des robots. Toutefois, même si les méthodes et les techniques utilisées diffèrent, la majorité des travaux se concentre sur la recherche de trajectoires sans collision et optimisées en temps pour les AGV [257, 258]. Comme affirmé dans cette étude [15], peu de travaux se sont intéressés à l'évitement de collisions entre les transporteurs lorsque ceux-ci effectuent des tâches de transport tout en suivant un chemin prédéfini, comme c'est le cas dans le JSSPT.

Par ailleurs, dans les systèmes à événements discrets, la concurrence entre processus peut engendrer des deadlocks, c'est-à-dire des situations d'impasse où des processus se bloquent mutuellement en attendant des ressources qu'ils ne détiennent pas ou qui ne sont pas encore disponibles. Cette situation a pour conséquence de gripper le système et d'empêcher la réalisation des tâches de chaque processus [259]. Selon Coffman Edward *et al.* [260], les deadlocks surviennent lorsque les quatre (04) conditions suivantes sont réunies : l'exclusion mutuelle, le « tenir et attendre », l'absence de préemption et l'attente circulaire. Ces quatre conditions peuvent paralyser l'ensemble du système si elles ne sont pas détectées et résolues à temps.

En outre, selon J. Zajac et W. Malopolski [259], les trois premières conditions de Coffman Edward *et al.* [260] sont toujours vérifiées dans les systèmes de transport basés sur les AGV. Ainsi, la prévention des attentes circulaires constitue un enjeu majeur. Ces attentes circulaires se produisent lorsque plusieurs AGV attendent chacun qu'un autre libère une ressource (par exemple, un tronçon de chemin), créant ainsi un cercle vicieux. Pour éviter ces situations, la littérature propose diverses stratégies de prévention des deadlocks, pouvant être classées en fonction du niveau de centralisation du système de contrôle :

- systèmes centralisés : Un contrôleur central gère l'allocation des ressources et peut mettre en place des algorithmes de prévention basés sur des graphes de ressources ou des réseaux de Petri. Par exemple, dans leur étude sur les systèmes d'entrepôts intelligents équipés d'AGV, Maoning Chen *et al.* [261] proposent d'éliminer les deadlocks en développant un algorithme de planification orientée vers le futur (FPP) qui prend en compte les routes futures que les AGV emprunteront via une simulation. Cet algorithme est accompagné d'une stratégie de détection et de résolution des deadlocks.

- systèmes distribués : Chaque AGV prend des décisions locales en fonction de l'information disponible, ce qui nécessite des protocoles de communication et des mécanismes de coordination efficaces. Dans ce contexte, Yuan Zhou *et al.* [262] proposent un algorithme distribué en temps réel pour éviter à la fois les collisions et les deadlocks dans les systèmes multi-robots, en arrêtant et en redémarrant les robots de manière répétée. Chaque robot peut exécuter l'algorithme de manière autonome et en temps réel en vérifiant son prochain état et les risques de deadlocks lors de ses déplacements.
- systèmes hybrides : Ces systèmes combinent les avantages des systèmes centralisés et distribués, en utilisant un contrôleur central pour gérer les ressources critiques et en laissant les AGV prendre des décisions locales pour les ressources moins importantes. Entre autres, dans son étude sur la prévention des deadlocks dans les systèmes de fabrication automatisés, J. Zajac [263] propose une méthode hybride basée sur une partition alternée des séquences de ressources nécessaires à la fabrication. La méthode identifie les séquences à risque et met en place une politique de contrôle dynamique adaptée aux systèmes de production distribués.

3.3 Modélisation du problème

À la confluence de l'informatique, des mathématiques, de la recherche opérationnelle et du génie industriel, le JSSPT a fait l'objet de nombreuses études [111, 113, 121]. Bien que les chercheurs aient abordé le problème sous différents angles, l'objectif principal reste la minimisation du makespan, c'est-à-dire la durée totale de l'ordonnancement. Ce faisant, ils ont souvent considéré l'ordonnancement des tâches de production et de transport comme un problème conjoint.

3.3.1 JSSPT

Le JSSPT consiste à déterminer l'ordre optimal d'exécution d'un ensemble de jobs $\mathcal{J} = \{J_1, J_2, \dots, J_I\}$ sur un ensemble de machines $\mathcal{M} = \{M_1, M_2, \dots, M_M\}$. Chaque job J_i est composée d'une séquence ordonnée d'opérations $(O_{i1}, O_{i2}, \dots, O_{in})$ à réaliser sur des machines spécifiques. Chaque opération O_{ij} nécessite un traitement sur une machine unique $M_k \in \mathcal{M}$, pendant une durée τ_k . Les machines sont mono-tâche et les opérations ne peuvent être interrompues avant leur terme. De plus, des buffers d'entrée B_k^I et de sortie B_k^O sont associés à chaque machine pour stocker les jobs en attente ou terminés. Les transferts de jobs entre les machines sont assurés par des AGV à charge unique, et les tâches de

3.3. MODÉLISATION DU PROBLÈME

transport ne peuvent être interrompues. On désigne une tâche de transport par $T_{i,j}$, qui signifie le transport du job J_i vers la machine sélectionnée pour effectuer l'opération $O_{i,j}$.

Symbole	Description
$\mathcal{J} = \{J_1, J_2, \dots, J_I\}$	L'ensemble des jobs.
J_i	Le job i .
O_{ij}	la j^e opération du job i .
n	Le nombre d'opérations du job i .
a_{ijk}	L'instant de début de la j^e opération du job i sur la machine k .
$N(N \rightarrow +\infty)$	Un nombre très grand.
T_{ij}	Le transport du job i subir sa j^e opération.
$tt_{dd'}$	Le temps de transport entre deux destinations d et d' .
C_i	Le temps de réalisation de toutes les opérations du job i .
$C_{max} = \max\{C_1, C_2, \dots, C_I\}$	Le temps maximum de réalisation de tous les job (makespan).
$\mathcal{M} = \{M_1, M_2, \dots, M_M\}$	L'ensemble des machines.
M_k	La machine k .
e_{ijk}	Une variable binaire qui est égale à 1 si O_{ij} peut être réalisée sur M_k .
$g_{i'j'k}^{ij}$	Une variable binaire qui est égale à 1 si O_{ij} précède $O_{i'j'}$ sur M_k .
τ_{ijk}	Le temps de traitement de la j^e opération du job i sur la machine k .
L/U	la station L/U.
d	Une destination (une machine ou la station L/U).
$\mathcal{V} = \{v_1, v_2, \dots, v_v\}$	L'ensemble des véhicules de transport.
c_{uv}	L'instant de début de la u^e tâche de transport du transporteur v .
d_{uv}	L'instant de fin de la u^e tâche de transport du transporteur v .
h_{iv}	Une variable binaire qui est égale à 1 si le job i est transporté par le transporteur

TABLE 3.1 – Notations

Plusieurs études [264, 265, 127] consacrées au JSSPT utilisent les instances de référence proposées par Bilge et Ulusoy [1] comme banc d'essai. Ces instances, largement reconnues dans la communauté scientifique, se composent typiquement de quatre machines, d'une station de chargement/déchargement et de deux AGV assurant les transports. Les auteurs modélisent le problème sous la forme d'un programme mathématique en nombres entiers mixtes (MILP), avec pour objectif de minimiser le makespan. En se basant sur leurs travaux, Moussa Abderrahim *et al.* [14] abordent le JSSPT dans le contexte d'un environnement de production où les tâches de fabrication et de transport doivent être coordonnées. En considérant une cellule de production équipée de machines et d'AGV, ils développent un algorithme de recherche par voisinage variable (VNS) pour optimiser l'ordonnancement des tâches et des transports. De plus, ils améliorent les bornes inférieures proposées par Bilge et Ulusoy [1], renforçant ainsi l'efficacité de leur approche. Par conséquent, nous reprenons ici leur formulation du

3.3. MODÉLISATION DU PROBLÈME

problème :

$$\text{Minimiser } C_{max} \quad (3.1)$$

Sous les contraintes :

$$a_{ijk} + \tau_{ijk} \leq e_{ijk} \times N; \quad \forall J_i \in \mathcal{J}, \forall j \leq n, \forall M_k \in \mathcal{M} \quad (3.2)$$

$$\sum_{M_k \in \mathcal{M}} a_{ijk} \geq \sum_{M_k \in \mathcal{M}} (a_{i-1jk} + \tau_{i-1jk}); \quad \forall J_i \in \mathcal{J}, \forall 0 < j \leq n \quad (3.3)$$

$$\sum_{M_k \in \mathcal{M}} e_{ijk} = 1; \quad \forall J_i \in \mathcal{J}, \forall j \leq n \quad (3.4)$$

$$g_{i'j'k}^{ij} + g_{ijk}^{i'j'} \leq 1; \quad \forall J_i, J_{i'} \in \mathcal{J}, J_i \neq J_{i'}, \forall j, j' \leq n, \forall M_k \in \mathcal{M} \quad (3.5)$$

$$e_{ijk} \times e_{i'j'k} = g_{i'j'k}^{ij} + g_{ijk}^{i'j'}; \quad \forall J_i, J_{i'} \in \mathcal{J}, J_i \neq J_{i'}, \forall j, j' \leq n, \forall M_k \in \mathcal{M} \quad (3.6)$$

$$a_{i'j'k} \geq a_{ijk} + \tau_{ijk} - (1 - g_{i'j'k}^{ij}) \times N; \quad \forall J_i, J_{i'} \in \mathcal{J}, J_i \neq J_{i'}, \forall j, j' \leq n, \forall M_k \in \mathcal{M} \quad (3.7)$$

$$g_{ijk}^{ij} = 0; \quad \forall J_i \in \mathcal{J}, \forall j \leq n, \forall M_k \in \mathcal{M} \quad (3.8)$$

$$C_{max} \times e_{is_j-1k} \geq a_{is_j-1k} + t_{is_j-1k}; \quad \forall J_i \in \mathcal{J}, \forall s_j \in \{O_{ij}\}, \forall M_k \in \mathcal{M} \quad (3.9)$$

$$c_{0v} = 0; \quad \forall v \in \mathcal{V} \quad (3.10)$$

$$d_{uv} \geq c_{uv}; \quad \forall v \in \mathcal{V}, \forall u \in \{T_{ij}\}_v \quad (3.11)$$

$$d_{uv} \geq c_{uv} + tt_{k'k} + tt_{kk''} \quad \forall v \in \mathcal{V}, \forall u > 0, \forall M_k, M_{k'}, M_{k''} \in \mathcal{M}, \quad (3.12)$$

$$- (2 - h_{ij}^{uv} \times e_{i-1jk} \times e_{ijk''} - h_{i'j'}^{u-1v} \times e_{i'j'k'}); \quad \forall J_i, J_{i'} \in \mathcal{J}, \forall 0 < j \leq n, \forall 0 < j' \leq n$$

$$d_{uv} \geq c_{uv} + tt_{k'L/U} + tt_{L/Uk} - (2 - h_{0j}^{uv} \times e_{0jk} \quad \forall v \in \mathcal{V}, \forall u > 0, \forall M_k, M_{k'} \in \mathcal{M}, \quad (3.13)$$

$$- h_{i'j'}^{u-1v} \times e_{i'j'k'}); \quad \forall J_i, J_{i'} \in \mathcal{J}, \forall 0 < j \leq n, \forall 0 < j' \leq n$$

$$d_{0v} \geq tt_{L/Uk} - (1 - h_{0j}^{0v} \times e_{0jk}) \times N; \quad \forall v \in \mathcal{V}, \forall J_i \in \mathcal{J}, \forall M_k \in \mathcal{M} \quad (3.14)$$

$$\sum_{u \in \{T_{ij}\}_v, v \in \mathcal{V}} h_{ij}^{uv} = 1; \quad \forall J_i \in \mathcal{J}, \forall j \leq n \quad (3.15)$$

$$\sum_{J_i \in \mathcal{J}} h_{ij}^{uv} \leq 1; \quad \forall u \in \{T_{ij}\}_v, \forall v \in \mathcal{V} \quad (3.16)$$

$$c_{uv} \geq d_{u-1v}; \quad \forall u \in \{T_{ij}\}_v, \forall v \in \mathcal{V} \quad (3.17)$$

$$\sum_{J_i \in \mathcal{J}} h_{ij}^{uv} \leq 0 \Rightarrow d_{uv} = c_{uv}; \quad \forall u \in \{T_{ij}\}_v, \forall v \in \mathcal{V} \quad (3.18)$$

$$\sum_{J_i \in \mathcal{J}} h_{ij}^{uv} \leq 0 \Rightarrow \sum_{u' \in \{T_{ij}\}_v, J_{i'} \in \mathcal{J}} h_{i'j'}^{u'v} \leq 0; \quad \forall u \in \{T_{ij}\}_v, u' > u, \forall v \in \mathcal{V} \quad (3.19)$$

$$a_{ijk} \geq d_{uv} + (h_{ij}^{uv} \times e_{ijk} - 1) \times N; \quad \forall J_i \in \mathcal{J}, \forall j \leq n, \forall M_k \in \mathcal{M}, \forall u \in \{T_{ij}\}_v, \forall v \in \mathcal{V} \quad (3.20)$$

3.3. MODÉLISATION DU PROBLÈME

$$d_{uv} \geq a_{i-1jk} + \tau_{i-1jk} + tt_{kk'} + (h_{ij}^{uv} \times e_{i-1jk} \times e_{ijk'} - 1) \times N; \quad \forall J_i \in \mathcal{J}, \forall 0 < j \leq n, M_k, M_{k'} \in \mathcal{M},$$

$$\forall u \in \{T_{ij}\}_v, \forall v \in \mathcal{V} \quad (3.21)$$

- La contrainte (3.2) représente la contrainte d'intervalle de temps pour les tâches de traitement sur les machines.
- La contrainte (3.3) désigne la contrainte de précédence. Elle permet de s'assurer que le traitement suivant ne puisse pas avoir lieu avant la fin du précédent.
- Les contraintes (3.4) à (3.8) sont les contraintes disjonctives de traitement, c'est-à-dire qu'elles vérifient qu'une tâche de traitement n'est exécutée que par une seule machine, que les jobs sont traités les uns après les autres sur une machine (une machine ne peut traiter qu'un seul job à la fois) et qu'aucun job ne peut se précéder lui-même.
- La contrainte (3.9) permet de calculer le makespan, c'est-à-dire le moment où s'achève la dernière opération du dernier job. $s_j - 1$ correspond à l'index de la dernière opération du job i .
- Les contraintes (3.10) à (3.14) représentent les contraintes d'intervalle de temps pour les tâches de transport. Toutes les tâches de transport débutent initialement à la station L/U, à l'instant $t = 0$. De plus, l'instant de fin d'une tâche de transport doit être supérieure à son instant de début.
- Les contraintes (3.15) à (3.17) garantissent qu'un job ne puisse être transporté que par un seul transporteur à la fois, qu'une tâche de transport n'implique au maximum qu'une seule tâche de traitement et qu'aucun transporteur ne puisse transporter plus d'un job à la fois.
- Les contraintes (3.18) et (3.19) obligent tout transporteur à se déplacer que s'il a préalablement reçu une assignation de job. Cela permet aux contraintes (3.11) et (3.12) de retrouver aisément la dernière position d'un transporteur.
- Les contraintes (3.20) et (3.21) garantissent la précédence entre les tâches de traitement et leur transport. En effet, une tâche de traitement ne peut avoir lieu que si le job a été préalablement transporté. Réciproquement, le transport d'un job ne peut se faire que si son traitement est achevé (excepté le cas où le job se trouve à la station L/U, en début de traitement).

3.3.2 Description de l'évitement de collisions et des deadlocks

L'évitement de collisions entre AGV requiert la considération des contraintes de forme et d'espace (voir section 2.4.5), compte tenu des contraintes géométriques et des mouvements de chaque robot. S'inspirant des comportements collectifs observés dans la nature, tels que les bancs de poissons ou les nuées d'oiseaux, nous avons défini des zones de sécurité autour de chaque AGV. Ces zones, détaillées dans la section 2.4.4.2, représentent une distance minimale à respecter entre les robots pour éviter tout contact. Ainsi, lorsqu'un AGV pénètre dans la zone de sécurité d'un autre, il déclenche un mécanisme d'évitement, typiquement un arrêt du robot suivant. Cette stratégie est particulièrement efficace lorsque les AGV sont amenés à suivre des trajectoires précises sur des chemins prédéfinis et à se déplacer en convoi. Cependant, dans des environnements complexes avec des carrefours, des adaptations peuvent être nécessaires pour garantir une fluidité du trafic. En l'occurrence, il s'agit d'implémenter un système de réservation de portions de chemin afin d'éviter les collisions frontales et latérales. Ce système est illustré sur les figures 2.13 et 2.14 dans la section 2.4.4.2.

Avec la contrainte d'évitement de collisions, le partage des couloirs de circulation et des jobs entre les AGV peut engendrer des situations d'impasse. Considérons le scénario suivant : l'AGV V_p attend le job J_i pour effectuer la tâche de transport $T_{i,j}$. Or, avant de pouvoir réaliser $T_{i,j}$, le job J_i doit être transporté par un autre AGV, V_q , pour la tâche $T_{i,j-1}$, c'est-à-dire le transport de J_i pour subir l'opération $O_{i,j-1}$. Si V_q arrive après V_p alors que ce dernier est en attente, les mécanismes d'évitement de collision vont immobiliser V_q derrière V_p . Cette situation crée une interdépendance : V_p ne peut pas récupérer J_i tant que ce dernier n'a pas préalablement subi l'opération $O_{i,j-1}$. Mais V_q ne peut pas non plus récupérer J_i tant qu'il est toujours bloqué derrière V_p . Ainsi, les deux AGV se retrouvent dans une situation d'attente circulaire, empêchant la réalisation des tâches $T_{i,j}$ et $T_{i,j-1}$. Par conséquent, $O_{i,j}$ ne se produira jamais car $O_{i,j-1}$ ne peut pas non plus être réalisée.

3.3.3 Résolution des deadlocks

Pour résoudre les deadlocks et les situations de blocage, nous avons développé une approche collaborative entre les AGV. Lorsque deux robots se trouvent dans une situation de blocage mutuel, ils procèdent à un échange de leurs tâches restantes. Cette stratégie repose sur le fait que chaque AGV dispose d'une connaissance précise de l'état d'avancement du traitement de chaque job, notamment du

3.3. MODÉLISATION DU PROBLÈME

nombre d'opérations à effectuer. Ainsi, un AGV peut déterminer si un job est prêt pour une opération donnée en vérifiant si toutes les opérations précédentes ont été réalisées. Cet échange d'informations permet de réorganiser les tâches de transport et de débloquer la situation. L'algorithme 2 implémente cette approche de manière décentralisée, sans nécessiter l'intervention d'un contrôleur central. Cette caractéristique est particulièrement intéressante pour les systèmes de transport flexibles et évolutifs. De plus, en évitant la monopolisation du passage par un seul robot, cet algorithme contribue à optimiser l'utilisation des ressources et à réduire les temps d'attente.

Algorithme 2 Pseudo-code de l'algorithme de résolution des deadlocks

Requiert : tous les transporteurs disponibles

```
1: procédure DEADLOCKS_RESOLUTION( $t_p$ ) :
2:   Si un autre transporteur  $t_q$  pénètre dans le rayon de détection alors
3:     établir la communication avec  $t_q$ 
4:     demander current_order de  $t_q$ 
5:     Si  $t_q$  est en attente de  $J_i$  pour réaliser  $T_{i,j}$  alors
6:       vérifier l'état d'avancement du traitement de  $J_i$ 
7:       Si  $J_i$  a déjà subi  $O_{i,j-1}$  alors
8:         EVITER_COLLISIONS( $t_p$ )
9:       Sinon
10:        ECHANGER_TACHES_TRANSPORT( $t_p, t_q$ )
11:      Fin si
12:    Fin si
13:  Fin si
14: Fin procédure
```

Par ailleurs, il arrive fréquemment qu'un AGV, disons t_q , doive attendre la fin du traitement d'un job qu'il vient de livrer avant de pouvoir entamer une nouvelle tâche impliquant ce même job. Cette attente peut créer des situations de blocage, notamment si un autre AGV, t_p , arrive pendant cette phase d'inactivité. Pour garantir la sécurité et éviter les collisions, t_p devra alors patienter à son tour. Ce comportement est explicitement géré à la huitième ligne de l'algorithme 2. Aussi, il est à noter que l'échange de tâches entre AGV, bien qu'une solution efficace pour la plupart des impasses, peut parfois conduire à de nouveaux deadlocks. Si, malgré plusieurs tentatives de résolution, le problème persiste, l'AGV qui bloque le passage sera contraint d'effectuer un trajet à vide (ou « deadheading ») pour permettre à l'autre d'achever sa tâche. Bien que ce scénario soit relativement rare, il souligne la nécessité de concevoir des mécanismes de coordination des AGV capables de gérer les situations complexes et imprévues.

3.4 Expérimentations

Nos expérimentations visent à évaluer, dans un premier temps, la capacité de notre simulateur à reproduire fidèlement des ordonnancements optimisés, voire optimaux, issus de la littérature, puis dans un second temps, à étudier la faisabilité de ces ordonnancements dans un environnement simulé intégrant la contrainte d'évitement de collisions. À cette fin, nous nous sommes appuyés sur les ordonnancements issus de l'algorithme d'optimisation proposé par Abderrahim *et al.* [14], reconnu pour la qualité de ses solutions sur les instances de référence de Bilge et Ulusoy [1].

Abderrahim *et al.* [14] ont développé une approche, basée sur la recherche de voisinage variable (VNS), pour optimiser conjointement les tâches de production et de transport. Leurs travaux, qui s'appuient sur les instances de référence de Bilge et Ulusoy [1], ont permis d'améliorer significativement les bornes inférieures connues. Néanmoins, les travaux de Bilge et Ulusoy ne considèrent pas le retour des jobs à la station L/U, se concentrant uniquement sur la réalisation des tâches de production. Ainsi, dans notre étude, nous avons étendu l'analyse aux tâches de transport, en considérant le makespan comme le temps maximum d'achèvement de toutes les tâches (production et transport). Cela revient à mesurer le temps écoulé entre le début de la première tâche de transport et le retour du dernier job à la station L/U. Par conséquent, les résultats d'Abderrahim *et al.* [14] ont été générés dans deux configurations : avec et sans prise en compte du retour des jobs. Cette double évaluation nous permet d'une part, de confirmer l'efficacité de leur approche, et d'autre part, de générer des ordonnancements utilisables pour nos simulations.

3.4.1 Description de l'algorithme VNS

L'algorithme de voisinage variable (VNS), introduit par N. Mladenovic et P. Hansen en 1997 [266], constitue le cœur de la méthode développée par Abderrahim *et al.* [14]. En effet, leur implémentation repose sur une stratégie de permutation de voisinages visant à explorer de manière systématique l'espace de recherche, en alternant entre une recherche locale verticale et une recherche locale horizontale, afin de s'extraire des minima locaux et de converger vers une solution optimale.

Pour ce faire, les ordonnancements sont représentés sous forme de deux chaînes, l'une dédiée aux tâches de production et l'autre aux tâches de transport (voir figure 3.1a). Cette double représentation permet de modéliser de manière fine les interactions entre les différentes activités. À partir de

3.4. EXPÉRIMENTATIONS

cette représentation, l'algorithme VNS explore l'espace de recherche en effectuant des permutations successives sur les chaînes de transport et de production. Ces permutations sont guidées par une heuristique qui vise à obtenir le meilleur makespan possible. Plus précisément, l'algorithme procède en deux étapes :

- Recherche locale sur le voisinage de transport : une recherche locale est appliquée à la chaîne de transport actuelle afin d'identifier une nouvelle chaîne qui améliore la solution globale.
- Recherche locale sur le voisinage de production : pour chaque nouvelle chaîne de transport trouvée, une recherche locale est effectuée sur la chaîne de production afin de déterminer la meilleure combinaison production-transport.

Ce processus de recherche en deux étapes est répété pour toutes les options de transport possibles jusqu'à ce qu'un critère d'arrêt soit atteint (par exemple, aucune amélioration trouvée) ou qu'une meilleure combinaison des tâches de production et de transport soit identifiée. L'ensemble du processus est ensuite itéré avec de nouvelles options de transport jusqu'à ce qu'un autre critère d'arrêt soit atteint. Un organigramme détaillé est présenté sur la figure 3.1b pour illustrer le déroulement de l'algorithme.

3.4.2 Expérience n°1

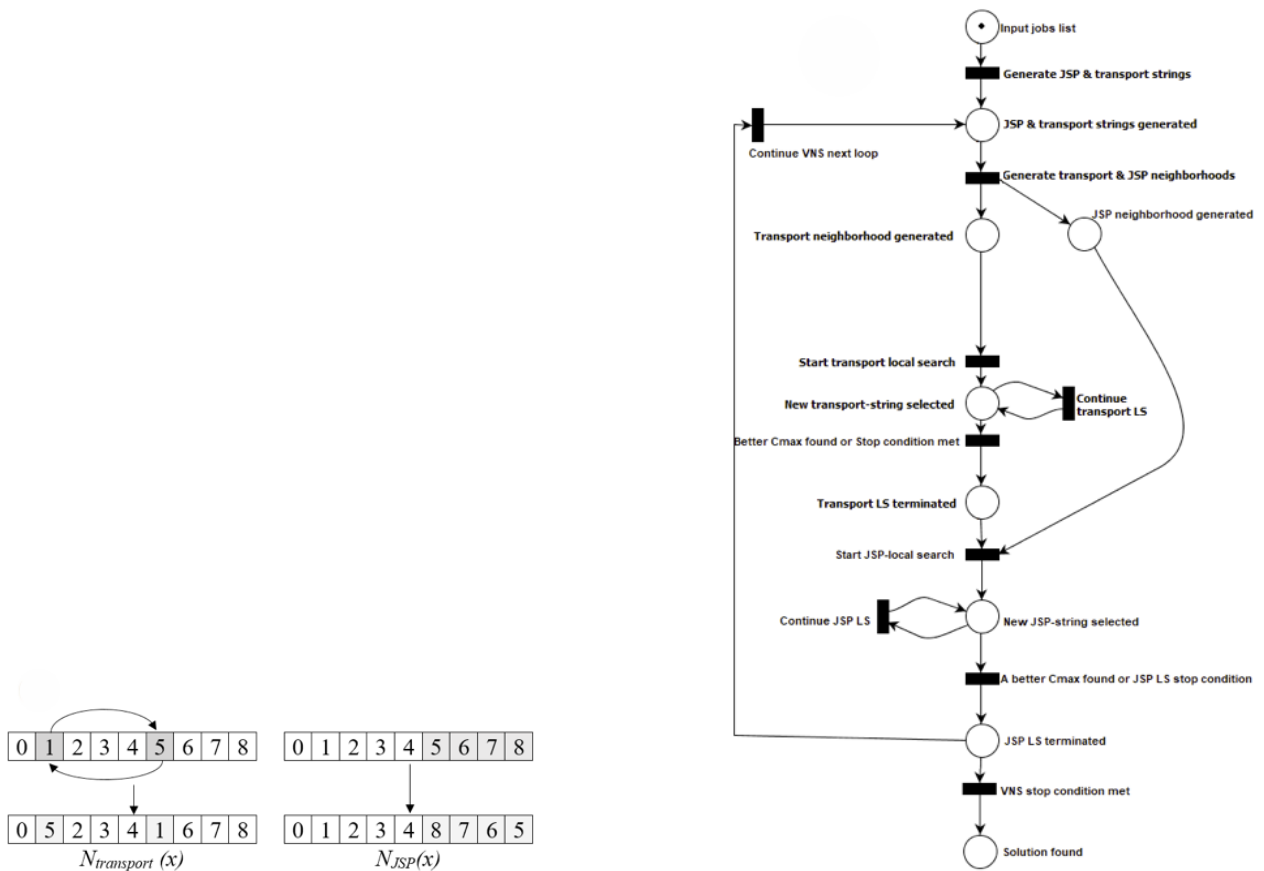
Le principal objectif de cette première expérience est de démontrer la faisabilité d'ordonnancements optimisés en simulation, d'évaluer l'impact de l'évitement des collisions entre AGV sur le makespan (ou C_{max}), et d'identifier, puis résoudre les deadlocks grâce notamment à l'algorithme 2.

3.4.2.1 Démarche expérimentale

Pour atteindre les objectifs fixés, nous avons mis en place une démarche expérimentale détaillée dans le logigramme de la figure 3.2. Cette démarche s'articule autour des étapes suivantes :

1. Génération d'ordonnancements optimisés : nous utilisons l'algorithme d'Abderrahim *et al.* [14] pour générer un ordonnancement pour chaque instance de problème. Il est à noter qu'on sélectionne le meilleur ordonnancement, c'est-à-dire celui avec le plus petit makespan, après 50 itérations. L'algorithme explore 30 voisinages à chaque itération, effectuant 50 et 15 répétitions de recherche locale respectivement sur les voisinages de transport et de production.
2. Simulation sans évitement de collisions : les ordonnancements générés sont simulés dans un

3.4. EXPÉRIMENTATIONS



(a) Représentation des chaînes de transport et de production

(b) Organigramme représentant le déroulement de la recherche VNS

FIGURE 3.1 – Représentation de la méthode VNS proposée par Abderrahim *et al.* [14]

premier temps sans considérer la contrainte de collision afin d'identifier les situations à risque et de vérifier la capacité du simulateur à reproduire fidèlement l'ordonnancement reçu.

3. Simulation avec évitement de collisions : les ordonnancements sont ensuite re-simulés en intégrant la contrainte d'évitement de collisions. En cas de deadlocks, l'algorithme 2 est utilisé pour les résoudre.
4. Analyse des résultats : les makespans obtenus dans les deux scénarios sont comparés afin d'évaluer l'impact de l'évitement des collisions sur les performances du système.

3.4. EXPÉRIMENTATIONS

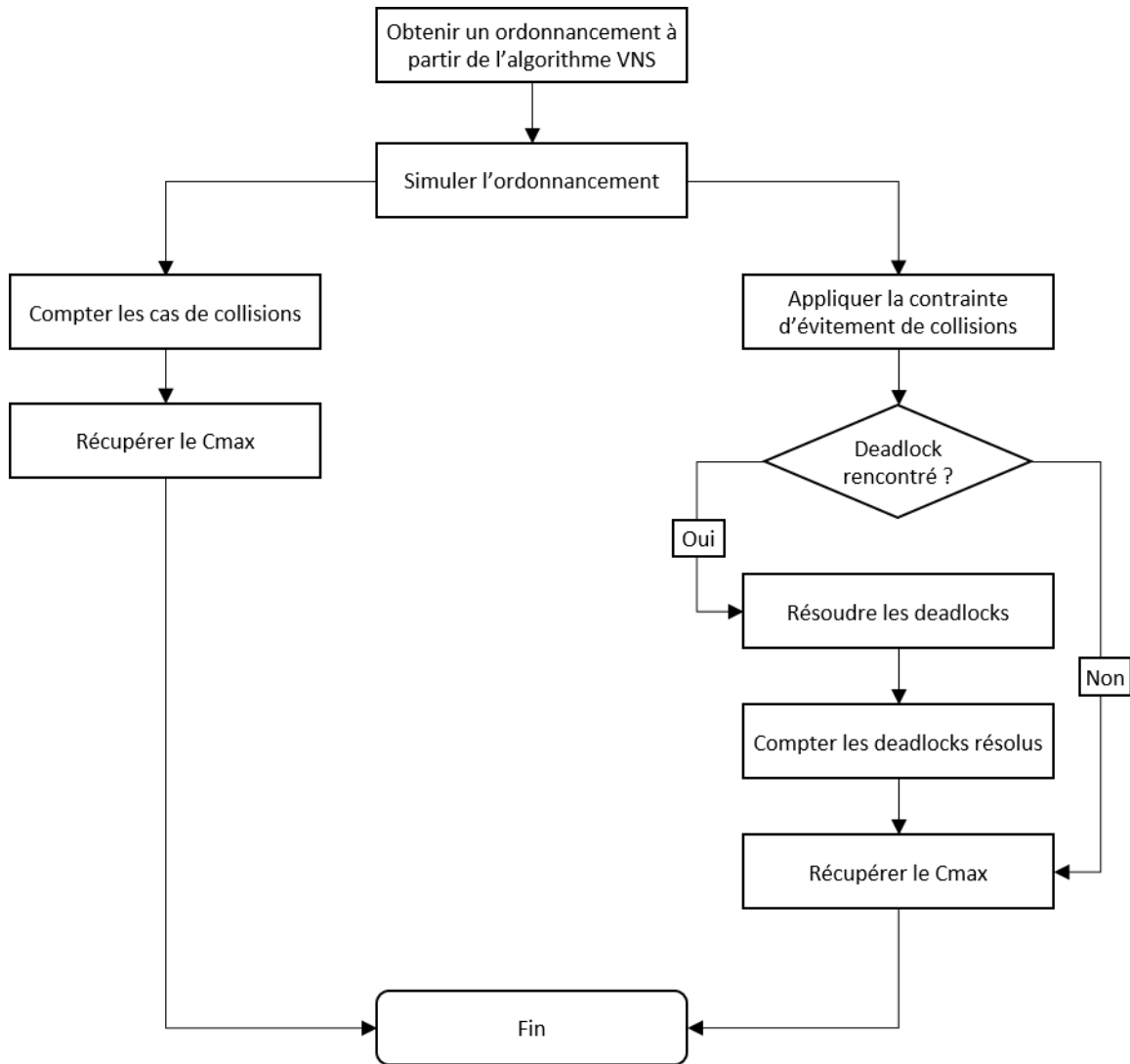


FIGURE 3.2 – Logigramme de la démarche expérimentale de l'expérience n°1

3.4.2.2 Résultats

Les résultats obtenus durant cette expérience ont été consignés dans le tableau 3.2. Les instances de référence sont indiquées dans la colonne *Instance*. Les résultats obtenus par Bilge et Ulusoy [1] sont répertoriés dans la colonne *B&U*. La colonne *BI* montre les bornes inférieures obtenues par Abderrahim *et al.* [14] grâce à leur nouvelle méthode de calcul. La colonne *MSO* affiche la meilleure solution trouvée avec l'algorithme VNS après 50 itérations, sans tenir compte du retour des jobs à la station L/U, à l'instar de la colonne *B&U*. Cette exclusion permet de comparer directement les deux colonnes et de souligner l'efficacité de la solution *MSO*, qui produit de meilleurs résultats dans

la plupart des cas.

La colonne *MSO-L/U* présente la meilleure solution obtenue après 50 itérations de l'algorithme VNS, tout en tenant compte du retour des jobs à la station L/U. Cela permet d'intégrer le temps de retour des jobs dans le calcul du makespan et de mieux identifier les risques de collision des AGV. La colonne *Cas 1* indique le makespan obtenu après la simulation de l'ordonnancement MSO-L/U, tandis que la colonne *Cas 2* montre le makespan obtenu en tenant compte de l'évitement des collisions lors de la simulation de ce même ordonnancement. Enfin, la colonne *#Collisions* recense le nombre de collisions survenues lors de la simulation du cas 1, et la colonne *#Deadlocks* indique le nombre d'impasses résolues durant la simulation du cas 2.

3.4.2.3 Analyse des résultats

A première vue, le tableau 3.2 révèle une quasi-parfaite adéquation entre les makespans prédits par l'algorithme VNS (colonne *MSO-L/U*) et ceux observés en simulation (colonne *Cas 1*). Les écarts observés, inférieurs à 0,2 secondes dans tous les cas, sont principalement attribuables aux approximations numériques inhérentes à la modélisation du simulateur. Cette concordance remarquable valide la fiabilité de notre simulateur et confirme sa capacité à reproduire fidèlement les ordonnancements générés en amont par optimisation. En effet, les simulations effectuées respectent scrupuleusement les séquences de tâches, les temps de traitement et les trajets des AGV définis par l'ordonnancement.

Une analyse plus approfondie du tableau 3.2 permet de faire les observations qui suivent.

1. **L'évitement des collisions entraîne une augmentation du makespan** : Ce phénomène est particulièrement visible sur la figure 3.3. L'introduction de l'évitement des collisions a un impact significatif sur le makespan (augmentation de plus d'une seconde). En effet, nous observons une augmentation moyenne de 50% sur les layouts 1 et 2, de 70% sur le layout 3 et de 30% sur le layout 4. Cette hausse s'explique par les retards induits par les évitements de collisions. Les AGV se retrouvent immobilisés en attente, créant ainsi des effets de dominos sur l'ensemble de l'ordonnancement. La figure 3.4, qui présente les diagrammes de Gantt combinés des simulations du cas 1 et du cas 2 (l'ordonnancement du cas 2 est marqué d'un astérisque (*)) pour l'instance EX12, illustre parfaitement ce phénomène. En observant l'ordonnancement d'AGV2 après la simulation du cas 2, on remarque un délai plus long entre $T_{O_{4,2}}^*$ et $T_{O_{5,2}}^*$ comparé à l'ordonnancement de la simulation du cas 1. Cela s'explique par le fait qu'AGV2 a été immobilisé derrière

3.4. EXPÉRIMENTATIONS

AGV1, pendant que ce dernier était en attente du job J_2 . Contraint d'attendre, l'AGV2 a vu donc son ordonnancement globalement retardé.

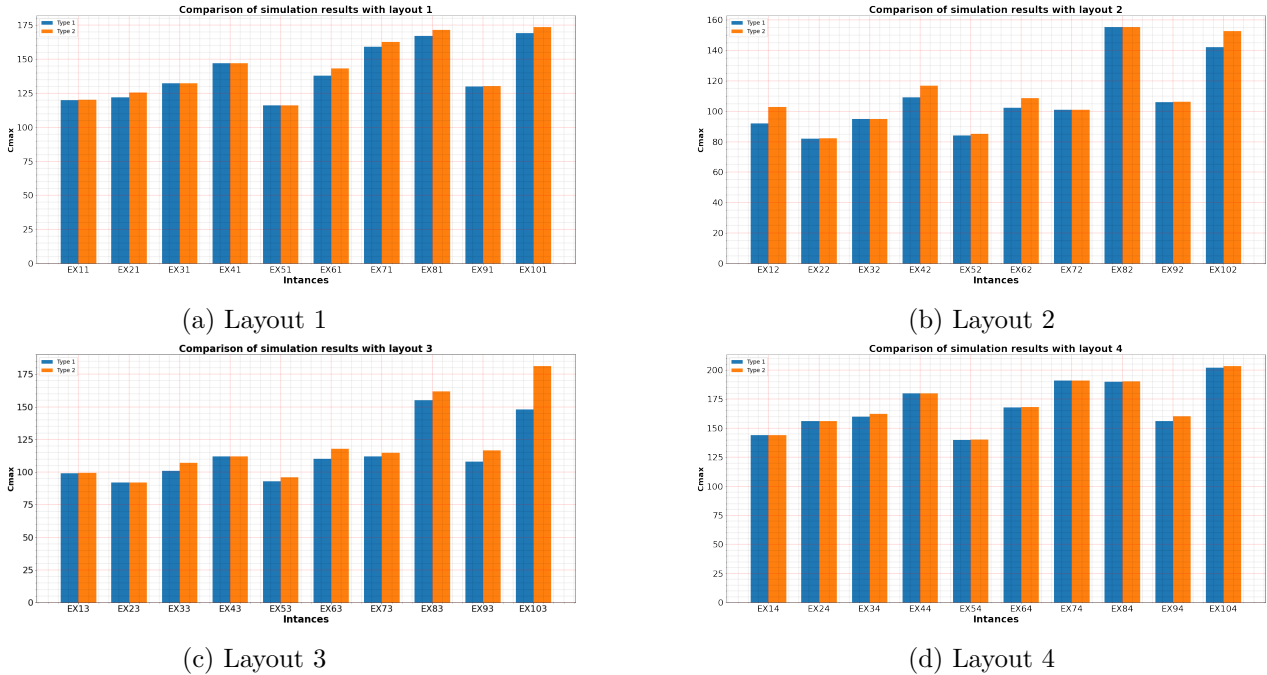


FIGURE 3.3 – Comparaison des résultats des simulations sur chaque layout (Kader Sanogo *et al.*, 2023) [15]

2. L'ordonnancement n'est pas retardé lorsque le traitement du job dure plus longtemps que le trajet de l'AGV : Ce constat constitue une exception notable à la tendance générale observée précédemment. En effet, lorsque le temps de traitement d'un job est suffisamment long, c'est-à-dire plus long que la durée du trajet de l'AGV, les retards induits par l'évitement des collisions n'ont plus d'impact sur le makespan global. Autrement dit, si l'AGV arrive sur une machine alors que le job à charger est encore en cours de traitement, il sera systématiquement immobilisé en attente, indépendamment des éventuels retards encourus lors de son trajet. Par conséquent, arriver un peu plus tôt n'aurait rien changé car il faudrait absolument attendre la fin du traitement du job avant de le récupérer. Cette situation est illustrée par l'instance EX32 (figure 3.5) où, malgré une collision, le makespan reste inchangé. Le zoom sur les diagrammes de Gantt (figure 3.6) montre que :

- dans le cas 1 : après avoir rencontré AGV2 à la 77^e seconde, l'AGV1 arrive à destination à la 81^e seconde, mais est contraint d'attendre la fin du traitement de J_1 jusqu'à la 83^e seconde ;

3.4. EXPÉRIMENTATIONS

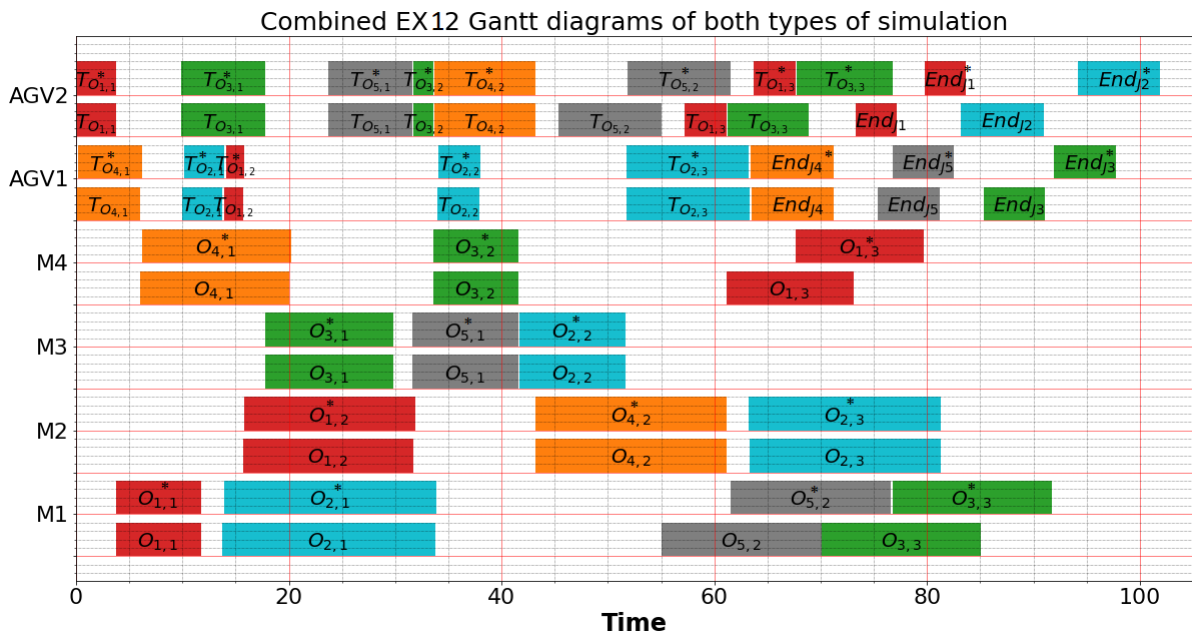


FIGURE 3.4 – Les diagrammes de Gantt combinés des simulations des cas 1 et 2 de EX12 (Kader Sanogo *et al.*, 2023) [15]

- dans le cas 2 : alors que l'AGV1 est resté bloqué derrière l'AGV2 entre la 77^e et la 79^e seconde, il arrive cependant à destination avant la 83^e et récupère J_1 .

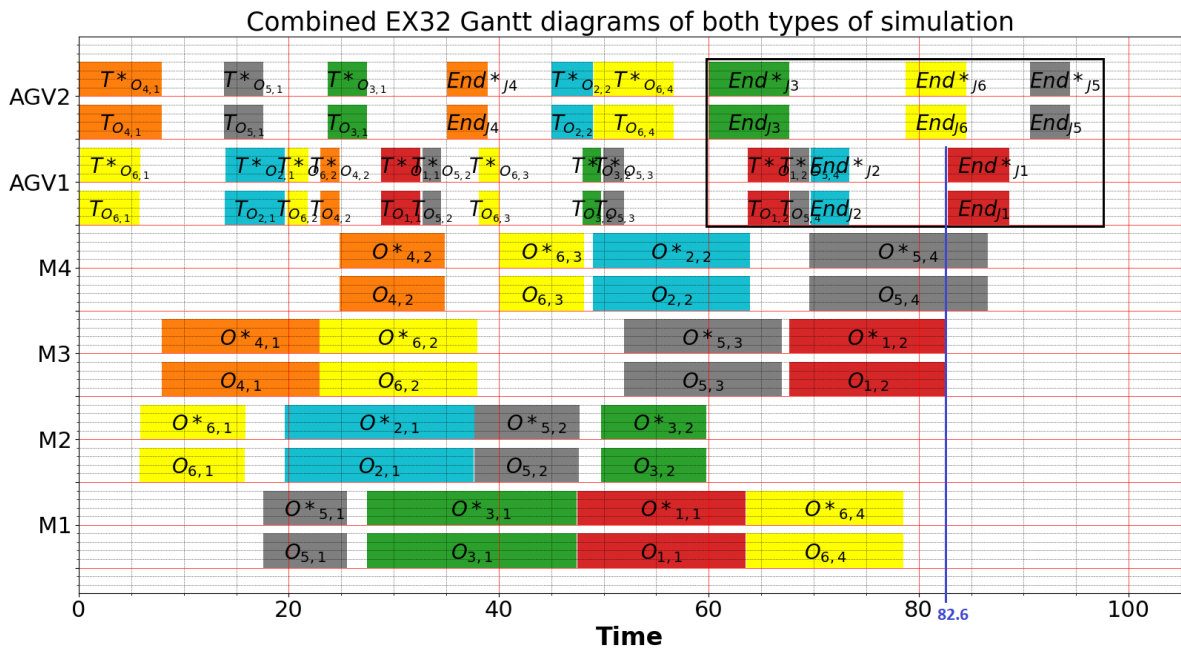


FIGURE 3.5 – Les diagrammes de Gantt combinés des simulations des cas 1 et cas 2 de EX32 (Kader Sanogo *et al.*, 2023) [15]

3.4. EXPÉRIMENTATIONS

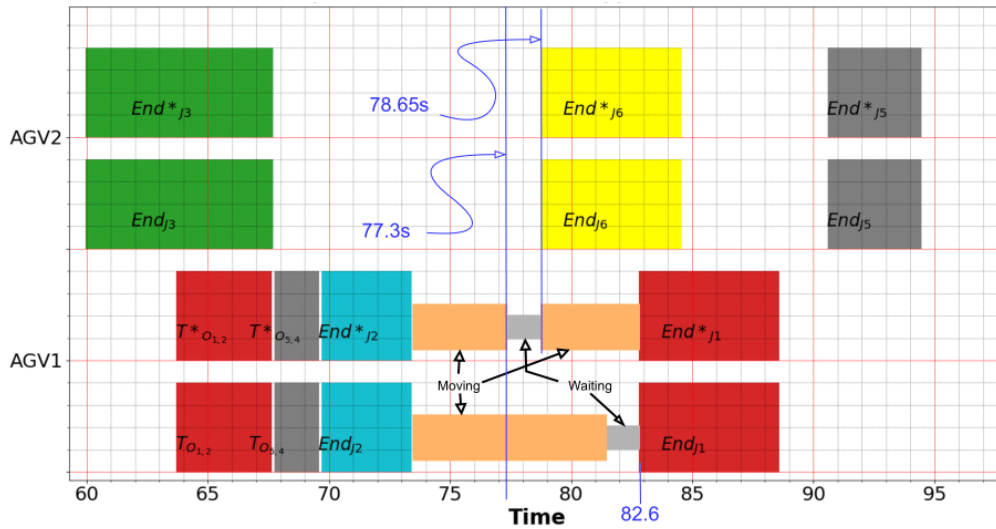


FIGURE 3.6 – Un aperçu détaillé de l’ordonnancement des AGV sur l’instance EX32 (Kader Sanogo *et al.*, 2023) [15]

3. **Un ordonnancement optimisé peut être perturbé :** L’évitement des collisions et les temps d’attente des jobs peuvent perturber significativement un ordonnancement initialement optimisé. En effet, ces événements peuvent modifier l’ordre d’exécution des opérations, entraînant parfois des conflits de ressources, et donc des deadlocks. Par exemple, si l’opération O_{ij} était initialement prévue avant l’opération $O_{i'j'}$ et que la première O_{xy} est retardée, la séquence initiale peut ne plus être respectée. Comme le montre le tableau 3.2, le nombre de deadlocks est généralement inférieur ou égal au nombre de collisions, soulignant ainsi le lien direct entre ces deux phénomènes. Le cas particulier de l’instance EX82, où le nombre de deadlocks excède le nombre de collisions, sera détaillé dans le point ci-après.
4. **La résolution d’un deadlock peut en entraîner d’autres :** L’instance EX82 illustre parfaitement comment la résolution d’une impasse peut en engendrer de nouvelles. Bien que seulement trois (03) collisions aient été initialement identifiées (aux 45^e, 60^e et 145^e secondes), cinq (05) deadlocks ont été observés au total. Des deadlocks supplémentaires sont survenus (aux 45^e, 60^e, 75^e, 91^e et 124^e secondes) suite aux échanges de tâches de transport effectués pour résoudre les collisions initiales. Ces échanges, bien que nécessaires pour débloquer la situation, ont entraînés une cascade d’événements indésirables.
5. **L’ordonnancement d’un AGV avec de longs temps d’attente augmente les risques de deadlocks :** Il peut arriver qu’un ordonnancement oblige un AGV à observer une longue

3.4. EXPÉRIMENTATIONS

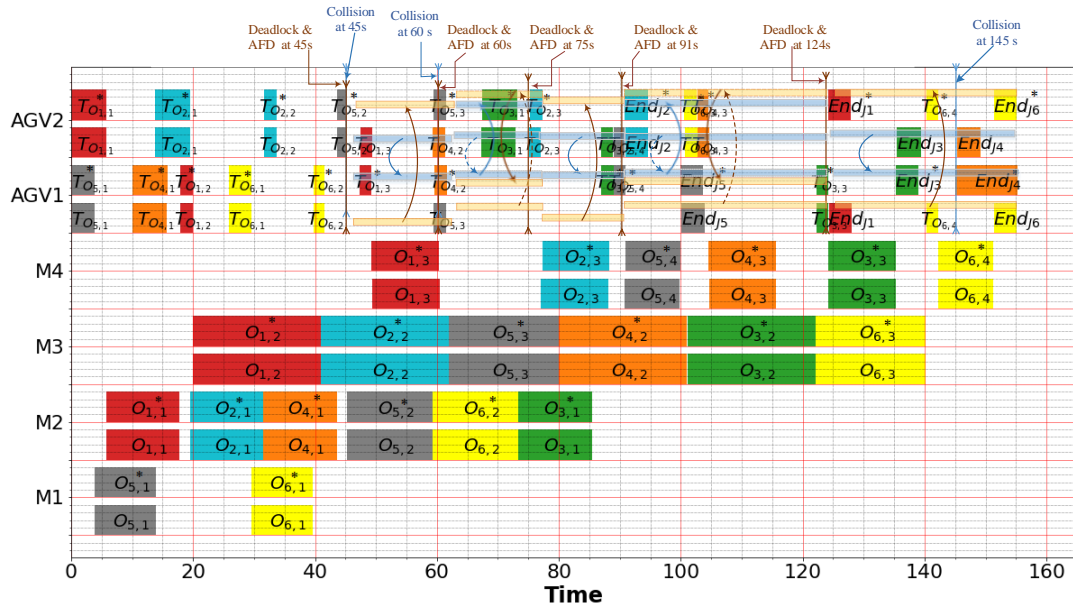


FIGURE 3.7 – Les diagrammes de Gantt combinés des simulations des cas 1 et cas 2 de EX82 (Kader Sanogo *et al.*, 2023) [15]

période d'attente afin de récupérer un job particulier. Durant toute cette période d'inactivité, l'AGV en question représente un obstacle pour les autres et bloque ainsi le passage. Cela à pour conséquence de perturber les activités des autres AGV. C'est le cas, par exemple, pour les instances EX82, EX63 et EX83.

- 6. La flexibilité de routage est un facteur préventif des deadlocks :** Une plus grande flexibilité de routage réduit significativement le risque de deadlocks en limitant les points de rencontre entre AGV. C'est pourquoi les layouts 1 et 4, offrant davantage de possibilités de circulation, n'ont généré aucun deadlock. Cependant, la multiplicité des intersections sur le layout 1 a favorisé un nombre élevé de collisions.

Par ailleurs, il faut souligner l'efficacité de l'algorithme de résolution des deadlocks (algorithme 2) car tous les problèmes identifiés lors des simulations ont été résolus. Toutefois, étant basé sur les perceptions locales des AGV, il n'a pas suffisamment d'informations sur un long historique pour évaluer la pertinence des échanges de tâches. Par conséquent, comme on a pu le constater avec l'instance EX82, la résolution d'un deadlock peut en générer un autre dans le futur. En outre, l'augmentation du makespan observée avec les simulations du cas 2 est la conséquence de la prise en compte de

3.4. EXPÉRIMENTATIONS

contraintes opérationnelles réelles, notamment l'évitement des collisions entre AGVs. Ceci rend les résultats plus réalistes et remet en cause les solutions optimisées proposées dans la littérature. En effet, de nombreux ordonnancement proposés ne seraient pas réalisables en l'état sans l'algorithme de résolution des deadlocks. Ainsi, ces résultats seraient difficilement exploitables s'ils ne tiennent pas compte des réalités industrielles.

3.4.3 Expérience n°2

Dans cette deuxième expérience, notre objectif est d'identifier des ordonnancements minimisant, voire éliminant totalement, les collisions entre AGV. Pour ce faire, nous avons adopté une approche combinant simulation et optimisation. Cette approche consiste à générer plusieurs ordonnancements optimisés, puis à les simuler dans un environnement sans collisions afin d'évaluer leur robustesse. L'analyse des résultats de ces simulations nous permettra d'identifier les ordonnancements les plus prometteurs.

3.4.3.1 Démarche expérimentale

Le logigramme de la figure 3.8 illustre la démarche adoptée pour cette deuxième expérience. Celle-ci consiste à générer successivement différents ordonnancements en intégrant systématiquement la contrainte d'évitement de collisions dans la simulation. En cas de retard dans l'ordonnancement (c'est-à-dire que le makespan est allongé) lié à cette contrainte, un nouvel ordonnancement est recherché afin d'améliorer la robustesse du système. Il est à noter que la priorité est donnée à la faisabilité de l'ordonnancement plutôt qu'à l'optimisation du makespan. Afin de limiter le temps de calcul, le processus est répété cinq (05) fois au maximum. Au bout des cinq répétitions, si aucun ordonnancement alternatif plus robuste n'est trouvé (c'est-à-dire le retard est supérieur au retard de la solution de départ), on garde alors la solution de départ. Cette approche vise à identifier des ordonnancements capables de s'exécuter normalement malgré la contrainte d'évitement de collision.

3.4.3.2 Résultats

Le tableau 3.3 présente les résultats de la deuxième expérience. Les instances marquées d'un astérisque (*) sont celles pour lesquelles la démarche de simulation-optimisation a permis de trouver un ordonnancement alternatif plus robuste. La colonne *OA* indique le makespan de l'ordonnancement

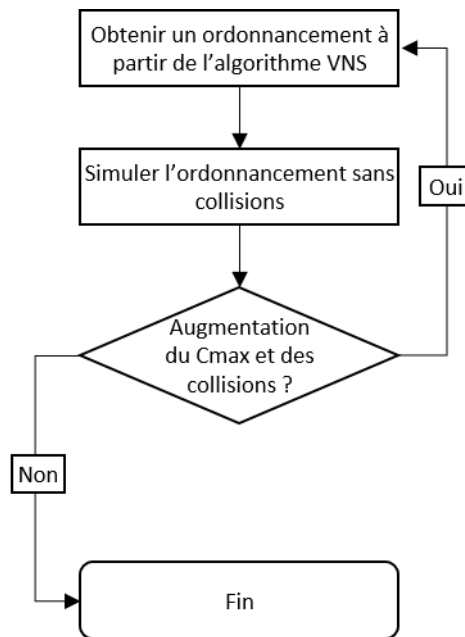


FIGURE 3.8 – Logigramme de la démarche expérimentale de l’expérience n°2

alternatif trouvé, tandis que la colonne *OA-SC* présente le résultat de la simulation de cet ordonnancement en intégrant la contrainte d’évitement des collisions. Les deux dernières colonnes (*#Collisions* et *#Deadlocks*) montrent l’évolution du nombre de collisions et de deadlocks après la simulation de l’ordonnancement alternatif. Par exemple, pour l’instance EX31, l’ordonnancement alternatif a permis d’éliminer complètement les collisions, alors qu’il y en avait deux avec l’ordonnancement initial.

3.4.3.3 Analyse des résultats

Les résultats présentés dans le tableau 3.3 permettent de tirer conclusions ci-après.

1. **Les ordonnancements ayant la même makespan peuvent avoir des comportements différents** : les résultats mettent en évidence la diversité de l’espace des solutions. En effet, plusieurs ordonnancements peuvent présenter un makespan identique tout en affichant des comportements très différents en simulation. Par exemple, pour l’instance EX62, l’ordonnancement initial avec un makespan de 102 secondes a engendré deux (02) collisions lors de la simulation du cas 1, conduisant à un allongement du makespan observé à 108,5 secondes et un (01) deadlock lors de la simulation du cas 2. Cependant, après le processus de simulation-optimisation, un ordonnancement alternatif de 102 secondes a été obtenu, présentant un makespan de 102,4

3.4. EXPÉRIMENTATIONS

secondes en simulation sans collisions avec une seule collision et aucun deadlock. Ce phénomène souligne l'importance de la simulation pour évaluer la qualité d'un ordonnancement et identifier les solutions les plus robustes et efficaces en pratique. Les figures 3.9 et 3.10 illustrent cette différence de comportement entre les deux ordonnancements.

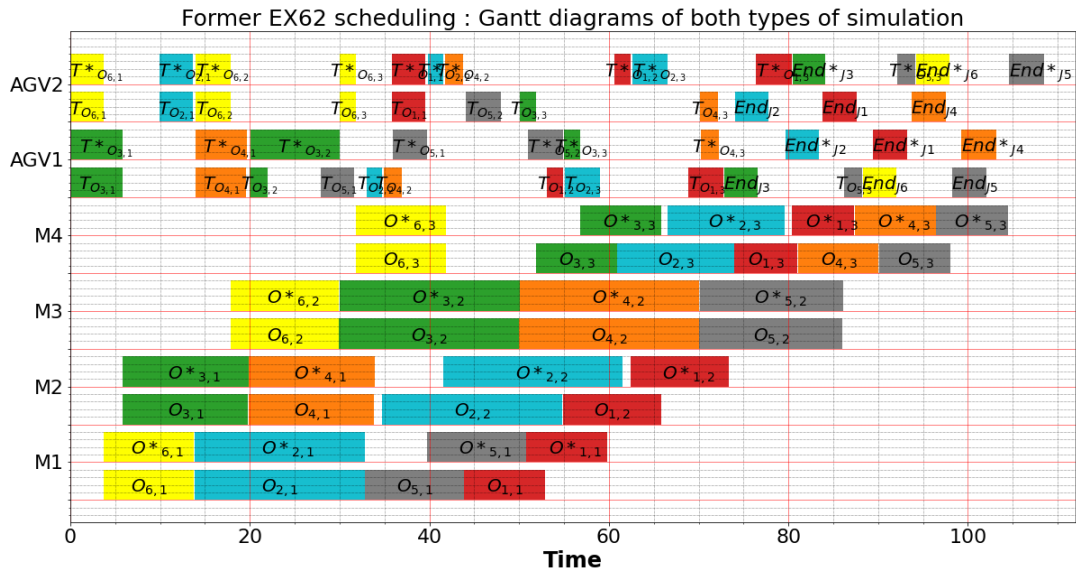


FIGURE 3.9 – Les diagrammes de Gantt combinés des simulations de l'ordonnancement initial de EX62 (Kader Sanogo *et al.*, 2023) [15]

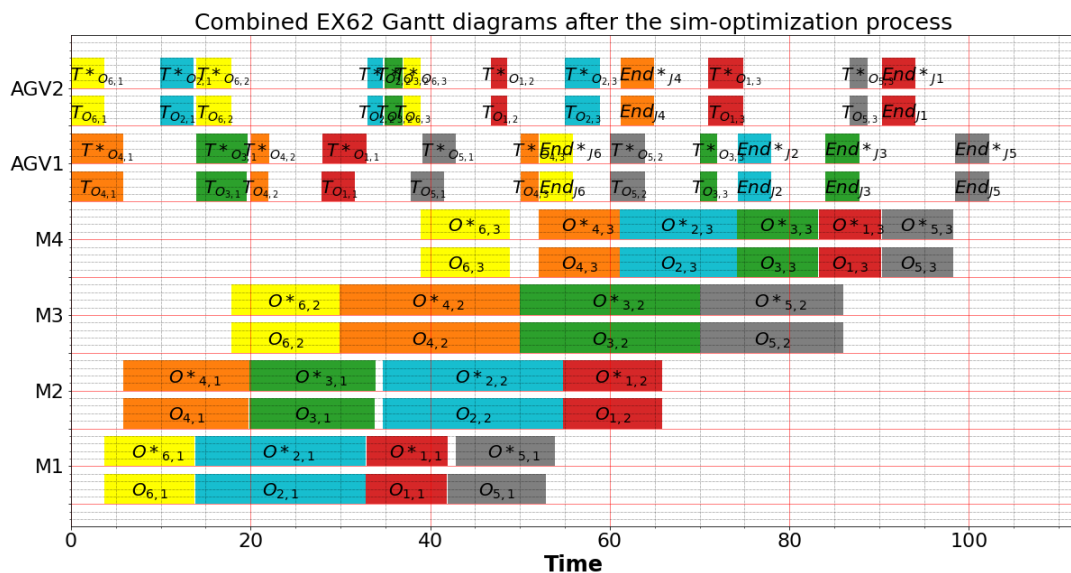


FIGURE 3.10 – Les diagrammes de Gantt combinés des simulations de l'ordonnancement alternatif de EX62 (Kader Sanogo *et al.*, 2023) [15]

2. **Les ordonnancements optimaux ne sont pas toujours les plus robustes** : Comme le montre le tableau 3.3, l'optimisation du makespan seul ne garantit pas systématiquement les meilleures performances en conditions réelles. En effet, certains ordonnancements avec un makespan légèrement dégradé peuvent offrir de meilleurs résultats en simulation, notamment en termes de fluidité et de robustesse. Par exemple, pour l'instance EX12, le processus de simulation-optimisation a permis de trouver un ordonnancement alternatif avec un makespan de 99 secondes, supérieur à celui du *MSO-L/U* (92 secondes). Cependant, en simulation sans collisions, l'ordonnancement alternatif conserve son makespan, soit une amélioration par rapport aux 102,8 secondes du **MSO-L/U**. Ce phénomène, observé également sur les instances EX94, EX61 et EX71, souligne l'importance d'évaluer les ordonnancements non seulement sur leur makespan théorique, mais aussi sur leur robustesse.

3.5 Conclusion

Pour conclure, ce chapitre a permis d'approfondir le problème complexe de l'ordonnancement conjoint des tâches de production et de transport (JSSPT), en examinant les travaux existants à travers un état de l'art et en explorant diverses méthodes de résolution. Il a aussi permis de mettre en évidence l'importance de considérer les contraintes opérationnelles telles que l'évitement des collisions et la résolution des deadlocks dans le cadre du JSSPT. Ces contraintes, difficiles à modéliser de manière formelle, ont été intégrées dans notre simulateur, offrant ainsi un environnement réaliste pour tester les ordonnancements optimisés.

Les simulations ont démontré non seulement l'efficacité de notre simulateur à reproduire fidèlement les ordonnancements théoriques, mais aussi les limites des approches d'optimisation qui ne considèrent pas généralement les collisions, pouvant aboutir à des inefficacités significatives comme les deadlocks, et rendant certaines solutions inapplicables dans un contexte réel. L'algorithme développé pour résoudre ces blocages a prouvé son efficacité en améliorant la faisabilité des ordonnancements tout en minimisant les risques de collisions entre les AGV. Ces résultats renforcent l'idée qu'une approche intégrée, combinant optimisation et simulation, est nécessaire pour atteindre des performances robustes et réalistes dans les systèmes de production actuels.

Enfin, ce travail ouvre la voie à des recherches futures visant à optimiser davantage les systèmes de

3.5. CONCLUSION

production dans le cadre du JSSPT, notamment en explorant l'utilisation de véhicules de transport plus flexibles, robustes et adaptatifs dans des environnements de production de plus en plus complexes, intégrant également des travailleurs humains. Le chapitre suivant abordera ainsi la transition des AGV vers les véhicules autonomes intelligents (AIV) dans des environnements de production plus dynamiques et inclusifs, dans le cadre de l'Industrie 5.0 (I5.0).

3.5. CONCLUSION

Instance	B&U[1]	BI[14]	MSO[14]	MSO-L/U	Cas 1	Cas 2	#Collisions	#Deadlocks
EX11	96.0	76.0	96.0	120.0	120.0	120.0	0	0
EX21	105.0	86.0	100.0	122.0	122.0	125.7	1	0
EX31	105.0	88.0	102.0	132.0	132.2	132.2	2	0
EX41	118.0	78.0	116.0	147.0	147.0	147.1	3	0
EX51	89.0	65.0	87.0	116.0	116.0	116.15	1	0
EX61	120.0	108.0	120.0	138.0	138.0	143.2	2	0
EX71	119.0	77.0	121.0	159.0	159.0	162.6	1	0
EX81	161.0	161.0	161.0	167.0	167.1	171.5	5	0
EX91	120.0	105.0	117.0	130.0	130.1	130.1	1	0
EX101	153.0	133.0	152.0	169.0	169.0	173.5	4	0
EX12	82.0	76.0	82.0	92.0	92.0	102.8	1	0
EX22	80.0	76.0	76.0	82.0	82.0	82.2	1	0
EX32	88.0	80.0	85.0	95.0	95.0	95.0	1	0
EX42	93.0	70.0	92.0	109.0	109.1	116.8	1	1
EX52	69.0	64.0	69.0	84.0	84.0	85.05	1	0
EX62	100.0	98.0	98.0	102.0	102.2	108.5	2	1
EX72	90.0	74.0	84.0	101.0	101.0	101.0	0	0
EX82	151.0	151.0	151.0	155.0	155.2	155.3	3	5
EX92	104.0	98.0	102.0	106.0	106.0	106.1	1	0
EX102	139.0	128.0	139.0	142.0	142.0	156.2	3	2
EX13	84.0	74.0	84.0	99.0	99.0	99.4	1	0
EX23	86.0	82.0	86.0	92.0	92.0	92.0	0	0
EX33	86.0	82.0	86.0	101.0	101.0	107.0	1	1
EX43	95.0	71.0	92.0	112.0	112.0	112.0	0	0
EX53	76.0	63.0	74.0	93.0	93.0	96.05	2	0
EX63	104.0	100.0	104.0	110.0	110.1	117.8	2	2
EX73	91.0	76.0	87.0	112.0	112.0	114.8	2	1
EX83	153.0	153.0	153.0	155.0	155.0	161.9	2	2
EX93	110.0	100.0	105.0	108.0	108.0	116.65	2	0
EX103	143.0	133.0	143.0	148.0	148.0	181.2	4	1
EX14	108.0	76.0	103.0	144.0	144.0	144.0	0	0
EX24	116.0	84.0	113.0	156.0	156.0	156.15	0	0
EX34	116.0	87.0	113.0	160.0	160.0	162.4	1	0
EX44	126.0	81.0	131.0	180.0	180.0	180.0	0	0
EX54	99.0	62.0	97.0	140.0	140.0	140.2	0	0
EX64	120.0	103.0	129.0	168.0	168.0	168.2	0	0
EX74	136.0	78.0	134.0	191.0	191.0	191.0	0	0
EX84	163.0	163.0	163.0	190.0	190.0	190.3	1	0
EX94	125.0	102.0	123.0	156.0	156.0	160.2	1	0
EX104	171.0	136.0	169.0	202.0	202.0	202.0	1	0

TABLE 3.2 – Résultats de l'expérience n°1

3.5. CONCLUSION

Instance	MSO-L/U	Cas 1	Cas 2	OA	OA-SC	#Collisions	#Deadlocks
EX11	120.0	120.0	120.0	-	-	0	0
EX21	122.0	122.0	125.7	-	-	1	0
EX31*	132.0	132.2	132.2	132.0	132.2	2 → 0	0
EX41	147.0	147.0	147.1	-	-	3	0
EX51	116.0	116.0	116.15	-	-	1	0
EX61*	138.0	138.0	143.2	139	142.9	2	0
EX71*	159.0	159.0	162.6	160.0	160.2	1	0
EX81*	167.0	167.1	171.5	167.0	171.5	5 → 3	0
EX91	130.0	130.1	130.1	-	-	1	0
EX101*	169.0	169.0	173.5	169.0	173.3	4 → 3	0
EX12*	92.0	92.0	102.8	99.0	99.0	1 → 0	0
EX22*	82.0	82.0	82.2	82.0	82.2	1 → 0	0
EX32	95.0	95.0	95.0	-	-	1	0
EX42*	109.0	109.1	116.8	109.0	109.0	1 → 0	1 → 0
EX52*	84.0	84.0	85.05	84.0	84.0	1 → 0	0
EX62*	102.0	102.2	108.5	102.0	102.4	2 → 1	1 → 0
EX72*	101.0	101.0	101.0	101.0	101.0	0	0
EX82	155.0	155.15	155.3	-	-	3	5
EX92	106.0	106.0	106.1	-	-	1	0
EX102*	142.0	142.0	156.2	145.0	145.0	3 → 1	2 → 0
EX13	99.0	99.0	99.4	-	-	1	0
EX23	92.0	92.0	92.0	-	-	0	0
EX33*	101.0	101.0	107.0	104.0	104.1	1 → 0	1 → 0
EX43	112.0	112.0	112.0	-	-	0	0
EX53*	93.0	93.0	96.05	93.0	93.1	2 → 1	0
EX63*	110.0	110.1	117.8	110.0	117.8	2	2
EX73*	112.0	112.0	114.8	112.0	112.0	2 → 1	1 → 0
EX83*	155.0	155.0	161.9	155.0	155.2	2	2 → 3
EX93*	108.0	108.0	116.65	109.0	109.2	2 → 0	0
EX103*	148.0	148.0	181.2	148.0	148.1	4 → 2	1 → 0
EX14	144.0	144.0	144.0	-	-	0	0
EX24	156.0	156.0	156.15	-	-	0	0
EX34	160.0	160.0	162.4	-	-	1	0
EX44	180.0	180.0	180.0	-	-	0	0
EX54	140.0	140.0	140.2	-	-	0	0
EX64	168.0	168.0	168.2	-	-	0	0
EX74	191.0	191.0	191.0	-	-	0	0
EX84	190.0	190.0	190.3	-	-	1	0
EX94*	156.0	156.0	160.2	158.0	158.0	1 → 0	0
EX104*	202.0	202.0	202.0	202.0	203.3	1	0

TABLE 3.3 – Résultats de l'expérience n°2

3.5. CONCLUSION

Chapitre 4

Transition des AGV vers les AIV dans le contexte de l'industrie 5.0

Contenu

4.1	Introduction	139
4.2	État de l'art	141
4.2.1	Modélisation du comportement des humains	141
4.2.2	L'utilisation des AIV dans les systèmes de production	145
4.3	Description du problème	147
4.3.1	Transition des AGV vers les AIV	147
4.3.2	Intégration des humains dans l'atelier	148
4.4	Modélisation du comportement des opérateurs humains	149
4.5	Mise à jour du modèle de simulation	153
4.6	Expérimentations	155
4.6.1	Expérience n°1	155
4.6.2	Expérience n°2	161
4.7	Conclusion	168

4.1 Introduction

Les véhicules terrestres autonomes (UGV), largement utilisés dans les industries modernes, représentent un domaine de recherche actif, marqué par des avancées continues et une évolution constante des capacités technologiques [117]. Parmi ces UGV, deux types principaux se distinguent : les véhicules autonomes guidés (AGV) et les véhicules autonomes intelligents (AIV), chacun ayant des applications et des fonctionnalités spécifiques. Les AGV, en particulier, ont démontré leur efficacité dans le transport de matériel au sein des ateliers de fabrication et des entrepôts logistiques depuis plusieurs décennies [118]. Toutefois, leur déploiement comporte un inconvénient majeur : l'installation

d'AGV exige des modifications coûteuses de l'agencement des ateliers, nécessitant la création de zones de déplacement dédiées [119]. En tant que robots guidés, toute modification de la disposition de l'atelier impose une mise à jour des cartes et de l'environnement dédié. Pour surmonter ces limitations, les AIV, des robots plus intelligents, flexibles et adaptatifs, émergent comme une alternative prometteuse [117]. Contrairement aux AGV, les AIV n'ont pas besoin de chemins prédéfinis et peuvent naviguer aisément parmi les obstacles statiques et dynamiques, y compris les travailleurs humains. Cette capacité les rend particulièrement adaptés à l'Industrie 5.0 (I5.0), un paradigme où l'interaction homme-robot est intensifiée [94].

Par ailleurs, dans le cadre de la transition vers l'I5.0, l'humain occupe une place centrale, contribuant à la gestion des défis et des ajustements contextuels avec une efficacité et une flexibilité accrues. En effet, les compétences humaines enrichissent les technologies d'automatisation en exploitant les capacités cognitives et adaptatives des travailleurs [267]. Bien que la perspective de créer un environnement industriel collaboratif, où humains et robots coexistent et se complètent, soit prometteuse, la présence humaine peut influencer le fonctionnement des robots mobiles de transport dans les environnements partagés. Il devient donc important d'intégrer et d'anticiper les mouvements des travailleurs en analysant leurs comportements pour optimiser les trajets des robots et améliorer l'efficacité globale du système de production [268]. Toutefois, si les recherches sur le JSSPT ont principalement porté sur l'optimisation des processus de production, il apparaît pertinent d'étendre ces travaux en intégrant les interactions humains-robot dans le processus d'optimisation des performances du système. D'ailleurs, pour une utilisation efficace des robots de transport dans des environnements dynamiques, il est essentiel d'évaluer l'impact des interactions robot-robot et humain-robot sur les temps de transport et sur la durée totale de fabrication (makespan). Une étude analytique de ces interactions peut s'avérer complexe. Néanmoins, la simulation constitue une alternative pour leur représentation et leur évaluation.

Dans ce chapitre, nous proposons donc une approche de simulation pour aborder le problème d'ordonnancement joint des tâches de transport et de production, en intégrant les AIV comme véhicules de transport, tout en tenant compte de la présence des travailleurs humains dans l'atelier. L'objectif de cette étude est double : d'une part, mettre en lumière les améliorations potentielles liées à la transition des AGV vers les AIV, et d'autre part, évaluer l'impact de la présence humaine sur l'ordonnancement des tâches. À cette fin, nous utilisons les chaînes de Markov à temps discret (DTMC) pour modéliser

le comportement stochastique des travailleurs sous forme d'événements discrets. Étant donné que les AIV et les humains partagent les mêmes couloirs de déplacement dans l'atelier, cette étude permettra d'analyser l'influence de la présence humaine sur les opérations des AIV.

La structure du chapitre est la suivante : dans un premier temps, un état de l'art sur les AIV et la modélisation du comportement humain est présenté. Ensuite, nous détaillons notre approche de modélisation des interactions entre les travailleurs humains et les AIV dans l'atelier. Enfin, le chapitre se conclut par la présentation des expériences menées, l'analyse des résultats obtenus et les conclusions qui en découlent.

4.2 État de l'art

4.2.1 Modélisation du comportement des humains

La modélisation du comportement humain demeure un défi majeur en raison de la complexité intrinsèque des individus et de l'influence de multiples facteurs contextuels. L'imprévisibilité des actions humaines, façonnée par des éléments cognitifs, émotionnels et sociaux, rend cette tâche particulièrement ardue. Néanmoins, face à cette complexité, les chercheurs ont développé un ensemble d'approches méthodologiques pour modéliser certains aspects du comportement humain. Ces approches peuvent être regroupées en quatre grandes catégories :

- **Méthodes mathématiques** : Ces méthodes s'appuient sur des modèles statistiques et probabilistes pour décrire les tendances et les schémas observés dans le comportement humain. Elles permettent de quantifier les incertitudes et de formuler des prédictions.
- **Méthodes basées sur la connaissance de l'environnement** : Ces approches prennent en compte l'influence de l'environnement physique et social sur le comportement humain. Elles utilisent des modèles cognitifs et adoptent une approche plus symbolique de la modélisation du comportement humain. Ces approches comprennent généralement des bibliothèques ou des bases de données sur ce que les gens font en général lorsqu'ils sont confrontés à une certaine situation. Cela leur permet de saisir le comportement humain à des niveaux plus élevés que les modèles mathématiques, ce qui permet une compréhension plus riche et hiérarchique de ce que font les gens [269].
- **Méthodes basées sur l'apprentissage automatique** : Ces méthodes exploitent de grandes

quantités de données pour apprendre des modèles complexes de comportement humain. Les techniques d'apprentissage profond, en particulier, permettent de développer des modèles capables de reconnaître des configurations subtiles dans les données et de générer des comportements réalistes.

- **Méthodes déterministes** : Bien que le comportement humain soit souvent considéré comme aléatoire, certaines situations peuvent être modélisées à l'aide de modèles déterministes. Ces modèles sont particulièrement adaptés à la description de comportements routiniers ou de réactions à des stimuli spécifiques.

Par ailleurs, il est important de noter que ces différentes approches ne s'excluent pas mutuellement et sont souvent combinées pour construire des modèles plus précis et robustes. Le choix de la méthode dépendra de la nature du comportement à modéliser, du degré de précision, de la disponibilité des données et des objectifs de la modélisation.

4.2.1.1 Méthodes mathématiques

Dans [270], Trevor S. *et al.* soulignent l'absence de méthodes systématiques pour intégrer des modèles humains variés dans la prise de décision robotique en temps réel. Ils proposent une approche basée sur un processus de décision markovien (MDP) couplée à une fonction d'appariement pour prédire les intentions humaines. Cependant, leur modèle se limite à des interactions simples entre deux agents sans communication directe. De même, Lin H. et Chen W. [271] proposent un framework basé sur un MDP pour reconnaître l'intention humaine à partir des informations obtenues grâce aux objets et au plancher. Ce framework est associé à une fonction de récompense basée sur la fréquence pour prédire l'intention humaine dans l'exécution d'une tâche en fonction de la fréquence à laquelle la tâche a été exécutée dans le passé. La position des objets et le type de gestes de la main sont utilisés comme les états du MDP.

Tomislav P. *et al.* [272, 273] proposent un algorithme de reconnaissance et d'estimation des intentions des travailleurs dans un entrepôt flexible robotisé. Cette estimation est faite à l'aide d'un modèle de Markov caché (HMM), où la probabilité de désirer chaque objectif est modélisée comme un état caché. Dans [274], les auteurs étudient les problèmes de planification de mouvement avec l'incertitude de l'intention humaine. Pour ce faire, ils supposent que le robot interagit avec un agent intentionnel, à savoir un humain. L'agent a un ensemble fini d'intentions, chacune incarnée par un comportement

observable, et fixes dans le temps. Compte tenu de cela, la dynamique comportementale de l'agent est modélisée à l'avance et connue du robot. La modélisation se fait à travers un processus de décision de Markov à observabilité mixte (MOMDP).

Thompson S. *et al.* [275] utilisent une approche probabiliste basée sur des grilles de probabilité de mouvement pour représenter le mouvement probable d'un être humain se déplaçant vers une cellule cible particulière. Les grilles de probabilité sont construites sur un nombre discret d'étapes dans le temps pour chaque emplacement possible de la cible et pour une plage de vitesses initiales. L'intérêt de ce travail est de déterminer l'intention de déplacement probable des humains et à l'utiliser pour prédire les mouvements humains pour un système de robot mobile autonome.

4.2.1.2 Méthodes basées sur la connaissance

Fazel Ansari *et al.* [276] proposent une méthodologie pour la sélection optimale d'un ou de plusieurs travailleurs compétents pour effectuer une certaine tâche par la modélisation sémantique et l'analyse des profils des travailleurs (humains et machines) correspondant aux caractéristiques de la tâche et aux exigences d'apprentissage, y compris les connaissances, les aptitudes et les compétences. Cette approche, basée sur la connaissance, comprend une modélisation sémantique et des méthodes quantitatives axées sur la mesure et la corrélation du niveau des compétences humaines et de l'autonomie de la machine, ainsi que sur l'identification du degré de complémentarité homme-machine dans l'exécution d'une tâche assignée. Dans une étude similaire, Mary Cummings [277] présente un cadre permettant de relier le raisonnement basé sur les compétences, les règles et les connaissances à l'expertise et à l'incertitude. Cette taxonomie est conçue pour unifier la conceptualisation de toutes les attributions fonctionnelles pour les systèmes autonomes interagissant avec les décideurs humains afin de mieux comprendre les implications potentielles sur la conception.

Cynthia Breazeal *et al.* [278] présentent une architecture socio-cognitive intégrée pour doter un robot anthropomorphe de la capacité de déduire des états mentaux tels que des croyances, des intentions et des désirs à partir du comportement observable de son partenaire humain. Ils évaluent les capacités de compréhension du robot sur une série de tâches de référence où le robot interagit avec un partenaire humain dans un scénario de coopération et un scénario d'apprentissage, puis reprennent les mêmes expériences avec des humains afin de comparer les performances du robot avec celles des humains.

4.2.1.3 Méthodes basées sur l'apprentissage automatique

Les techniques d'apprentissage automatique (ou *machine learning*, en anglais) sont largement utilisées depuis ces dernières années pour la modélisation du comportement humains. Par exemple, dans [279], les auteurs utilisent le traitement d'images afin d'identifier les actions des travailleurs et exploitent les connaissances liées aux processus, par le biais de modèles de Markov cachés, afin d'identifier les erreurs possibles, les écarts par rapport à l'exécution planifiée ou les situations dangereuses. Dans [280], Dimitris P. *et al.* utilisent un réseau de neurones convolutionnel (CNN) pour effectuer la reconnaissance des gestes et l'estimation de la pose, en vue d'une meilleure introduction des cobots (robots collaboratifs) dans une chaîne de montage industrielle. La reconnaissance des gestes, l'estimation de la pose, l'interaction physique et la notification sonore permettent de créer un système d'interaction multimodale homme-robot.

Al-Yacoub *et al.* [281] utilisent une approche basée sur la co-manipulation homme-robot pour enseigner à un robot industriel comment réagir avec un humain dans une tâche de co-manipulation. Le principal avantage de cette approche est la modélisation intuitive, grâce à des données, qui permet au robot d'avoir un comportement similaire à celui de l'homme. Il est donc possible d'estimer la force provenant de l'humain et les forces causées par le contact avec l'environnement. Un capteur d'activité musculaire sous la forme d'un électromyographe permet d'améliorer la correspondance entre la force/le couple et les déplacements dans les tâches de co-manipulation. Dans une démarche similaire, Yoichi M. *et al.* [282] présentent un modèle computationnel pour la marche côte à côte d'un humain et d'un robot. Pour créer ce modèle, ils observent d'abord comment les humains marchent côte à côte dans un environnement étroit comportant des coins et des obstacles. À partir de ces données, un modèle est conçu pour calculer les positions futures et prédire l'utilité attendue. Le modèle est ensuite implémenté dans un robot pour des évaluations expérimentales. De même, dans [283], les auteurs proposent un algorithme qui apprend les mouvements d'un humain. Pour ce faire, ils enregistrent plusieurs trajectoires typiques caractérisant les modèles de mouvement d'une personne. Les données sont recueillies grâce à des robots mobiles, puis regroupées en différents types de mouvements.

4.2.1.4 Méthodes déterministes

L'une des méthodes déterministes les plus utilisées est le réseau de Petri (*Petri nets (PN)* en anglais). Les PN modélisent le monde en termes de ressources, permettant de gérer efficacement la

concurrence (par exemple en représentant deux agents comme des ressources différentes). Selon Laura M. Hiatt *et al.* [269], cette propriété fait qu'ils sont bien adaptés pour la capture d'activités telles que les mouvements de personnes (ou de groupes de personnes) dans le temps. De ce fait, La plupart des recherches sur la reconnaissance de l'activité humaine sont basées sur la reconnaissance d'actions à partir de données vidéo à l'aide de méthodes de vision par ordinateur [284]. Par exemple, dans [285, 286], les auteurs proposent un framework de reconnaissance des événements de surveillance utilisant les réseaux de Petri, ce qui permet d'exprimer de manière robuste les connaissances sémantiques sur le domaine des événements, ainsi que d'utiliser des algorithmes efficaces pour reconnaître les événements au fur et à mesure qu'ils se produisent dans une séquence vidéo donnée. De même, Z. Xiao *et al.* [287] proposent un framework pour la description, la reconnaissance et la compréhension automatiques d'événements à partir d'enregistrements vidéos.

4.2.2 L'utilisation des AIV dans les systèmes de production

Les AIV, connus également sous l'appellation de robots mobiles autonomes (*Autonomous Mobile Robots - AMR*, en anglais) ne cessent de gagner en attractivité ces dernières années en raison de leur pertinence et de leurs applications dans les systèmes industriels actuels. Selon Alatis M. B. et Hancke G. P. [288], leur capacité à naviguer dans les environnements complexes, sans nécessiter de dispositifs de guidage (physiques ou électromécaniques), les rend plus prometteurs et utiles. Bien qu'on ne puisse dater avec précision l'avènement des AIV, on note que les années 2000 constituent une période charnière dans leur développement et leur vulgarisation à travers le monde. De ce fait, ils bénéficient des dernières avancées en matière d'intelligence artificielle, de capteurs et de traitement des données, offrant ainsi une solution plus agile et adaptable aux besoins de l'I5.0. La figure 4.1 présente un exemple d'AIV actuellement disponible sur le marché.

Soovadeep Bakshi *et al.* [189] traitent le problème de l'optimisation de la planification des tâches pour une flotte d'AIV dans un environnement de production. Face à la complexité croissante de ce problème, les auteurs proposent une nouvelle approche en deux étapes : un algorithme de clustering des tâches suivi d'un algorithme de planification basé sur l'apprentissage par modèle. Leur méthode permet de générer des plannings en temps réel, même pour un nombre important d'AIV et de tâches. Milica Petrović *et al.* [143], quant à eux, développent une méthode d'optimisation multi-objectif basée sur l'algorithme Loup Gris (*Grey Wolf*, en anglais) pour planifier efficacement le transport de ma-

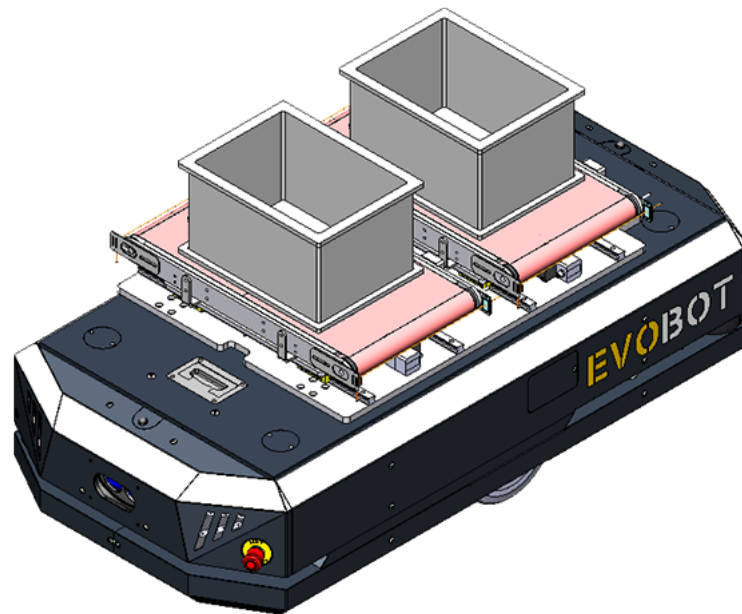


FIGURE 4.1 – Illustration d'un AIV (produit par MG Tech [16])

tériaux dans un système de production utilisant un seul robot mobile. Les résultats obtenus sur un ensemble de benchmarks, ainsi que les expérimentations en environnement réel, soulignent l'efficacité de la méthode. Dans un autre cadre, Paweł Stączek *et al.* [182] soulignent l'importance du jumeau numérique dans l'optimisation des processus de production et de logistique. En se focalisant sur le cas d'un AIV, les auteurs démontrent l'utilité de cette technologie pour tester et améliorer la conception de l'environnement opérationnel avant déploiement réel. Grâce à la simulation de différents scénarios dans un environnement virtuel, il est possible de réduire les coûts et les délais de mise en œuvre tout en augmentant la fiabilité du système final. De leur côté, Chaima Bensaci *et al.* [190] traitent de la complexité inhérente au contrôle d'un système multi-robots opérant dans un environnement dangereux comme un laboratoire d'analyse chimique. Deux architectures de contrôle des AIV, distribuée et hybride, sont étudiées à l'aide de l'analyse théorique des processus système (STPA) afin d'identifier les scénarios de défaillance potentiels. La méthode du processus analytique hiérarchique (AHP) est ensuite utilisée pour sélectionner l'architecture la plus adaptée en fonction des risques identifiés.

Bien que les AIV soient prometteurs, force est de constater qu'ils sont peu employés dans les

problèmes d’ordonnancement joint des tâches de transport et de production (JSSPT). Toutefois, on note les travaux de Jiaojiao Liu *et al.* [138] qui proposent une nouvelle approche pour optimiser simultanément l’ordonnancement de la production et du transport dans un atelier flexible utilisant des AIV. En combinant un modèle de programmation mathématique bi-niveau avec un algorithme génétique auto-apprenant et un algorithme de Dijkstra, les auteurs développent une méthode pour minimiser le temps total de production et de transport. Dans la même veine, You-Jie Yao *et al.* [191] présentent un nouveau modèle de programmation linéaire en nombres entiers mixtes (MILP) pour résoudre le JSSPT intégrant les déplacements des AIV. Ce modèle, basé sur un graphe disjonctif modifié, permet d’obtenir des solutions optimales pour des instances de taille moyenne et démontre son efficacité sur des instances de grandes tailles. Kader Sanogo *et al.* [124] se distinguent en présentant une approche basée sur la simulation multi-agents qui permet de résoudre le JSSPT dans un environnement sans collisions et de comparer les performances des AGV et des AIV sur un échantillon de test réduit. L’étude évalue également différentes stratégies de navigation des AIV et étudie l’impact de plusieurs critères intégrés dans un système de gestion des priorités de passage.

4.3 Description du problème

Comme annoncé plus tôt dans ce chapitre, nous nous intéressons au problème du JSSPT dans lequel les tâches de transport sont réalisées par les AIV, dans un environnement inclusif et partagé avec les opérateurs humains. La présence de travailleurs humains, partageant les mêmes couloirs de circulation que les robots, introduit une dimension dynamique et imprévisible dans le JSSPT classique. Nous nous concentrerons spécifiquement sur la modélisation des déplacements des opérateurs et sur les interactions homme-robot.

4.3.1 Transition des AGV vers les AIV

Dans le problème du JSSPT classique, les tâches de transport sont réalisées par les AGV qui se déplacent en suivant des chemins prédéfinis. Cependant, contrairement à ces derniers, les AIV n’ont pas besoin de guidage et peuvent naviguer de manière autonome et intelligente en planifiant eux-mêmes leurs trajets. Pour ce faire, l’atelier est modélisé par un graphe connexe non orienté, où chaque point d’intérêt (machine, stock, coin et intersection) est représenté par un nœud, et les arêtes représentent les couloirs de déplacement. Lorsqu’un AIV doit effectuer une tâche de transport T_{ij} , celui-ci détermine

4.3. DESCRIPTION DU PROBLÈME

son itinéraire en employant un algorithme inspiré de l'algorithme de Dijkstra pour identifier le chemin le plus rapide.

Vu que les AIV utilisent les couloirs dans les deux sens, Il est nécessaire de prévenir et d'éviter les collisions frontales. Par conséquent, comme décrit dans la section 2.4.4.2, le mécanisme d'évitement des collisions fait intervenir un système de gestion des priorités de passage. En effet, lorsque deux AIV se détectent mutuellement à l'intérieur de leurs rayons de détection, ils réduisent leur vitesse. Puis, une fois à l'intérieur du rayon critique, l'un des AIV s'arrête, tandis que l'autre AIV qui a la priorité initie une manœuvre de contournement. Dès que le chemin se libère, les deux robots accélèrent progressivement jusqu'à retrouver leur vitesse de pointe (1 m/s). La priorité est accordée selon différents critères (la distance, le niveau de batterie, l'instant de début de la tâche ou un tirage aléatoire). Cependant, l'étude menée dans [124] a permis d'identifier le critère de la distance comme étant le plus efficace. En conséquence, dans le reste de ce document, la priorité sera accordée au robot le plus proche de sa destination.

De plus, le remplacement des AGV par des AIV induit une réévaluation des temps de parcours au sein de l'atelier. Grâce à leur grande flexibilité de déplacement, notamment leur capacité à circuler librement dans les deux sens, les AIV ne sont pas limités à des trajets prédéfinis et peuvent emprunter les chemins les plus courts. Par exemple, le trajet entre la machine 1 et la station L/U , qui nécessitait 6 secondes avec les AGV (*cf.* tableau 2.2), est désormais parcouru en seulement 4 secondes par les AIV. Le tableau 4.1 détaille les nouveaux temps de trajet pour l'ensemble des destinations, en tenant compte de la configuration actuelle de l'atelier.

4.3.2 Intégration des humains dans l'atelier

L'atelier est un environnement dans lequel cohabitent des machines, des AIV et des travailleurs humains. La main-d'œuvre humaine, notée $\mathcal{W} = \{W_1, W_2, \dots, W_w\} = W^T \cup W^S \cup W^{PA}$, est divisée en trois catégories : les techniciens ($W^T \subset \mathcal{W}$), assignés à des machines spécifiques M_k ; les superviseurs ($W^S \subset \mathcal{W}$), responsables de la surveillance des activités deux machines distinctes $M_k \neq M_{k'} \in \mathcal{M}$; et le personnel administratif ($W^{PA} \subset \mathcal{W}$), basé dans un bureau commun \mathcal{O} . L'atelier comprend également une salle de pause \mathcal{SP} où les travailleurs peuvent se détendre, mais il n'est pas nécessaire que tous les travailleurs prennent leur pause en même temps.

Les déplacements des travailleurs contribuent à la dynamique de l'atelier. Ils se déplacent entre

4.4. MODÉLISATION DU COMPORTEMENT DES OPÉRATEURS HUMAINS

	L/U	M1	M2	M3	M4
L/U	0	6	8	8	6
M1	6	0	6	8	10
M2	8	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

(a) Layout 1

	L/U	M1	M2	M3	M4
L/U	0	2	4	4	2
M1	2	0	2	6	4
M2	4	2	0	6	6
M3	4	6	6	0	2
M4	2	4	6	2	0

(c) Layout 3

	L/U	M1	M2	M3	M4
L/U	0	4	6	6	4
M1	4	0	2	4	2
M2	6	2	0	2	4
M3	6	4	2	0	2
M4	4	2	4	2	0

(b) Layout 2

	L/U	M1	M2	M3	M4
L/U	0	4	8	10	14
M1	4	0	4	6	10
M2	8	4	0	6	6
M3	10	6	6	0	6
M4	14	10	6	6	0

(d) Layout 4

TABLE 4.1 – Les temps de trajets minimums requis pour les AIV

leurs postes de travail, la salle de pause et le bureau administratif en empruntant les mêmes couloirs de déplacement que les AIV, ce qui peut potentiellement occasionner des rencontres humain-robot comme illustrée dans la figure 4.2. Lorsqu'un humain pénètre dans le rayon de détection d'un AIV, ce dernier réduit de moitié sa vitesse pour éviter toute collision. Si l'humain pénètre dans la zone de sécurité critique, l'AIV s'immobilise immédiatement pendant au moins 5 secondes. Cet arrêt peut se prolonger tant que l'humain reste dans le rayon critique. Toutefois, on suppose que l'humain quittera cette zone dans un délai maximal de 10 secondes, permettant à l'AIV de reprendre sa tâche. Cette hypothèse est inspirée de l'étude de Martin L. *et al.* [123] dans laquelle les auteurs s'intéressent à l'impact des interactions humain-robot sur la taille d'une flotte d'AIV.

Il est important de noter que la présence des humains, en particulier celle des techniciens et des superviseurs, n'impacte pas le travail des machines car on suppose qu'ils réalisent des tâches connexes indépendantes de celles des machines. Néanmoins, il n'en demeure pas moins que leur présence est essentielle au bon fonctionnement global de l'atelier.

4.4 Modélisation du comportement des opérateurs humains

Notre modélisation du comportement des humains se limite essentiellement à la modélisation de leurs déplacements au sein de l'atelier. Pour ce faire, nous utilisons des chaînes de Markov à temps discret (DTMC) qui permettent de représenter les déplacements grâce à un ensemble de probabilités.

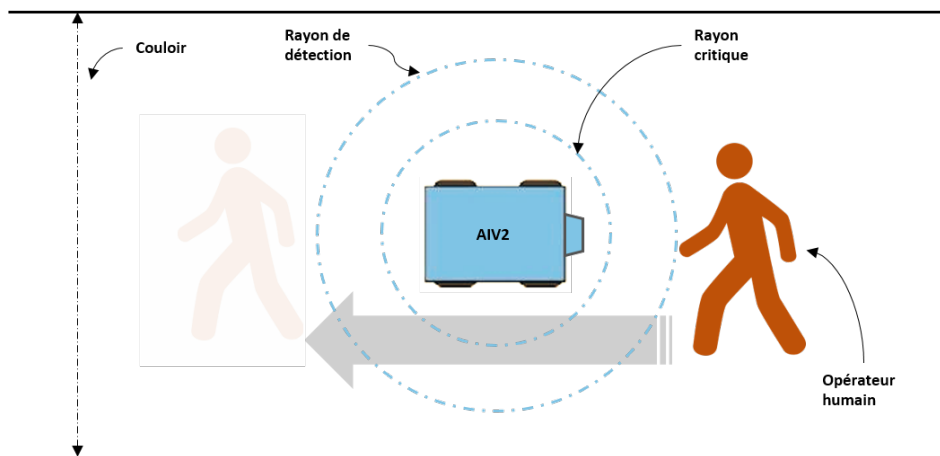


FIGURE 4.2 – Illustration d’une interaction humain-robot

On suppose que les processus cognitifs et/ou les motivations inhérents au déplacement des humains sont pleinement intégrés dans leur désir de se déplacer. Ainsi, ce mouvement peut être caractérisé comme un processus stochastique, qui peut être efficacement modélisé en utilisant une chaîne de Markov pour générer un modèle de simulation par événements discrets. Cette démarche est d’ailleurs évoquée par Rashmi S. et Mathirajan M. [289] et employée par Ki-Hwan B. et Hyun-Jeung Ko [290] dans leur étude.

Bien que cette approche de modélisation puisse sembler simple et restrictive, elle englobe plusieurs informations sur le comportement des opérateurs humains, y compris : leur engagement dans le travail, la nature de leurs tâches, leur adaptation à leur poste de travail, ainsi que leurs compétences. Cette méthode de modélisation a été utilisée dans d’autres études, où elle a été validée par la collecte de données réelles et une mise en œuvre pratique. C’est notamment le cas dans les travaux de Bouaziz N. *et al.* [234, 291] où les auteurs modélisent la nature probabiliste des mouvements humains au sein d’un atelier de production grâce aux chaînes de Markov, afin d’évaluer l’impact des divers comportements humains sur les performances du système de production.

Comme définit par Oswald R. *et al.* [292], un processus stochastique à temps discret est une séquence d’événements discrets potentiels, où la probabilité de chaque événement est uniquement déterminée par l’état atteint lors de l’événement précédent. Dans notre cas, les événements, c’est-à-dire les lieux potentiels qu’un travailleur peut visiter, sont connus et définis en fonction de son rôle dans l’atelier :

- les techniciens ne peuvent travailler que sur les machines sur lesquelles ils ont été affectés. Ils peuvent néanmoins se rendre dans la salle de repos pour prendre une pause ou dans le bureau administratif en cas de besoin ;
- les tâches des superviseurs se limitent à la supervision des activités des techniciens qui ont été affectés sur des machines données. Chaque superviseur a sous sa responsabilité deux machines distinctes. Toutefois, ils peuvent prendre des pauses dans la salle de repos ou à se rendre dans le bureau administratif si nécessaire ;
- le personnel administratif est regroupé dans un « open space » qui sert de bureau administratif. Cependant, comme les autres travailleurs, ils peuvent prendre des pauses dans la salle de repos. Ils peuvent aussi se rendre aux machines de l’atelier pour discuter avec les travailleurs qui y sont, lorsque cela est nécessaire.

En se basant sur cette description, nous avons établi des matrices de transition pour chaque type de travailleur qui fournissent la probabilité de transition entre deux lieux. La matrice P 4.1 est une illustration d’une matrice de transition dans laquelle chaque élément $p_{x,y}, \forall x, y \in \{1, 2, \dots, \lambda\}$ représente la probabilité de passer du lieu x au lieu y en une seule étape.

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,y} & \dots & p_{1,\lambda} \\ p_{2,1} & p_{2,2} & \dots & p_{2,y} & \dots & p_{2,\lambda} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{x,1} & p_{x,2} & \dots & p_{x,y} & \dots & p_{x,\lambda} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{\lambda,1} & p_{\lambda,2} & \dots & p_{\lambda,y} & \dots & p_{\lambda,\lambda} \end{pmatrix}, \quad \text{avec} \quad \sum_{y=1}^{\lambda} p_{x,y} = \sum_{y=1}^{\lambda} Pr(x|y) = 1. \quad (4.1)$$

De plus, il est possible d’associer un graphe à une matrice de transition pour représenter les transitions entre les états du système. On peut l’illustrer avec le graphe G 4.2 qui est un graphe orienté et valué, où V désigne l’ensemble des états, A représente les transitions entre ces états et ϕ indique la probabilité de transition.

$$\begin{aligned} G &= (V, A, \phi), \\ \phi : A &\longrightarrow [0; 1] \\ a &\longmapsto \phi(a). \end{aligned} \quad (4.2)$$

Les figures 4.3, 4.4 et 4.5 illustrent notre modélisation, où chaque type de travailleurs (techniciens, superviseurs et personnel administratif) est représenté par une chaîne de Markov spécifique. Chaque état de la chaîne correspond à une localisation spécifique (machine, salle de pause, bureau administratif) et les transitions entre les états représentent les déplacements des individus. Ainsi, les techniciens

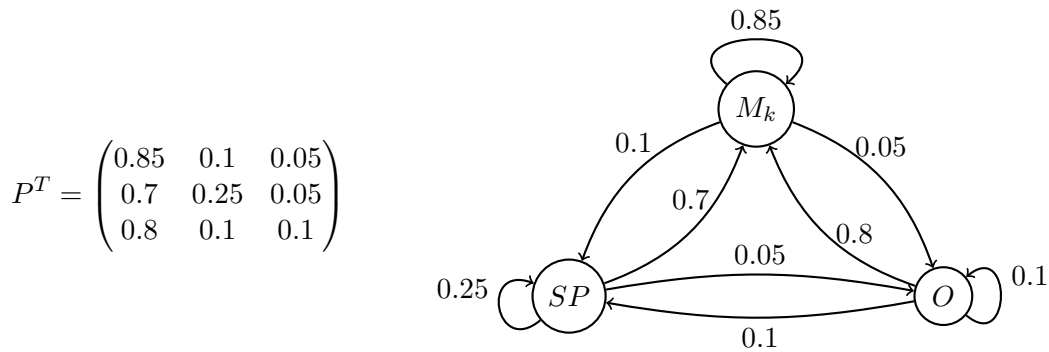


FIGURE 4.3 – La chaîne de Markov des techniciens

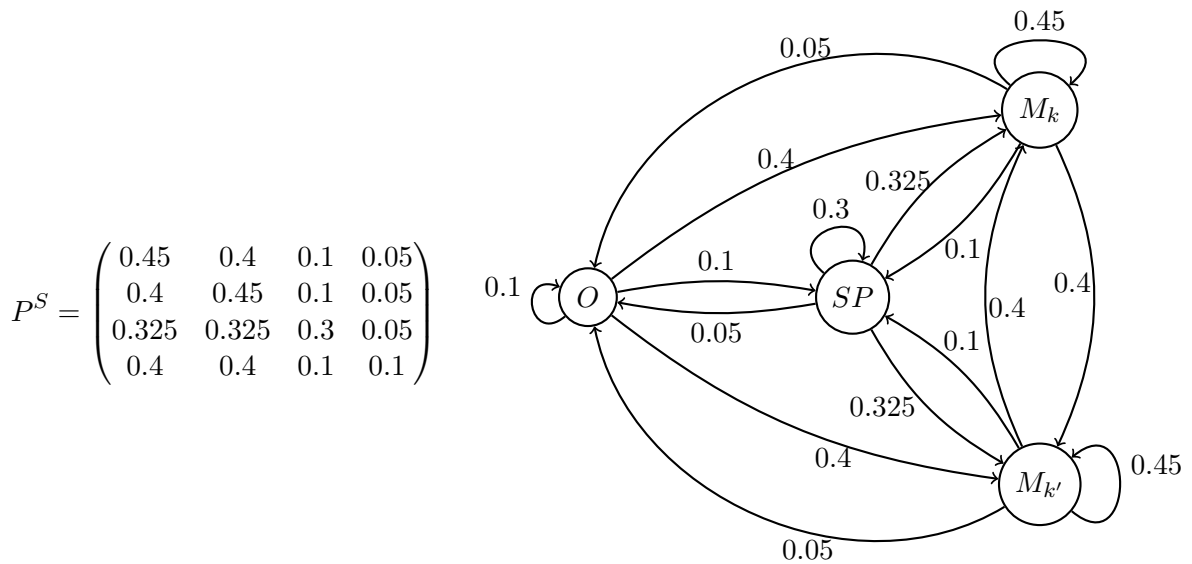


FIGURE 4.4 – La chaîne de Markov des superviseurs

alternent entre leur poste de travail (M_k), la salle de pause (SP) et le bureau administratif (O). Les superviseurs, quant à eux, circulent entre les deux machines (M_k et $M_{k'}$) qu'ils supervisent, la salle de pause et le bureau administratif. Enfin, le personnel administratif se déplace principalement entre leur bureau et la salle de pause, avec des déplacements occasionnels vers les machines. En outre, nous supposons, d'une part, que les travailleurs ont une attitude rationnelle et qu'ils iront jusqu'au bout de leur trajet lorsqu'ils décideront de se déplacer. D'autre part, nous supposons également qu'ils se déplacent toujours suivant le chemin le plus rapide, avec une vitesse constante de 0,5 m/s, et que rien ne vient perturber leurs déplacements.

Par ailleurs, il est important de noter que les probabilités utilisées dans cette modélisation ont été

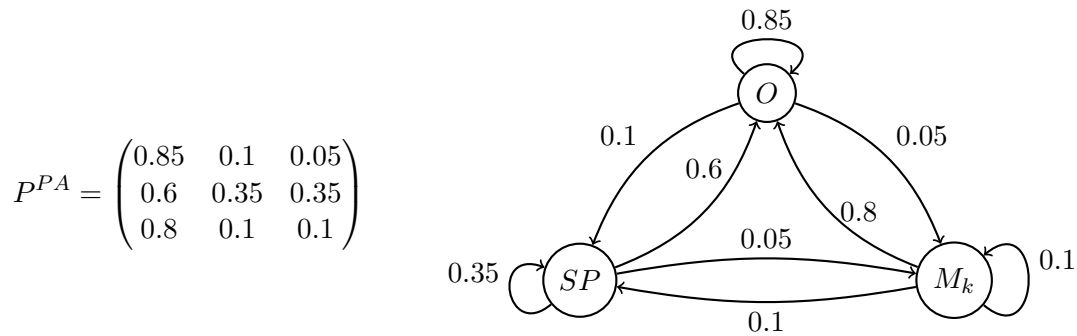


FIGURE 4.5 – La chaîne de Markov du personnel administratif

établies par conjecture, à partir de l’observation d’un cas d’atelier générique. Bien que cette approche exploratoire permette de faire des observations intéressantes, elle souligne la nécessité de mener des études empiriques plus approfondies pour affiner les modèles et valider les résultats. En effet, l’absence de données réelles provenant d’un cas d’étude concret limite la précision de nos modèles. De plus, jusqu’au moment de la rédaction de ce manuscrit, nous n’avions pas connaissance d’un jeu de données benchmark pour ce type de problème. Cela aurait permis de comparer les performances de différentes approches et de favoriser la reproductibilité des recherches.

4.5 Mise à jour du modèle de simulation

L’intégration des travailleurs humains dans le modèle de simulation introduit de nouvelles interactions dans le système multi-agents (MAS) sur lequel repose le simulateur. Deux types d’agents supplémentaires sont ainsi considérés :

- les agents *humains* : Ils participent activement au bon fonctionnement global de l’atelier, même s’ils ne sont pas directement impliqués dans les processus de production. ils exercent des activités connexes (maintenance des machines, préparation des jobs en amont, conception des procédés de fabrication, etc.) aux tâches de transport et de production et sont, de ce fait, aussi importants que les robots et les machines. Ils sont caractérisés par leurs fonctions et leurs besoins de déplacements ;
- les agents *bureaux* : Ces entités représentent les espaces de travail et de repos des humains. Ils jouent un rôle dans la dynamique du système, car ils sont des points de référence pour les déplacements des flux de personnes.

4.5. MISE À JOUR DU MODÈLE DE SIMULATION

La figure 4.6, accompagnée du tableau 4.2, présente la nouvelle version de notre modèle de simulation, qui prend désormais en compte les nouvelles interactions entre les différents agents. Il est important de noter que les risques de collision ne se limitent plus aux interactions robot-robot, mais s'étendent désormais aux interactions humain-robot. Le mécanisme d'évitement de collisions permet donc d'assurer la sécurité de tous les acteurs au sein de l'atelier.

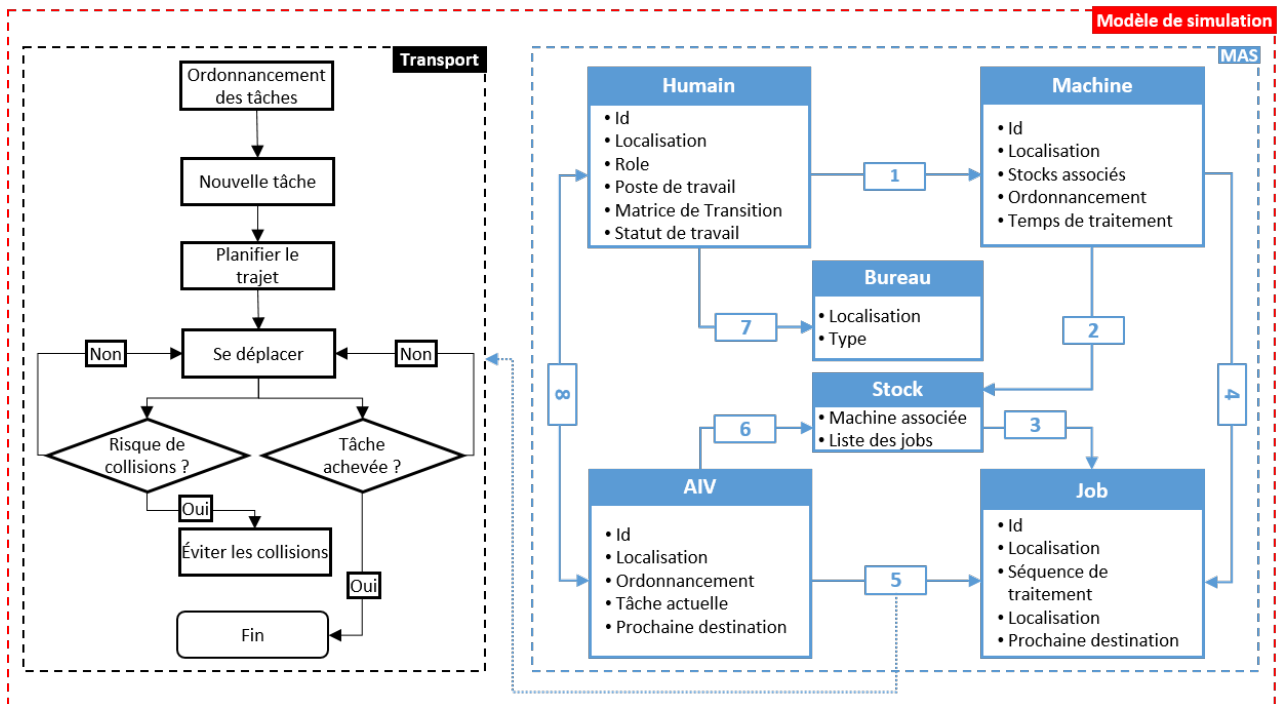


FIGURE 4.6 – Mise à jour du modèle de simulation

Interaction	Description
1	Les humains travaillent sur des poste de travail, près des machines.
2	Les machines récupèrent et déposent les jobs dans leurs buffers d'entrée et de sortie (stocks).
3	Les stocks assurent le stockage des jobs.
4	Les machines assurent le traitement des opérations des jobs.
5	Les AIV se chargent de transporter les jobs.
6	Les AIV déposent ou récupèrent les jobs dans les stocks.
7	Les humains travaillent dans des bureaux et se repose à la salle de pause.
8	Les humains et les AIV partagent les mêmes couloirs de déplacement.

TABLE 4.2 – Description des interactions entre les agents du MAS

4.6 Expérimentations

Les expérimentations de ce chapitre visent, dans un premier temps, à réaliser une évaluation comparative des performances des AGV et des AIV dans le cadre du JSSPT. Dans un second temps, nous nous intéressons à l'influence de la présence humaine sur le comportement et les performances des AIV dans le cadre du JSSPT.

Il faut préciser que les ordonnancements simulés sont issus de l'algorithme d'optimisation proposé par M. Abderrahim *et al.* [14], en considérant le retour des jobs à la station L/U à la fin de leurs traitements respectifs.

4.6.1 Expérience n°1

L'objectif de cette première expérience est de comparer les performances des AIV et des AGV dans le cadre du JSSPT, en intégrant systématiquement la contrainte d'évitement de collisions dans les simulations des ordonnancements issus de l'algorithme de VNS [14].

4.6.1.1 Démarche expérimentale

Les simulations réalisées durant cette expérience ont été faites dans deux configurations : avec des AGV et avec des AIV, en considérant systématiquement les mécanismes d'évitement de collisions. Puis, afin d'étudier l'impact de la taille de la flotte sur les performances du système, nous faisons successivement varier le nombre d'AGV et d'AIV de deux (02) à cinq (05). Cette plage a été choisie car nous observons une stabilisation du makespan, dans 65% des instances de test, à partir de quatre (04) robots. Ce qui suggère que l'ajout de robots supplémentaires n'apportera pas nécessairement de gains significatifs en termes de performance. De plus, un excès de robots par rapport au nombre de jobs, notamment sur les ensembles de jobs 1, 4, 5 et 9, (voir Annexe A) peut engendrer des situations d'inactivité pour les robots en trop.

Les instances de référence de Bilge et Ulusoy [1], initialement conçus pour deux (02) AGV, ont été adaptées aux besoins de l'expérience. Sachant que l'algorithme du VNS offre la possibilité d'augmenter le nombre d'AGV disponibles pour exécuter les tâches de transport, nous avons donc généré des ordonnancements avec trois (03), quatre (04) et cinq (05) robots. Ces ordonnancements ont été également simulés selon les configurations décrites plus haut.

4.6.1.2 Résultats

Le tableau 4.3 présente les résultats de l'étude comparative des performances des AGV et des AIV. Les colonnes *VNS* répertorient les résultats des ordonnancements obtenus grâce à l'algorithme du VNS. Les colonnes *AGV* et *AIV* désignent les résultats des simulations pour chaque type de robots utilisé.

4.6.1.3 Analyse des résultats

L'analyse des résultats présentés dans le tableau 4.3 nous permet de faire les observations ci-dessous.

1. **Dans la quasi-totalité des cas, les AIV présentent de meilleures performances que les AGV** : Les résultats démontrent que la transition des AGV vers les AIV conduit à une réduction significative du makespan. À l'exception des instances EX22, EX62 et EX82 avec deux (02) robots, cette réduction est systématique pour toutes les instances, et même plus prononcée sur les layouts 1 et 4 qui offre plus de flexibilité de routage du trafic. Par exemple, dans le cas de l'EX44 avec 2 robots, la réduction du makespan est de près de 31 secondes, ce qui représente une amélioration de 17%. En moyenne, sur l'ensemble des simulations réalisées avec 2 robots, l'amélioration du makespan est de 9%, ce qui se traduit par un gain de temps moyen d'environ 12 secondes. L'amélioration persiste même avec une augmentation du nombre de robots, comme le montre la figure 4.7.

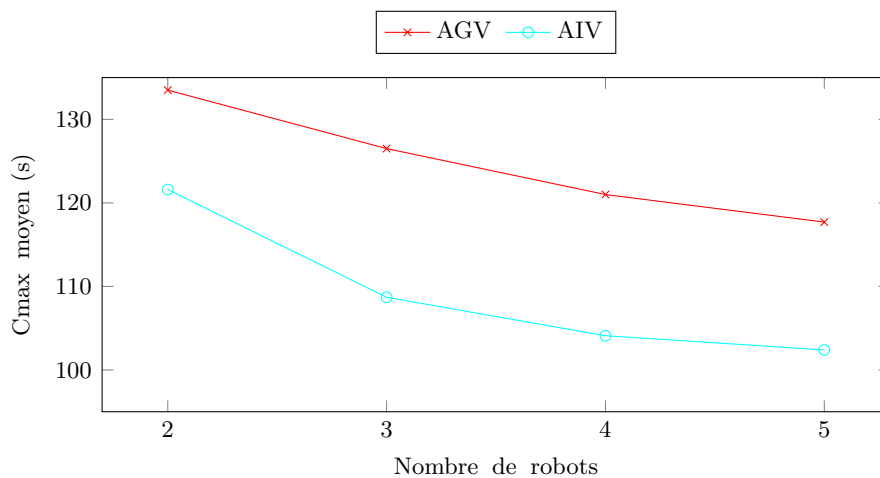


FIGURE 4.7 – Évolution du makespan moyen en fonction du nombre de robots.

4.6. EXPÉRIMENTATIONS

Instance	2 robots			3 robots			4 robots			5 robots		
	VNS	AGV	AIV	VNS	AGV	AIV	VNS	AGV	AIV	VNS	AGV	AIV
EX11	120	120.0	118.5	96	108.2	90.3	88	92.2	84.7	86	93.6	84.2
EX21	122	125.6	119.1	99	119.7	95.0	92	128.1	92.4	92	103.6	92.9
EX31	132	132.2	122.2	105	110.4	104.1	95	108.1	94.5	94	101.3	94.8
EX41	147	147.1	140.1	115	131.0	114.5	100	119.6	101.3	94	100.9	95.5
EX51	116	116.2	103.1	87	96.2	83.2	75	75.5	73.4	75	80.0	73.8
EX61	138	142.8	135.6	116	124.3	114.5	114	140.7	114.5	114	119.0	114.5
EX71	159	160.2	144.8	116	123.2	112.9	99	129.0	95.4	91	107.7	91.3
EX81	167	171.5	154.8	167	220.7	167.7	167	178.9	168.0	167	203.7	167.7
EX91	130	130.1	123.5	115	144.9	115.1	112	137.3	111.2	111	129.1	111.3
EX101	169	173.5	159.7	147	176.6	147.8	142	159.5	140.6	139	147.3	139.7
EX12	99	99.2	76.5	84	93.3	72.8	84	90.4	72.8	84	94.3	73.3
EX22	82	82.6	82.8	81	103.6	81.9	81	82.4	82.1	81	82.8	82.3
EX32	95	95.5	87.7	84	89.5	84.6	84	100.5	84.9	84	96.0	84.9
EX42	109	109.6	94.8	93	100.5	79.9	79	82.0	72.3	79	88.9	72.2
EX52	84	84.2	68.5	72	84.7	64.7	72	87.8	64.0	72	75.2	63.5
EX62	102	102.7	102.8	102	115.8	102.5	102	115.3	102.8	102	121.2	103.3
EX72	101	101.2	96.3	83	99.4	79.1	80	101.5	79.2	80	90.0	78.7
EX82	155	155.7	155.6	155	160.4	155.7	155	156.8	156.2	155	156.1	156.0
EX92	106	106.5	99.5	104	105.8	100.1	102	129.5	102.6	102	120.9	100.1
EX102	145	145.6	143.6	135	139.9	135.9	132	133.5	131.4	132	146.1	132.2
EX13	99	99.2	80.7	90	96.4	74.6	84	98.9	74.7	84	100.7	74.8
EX23	92	92.3	84.4	86	89.6	80.1	84	97.9	83.0	84	111.5	90.9
EX33	104	104.2	89.3	88	94.2	82.8	84	84.9	83.1	84	95.5	84.8
EX43	112	112.2	87.8	92	115.2	84.9	84	109.6	80.8	79	81.6	76.1
EX53	93	93.1	90.3	77	98.1	63.6	73	75.6	64.2	73	85.3	63.6
EX63	110	117.8	100.9	102	110.3	100.9	102	143.1	130.1	102	118.5	101.0
EX73	112	112.1	88.9	91	107.4	80.9	80	82.9	77.7	80	86.8	77.6
EX83	155	155.2	155.5	155	182.4	156	155	159.5	155.4	155	163.7	155.8
EX93	109	109.2	103	102	116.4	98.3	102	148.3	97.7	102	127.5	97.8
EX103	148	148.1	137.9	136	146.1	136.3	135	150.0	133.8	135	160.5	134.0
EX14	144	144.3	128.7	110	121.6	95.5	100	117.7	85.5	96	113.6	82.3
EX24	156	156.3	138.2	119	125.6	112.9	102	109.7	98.5	98	135.3	100.1
EX34	160	162.2	151.6	132	131.7	124.6	107	121.0	102.1	102	121.0	102.7
EX44	180	180.1	149.3	136	157.0	127.0	114	127.6	103.3	105	118.1	92.8
EX54	140	140.2	112.6	104	104.2	86.3	89	99.6	83.9	82	86.2	69.7
EX64	168	168.2	154.4	137	156.1	132.8	119	138.9	124.3	118	134.3	118.5
EX74	191	191.3	169.6	150	148.1	137.8	120	129.1	113	103	119.8	102.8
EX84	190	190.3	166.5	177	194.7	175.2	177	204.5	175.1	177	184.8	180.3
EX94	158	158.1	148.4	131	138.1	130.3	116	135.0	117.1	116	139.6	121.7
EX104	202	203.3	197.2	167	177.2	165.3	152	158.0	154.7	152	164.8	154.7

TABLE 4.3 – Résultats de l'expérience n°1 : comparaison AGV vs AIV

La performance générale des AIV s'explique par leur capacité à ne pas suivre des chemins prédéfinis, permettant ainsi de gagner du temps pendant le trajet. Cependant, si les AIV peuvent optimiser leurs itinéraires en utilisant le chemin le plus rapide, cette approche augmente le risque de collisions, ce qui nécessite des manœuvres d'évitement des collisions. Nos mesures révèlent que

4.6. EXPÉRIMENTATIONS

les AIV peuvent effectuer des manœuvres d'évitement des collisions en 0,8 seconde en moyenne. Cela peut entraîner de légers retards même si, en général, ils n'ont pas d'impacts significatifs.

2. **L'influence de la taille de la flotte dépend du type de robots** : Conformément à l'analyse précédente, l'augmentation du nombre de robots réduit généralement le makespan, comme le témoigne la figure 4.7. Toutefois, un constat apparaît lorsque l'on compare les AGV et les AIV. On remarque qu'une flotte de seulement 2 AIV obtient des résultats presque identiques (C_{\max} moyen = 121,6 s) à ceux d'une flotte de 4 AGV (C_{\max} moyen = 121 s). De plus, même avec une flotte de 5 AGV, le gain sur le makespan est de seulement 3,2% par rapport à 2 AIV. Ce résultat, révélateur de la performance des AIV, suggère donc que l'augmentation de la taille de la flotte d'AGV produit peu, voire aucun, avantage comparé à une flotte de seulement 2 AIV. Bien au contraire, l'augmentation de la flotte va entraîner une augmentation du trafic, accroître les risques de collisions et donc, réduire le gain marginal sur le maksepan. La figure 4.8 illustre bien ce phénomène. On note qu'une augmentation incrémentale du nombre de robots se traduit par une baisse importante sur la moyenne du gain sur le makespan.

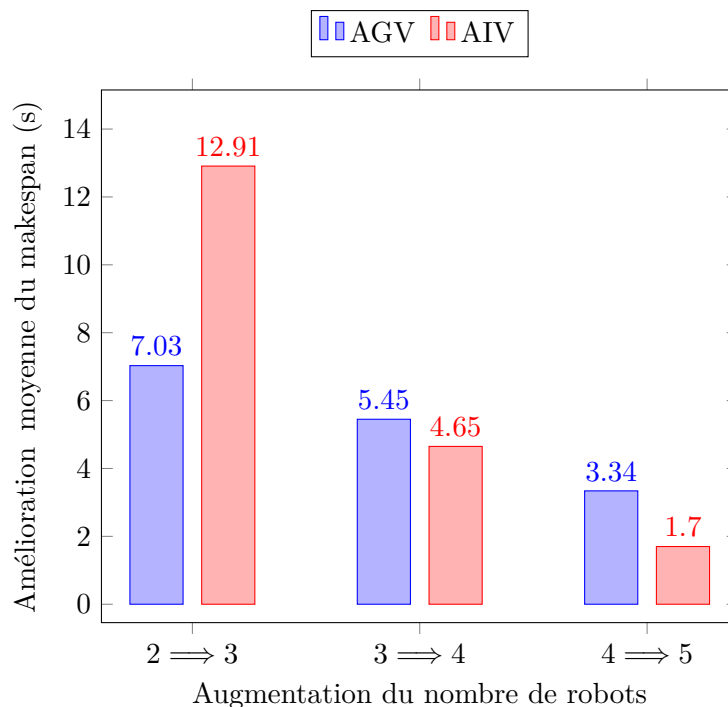


FIGURE 4.8 – Évolution du gain moyen sur le makespan après augmentation du nombre de robot

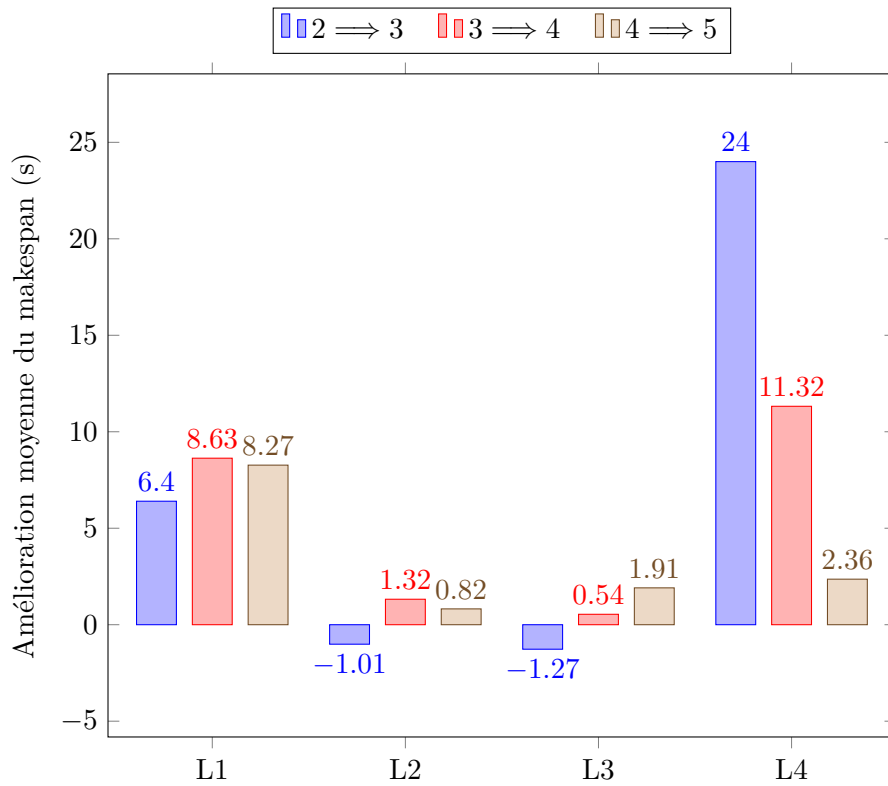
3. **Les layouts des ateliers ont un impact significatif sur le comportement des flottes**

de robots : En plus de l'impact de la taille de la flotte de robots, le layout de l'atelier lui-même joue un rôle important en influençant leur efficacité, comme le montrent la figure 4.9. Dans les simulations réalisées avec les AGV, l'augmentation de la flotte de 2 à 3 AGV entraîne en moyenne, un gain de 6,40 secondes (soit une amélioration de 4,5%) pour le layout 1 et de 24 secondes (soit une amélioration de 14,2%) pour le layout 4, mais une perte 1,01 secondes (soit une dégradation de -0,9%) pour le layout 2 et de 1,27 secondes (soit une dégradation -1,1%) pour le layout 3. Cela révèle que les layouts 1 et 4, qui offrent une plus grande flexibilité de routage que les layouts 2 et 3, supportent mieux l'augmentation du trafic des AGV. Cette tendance persiste même quand on passe successivement de 3 à 4 et de 4 à 5 AGV. De même, le constat se confirme sur les simulations réalisées avec les AIV, où quelle que soit l'augmentation incrémentale de la flotte, les layouts 1 et 4 améliorent beaucoup plus le makespan que les layouts 2 et 3. Ainsi, même si en moyenne, l'augmentation du nombre de robots améliore le système comme le suggère la figure 4.8, l'influence de la taille de flotte est plutôt mitigée en faisant une analyse plus en profondeur, layout par layout, comme le montre figure 4.9.

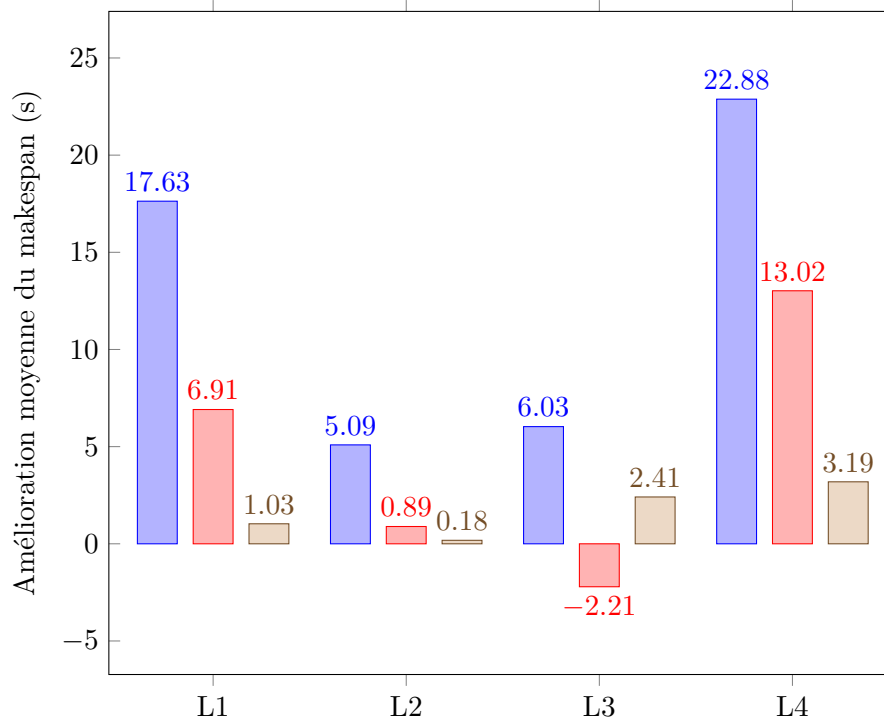
4. **Les ordonnancements avec de longs temps d'attente peuvent perturber les performances des AIV** : L'analyse des diagrammes de Gantt de l'EX82 avec 2 AGV et 2 AIV montre clairement le phénomène des longs temps d'attente. Pour faciliter la lecture, nous avons regroupé ces deux diagrammes dans la figure 4.10, où le diagramme avec les AIV contient un astérisque (*). La première observation est que les AIV optent systématiquement pour le chemin le plus rapide. Par exemple, après avoir effectué $T_{O_{1,1}}^*$, l'AIV2 met 6 secondes pour retourner à la station L/U , alors que l'AGV2 met 8 secondes pour la même tâche. Cependant, le gain de temps est rapidement annihilé par des périodes d'attente prolongées, comme celle entre $T_{O_{4,3}}^*$ et $End_{J_3}^*$. De plus, pendant qu'il est en attente, l'AIV2 obstrue le passage à l'AIV1 qui mettra environ 1s pour l'éviter. Ceci explique pourquoi End_{J_1} et $End_{J_1}^*$ démarrent à peu près au même moment, mais que $End_{J_1}^*$ est légèrement plus long (environ 1s).

Au-delà des considérations techniques abordées dans les analyses ci-dessus, l'un des défis importants associés à l'adoption de toute nouvelle technologie dans les industries, est de pouvoir surmonter la résistance due à l'aversion au changement et à l'incertitude [126]. À cet égard, notre simulateur constitue un outil d'aide à la prise de décision qui permet d'explorer les AIV dans le contexte du JSSPT et de quantifier les gains de productivité potentiels. Les résultats montrent, par exemple,

4.6. EXPÉRIMENTATIONS



(a) Simulations réalisées avec les AGV



(b) Simulations réalisées avec les AIV

FIGURE 4.9 – Évolution du gain moyen sur le makespan par layout.

4.6. EXPÉRIMENTATIONS

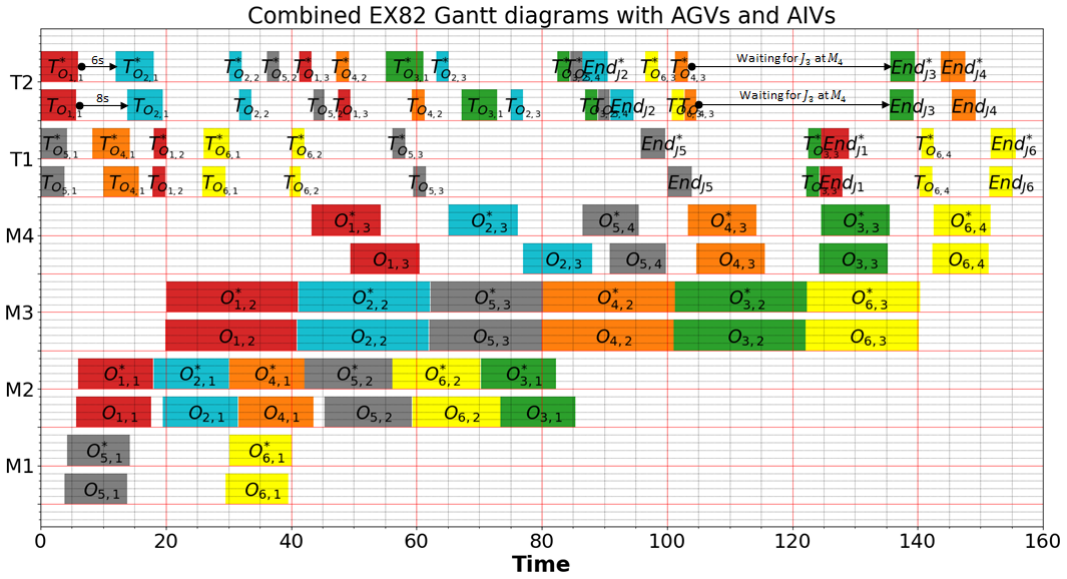


FIGURE 4.10 – Diagrammes de Gantt combinés de EX82 (Kader Sanogo *et al.*, 2024) [17]

qu’une flotte de 3 AIV est plus performante qu’une flotte de 5 AGV. Ces informations peuvent être utiles aux décideurs pour orienter les futurs investissements et optimiser l’affectation des ressources. Néanmoins, malgré des coûts d’investissement initiaux potentiellement plus élevés que ceux des AGV, les AIV peuvent offrir une structure de coûts globale plus favorable. Cela tient à deux facteurs : l’absence de modifications d’infrastructure pour les AIV en raison de leur flexibilité inhérente, et la possibilité d’obtenir des performances similaires avec une plus petite flotte d’AIV comparé aux AGV. Cependant, la mise en œuvre des AIV nécessite un investissement dans la formation et l’éducation du personnel, car ces robots partageront l’espace de travail avec des travailleurs humains.

4.6.2 Expérience n°2

La deuxième expérience consiste à faire cohabiter, dans le même environnement de travail, les travailleurs humains et les AIV. L’objectif de cette expérience est d’étudier l’impact des travailleurs sur les activités des AIV, lorsque tous deux partagent les mêmes couloirs de déplacement. Les AGV ont été exclus de cette expérience car leur fonctionnement nécessite en général qu’ils soient en vase clos.

4.6.2.1 Démarche expérimentale

Pour cette expérience, nous avons utilisés les simulations des ordonnancements réalisés avec deux (02) AIV, en intégrant les travailleurs humains. Les simulations ont été réalisées selon deux (02) scénarios.

1. **Planification statique du chemin (S-AIV)** : c'est-à-dire que les AIV déterminent leurs itinéraires de manière définitive avant de commencer leur déplacement. Ils ne sont pas capables de recalculer leur trajet en temps réel en fonction des événements survenant sur leur parcours.
2. **Planification dynamique du chemin (D-AIV)** : dans ce scénario, les AIV ajustent leurs itinéraires en temps réel en évaluant le niveau de congestion de chaque couloir, privilégiant les zones moins fréquentées. Ainsi, plutôt que de planifier le chemin définitif de sa position initiale à sa prochaine destination, l'AIV navigue dynamiquement à travers des couloirs jusqu'à ce qu'il atteigne sa destination. L'objectif de ce processus est de minimiser la somme $TT + (E_{NH} \times \mu_{tp})$, où : TT est le temps minimum requis pour traverser le couloir ; E_{NH} est l'estimation du nombre d'humains dans le couloir ; et μ_{tp} est le temps moyen perdu lors de la rencontre d'un humain.

La démarche expérimentale adoptée dans chaque scénario est décrite comme suit.

- **Simulation des ordonnancements avec les AIV, sans présence des humains (AIV-SH)** : cette première étape permet de connaître les performances intrinsèques des AIV, sans perturbations extérieures. Les résultats obtenus serviront de base de comparaison.
- **Simulation des ordonnancements des AIV, avec un nombre fixe d'humains dans l'atelier (AIV-FH)** : les humains sont désormais intégrés dans les simulations afin d'étudier les effets de leur présence. Nous avons définis différentes configurations, où le nombre d'humains varie entre 12 et 24 par des incréments successives de 3 humains.

Dans chaque scénario de la configuration AIV-FH, on suppose qu'il y a au moins 2 techniciens par machine, un (01) superviseur pour deux machines et 1 membre administratif pour s'occuper d'un superviseur ainsi que des techniciens sous sa responsabilité. Par exemple, dans la configuration avec 12 travailleurs, on a au total huit (08) techniciens, 2 superviseurs et 2 membres administratifs. Il est important de préciser que les travailleurs ne sont pas synchronisés (ils sont indépendants les uns des autres) et sont susceptibles de se déplacer toutes les 5 secondes. Nous avons aussi établi une corrélation entre le rôle d'un individu et la durée des interactions avec les AIV. Les techniciens et superviseurs, en contact fréquent avec les robots, occasionnent des pertes de temps estimées entre 1 et 1,5 seconde par

4.6. EXPÉRIMENTATIONS

rencontre, tandis que les membres administratifs, moins habitués à cet environnement, entraînent des pertes de temps légèrement supérieures, allant de 1,5 à 2 secondes. Enfin, étant donné la variabilité dans les déplacements des travailleurs, nous avons exécuté chaque simulation 50 fois, tout en retenant le makespan après chaque essai.

4.6.2.2 Résultats

Le tableau 4.4 présente les moyennes du makespan, calculées sur les 50 répétitions d'une simulation, pour chaque instance. On distingue trois (03) cas de simulation :

- Simulation de référence sans présence humaine (AIV-SH) : La colonne *AIV-SH* présente les résultats de la simulation des ordonnancements avec deux AIV, sans la moindre présence humaine.
- Simulation avec planification statique du chemin et présence humaine (S-AIV) : les colonnes *S-AIV* indiquent les résultats des simulations des ordonnancements avec deux AIV, dans la configuration de la planification statique du chemin, en présence de travailleurs humains.
- Simulation avec planification dynamique du chemin et présence humaine (D-AIV) : les colonnes *S-AIV* révèlent les résultats des simulations des ordonnancements avec deux AIV, intégrant la présence humaine, dans la configuration de la planification dynamique du chemin des AIV.

Cette étude permet d'évaluer l'influence de la présence des humains et de comparer l'efficacité des différentes stratégies de planification de chemin face à aux perturbations causées par les travailleurs.

4.6.2.3 Analyse des résultats

La lecture des résultats consignés dans le tableau 4.4 nous permet de faire les observations qui suivent.

1. La présence des humains entraîne inévitablement une augmentation du makespan :

La présence d'humains dans l'atelier augmente de manière significative le makespan, comme le montre le tableau 4.4. Ce phénomène s'explique par la nature perturbatrice de la présence humaine, car chaque rencontre nécessite une interruption des déplacements des AIV pendant environ 1 à 2 secondes. Dans certains cas, ces rencontres peuvent même tripler le temps de trajet entre deux endroits. Pour illustrer cela, nous pouvons examiner la figure 4.11 qui montre les diagrammes de Gantt combinés de EX51 avec 12 humains et sans humains, où le diagramme

4.6. EXPÉRIMENTATIONS

Instance	AIV-SH	12 humains		15 humains		18 humains		21 humains		24 humains	
		S-AIV	D-AIV	S-AIV	D-AIV	S-AIV	D-AIV	S-AIV	D-AIV	S-AIV	D-AIV
EX11	118.5	181.8	166.7	204.2	189.5	229.4	209.4	241.9	231.7	267.6	242.9
EX21	119.1	303.4	293.2	330.8	312.1	352.4	341.2	376.3	357.9	391.7	384.2
EX31	122.2	213.7	198.4	237.2	222.8	254.2	248.1	282.8	269.3	319.8	294.7
EX41	140.1	247.0	228.4	285.2	257.8	311.0	289.5	338.9	304.4	347.8	337.0
EX51	103.1	181.6	168.2	200.1	195.3	219.9	206.7	237.8	233.5	258.3	251.2
EX61	135.6	284.9	269.6	308.4	288.3	343.4	316.8	354.9	342.0	380.9	363.4
EX71	144.8	271.0	253.5	312.7	292.5	335.4	331.0	359.4	358.9	384.2	383.1
EX81	154.8	270.6	250.1	295.4	271.8	314.9	313.9	360.4	316.3	388.8	353.0
EX91	123.5	211.5	190.8	241.1	228.1	265.2	248.0	282.1	260.2	302.3	299.8
EX101	159.7	280.0	245.8	302.3	265.6	335.1	290.2	364.2	318.1	400.5	340.5
EX12	76.5	133.3	121.6	163.4	151.2	176.6	175.1	191.3	198.0	201.8	223.4
EX22	82.8	156.6	138.2	181.6	162.3	202.6	197.7	220.7	214.6	240.4	245.0
EX32	87.8	159.2	156.1	184.8	172.1	206.4	203.9	225.8	233.0	240.8	273.9
EX42	94.8	193.7	183.3	223.4	217.7	250.8	247.6	264.3	290.3	286.6	330.5
EX52	68.5	126.8	122.8	156.1	150.0	172.5	167.2	194.2	202.5	209.4	231.6
EX62	102.9	173.0	160.8	205.5	195.0	231.3	216.1	257.8	254.9	270.9	297.5
EX72	96.3	219.7	190.3	228.9	223.8	273.7	256.0	292.1	290.9	319.5	344.8
EX82	155.6	223.4	204.9	231.9	238.7	291.7	261.9	314.0	291.6	343.1	335.6
EX92	99.5	158.9	149.2	188.7	177.2	212.0	204.8	229.8	227.2	246.6	270.8
EX102	143.6	218.7	199.3	239.7	222.0	276.5	246.5	291.1	286.6	308.6	324.9
EX13	80.7	127.0	128.9	150.0	158.4	162.6	168.5	173.9	191.1	189.9	220.5
EX23	84.4	125.4	129.5	143.1	156.3	163.3	171.4	168.3	187.9	187.0	218.7
EX33	89.3	146.7	147.8	164.8	175.3	186.2	205.8	204.8	229.5	214.5	253.6
EX43	87.8	175.4	191.9	206.9	218.9	221.2	245.4	246.0	268.2	266.0	296.3
EX53	90.3	149.6	148.3	165.4	174.0	179.4	189.7	196.9	202.9	206.7	237.6
EX63	100.9	161.3	162.5	170.7	194.4	195.3	217.9	221.6	236.0	228.2	274.4
EX73	88.9	166.0	171.7	196.4	221.0	220.4	230.2	250.6	270.7	265.2	309.6
EX83	155.5	239.4	258.0	264.7	281.7	290.0	300.4	313.5	334.6	336.0	386.6
EX93	103.0	163.1	162.3	188.0	199.5	208.7	234.8	228.7	263.0	244.8	285.2
EX103	137.9	206.4	217.1	245.1	259.9	270.2	294.6	303.3	331.5	317.3	366.6
EX14	128.7	208.7	183.8	242.3	212.1	267.9	235.3	272.3	258.2	290.7	262.8
EX24	138.2	245.6	213.2	274.1	238.4	300.8	269.9	311.6	274.0	335.7	294.3
EX34	151.6	244.8	216.4	281.5	239.5	307.4	266.4	332.9	291.4	350.8	303.8
EX44	149.3	266.6	242.9	301.4	273.1	325.4	299.9	343.5	315.3	368.7	331.7
EX54	112.6	193.3	166.5	211.7	191.1	236.9	217.6	257.6	235.9	257.7	250.1
EX64	154.4	269.9	226.6	292.1	253.3	333.9	263.2	341.9	280.3	368.4	297.1
EX74	169.6	308.8	282.5	337.7	306.8	389.5	334.0	433.5	349.8	445.1	376.4
EX84	166.5	311.5	285.4	355.3	303.6	369.0	332.8	407.9	354.9	430.0	366.3
EX94	148.4	256.5	224.1	291.0	247.2	307.9	269.8	335.7	292.5	351.0	320.8
EX104	197.2	315.0	285.9	359.9	310.6	392.0	339.6	440.0	360.7	438.7	378.4

TABLE 4.4 – Résultats de l'expérience n°1 : intégration des humains dans l'atelier

avec humains contient la lettre « h ». Par exemple, AIV1 met 8 secondes pour réaliser la tâche $T_{O_{1,3}}$ dans un environnement sans humain, alors que la même tâche ($T_{O_{1,3}}^h$) prend 21 secondes en présence d'humains. Ces délais conduisent inexorablement à un allongement du makespan.

- L'augmentation du makespan est corrélée au nombre d'humains dans l'atelier :** En complément du point précédent, une analyse large sur toutes les instances révèle une corrélation

4.6. EXPÉRIMENTATIONS

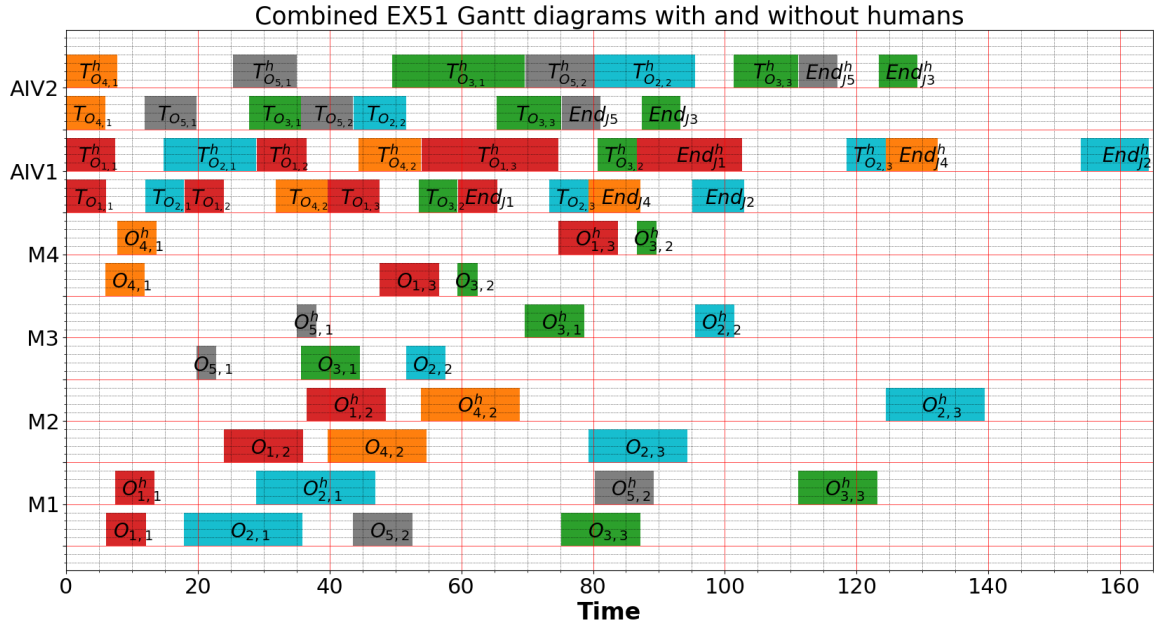


FIGURE 4.11 – Diagrammes de Gantt combinés de EX51 dans les cas AIV-SH et S-AIV avec 12 humains (Kader Sanogo *et al.*, 2024) [17]

directe entre le nombre d'humains présents dans l'atelier et l'allongement du makespan. Par exemple, dans le cas D-AIV avec 12 humains, on observe une augmentation moyenne du makespan de 65% comparé au cas AIV-SH. De plus, cette augmentation grimpe jusqu'à atteindre 154% quand on double le nombre d'humains (c'est-à-dire avec 24 humains dans l'atelier). Ce qui signifie que doubler le nombre d'humains présents dans l'atelier à pour conséquence, a minima, de doubler le retard causé par leur présence. Néanmoins, on observe une augmentation quasi-linéaire du makespan lorsque le nombre d'humains augmente progressivement. Cette observation est également corroborée par les résultats des simulations S-AIV, où même si le retard est plus grand, l'évolution de l'augmentation moyenne du makespan suit la même tendance. La figure 4.12 illustre ce constat.

3. **La planification dynamique du chemin permet d'atténuer l'influence de la présence des humains sur le makespan** : La figure 4.12, qui montre l'évolution de l'augmentation moyenne du makespan en fonction du nombre d'humains, met en évidence l'avantage significatif de la planification dynamique par rapport à la planification statique. En effet, une lecture détaillée des résultats permet de constater que la planification dynamique est toujours plus performante que la planification statique sur les layouts 1 et 4, quel que soit le nombre d'humains

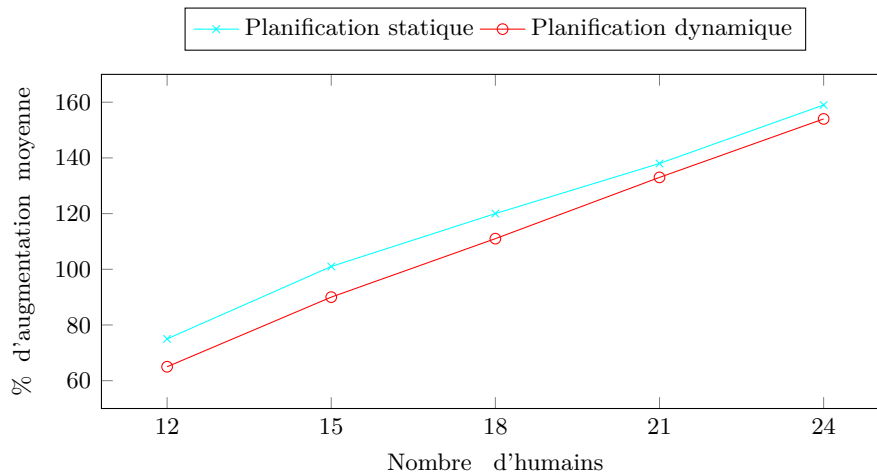


FIGURE 4.12 – Évolution de l’augmentation moyenne du makespan en fonction du nombre d’humains.

présents dans l’atelier. Cependant, cette supériorité n’est pas systématique. Sur le layout 2, la planification dynamique conserve un avantage dans 72% des cas, tandis que sur le layout 3, cet avantage se réduit à seulement 4%. Cette variation de performance peut s’expliquer par la flexibilité de routage des layouts 1 et 4, qui permet aux robots de s’adapter plus facilement aux mouvements imprévus des humains. À l’inverse, la structure plus rigide du layout 3 limite les possibilités de réajustement des trajectoires. Ces résultats suggèrent que le choix de la méthode de planification doit être adapté à la configuration de l’atelier. Pour les environnements dynamiques offrant une grande flexibilité de routage du trafic, la planification dynamique apparaît comme la solution la plus performante.

4. **La planification dynamique des chemins réduit la fréquence de rencontre avec les humains** : La planification dynamique permet aux AIV d’adapter leurs trajectoires en temps réel pour éviter les zones où la densité humaine est élevée. Cette capacité se traduit par une réduction significative du nombre de rencontres entre les robots et les humains dans 95% des cas étudiés, regroupés par layout, comme l’illustre la figure 4.13. Plus précisément, nous avons observé des diminutions de 24%, 17% et 40% du nombre de rencontres respectivement pour les layouts 1, 2 et 4. Seul le layout 3, plus rigide, fait exception à cette tendance, avec une légère augmentation de 2% du nombre de rencontres. Cette observation souligne encore une fois, l’importance de la configuration de l’atelier dans l’efficacité de la planification dynamique. Cependant, bien que généralement bénéfique, la planification dynamique peut entraîner des

4.6. EXPÉRIMENTATIONS

comportements inattendus. En effet, sur les layouts 2 et 3, les AIV ont tendance à effectuer des détours importants pour éviter les zones fortement congestionnées, même si cela allonge leur trajet. Ces détours sont souvent répétés, les AIV étant contraints de rebrousser chemin en raison de l'évolution dynamique de la densité humaine dans les couloirs. Ce phénomène est particulièrement marqué sur le layout 3, où l'augmentation globale de 2% des rencontres d'humains est principalement due à un pic de 6,6% du nombre de rencontres entre les robots et les humains lorsque 24 personnes étaient présentes dans l'atelier. Cette hausse, bien que faible au niveau global, suggère que les efforts des robots pour éviter les collisions peuvent, dans certains cas, dégrader leurs performances.

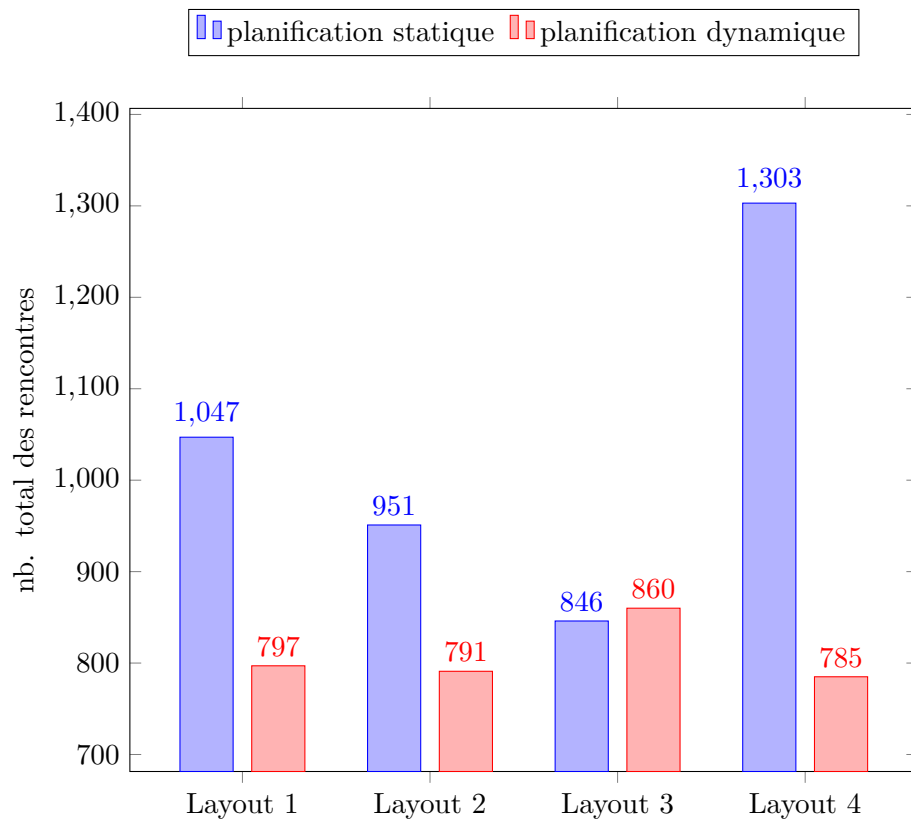


FIGURE 4.13 – Comparaison du nombre d'humains rencontrés sur chaque layout

5. **Les types de travailleurs ont un impact disproportionné sur les retards** : En examinant les différences de comportement des travailleurs, leur fréquence de rencontre avec les AIV et leur présence générale dans l'atelier, nous avons découvert des disparités significatives dans l'impact des différents types de travailleurs sur le temps perdu par rencontre. En moyenne, sur l'ensemble

4.7. CONCLUSION

des simulations, un AIV perd 0,73 seconde lorsqu'il évite un technicien, 0,75 seconde lorsqu'il évite un superviseur et 1,5 secondes lorsqu'il évite un membre administratif. En outre, si l'on observe par exemple la figure 4.14a, dans le cas S-AIV avec 24 humains présents sur le layout 1, un AIV perd en moyenne, environ 91 ($124 \times 0,73$) secondes en évitant les techniciens et environ 71 ($47 \times 1,5$) secondes en évitant les membres administratifs. Étant donné qu'il y a en moyenne, 12 techniciens et 6 membres administratifs présents dans le cas avec 24 humains, cela signifie que même s'il y a deux fois moins de membres administratifs que de techniciens, le fait de les éviter représente 78% du temps perdu à éviter les techniciens. Cet impact disproportionné, lié aux types de travailleurs, se vérifie également sur les autres graphiques de la figure 4.14.

Par ailleurs, il est important de noter que le rôle et l'importance des humains ne sont pas remis en question dans cette étude. Bien au contraire, elle souligne la nécessité de penser et de développer des approches centrées sur l'humain. Malgré une possible augmentation du makespan, l'intégration des humains parmi les technologies employées dans les systèmes industriels devrait améliorer le bien-être, la motivation et la productivité des travailleurs, selon certaines études [293, 294].

4.7 Conclusion

En définitive, ce chapitre a permis d'explorer la transition des AGV vers les AIV dans le cadre du JSSPT, tout en intégrant la présence des travailleurs humains dans l'atelier. En transformant l'atelier de production en un environnement inclusif et collaboratif, où humains et robots coexistent et interagissent dans un espace commun, ce chapitre a contribué à concrétiser l'une des principales ambitions de la transition vers l'Industrie 5.0 (I5.0).

Les expériences menées ont mis en évidence l'efficacité des AIV, qui, même en nombre réduit, surpassent les AGV en termes de performance. Elles ont également permis de quantifier l'impact de la présence humaine sur l'ordonnancement global de l'atelier, révélé par une augmentation significative du makespan. De plus, le passage aux AIV, en remplacement des AGV, représente un tournant technologique stratégique, nécessitant une gestion prudente et des ajustements managériaux. Par conséquent, l'adoption d'une approche basée sur les données s'avère nécessaire pour évaluer les impacts et identifier les implications associées à ce changement. En ce sens, notre simulateur et les expériences réalisées constituent de bons outils pour éclairer les décisions stratégiques.

4.7. CONCLUSION

Toutefois, bien que les modèles proposés dans ce chapitre cherchent à reproduire fidèlement les ateliers de production, ils restent limités par leur représentation des comportements humains qui se concentrent principalement sur les déplacements physiques des individus dans l'espace de travail. Pour une meilleure adéquation avec la réalité, il serait nécessaire d'intégrer dans ces modèles une dimension plus nuancée de l'individu, en tenant compte de l'hétérogénéité des profils (âge, expérience, compétences, etc.), des comportements de groupe d'individus, des processus cognitifs et des facteurs socio-professionnels qui influencent les performances et les comportements des travailleurs.

En outre, ce chapitre met en lumière l'importance de développer de nouvelles méthodes d'optimisation de l'ordonnancement, qui soient à la fois robustes, résilientes et efficaces, en intégrant directement le facteur humain dans la prise de décision des robots de transport. Cette problématique sera approfondie dans le chapitre suivant, où nous nous pencherons sur le développement d'un algorithme d'ordonnancement dynamique, conçu pour intégrer le facteur humain dans le processus décisionnel.

4.7. CONCLUSION

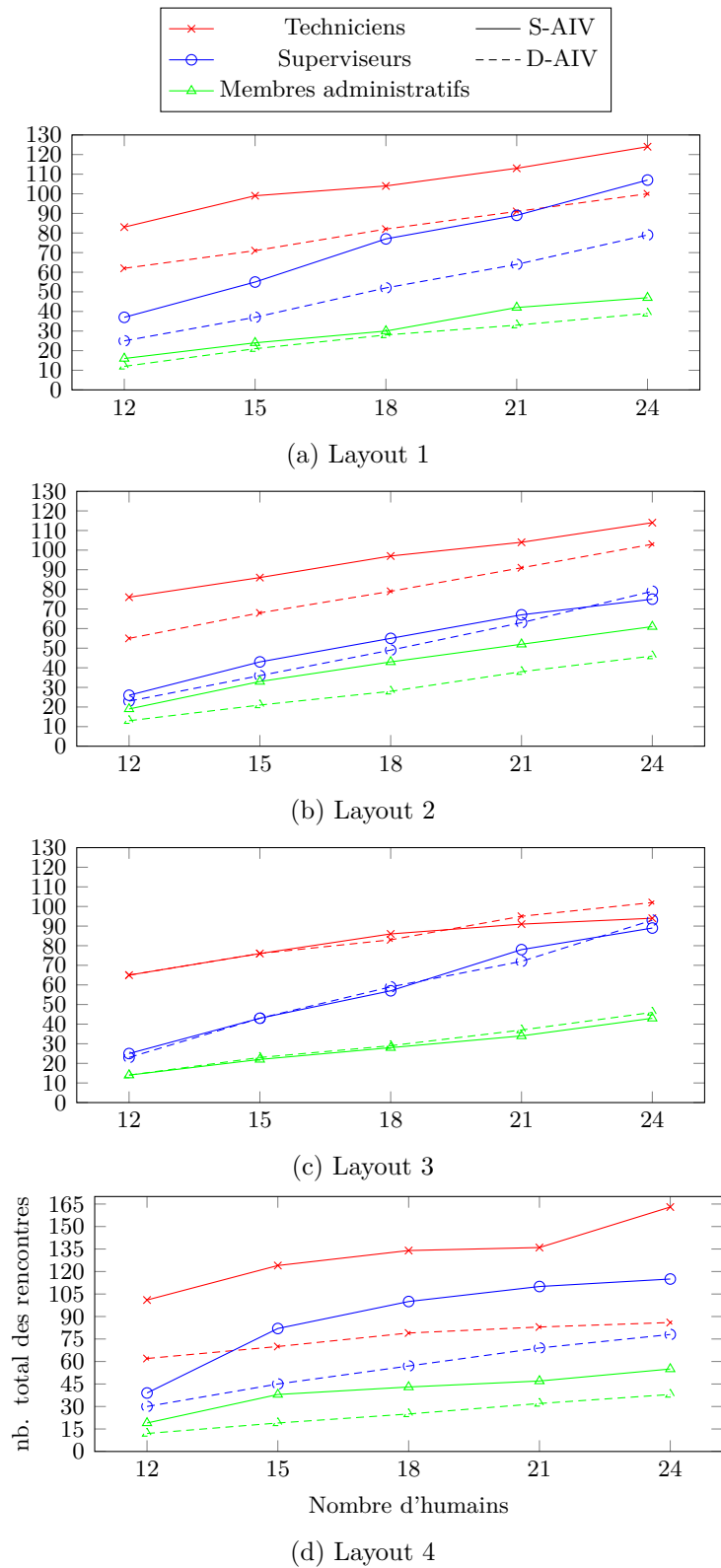


FIGURE 4.14 – Évolution du nombre de rencontres par type de travailleurs

Chapitre 5

Optimisation dynamique de l'ordonnancement dans le contexte de l'industrie 5.0

Contenu

5.1	Introduction	171
5.2	État de l'art	173
5.2.1	Ordonnancement dynamique	173
5.2.2	Allocation des tâches dans un système multi-robots (MRTA)	177
5.3	Ordonnancement dynamique par la théorie des jeux	181
5.3.1	Rappel des concepts théoriques	182
5.3.2	Modélisation de l'ordonnancement	185
5.3.3	Définition des fonctions d'utilités	187
5.3.4	Optimisation de l'ordonnancement	190
5.3.5	Caractéristiques de la fonction d'utilité globale	191
5.4	Implémentation de l'algorithme d'ordonnancement	194
5.5	Expérimentations	195
5.5.1	Description de l'algorithme d'ordonnancement dynamique par entropie	197
5.5.2	Expérience n°1	198
5.5.3	Expérience n°2	206
5.6	Conclusion	211

5.1 Introduction

L'avènement de l'Industrie 5.0 (I5.0), caractérisée entre autres par un environnement industriel inclusif et collaboratif, où humains et robots interagissent de manière étroite, suscite de nombreux espoirs quant à l'amélioration des processus de production et le bien-être des travailleurs. Cependant,

son déploiement s'accompagne aussi de nombreux défis, dont l'un des plus importants reste l'optimisation de l'ordonnancement de l'atelier de production dans un contexte de cohabitation entre les humains et les robots. En effet, alors que l'I5.0 redéfinit les contours de la production industrielle en plaçant l'humain au cœur des processus, il devient impératif de repenser les approches traditionnelles d'ordonnancement pour les adapter à ce nouveau paradigme.

Les méthodes classiques d'ordonnancement, bien qu'efficaces dans des environnements fortement automatisés, montrent leurs limites lorsqu'il s'agit de prendre en compte les interactions complexes entre travailleurs humains et robots mobiles. Assurément, l'introduction des véhicules intelligents autonomes (AIV), combinée à la présence humaine, génère de nouvelles dynamiques qui influencent la performance globale du système de production. Les comportements humains, imprévisibles par nature, ajoutent une couche de complexité supplémentaire, nécessitant des modèles d'ordonnancement plus flexibles et résilients.

Ce chapitre propose donc de développer un algorithme d'ordonnancement dynamique qui intègre les particularités des interactions humain-robot. L'objectif est d'élaborer une solution capable d'adapter en temps réel les plans de production en fonction des mouvements des travailleurs, tout en optimisant l'utilisation des AIV. Pour ce faire, nous proposons de développer un algorithme d'ordonnancement dynamique, basé sur la théorie des jeux, qui sera intégré dans notre simulateur, en plus des modèles des comportements des travailleurs présentés dans le chapitre précédent. Ainsi, nous nous appuyerons sur une approche de simulation-optimisation pour optimiser l'ordonnancement de l'atelier, tout en intégrant les interactions humain-robot dans le processus décisionnel.

La structure du chapitre est organisée comme suit : nous commencerons par une revue bibliographique des approches d'ordonnancement dynamique, en nous concentrons particulièrement les travaux relatifs au JSSPT, ainsi que des approches existantes en matière d'allocation des tâches dans les systèmes multi-robots. Ensuite, nous présenterons notre approche algorithmique, en détaillant les principes de modélisation et la méthode d'optimisation employée. Enfin, nous discuterons des résultats obtenus à travers des études de cas et des simulations, en mettant l'accent sur les performances du nouvel algorithme en termes de réduction du makespan et d'amélioration de la collaboration humain-robot.

5.2 État de l'art

L'ordonnancement dynamique des tâches dans les environnements industriels, notamment ceux impliquant des systèmes multi-robots, est un domaine de recherche actif en constante évolution. Il se distingue de l'ordonnancement statique par sa capacité à s'adapter en temps réel aux changements de l'environnement, tels que les pannes de machines, les arrivées imprévues de nouveaux jobs ou les modifications des priorités dans la production. Les approches dynamiques sont généralement basées sur des algorithmes d'optimisation en ligne, qui prennent des décisions en fonction de l'information disponible à un instant donné.

5.2.1 Ordonnancement dynamique

Face à la dynamique inhérente aux environnements industriels, l'ordonnancement dynamique permet d'adapter l'ordonnancement, en temps réel, en fonction de l'information disponible. En effet, les environnements industriels sont soumis à diverses perturbations qui peuvent modifier l'état du système et affecter ses performances. Dans la littérature [109], ces perturbations sont classées en deux (02) groupes :

- les perturbations liées aux ressources : il s'agit des événements affectant la disponibilité ou les capacités des ressources. On peut citer notamment : les pannes de machine, les indisponibilités d'opérateurs, les indisponibilités ou défaillances d'outils, les limites de chargement, les retards dans l'arrivée ou les pénuries de matériaux, les matériaux défectueux, etc. ;
- les perturbations liées aux jobs : ce sont des événements modifiant les caractéristiques des tâches à exécuter et/ou des jobs à traiter. Ces perturbations concernent les annulations de jobs, les modifications de date d'échéance, les arrivées anticipées ou tardives des jobs, les modifications de priorité des jobs, les modifications du temps de traitement des jobs, etc.

Dans leur étude [295], Zuo-Jun M. S. *et al.* explorent le problème d'ordonnancement des tâches dans des systèmes où l'information sur le type de job peut être imparfaite et évoluer au fil du temps. Ils proposent donc une méthode d'ordonnancement dynamique qui tient compte de l'incertitude des types de jobs et des connaissances acquises à partir des erreurs d'assignation. Une politique optimale, appelée « less-uncertainty-first », est identifiée pour le cas de deux types de jobs, et des heuristiques sont développées pour des cas plus généraux. Zhaojun Qin *et al.* [296] explorent le problème de l'ordon-

nancement dynamique dans le contexte de la personnalisation de masse. Ils proposent une approche basée sur l'apprentissage par renforcement pour résoudre le JSSP dynamique. Les politiques d'ordonnement sont apprises à partir d'instances d'ordonnement statiques en utilisant une approche multi-agent d'apprentissage par renforcement profond (Deep Q-Network - DQN). De nouvelles représentations de l'observation, de l'action, de la récompense et des mécanismes de coopération entre agents sont ainsi proposées. Les politiques d'ordonnement apprises sont ensuite déployées dans un système d'ordonnement dynamique où des temps de traitement stochastiques et des pannes de machines non planifiées se produisent aléatoirement.

Yong Gui *et al.* [297], quant à eux, abordent le problème du FJSSP, une extension du JSSP classique dans un environnement flexible, avec une méthode d'ordonnement dynamique. Plutôt que d'utiliser une règle d'ordonnement unique, la méthode sélectionne dynamiquement la règle la plus appropriée parmi plusieurs options, en utilisant le DQN. Ils introduisent un espace d'action continu en combinant des règles d'ordonnement discrètes avec des poids continus. Une fonction de récompense est définie pour minimiser le retard moyen, et un réseau de politique est entraîné pour générer les décisions d'ordonnement. De même, Du Yu *et al.* [298] proposent une solution basée sur le DQN pour résoudre le même problème, mais en considérant le transport de grue et les temps de préparation. Le modèle proposé optimise simultanément deux objectifs : le makespan et la consommation totale d'énergie. Pour mieux représenter la réalité du problème, huit étapes différentes de transport de grue et trois états typiques des machines sont considérés. Une règle d'identification est conçue pour organiser le transport de grue dans le processus de décodage des solutions. Le modèle DQN utilise 12 caractéristiques d'état et 7 actions pour décrire le processus d'ordonnement.

Xu Yiyi *et al.* [18] explorent le FJSSPT, problème peu abordé dans la littérature, en présence d'incertitudes dans l'environnement de fabrication. Dans une démarche de simulation-optimisation, ils propose une méthode d'optimisation multi-objectifs assurant les affectations de tâches de transport. Cette méthode, s'inscrit dans le cadre des approches Pareto, où tous les objectifs sont améliorés simultanément grâce à un système de classement lexicographique dynamique avec l'entropie et la programmation par objectifs.

5.2.1.1 Approches d'ordonnancement dynamique

D'après Ouelhadj D. et Petrovic S. [109], et Priore Paolo *et al.* [299], on distingue trois (03) approches d'ordonnancement dynamique qui sont : l'ordonnancement entièrement réactif, l'ordonnancement prédictif-réactif et l'ordonnancement pro-actif robuste.

5.2.1.1.1 Ordonnancement entièrement réactif : il se distingue par l'absence de toute planification prévisionnelle. Les décisions d'ordonnancement sont prises localement et en temps réel, en se basant sur des règles de priorité simples. Ces règles, souvent intuitives et faciles à mettre en œuvre, sélectionnent la tâche la plus prioritaire parmi celles en attente. Cependant, cette approche locale présente des limites. En effet, en se focalisant uniquement sur les décisions immédiates, il n'est pas évident d'optimiser les performances globales du système. À l'inverse, les approches d'ordonnancement prévisionnel, bien que plus complexes à mettre en œuvre, offrent une vision plus globale du système et permettent de prendre des décisions qui maximisent les performances à long terme. Toutefois, ces approches ne sont pas robustes face aux perturbations et peuvent très vite s'avérer inexploitable dans un environnement dynamique. Il faut donc trouver un compromis entre la robustesse et la performance

5.2.1.1.2 Ordonnancement prédictif-réactif : c'est l'approche la plus répandue, combinant à la fois la planification à long terme et l'adaptation en temps réel. Ce processus se décompose en deux phases distinctes : d'abord, un ordonnancement prévisionnel initial est généré hors ligne, afin d'optimiser les performances globales du système, sans tenir compte des perturbations qui pourraient survenir. Ensuite, cet ordonnancement initial est régulièrement révisé et ajusté en fonction des événements inattendus, afin de le rendre robuste et de maintenir les performances du système à un niveau satisfaisant.

5.2.1.1.3 Ordonnancement pro-actif robuste : il consiste à élaborer des ordonnancements qui résistent aux perturbations inévitables des environnements industriels. En insérant des marges de manœuvre dans les plannings, cette méthode cherche à garantir un niveau de performance prévisible, généralement mesuré par la robustesse ou la fiabilité de l'ordonnancement. Cependant, la détermination des mesures de prévisibilité adaptées, la gestion de la complexité calculatoire et l'exploration de compromis entre différents critères de performance constituent autant de défis à relever.

5.2.1.2 Techniques d'ordonnancement dynamique

L'ordonnancement dynamique vise à optimiser l'allocation de tâches à des ressources, en temps réel. Contrairement à l'ordonnancement statique, l'ordonnancement dynamique requiert une prise de décision continue et adaptative en fonction de critères variables tels que les délais, les priorités, les ressources disponibles et les événements imprévus. Pour ce faire, plusieurs techniques ont été développées [109], parmi lesquelles on retrouve : les règles de priorité, les heuristiques et les métaheuristiques, les techniques d'intelligence artificielle et les systèmes multi-agents.

5.2.1.2.1 Les règles de priorité : ces règles attribuent une priorité à chaque tâche en fonction de différents critères (date d'échéance, temps de traitement, etc.). Les tâches sont ensuite exécutées par ordre de priorité décroissante.

5.2.1.2.2 Les heuristiques : dans ce contexte, les heuristiques sont des outils d'adaptation spécifiques à un problème. Ils ne garantissent pas de trouver un ordonnancement optimal, mais ont la capacité de trouver des solutions raisonnablement bonnes en peu de temps. D'après Ouelhadj D. et Petrovic S. [109], les heuristiques les plus courantes sont l'heuristique Right-shift (décalage vers la droite), l'heuristique de concordance et l'heuristique de réparation partielle.

5.2.1.2.3 Les métaheuristiques : ce sont des mécanismes plus sophistiqués que les heuristiques, offrant ainsi une plus grande capacité d'exploration de l'espace de recherche. Contrairement aux heuristiques locales qui explorent un voisinage restreint de la solution courante, les métaheuristiques adoptent une perspective plus globale, guidant la recherche vers des régions prometteuses de l'espace de solutions. Cela les rend particulièrement adaptées aux problèmes complexes et de grande taille.

5.2.1.2.4 Les techniques d'intelligence artificielle (IA) : il s'agit notamment des techniques d'apprentissage automatique, qui permettent de prendre des décisions à partir d'un historique d'apprentissage. Le *deep learning*, à travers des réseaux de neurones artificiels, permettent d'apprendre des configurations complexes à partir de données historiques et de prendre des décisions en temps réel. Le *reinforcement learning*, lui consiste à apprendre par essais et erreurs en interagissant avec un environnement. L'agent apprend à prendre les meilleures décisions pour maximiser une récompense.

5.2.1.2.5 Les techniques basées sur les agents : Les techniques d'ordonnement basées sur les agents offrent une approche décentralisée et flexible pour résoudre les problèmes d'ordonnement complexes et dynamiques. Dans ce paradigme, chaque tâche ou ressource est représentée par un agent autonome, capable de prendre des décisions localement et d'interagir avec d'autres agents pour coordonner leurs actions. Les systèmes multi-agent offrent donc une alternative à l'ordonnement centralisé et hiérarchique qui est parfois complexe, difficile à maintenir et à reconfigurer, et lent à satisfaire les besoins des environnements de fabrication complexes et dynamiques.

Face à la complexité et à l'évolution constante de nos environnements de production, nous avons opté pour des techniques d'ordonnement basées sur les agents. Cette approche nous permet de bénéficier d'une grande flexibilité, les agents étant capables de s'adapter en temps réel aux changements de conditions, aux pannes éventuelles et aux nouvelles contraintes. De plus, la décentralisation des décisions renforce la robustesse de notre système. Chaque robot mobile étant représenté par un agent, nous pouvons optimiser en temps réel l'allocation des tâches et minimiser les temps d'arrêt. En outre, l'allocation de tâches dans les systèmes multi-agents, particulièrement dans le contexte des systèmes multi-robots, est un domaine de recherche dynamique, souvent désigné par l'acronyme MRTA (Multi-Robot Task Allocation). La suite de cette section explore les différentes méthodes utilisées pour résoudre le problème du MRTA et justifie le choix de l'approche adoptée dans le cadre de cette thèse.

5.2.2 Allocation des tâches dans un système multi-robots (MRTA)

Comme définit par Hamza Chakraa *et al.* [300], l'allocation de tâches dans les systèmes multi-robots (MRTA) consiste à affecter de manière optimale un ensemble de tâches à un groupe de robots. Ce problème est NP-complet, ce qui signifie qu'il est extrêmement compliqué de trouver une solution optimale, sauf pour des instances de très petite taille. Les contraintes liées à ce problème sont multiples : contraintes temporelles (délais à respecter), contraintes de ressources (énergie, communication), contraintes de compétences, etc. Les objectifs peuvent varier selon les applications, mais ils visent généralement à minimiser le temps total d'exécution des tâches, à maximiser l'utilisation des ressources ou à optimiser un compromis entre ces différents critères. Les applications du MRTA sont nombreuses, touchant divers domaines tels que l'agriculture [301], la sécurité et la défense [302, 303], la santé [304], la logistique [305], etc.

D'après la littérature [300, 306], on distingue deux approches d'allocation des tâches, à savoir :

l'approche centralisée et l'approche décentralisée. Dans le premier cas, il existe un coordinateur central qui dispose de canaux de communication avec tous les autres robots (ou agents). Ce coordinateur central gère les négociations des robots et décide des tâches qui doivent être attribuées à chacun. Dans le second cas, il n'y a pas de coordinateur central. Les robots prennent leurs décisions à partir de leurs perceptions locales de l'environnement et peuvent négocier entre eux. Par conséquent, l'attribution des tâches est faite localement, de manière distribuée. Ainsi, les méthodes de résolution du problème utilisent soit une approche centralisée, soit décentralisée ou même hybride (combinant les deux premières approches). Elles se répartissent généralement entre les méthodes basées sur un système d'enchères ou de consensus, les techniques d'optimisation, la théorie des jeux, les techniques d'apprentissage automatique et les méthodes hybrides.

Par exemple, dans leur étude [307], Whitbrook Amanda *et al.* développent une heuristique pour l'ordonnancement distribué et le réordonnancement dans le temps d'un système multi-agents. Cette heuristique est basée sur un algorithme de regroupement basé sur le consensus (*consensus-based bundle algorithm (CBBA)*) amélioré avec un algorithme d'impact de performance, qui consiste à faire en sorte que tous les agents impliqués dans l'allocation de tâches soient d'accord sur la solution. Leur solution intègre des modules additionnels qui permettent d'éviter de rester piégé dans les minima locaux. Tout comme eux, Keum-Seong Kim *et al.* [308] développent un algorithme d'allocation des tâches décentralisée dans un système avec plusieurs drones autonomes. Cet algorithme est une extension du CBBA qui permet d'améliorer l'efficacité de la communication dans le plan du processus de consensus. Wing Lok Yeung [309] adopte plutôt une allocation des tâches distribuée basée sur le *Contract Net Protocol (CNP)*. Afin de limiter la quantité de messages qui peuvent s'échanger durant l'allocation des tâches, il instaure une restriction d'audience. Bin Du et Shuang Li [310] adoptent aussi une allocation des tâches basée sur le CNP pour développer un modèle de collaboration multidimensionnelle et multi-agents, comprenant le pré-traitement, l'attribution, la programmation et la replanification des tâches en cas d'échec. Toutefois, comme le précisent Aleksis Liekna *et al.* [311], le CNP utilisé tout seul pour l'allocation des tâches, est insuffisant pour atteindre des résultats optimaux. Il faudrait donc qu'il soit couplé avec d'autres techniques pour améliorer ses performances.

De leur côté, Han Wu et Huiliang Shang [140] adoptent la théorie des jeux pour l'allocation des tâches. Ils proposent une méthode d'allocation dynamique des tâches dans un système multi-agents, de manière distribuée, basée sur un jeu de potentiel. Ils ont conçu un modèle théorique de répartition des

tâches avec des expressions des fonctions d'utilité. De même, Archie Chapman *et al.* [312] proposent une approche décentralisée pour l'ordonnancement des agents dans les problèmes d'attribution dynamique des tâches. Leur approche repose sur la formulation du problème comme un jeu de Markov. Efstathios Bakolas et Yoonjae Lee [313] proposent un cadre décentralisé de théorie des jeux pour les problèmes d'attribution dynamique des tâches dans les systèmes multi-agents. Ce cadre tient compte à la fois des récompenses et des coûts associés à l'accomplissement des tâches, ainsi que de l'état actuel de chaque agent. Ils proposent une approche gloutonne dans laquelle les agents négocient entre eux pour trouver un profil d'attribution des tâches mutuellement acceptable. Similairement, Hui Zou et Yan Xi [139] définissent le problème d'attribution des tâches comme un modèle de jeu non coopératif, dans lequel l'utilité de l'agent est dérivée de sa contribution marginale dans l'utilité globale. Ils prouvent que le schéma optimal global d'attribution des tâches réside dans l'équilibre de Nash du jeu.

Les techniques d'apprentissage automatique sont de plus en plus utilisées dans la résolution du MRTA, depuis plus d'une décennie maintenant. Zhu Xiaofei *et al.* [141] étudient un problème d'ordonnancement en temps réel d'un atelier flexible avec des robots. Leur approche consiste à utiliser un modèle mathématique pour optimiser l'ordonnancement des tâches, en tenant compte de la flexibilité du plan de processus. Ensuite, ils utilisent un algorithme d'apprentissage par renforcement pour prendre des décisions d'ordonnancement en temps réel, en fonction de l'état actuel du système. Chaque agent prend des décisions pour différentes tâches en fonction de ce qu'il observe dans son environnement, en suivant les règles que lui a apprises le réseau de politiques. Park Bumjin *et al.* [314] utilisent également l'apprentissage par renforcement pour l'allocation des tâches de plusieurs robots. Le problème est formulé grâce à un processus de décision de Markov (MDP), puis un algorithme d'apprentissage par renforcement profond est utilisé pour trouver la meilleure allocation pour chaque problème. Cette approche est également utilisée dans les travaux de Agrawal Aakriti *et al.* [315, 316].

Les techniques d'optimisation restent encore largement utilisées dans les problèmes de MRTA. Par exemple, Zitouni Farouq *et al.* [317] utilisent les techniques d'optimisation pour optimiser l'allocation des tâches aux robots. Pour ce faire, ils adoptent une méthodologie en deux étapes qui consiste d'une part à réaliser une allocation globale avec l'algorithme des lucioles (*firefly algorithm* en anglais), et de l'autre à faire l'allocation locale en combinant l'algorithme génétique quantique et l'optimisation par colonie d'abeilles artificielles. Dans [318], les auteurs utilisent l'optimisation par colonie de fourmis pour optimiser l'ordonnancement des tâches de plusieurs robots dans les conditions de l'Antarctique.

5.2. ÉTAT DE L'ART

Les auteurs de [319] utilisent une méthode d'optimisation hiérarchique basée sur une approche pas à pas. Cette méthode permet de résoudre le problème complexe d'allocation de tâches à plusieurs robots dans un environnement avec plusieurs stations de travail, en tenant compte de nombreuses contraintes internes et externes. Kong Xiangjun *et al.* [320] présentent une méthode combinée utilisant l'algorithme d'optimisation par essaim de particules et un algorithme glouton pour résoudre le problème d'allocation de tâches à plusieurs robots hétérogènes, en tenant compte de plusieurs objectifs, tels que la minimisation du temps d'exécution des tâches, l'équilibrage de la charge des robots et la recherche d'une solution quasi-optimale.

Par ailleurs, il existe des travaux adoptant une approche de résolution hybride. C'est le cas notamment de Zitouni Farouq *et al.* [321] qui adoptent une méthode hybride pour résoudre le problème d'allocation des tâches en le modélisant comme un problème de recherche et sauvetage. Chaque robot commence par construire un ensemble de survivants à sauver en utilisant l'algorithme de la colonie de fourmis. Puis, les robots résolvent les conflits dans leurs ensembles de survivants en utilisant un mécanisme de coordination basé sur le consensus. Wei Dai *et al.* [322], quant à eux, traitent l'allocation des tâches dans un contexte d'exploration et de destruction. Les robots doivent coopérer pour retrouver des cibles cachées dans l'environnement et les détruire. Les auteurs utilisent une méthode combinant l'approche basée sur les enchères et l'apprentissage automatique pour résoudre le problème.

5.2.2.1 Comparaison des différentes techniques de résolution du MRTA

Afin d'évaluer les différentes approches de résolution du problème MRTA, nous nous sommes appuyés sur l'état de l'art. En particulier, l'étude de Skaltsis George *et al.* [306] offre une synthèse détaillée des méthodes existantes, en les comparant selon différents critères et en discutant de leur pertinence dans divers contextes. Le tableau 5.1 présente une adaptation de cette comparaison à notre étude.

Méthode	Performance	Scalabilité	Robustesse	Coût de calcul
Optimisation exacte	Très Élevé	Faible	Moyen	Très Élevé
Optimisation approchée	Élevé	Élevé	Moyen	Moyen
Machine learning	Très Élevé	Élevé	Élevé	Élevé
Théorie des jeux	Élevé	Élevé	Élevé	Moyen
Enchère et Consensus	Moyen	Moyen	Moyen	Très Élevé

TABLE 5.1 – Comparaison des différentes techniques de résolution du MRTA

L'analyse comparative révèle que les méthodes d'optimisation exacte, bien que très efficaces pour trouver la solution optimale, souffrent d'une complexité algorithmique prohibitive, les rendant peu adaptées aux systèmes de moyenne et grande échelle. De même, les techniques d'enchères et de consensus, avec leur efficacité moyenne, présentent un coût computationnel élevé pour des problèmes de grande taille. En revanche, les méthodes d'optimisation approchée offrent de bonnes performances, en délivrant des solutions sous-optimales dans des temps acceptables. Cependant, elles montrent une robustesse moyenne face aux changements dynamiques qui peuvent survenir.

En outre, bien que les approches fondées sur le machine learning offrent des perspectives intéressantes, notamment pour des problèmes dynamiques, leur complexité en termes de données et de temps d'apprentissage peut constituer un frein dans certains contextes, en particulier pour les systèmes à très grande échelle. Le temps d'apprentissage et la quantité de données requises augmentent avec la taille du problème, et les modèles conçus sont difficilement généralisables à d'autres problèmes. À l'inverse, la théorie des jeux présente l'avantage d'être généralement moins gourmande en ressources computationnelles et de pouvoir traiter des problèmes de grande taille, comme le précise Yichuan Jiang [323]. Elle est particulièrement prometteuse pour les systèmes multi-agents, comme le soulignent Skaltsis George *et al.* [306], ainsi que Lu Yunlong et Yan Kai [324]. De plus, la modélisation des interactions entre les joueurs peut être extrapolée sur d'autres problèmes présentant les mêmes configurations. Par conséquent, nous avons privilégié cette approche dans le cadre de cette thèse. Notre choix se justifie également par le fait que, à notre connaissance, aucune étude n'a encore exploré l'application de la théorie des jeux dans la résolution du JSSPT, alors qu'elle pourrait offrir un bon compromis entre performance, robustesse et complexité. En conséquence, Cette étude constitue une contribution originale à ce domaine de recherche.

5.3 Ordonnancement dynamique par la théorie des jeux

Dans cette section, nous explorerons l'application de la théorie des jeux à l'ordonnancement dynamique des tâches dans un atelier de production. Nous commencerons par un rappel des concepts fondamentaux de la théorie des jeux, tels que les joueurs, les stratégies, les gains et les équilibres de Nash. Nous montrerons ensuite comment ces concepts peuvent être adaptés pour modéliser les interactions entre les transporteurs, considérés comme des agents autonomes. En particulier, nous nous intéresserons aux jeux de potentiel, offrant un cadre analytique particulièrement adapté à l'étude de

5.3. ORDONNANCEMENT DYNAMIQUE PAR LA THÉORIE DES JEUX

systèmes décentralisés où les agents interagissent. Enfin, nous présenterons le problème spécifique de l'ordonnancement dynamique dans ce contexte, en soulignant les défis liés à l'imprévisibilité des événements, en particulier les déplacements des humains, et à la nécessité de prendre des décisions en temps réel.

Symbole	Description
$\mathcal{J} = \{J_1, J_2, \dots, J_I\}$	L'ensemble des jobs.
J_i	Le job i .
O_{ij}	La j^e opération du job i .
t_{ij}^s	L'instant de début de j^e opération du job i .
t_{ij}^e	L'instant de fin de la j^e opération du job i .
t_{ij}^r	Le temps restant avant la fin de la j^e opération du job i .
T_{ij}	La tâche de transport du job i , en vue d'effectuer sa j^e opération.
C_i	Le temps de réalisation (traitement complet) de toutes les opérations du job i .
$C_{max} = \max\{C_1, C_2, \dots, C_I\}$	Le temps maximum de réalisation de tous les jobs (makespan).
$\mathcal{M} = \{M_1, M_2, \dots, M_M\}$	L'ensemble des machines.
M_k	La machine k .
τ_{ijk}	Le temps de traitement de la j^e opération du job i sur machine k .
t_{ijk}^w	Le temps d'attente du job i avant le début de sa j^e opération sur la machine k .
L/U	La station de chargement et de déchargement.
d	Une destination (soit une machine, soit la station L/U).
$\mathcal{V} = \{v_1, v_2, \dots, v_v\}$	L'ensemble des véhicules de transport.
$\mathcal{P} = \{p_1, p_2, \dots, p_p\}$	L'ensemble des joueurs.
$\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_p$	L'ensemble des actions conjointes de tous les joueurs.
$\mathcal{A}_p = \{a_1, a_2, \dots, a_{zp}\}$	L'ensemble des actions du joueur p .
U_g	La fonction d'utilité globale.
u_p	La fonction d'utilité du joueur p .
$\Gamma = \langle \mathcal{P}, \{\mathcal{A}_p\}, \{u_p\} \rangle$	Un jeu.
tt_{pi}	Le temps de trajet entre le joueur p et le job i .
tt_{id}	Le temps de trajet entre le job i et sa prochaine destination d .
$t_{p,end}^q$	L'instant de fin de la tâche de transport du joueur p à l'étape q .
ct_{pi}^q	Le coût (en temps) du choix du job i pour le joueur p à l'étape q .
R_{pi}^q	La réponse du joueur p après avoir choisi le job i à l'étape q .

TABLE 5.2 – Liste des symboles

5.3.1 Rappel des concepts théoriques

La théorie des jeux se concentre sur l'analyse des interactions entre des acteurs indépendants et rationnels (les joueurs) qui font des choix basés sur leurs stratégies, afin de maximiser leur propre utilité [325, 114]. La stratégie d'un joueur se compose de plusieurs actions, et le gain d'une action dépend des actions des autres joueurs et de l'état de l'environnement [326, 327].

5.3.1.1 Définition

Comme décrit par Jaleel Hassan et Shamma Jeff S. [327], un jeu est défini par l'ensemble des joueurs, leurs stratégies (ensemble d'actions) et leurs fonctions d'utilité. Soit $\mathcal{P} = \{p_1, p_2, \dots, p_p\}$ l'ensemble des joueurs, $\mathcal{A}_p = \{a_1, a_2, \dots, a_{z_p}\}$ l'ensemble des actions du joueur p , et $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_p$ l'ensemble des profils d'action conjoints des joueurs. Les préférences du joueur p sur les profils d'action sont définies par sa fonction d'utilité $u_p : \mathcal{A} \rightarrow \mathbb{R}$. Le joueur p choisira le profil d'action α s'il fournit une utilité plus élevée que le profil alternatif α' , c'est-à-dire si $u_p(\alpha) > u_p(\alpha')$. Un profil d'action α est souvent représenté du point de vue du joueur p comme $\alpha = (a_p, \alpha_{-p})$, où α_{-p} représente les actions des autres joueurs. De même, $\mathcal{A}_{-p} = \prod_{p' \neq p} \mathcal{A}_{p'}$ est l'ensemble des actions conjointes de tous les joueurs à l'exception du joueur p .

Les jeux se répartissent en deux catégories [325] : les jeux coopératifs, où les joueurs travaillent ensemble ou forment des coalitions pour atteindre un objectif commun, et les jeux non coopératifs, où chaque joueur privilégie la maximisation de son utilité personnelle. Dans tous les cas, un jeu peut être décrit par l'uplet suivant : $\Gamma = \langle \mathcal{P}, \{\mathcal{A}_p\}_{p \in \mathcal{P}}, \{u_p\}_{p \in \mathcal{P}} \rangle$.

5.3.1.2 Équilibre de Nash

Sous l'hypothèse de la rationalité des joueurs, un jeu peut atteindre un équilibre de Nash (EN). Cet équilibre représente un état stable dans lequel aucun joueur n'a d'intérêt stratégique à s'écarter de son action actuelle, sachant que les actions de tous les autres joueurs restent constantes. Cela signifie que si un profil d'action α^* est un EN, alors :

$$u_p(a_p^*, \alpha_{-p}^*) > u_p(a_p, \alpha_{-p}^*), \forall a_p \in \mathcal{A}_p.$$

Un jeu peut n'avoir aucun EN, ou en avoir au moins un. Un profil d'action α^* est appelé un **équilibre de Nash pur** si pour chaque joueur $p \in \mathcal{P}$, on a :

$$u_p(a_p^*, \alpha_{-p}^*) = \max_{a_p \in \mathcal{A}_p} u_p(a_p, \alpha_{-p}^*). \quad (5.1)$$

5.3.1.3 Jeux de potentiel

Les jeux de potentiels, qui constituent une sous-classe des jeux non coopératifs, sont caractérisés par la présence d'une « fonction de potentielle ». Cette fonction, qui associe à chaque état du jeu

une valeur numérique, possède un gradient qui s'aligne sur le gradient des fonctions d'utilité des joueurs [328]. En d'autres termes, cela signifie que toute variation locale dans la fonction d'utilité d'un joueur, se traduit par une variation identique dans la fonction de potentiel. Plus formellement, un jeu Γ est un jeu de potentiel s'il existe une fonction $\Phi : \mathcal{A} \rightarrow \mathbb{R}$ telle que pour chaque joueur $p \in \mathcal{P}$, pour chaque profil d'actions $\alpha \in \mathcal{A}$, et pour toutes actions quelconques a_p et $a'_p \in \mathcal{A}_p$, la condition suivante est remplie :

$$\Phi(a_p, \alpha_{-p}) - \Phi(a'_p, \alpha_{-p}) = u_p(a_p, \alpha_{-p}) - u_p(a'_p, \alpha_{-p}).$$

Comme Φ est une fonction globale, cela implique que toute amélioration locale de l'utilité du joueur p , en passant de a_p à a'_p , est directement et identiquement répercutée dans la fonction de potentiel [328, 327].

5.3.1.4 Problème d'ordonnement des tâches de l'atelier (JSSPT)

Le JSSPT est connu comme un problème de séquençage d'un ensemble de jobs $\mathcal{J} = \{J_1, J_2, \dots, J_I\}$ à traiter sur un ensemble de machines $\mathcal{M} = \{M_1, M_2, \dots, M_M\}$ afin de minimiser le temps maximum de réalisation de l'ensemble des tâches, c'est-à-dire le makespan (ou C_{max}). Chaque job J_i est composé d'une séquence d'opérations $(O_{i1}, O_{i2}, \dots, O_{in})$ à effectuer consécutivement. Chaque opération O_{ij} , c'est-à-dire l'opération j du job i , ne peut être exécutée que sur la machine $M_k \in \mathcal{M}$ habilitée à le faire, pendant une durée de traitement τ_k . Les machines traitent les opérations les unes après les autres, et rien ne vient interrompre leur traitement, une fois entamé. De plus, on suppose que chaque machine M_k est équipée d'un stock tampon (buffer) d'entrée B_k^I et de sortie B_k^O qui servent respectivement à stocker les jobs avant et après leur traitement.

Le processus de traitement du job i peut être représenté par la séquence ordonnée suivante $(T_{i1}, O_{i1}, T_{i2}, O_{i2}, \dots, T_{in}, O_{in}, T_{in+1})$, où $T_{i,j}$ désigne le transport du job i vers la machine choisie pour exécuter $O_{i,j}$. Ce transport est effectué par un robot mobile autonome (UGV) $v_v \in \mathcal{V}$ qui ne peut transporter qu'un job à la fois. T_{1i} représente la première tâche de transport du job i depuis la station de chargement/déchargement (L/U) jusqu'à la machine désignée pour effectuer O_{i1} . De même, T_{n+1i} désigne le dernier transport du job i depuis la machine désignée pour effectuer O_{in} jusqu'à la station L/U . Dans ce contexte spécifique, le makespan fait référence à la durée du traitement de tous les jobs. Cette durée est mesurée depuis le début de la première tâche de transport jusqu'au retour du dernier job à la station L/U . Il est à noter qu'on suppose que toutes les tâches de transport débutent

initialement à la station L/U .

5.3.2 Modélisation de l'ordonnancement

Dans notre modèle, les tâches de production et de transport fonctionnent de manière indépendante. Cela signifie que les robots peuvent transférer certains jobs tandis que les machines en traitent d'autres simultanément. Toutefois, une contrainte fondamentale demeure : aucun job ne peut être traité par une machine tant qu'un robot ne l'a pas d'abord livré. Dans ce contexte, le problème d'ordonnancement initial peut être réduit à un problème d'allocation de tâches dans un système multi-robots (MRTA). Nous supposons que les robots de transport gèrent collectivement les tâches de transport jusqu'à ce que le job final soit traité. Les machines de traitement fonctionnent selon une politique FIFO (premier entré, premier sorti) et traitent les jobs au fur et à mesure qu'ils sont livrés par les robots.

En outre, les problèmes imprévus (tels que les pannes de machines ou les défaillances de robots), l'arrivée de nouveaux jobs, les déplacements des opérateurs humains et les retards sont inhérents aux environnements dynamiques, ce qui nécessite une approche d'ordonnancement résiliente et adaptable. Cette approche doit réagir dynamiquement aux nouvelles conditions environnementales afin de proposer en permanence des solutions efficaces répondant à tous les critères requis. Par conséquent, nous modélisons l'allocation dynamique des tâches aux robots en mettant en œuvre une approche basée sur la théorie des jeux. Le problème est formulé comme un jeu non coopératif dans lequel les robots sélectionnent indépendamment les jobs, étape par étape, jusqu'à ce que le traitement de tous les jobs soit terminé.

Le problème est modélisé par le jeu suivant : $\Gamma = \langle \mathcal{P}, \{\mathcal{A}_p\}_{p \in \mathcal{P}}, \{u_p\}_{p \in \mathcal{P}} \rangle$, où :

- $\mathcal{P} = \mathcal{V}$ est l'ensemble des joueurs, c'est-à-dire les robots de transport.
- $\mathcal{A}_p = \{a_1, a_2, \dots, a_{z_p}\}$ est l'ensemble des actions du joueur p . Une action a_p peut être représentée par la paire (J_i, d) , qui désigne le transport du job i vers la destination d (une machine ou la station L/U).
- u_p est la fonction d'utilité du joueur p .

Le choix de l'action a_p permet d'intégrer directement la sélection du chemin dans le processus de prise de décision. Lorsque le transport est assuré par des AGV (Véhicules Autonomes Guidés) ou des AIV (Véhicules Autonomes Intelligents) en l'absence d'humains dans l'atelier, l'action a_p se résume principalement au choix du job i , car les AGV suivent des trajectoires prédéfinies et les AIV empruntent

systématiquement les chemins les plus rapides. Cependant, en présence d'humains dans l'atelier, choisir systématiquement le chemin le plus rapide n'est pas toujours la solution la plus adaptée, comme nous l'avons vu dans le chapitre précédent. En effet, lorsque les AIV doivent cohabiter avec des opérateurs humains, il devient nécessaire pour les robots de sélectionner de manière stratégique le meilleur couple $(job, chemin)$ afin de minimiser les perturbations liées au partage des couloirs de circulation. Ce modèle permet ainsi d'intégrer explicitement la présence humaine dans le processus d'allocation des tâches. Toutefois, pour simplifier les définitions et explications à venir, nous résumerons l'action a_p au seul choix du job i . Cette simplification est effectuée sans perte de généralité, car bien que le choix du chemin soit important en présence d'humains, il n'affecte pas la validité des principes fondamentaux du modèle.

Dans ce jeu, les joueurs (les robots) ont pour objectif de terminer le traitement de tous les jobs le plus rapidement possible, minimisant ainsi le makespan. Ils y parviennent en sélectionnant stratégiquement les jobs dans un ordre spécifique (ordonnancement). Chaque fois qu'une décision doit être prise, les joueurs « jouent » le jeu pour trouver le meilleur choix à faire. Ils répètent ce processus jusqu'à ce que tous les jobs soient traités. Ce type de jeu est connu sous le nom de *jeux répétés (ou jeux à étapes multiples)* [329]. De plus, dans notre cas, le jeu est répété un nombre fini de fois, et chaque décision est prise à des pas de temps discrets. C'est ce que l'on appelle un *jeu à répétition finie* [330].

Soit ct_{pi}^q le coût (en temps) que représente le choix du job i pour le joueur p , à l'étape q du jeu ($q = 1, 2, \dots, Q < \infty$). On a :

$$ct_{pi}^q = \begin{cases} \max \{t_{pi}, t_{ij-1}^r\} + t_{id} + t_{ijd} + t_{ijd}^w & \text{si } d \in \mathcal{M} \\ \max \{t_{pi}, t_{ij-1}^r\} + t_{id} & \text{si } d = L/U \end{cases} \quad (5.2)$$

où :

- tt_{pi} est la durée du trajet entre le joueur p et le job i ;
- t_{ij-1}^r est le temps restant avant la fin de la $j - 1^e$ opération du job i ;
- tt_{id} est la durée du trajet entre le job i et sa prochaine destination d ;
- t_{ijd} est le temps de traitement de la j^e opération du job i sur sa prochaine destination d . Si d est une machine, alors $t_{ijd} = \tau_{ijk}$. Sinon, $t_{ijd} = 0$;
- t_{ijd}^w est le temps d'attente du job i avant le début de sa j^e opération sur sa prochaine destination d . Si d est une machine, alors $t_{ijd}^w = t_{ijk}^w$. Sinon, $t_{ijd}^w = 0$.
- $\max \{t_{pi}, t_{ij-1}^r\}$ représente le temps nécessaire au joueur p pour récupérer le job i . Si J_i est

encore en cours de traitement sur la machine M_k et que le temps restant avant la fin du traitement est supérieur au temps de trajet entre le joueur p et M_k , alors le temps requis sera le temps de traitement restant. Dans le cas contraire, si J_i n'est pas en cours de traitement ou si le temps restant avant la fin du traitement est inférieur au temps de trajet, alors le temps requis sera le temps de trajet entre le joueur p et l'emplacement actuel du job i .

Connaissant l'instant de fin de sa précédente livraison (dernière tâche de transport), le joueur p peut prédire l'instant de fin de la prochaine opération du job qu'il a choisi. Cette prédiction, que nous appelons également *la réponse*, est exprimée par :

$$R_{pi}^q = t_{p,end}^{q-1} + ct_{pi}^q, \quad \text{où } t_{p,end}^{q-1} \text{ est l'instant de fin de la dernière livraison.} \quad (5.3)$$

Soit C^q le temps de fin maximum des opérations en cours ou qui sont sur le point de commencer à l'étape q du jeu :

$$C^q = \max \{C^{q-1}, \max_{p \in \mathcal{P}} \{R_{pi}^q\}\}, \quad \text{où } q = 1, 2, \dots, Q < \infty. \quad (5.4)$$

- $C^q = C^{q-1}$ si toutes les opérations de l'étape q seront terminées avant celles de l'étape $q - 1$.
- $C^q = \max_{p \in \mathcal{P}} \{R_{pi}^q\}$ si toutes les opérations de l'étape $q - 1$ se termineront avant la plus longue réponse de l'étape q .

On considère que le cas où $q = 0$ n'a pas d'intérêt, car il correspond à l'étape d'initialisation du jeu ($C^q = 0$). De plus, en se basant sur l'équation 5.4, on peut facilement déduire que C^Q , qui représente le temps d'exécution maximal des opérations à la dernière étape Q , est le makespan global du système ($C^Q = C_{max}$). Puisque C^Q détermine C_{max} , nous pouvons construire dynamiquement, étape par étape, un ordonnancement qui permet le traitement de tous les jobs.

5.3.3 Définition des fonctions d'utilités

Le jeu présenté à la section 5.3.2 vise à minimiser le makespan (C_{max}), en minimisant stratégiquement et dynamiquement le temps d'achèvement maximum (C^q) à chaque étape q du jeu. Pour ce faire, on définit la fonction d'utilité globale du système comme suit :

$$\begin{aligned} \mathcal{U}_g^q : \mathcal{A}^q &\longrightarrow \mathbb{R} \\ \alpha^q &\longmapsto \max \{C^{q-1}, \max_{p \in \mathcal{P}} \{R_{pi}^q\}\} \end{aligned} \quad (5.5)$$

Ainsi, à chaque étape q du jeu, l'objectif est de trouver le profil d'actions α^q qui minimise C^q , défini comme le makespan à cette étape. Pour inciter les joueurs à oeuvrer à la minimisation du makespan,

nous proposons d'utiliser la notion de *contribution marginale* (marginal contribution utility - MCU, en anglais), décrite dans les travaux de Jaleel Hassan et Shamma Jeff S. [327, 331], et de Archie Chapman *et al.* [331]. La MCU d'un joueur mesure l'amélioration apportée au système global lorsqu'il effectue une action particulière. En d'autres termes, elle quantifie la valeur ajoutée de chaque joueur à l'objectif commun. La fonction d'utilité de chaque joueur est donc définie comme suit :

$$u_p^q(a_p, \alpha_{-p}) = \mathcal{U}_g^q(a_p, \alpha_{-p}) - \mathcal{U}_g^q(a_p^0, \alpha_{-p}). \quad (5.6)$$

$a_p^0 \in \mathcal{A}_p$ est appelé une *action nulle*, c'est-à-dire que le joueur p ne choisit aucune action. Ainsi, la contribution marginale du joueur p reflète l'impact de sa décision sur le bénéfice de l'ensemble du jeu. Dans notre modèle, chaque joueur cherche à maximiser sa propre utilité, tout en tenant compte des actions des autres joueurs. Cette approche permet donc de concilier les intérêts individuels avec l'objectif global de minimisation du makespan.

Proposition 1. *En définissant la fonction d'utilité de chaque joueur à l'aide de la MCU, le jeu résultant est un jeu potentiel [327, 140].*

Démonstration. Soit a_p et a'_p deux actions différentes, toutes deux appartenant à l'ensemble d'actions \mathcal{A}_p^q . Le passage de a_p à a'_p peut être exprimé par :

$$\begin{aligned} u_p^q(a_p, \alpha_{-p}) - u_p^q(a'_p, \alpha_{-p}) &= \mathcal{U}_g^q(a_p, \alpha_{-p}) - \mathcal{U}_g^q(a_p^0, \alpha_{-p}) - \left(\mathcal{U}_g^q(a'_p, \alpha_{-p}) - \mathcal{U}_g^q(a_p^0, \alpha_{-p}) \right) \\ &= \mathcal{U}_g^q(a_p, \alpha_{-p}) - \mathcal{U}_g^q(a_p^0, \alpha_{-p}) - \mathcal{U}_g^q(a'_p, \alpha_{-p}) + \mathcal{U}_g^q(a_p^0, \alpha_{-p}) \\ &= \mathcal{U}_g^q(a_p, \alpha_{-p}) - \mathcal{U}_g^q(a'_p, \alpha_{-p}). \end{aligned}$$

Par conséquent, toute amélioration locale de la fonction d'utilité du joueur p , en passant de a_p à a'_p , entraînera une amélioration identique de la fonction d'utilité globale \mathcal{U}_g^q . \square

Définition 1. (Adrian Vetta, 2002) [332]

Soient : γ la fonction d'utilité globale du système ; $\bar{\gamma}$ l'espérance mathématique de γ ; $\bar{\gamma}'$ la dérivée discrète de $\bar{\gamma}$, c'est-à-dire $\bar{\gamma}'_D(X) = \bar{\gamma}(X \cup D) - \bar{\gamma}(X)$, $\forall D \subseteq \Omega \setminus X$; λ_p la fonction d'utilité du joueur p ; $\bar{\lambda}_p$ l'espérance mathématique de λ_p ; $S = (s_1, s_2, \dots, s_P)$ une séquence de stratégie ; et \mathcal{S} l'ensemble de toutes les séquences de stratégies possibles.

Un système d'utilité $(\gamma, \{\lambda_p\}_{p=1}^P)$ est dit « valide » s'il satisfait les deux conditions suivantes :

1. L'utilité du joueur p ($1 \leq p \leq P$) est au moins aussi importante que la perte constatée dans l'utilité globale quand le joueur p décide de ne pas jouer. Autrement dit, pour toute séquence de stratégie $S = (s_1, s_2, \dots, s_P) \in \mathcal{S}$, le système d'utilité $(\bar{\gamma}, \{\bar{\lambda}_p\}_{p=1}^P)$ vérifie l'inégalité suivante,

$$\bar{\lambda}_p(S) \geq \bar{\gamma}'_{s_p}(S_{-p}), \quad \forall 1 \leq p \leq P.$$

2. La somme des utilités de chaque joueur n'est pas supérieure à l'utilité globale du système. En d'autres termes, pour toute séquence de stratégies $S = (s_1, s_2, \dots, s_P) \in \mathcal{S}$, l'inégalité suivante est vraie :

$$\sum_{p=1}^P \bar{\lambda}_p(S) \leq \bar{\gamma}(S).$$

Proposition 2. Le système d'utilité $(\mathcal{U}_g, \{u_p\}_{p=1}^P)$ est un système d'utilité valide.

Démonstration. La première condition de la définition 1 se vérifie aisément. En effet, pour un profil d'actions $\alpha = (a_p, a_{-p})$, on a :

$$\bar{\mathcal{U}}'_g(\alpha) = \Delta \mathcal{U}_g(\alpha) = \mathcal{U}_g(a_p, a_{-p}) - \mathcal{U}_g(a_p^0, a_{-p}) = u_p(\alpha) \quad (5.7)$$

D'où : $u_p(\alpha) \geq \bar{\mathcal{U}}'_g(\alpha)$.

Pour ce qui est de la deuxième condition, elle se vérifie aussi aisément car l'utilité de chaque joueur est définie par sa contribution marginale. Donc la somme des utilités des joueurs ne peut pas excéder l'utilité globale du système. On peut également le vérifier par un raisonnement par l'absurde.

Soient PR la proposition de la deuxième condition et $nonPR$ sa proposition contraire. on a :

$$nonPR : \sum_{p=1}^P u_p(\alpha) > \mathcal{U}_g(\alpha).$$

Supposons que $nonPR$ est vraie. Cela signifie que la somme des contributions marginales des joueurs est supérieure à l'utilité globale du système. En particulier, dans un système avec un seul joueur, on aura :

$$u_p(\alpha) = \mathcal{U}_g(a_p, a_{-p}) - \mathcal{U}_g(a_p^0, a_{-p}) > \mathcal{U}_g(a_p, a_{-p}).$$

Or, $\mathcal{U}_g(a_p^0, a_{-p}) = 0$ dans un système avec un unique joueur. D'où :

$$\mathcal{U}_g(a_p, a_{-p}) > \mathcal{U}_g(a_p, a_{-p}).$$

On aboutit donc à une absurdité. Par conséquent, si $nonPR$ est fausse, on conclut que PR est vraie.

Alors :

$$\sum_{p=1}^P u_p(\alpha) \leq \mathcal{U}_g(\alpha). \quad (5.8)$$

Les équations 5.7 et 5.8 prouvent donc que $(\mathcal{U}_g, \{u_p\}_{p=1}^P)$ est un système d'utilité valide. \square

5.3.4 Optimisation de l'ordonnement

Comme démontré par Ragavendran G. *et al.* [333], dans un jeu de potentiel où les fonctions d'utilité des joueurs sont définies à l'aide de la MCU, l'existence d'un équilibre de Nash (NE) est toujours garantie. De plus, le jeu résultant est un *jeu de potentiel exact*, où la fonction de potentiel est la même que la fonction d'utilité globale. En outre, comme le prouvent Jaleel Hassan et Shamma Jeff S. [327], un profil d'action qui maximise la fonction de potentiel, c'est-à-dire la fonction d'utilité globale, est toujours un EN. Par ailleurs, Monderer Dov et Shapley Lloyd S. [328] prouvent que dans un jeu de potentiel, il existe au moins un EN pur, c'est-à-dire un équilibre tel que l'équation 5.1 est vérifiée pour chaque joueur. Mais, dans notre cas particulier, nous sommes dans un problème de minimisation. Alors, un EN pur est établi par :

$$u_p^q(a_p^*, \alpha_{-p}^*) = \min_{a_p \in \mathcal{A}_p^q} u_p^q(a_p, \alpha_{-p}^*). \quad (5.9)$$

Par conséquent, d'après Hui Zou et Yan Xi [139], et de tout ce qui précède, l'optimisation de l'allocation des tâches dans un système multi-robots, modélisé par un jeu de potentiel exact où l'utilité des joueurs est définie grâce à la MCU, consiste à trouver un EN dans le jeu.

Proposition 3. *Quel que soit l'étape q du jeu, et pour tout équilibre de Nash (EN) α^* , la fonction d'utilité globale \mathcal{U}_g vérifie la condition suivante :*

$$\mathcal{U}_g(\alpha^*) \leq \mathcal{U}_g(\alpha^{opt}), \quad (5.10)$$

où α^{opt} est le profil d'actions optimal.

Démonstration. En général, un EN n'est pas nécessairement une solution optimale, voire même sous-optimale. Néanmoins, supposons que la solution optimale du problème d'ordonnement ne soit pas un EN. Cela signifie qu'au moins un robot pourrait améliorer son utilité en modifiant son action. Étant donné que l'utilité individuelle de chaque robot est définie à partir de sa contribution marginale

et que la fonction d'utilité globale est la fonction de potentiel du jeu, cette amélioration individuelle entraînerait nécessairement une amélioration de l'utilité globale du système. Or, cela contredit le fait que la solution initiale était supposée être optimale. On en déduit donc qu'une allocation de tâches optimale doit nécessairement correspondre à un équilibre de Nash. En d'autres termes, dans notre modèle, la répartition idéale des jobs est toujours une situation stable où aucun robot n'a intérêt à changer d'action. \square

5.3.5 Caractéristiques de la fonction d'utilité globale

La fonction d'utilité globale \mathcal{U}_g^q présente une propriété fondamentale en optimisation combinatoire : la sous-modularité. Cette caractéristique est fréquemment rencontrée dans les problèmes d'optimisation de ressources. La sous-modularité de \mathcal{U}_g^q a des implications importantes pour la résolution du problème d'ordonnancement. Elle garantit l'existence d'algorithmes efficaces pour trouver des solutions approchées, avec des bornes de performance théoriques. De plus, elle permet d'utiliser des techniques spécifiques à l'optimisation de fonctions sous-modulaires. La démonstration détaillée de la sous-modularité de \mathcal{U}_g^q est présentée ci-après.

Définition 2. (Schrijver Alexander, 2003) [334]

Soit Ω un ensemble fini et $f : \Omega \rightarrow \mathbb{R}$ une fonction d'ensemble. f est appelée « fonction sous-modulaire » si l'une des conditions suivantes est satisfaite :

1. $f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$, $\forall X, Y \subseteq \Omega$, avec $X \subseteq Y$ et $x \in \Omega \setminus Y$.
2. $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$, $\forall X, Y \subseteq \Omega$.
3. $f(X \cup \{x_1\}) + f(X \cup \{x_2\}) \geq f(X \cup \{x_1, x_2\}) + f(X)$, $\forall X \subseteq \Omega$, $x_1 \neq x_2 \in \Omega \setminus X$.

Proposition 4. La fonction d'utilité globale $\mathcal{U}_g^q = \max \{C^{q-1}, \max_{p \in \mathcal{P}} \{R_{pi}^q\}\}$ est une fonction sous-modulaire.

Démonstration. Soit $X \subseteq \mathcal{P}$, un sous-ensemble de joueurs et $x_1, x_2 \in \mathcal{P} \setminus X$ deux joueurs distincts.

On a :

$$\begin{aligned} \mathcal{U}_g^q(X) &= \max \{C^{q-1}, \max_{p \in X} \{R_{pi}^q\}\} \\ \mathcal{U}_g^q(X \cup \{x_1\}) &= \max \{C^{q-1}, \max_{p \in X \cup \{x_1\}} \{R_{pi}^q\}\} \\ \mathcal{U}_g^q(X \cup \{x_1, x_2\}) &= \max \{C^{q-1}, \max_{p \in X \cup \{x_1, x_2\}} \{R_{pi}^q\}\} \end{aligned}$$

Or,

$$\mathcal{U}_g^q(X \cup \{x_1\}) = \begin{cases} U_g^q(X), & \text{si } R_{x_1i}^q \leq U_g^q(X) \\ R_{x_1i}^q, & \text{si } U_g^q(X) \leq R_{x_1i}^q \end{cases} \Rightarrow \mathcal{U}_g^q(X \cup \{x_1\}) \geq U_g^q(X). \quad (5.11)$$

Similairement,

$$\mathcal{U}_g^q(X \cup \{x_2\}) \geq U_g^q(X). \quad (5.12)$$

De plus,

$$\mathcal{U}_g^q(X \cup \{x_1, x_2\}) = \begin{cases} U_g^q(X), & \text{si } R_{x_1i}^q \leq U_g^q(X) \text{ et } R_{x_2i}^q \leq U_g^q(X) \quad (i) \\ R_{x_1i}^q, & \text{si } U_g^q(X) \leq R_{x_1i}^q \text{ et } R_{x_2i}^q \leq R_{x_1i}^q \quad (ii) \\ R_{x_2i}^q, & \text{si } U_g^q(X) \leq R_{x_2i}^q \text{ et } R_{x_1i}^q \leq R_{x_2i}^q \quad (iii) \end{cases} \quad (5.13)$$

Si le cas 5.13-(i) est vérifié, alors : $\mathcal{U}_g^q(X \cup \{x_1\}) = \mathcal{U}_g^q(X \cup \{x_2\}) = \mathcal{U}_g^q(X)$. D'où,

$$\mathcal{U}_g^q(X \cup \{x_1\}) + \mathcal{U}_g^q(X \cup \{x_2\}) \geq \mathcal{U}_g^q(X \cup \{x_1, x_2\}) + \mathcal{U}_g^q(X).$$

Si le cas 5.13-(ii) est vérifié, alors : $\mathcal{U}_g^q(X \cup \{x_1\}) = \mathcal{U}_g^q(X \cup \{x_1, x_2\}) = R_{x_1i}^q$. Cependant, d'après l'équation 5.12, $\mathcal{U}_g^q(X \cup \{x_2\}) \geq U_g^q(X)$. D'où,

$$\mathcal{U}_g^q(X \cup \{x_1\}) + \mathcal{U}_g^q(X \cup \{x_2\}) \geq \mathcal{U}_g^q(X \cup \{x_1, x_2\}) + \mathcal{U}_g^q(X).$$

Le cas 5.13-(iii) est similaire au cas 5.13-(ii), ce qui conduit au même résultat. Par conséquent, conformément à la troisième condition de la définition 2, $\forall X \subseteq \mathcal{P}, x_1 \neq x_2 \in \mathcal{P} \setminus X$,

$$\mathcal{U}_g^q(X \cup \{x_1\}) + \mathcal{U}_g^q(X \cup \{x_2\}) \geq \mathcal{U}_g^q(X \cup \{x_1, x_2\}) + \mathcal{U}_g^q(X). \quad (5.14)$$

On conclut donc que \mathcal{U}_g^q est une fonction d'ensemble sous-modulaire. \square

Définition 3. (Adrian Vetta, 2002) [332]

Une fonction d'ensemble f est non décroissante si :

$$f(X) \leq f(Y), \forall X \subseteq Y \subseteq \Omega.$$

Proposition 5. La fonction d'utilité globale \mathcal{U}_g^q est une fonction d'ensemble non décroissante.

Démonstration. En s'inspirant de l'équation 5.11, on démontre aisément que :

$$\forall X \subseteq Y \subseteq \mathcal{P}, \quad \mathcal{U}_g^q(X) \leq \mathcal{U}_g^q(Y). \quad (5.15)$$

\square

Proposition 6. *Quel que soit l'étape q du jeu, et pour tout équilibre de Nash α^* , la fonction d'utilité globale \mathcal{U}_g vérifie la condition suivante :*

$$\mathcal{U}_g(\alpha^*) \geq \frac{1}{2}\mathcal{U}_g(\alpha^{opt}), \quad (5.16)$$

où α^{opt} est le profil d'actions optimal.

Démonstration. (Théorème n°5, (Adrian Vetta, 2002) [332])

Soit γ une fonction d'ensemble sous-modulaire non décroissante. Si $(\gamma, \{\lambda_p\}_{p=1}^P)$ est un système d'utilité valide, alors pour tout équilibre de Nash $S \in \mathcal{S}$, on a :

$$OPT \leq (1 + \delta(\gamma))\bar{\gamma}(S) \leq 2\bar{\gamma}(S),$$

où $\delta(\gamma)$ est la courbure discrète de la fonction γ . □

La sous-modularité de la fonction d'utilité globale \mathcal{U}_g^q offre de nombreuses perspectives pour la résolution de notre problème d'ordonnancement. En effet, les fonctions sous-modulaires étant les analogues discrets des fonctions convexes [327, 334], de nombreux résultats théoriques de l'optimisation convexe peuvent être adaptés à notre contexte. Notamment, il a été démontré dans les travaux de Laszlo Lovasz [335] et d'Edmonds Jack [336], que le problème d'optimisation d'une fonction sous-modulaire peut être résolu en temps polynomial. Cette propriété ouvre la voie à l'utilisation d'algorithmes gloutons pour trouver des solutions approchées efficaces à notre problème d'ordonnancement. Par exemple, l'algorithme de la meilleure réponse, qui consiste à itérativement améliorer l'utilité d'un joueur en lui attribuant la tâche qui minimise sa contribution marginale (voir l'équation 5.1), peut être utilisé pour converger vers un équilibre de Nash pur. Bien que cet équilibre ne soit pas nécessairement optimal, il garanti au moins 50% de sous-optimalité car tout équilibre de Nash α^* vérifie l'inégalité suivante :

$$\frac{1}{2}\mathcal{U}_g(\alpha^{opt}) \leq \mathcal{U}_g(\alpha^*) \leq \mathcal{U}_g(\alpha^{opt}).$$

Par ailleurs, la relation entre la sous-modularité et l'optimisation convexe permet d'envisager des algorithmes distribués pour résoudre notre problème dans un contexte multi-agents. Ces algorithmes permettraient de répartir les calculs entre les différents robots et d'adapter en ligne les décisions d'ordonnancement en fonction de l'évolution de l'environnement.

5.4 Implémentation de l'algorithme d'ordonnancement

Nous avons formalisé le problème d'ordonnancement joint des tâches de transport et de production (JSSPT) comme un problème d'allocation de tâches dans un système multi-robots (MRTA). Dans cette formulation, chaque robot de transport correspond à un agent décisionnel et les tâches à allouer représentent les différentes opérations à effectuer. Cette transformation nous permet de tirer parti des outils théoriques de la théorie des jeux, notamment les équilibres de Nash. La modélisation du jeu garantit l'existence d'équilibres de Nash purs, ce qui permet d'adopter une approche gloutonne qui consiste à itérativement améliorer la situation d'un robot en lui attribuant la tâche qui minimise sa contribution marginale dans l'augmentation du makespan. Le pseudo-code de l'algorithme 3 et la figure 5.1 décrivent le fonctionnement global de l'algorithme d'ordonnancement qui est exécuté dynamiquement par chaque robot.

L'algorithme fonctionne en plusieurs étapes pour déterminer quelle action un robot doit entreprendre à un instant donné. Le processus commence par une vérification de l'état des stocks et se termine par la sélection optimale du job et d'un chemin, en tenant compte de divers facteurs tels que les temps de traitement, les déplacements, et l'utilité globale du système. L'algorithme commence par vérifier l'état de la station de déchargement. Si le nombre de jobs dans ce stock n'est pas égal au nombre total de jobs à traiter, le robot doit réaliser un choix afin de poursuivre le traitement des jobs restants. Ensuite, l'algorithme vérifie l'état de la station de chargement. Si tous les jobs ont quitté cette station, le robot devra faire un choix parmi les jobs qui ne sont pas encore ciblés par d'autres robots, et qui sont soit en cours de traitement ou soit dans les stocks de sortie des machines de l'atelier. Si des jobs sont encore présents dans la station de chargement, l'algorithme sélectionnera simplement l'un des jobs qui ne sont pas encore ciblés par d'autres robots.

Une fois les jobs restants identifiés, l'algorithme les parcourt un à un pour décider du meilleur job à transporter. Pour chacun d'eux, il vérifie s'il lui reste encore des opérations à effectuer, selon son planning de traitement. Si ce n'est pas le cas, il estime le temps nécessaire pour transporter ce job jusqu'à la station de déchargement. Sinon, il récupère les informations de l'opération suivante à réaliser et calcule la réponse ($R_{p_i}^q$) associée à ce choix. Puis, il procède à un calcul itéré de l'utilité ($u_p(a_p, \alpha_{-p})$) de l'action actuelle, en fonction de celle des autres robots. Au bout de ce processus, le système désigne la meilleure action et met à jour l'utilité globale du système (le makespan à cette

étape du jeu). Enfin, l'algorithme vérifie si plusieurs robots n'ont pas effectué le même choix. Si tel est le cas, le premier à avoir choisi conserve l'action, et les autres sont invités à effectuer d'autres choix. Si aucun conflit n'est détecté, le robot enregistre l'ordre à exécuter et se met en mouvement.

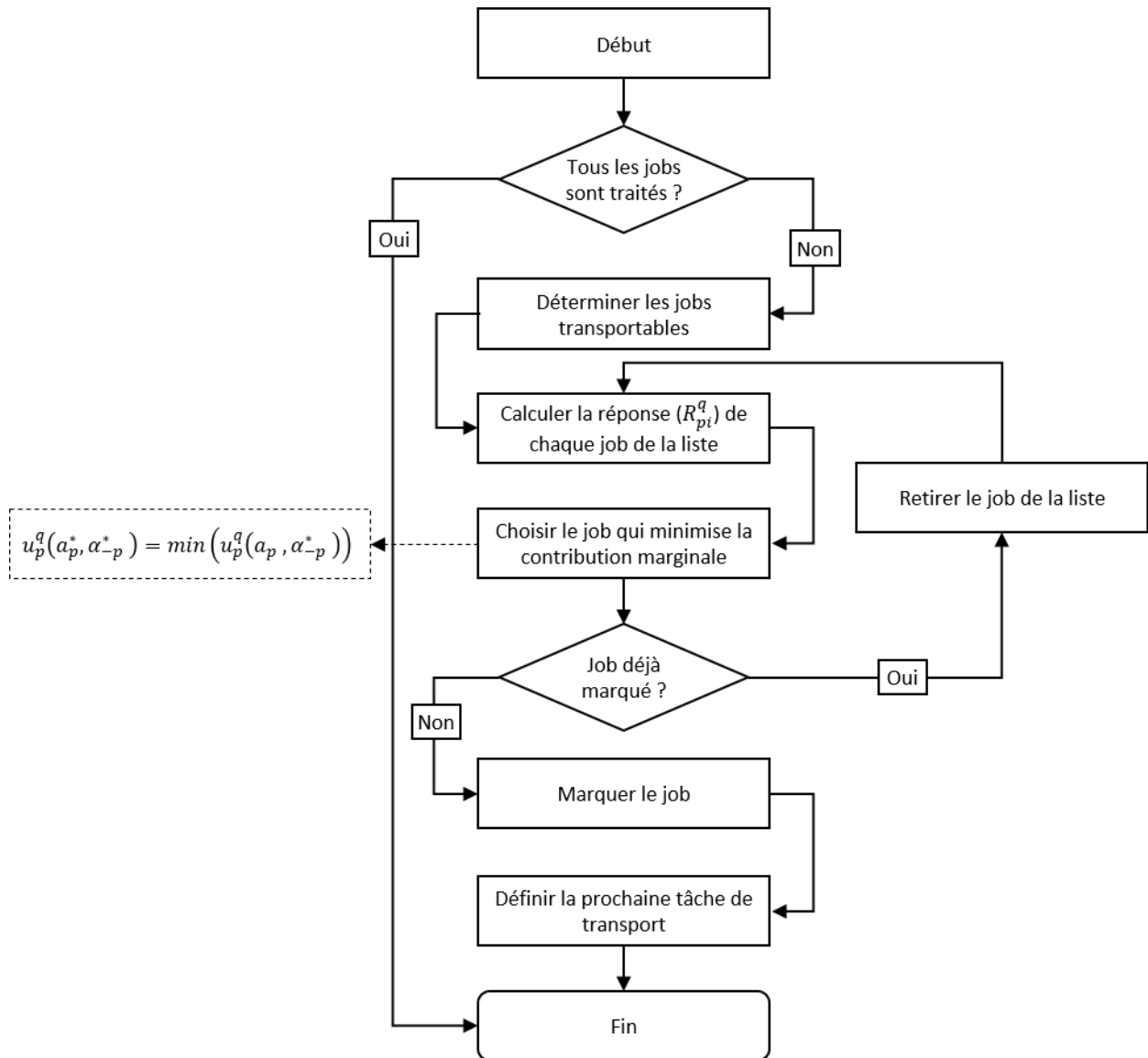


FIGURE 5.1 – Logigramme de l’algorithme d’ordonnement

5.5 Expérimentations

Afin d’évaluer les performances de notre algorithme d’ordonnement dynamique, nous avons mené une série d’expérimentations sur des scénarios de simulation variés.

Algorithme 3 Pseudo-code de l'algorithme d'ordonnancement**Requiert :** les jobs et robots disponibles

```

1: procédure TASKS_SCHEDULING(joueurp) :
2:   Si nb_total_jobs ≠ nb_jobs_traite alors
3:     jobs_restants ← liste des jobs pas encore traités ET pas encore marqués par d'autre robots
4:     up ← 100000                                ▷ initialisation de la MCU à une valeur très grande
5:     soit tmp1 ← 0                                ▷ initialisation
6:     soit tmp2 ← []                               ▷ initialisation
7:     Si jobs_restants ≠ [] alors                 ▷ liste non vide
8:       Pour jobi ∈ jobs_restants, faire :
9:         trouver tous les chemins entre la position de jobi et sa prochaine destination d ▷
          faire une recherche en profondeur (DFS)
10:        Pour chaque chemin pathn, faire :
11:          déterminer le nombre d'humains sur pathn
12:          estimer le temps de retard dû à la présence des humains ▷ avgretard × nbhumains
13:          calculer la réponse Rpiq de l'action ap = (jobi, pathn)
14:          calculer  $\mathcal{U}_g^q(a_p, \alpha_{-p}) - \mathcal{U}_g^q(a_p^0, \alpha_{-p})$ 
15:          Si  $u_p > \mathcal{U}_g^q(a_p, \alpha_{-p}) - \mathcal{U}_g^q(a_p^0, \alpha_{-p})$  alors
16:             $u_p \leftarrow \mathcal{U}_g^q(a_p, \alpha_{-p}) - \mathcal{U}_g^q(a_p^0, \alpha_{-p})$ 
17:             $tmp_1 \leftarrow R_{pi}^q$ 
18:             $tmp_2 \leftarrow a_p$ 
19:          Fin si
20:        Fin pour
21:      Fin pour
22:      Si jobi est déjà marqué par un autre robot alors
23:        TASKS_SCHEDULING(joueurp)
24:      Sinon
25:        current_order ← tmp2
26:         $R_{pi}^q \leftarrow tmp_1$ 
27:        mettre à jour  $\mathcal{U}_g^q(a_p, \alpha_{-p})$ 
28:        statut ← en activité
29:      Fin si
30:    Sinon
31:      current_order ← []
32:      statut ← en attente
33:    Fin si
34:  Fin si
35: Fin procédure

```

Dans un premier temps, nous avons considéré un environnement de production sans présence humaine. Les résultats de notre algorithme ont été comparés à ceux obtenus avec l'algorithme de VNS proposé par Moussa Abderrahim *et al.* [14], ainsi qu'avec l'algorithme d'ordonnancement dynamique par entropie développé par Xu Yiyi *et al.* [18]. Le critère de performance retenu pour cette comparaison est le makespan.

Dans un second temps, nous avons introduit les travailleurs humains dans la simulation, afin d'évaluer la capacité de notre algorithme à s'adapter à des environnements dynamiques et collaboratifs. Le même critère de performance a été utilisé pour comparer les trois algorithmes dans ce contexte plus complexe. Étant donné la variabilité dans les résultats des simulations (dans les deux cas), chaque instance a été simulée 30 fois et nous avons retenu la moyenne du makespan.

5.5.1 Description de l'algorithme d'ordonnancement dynamique par entropie

Nous avons utilisé et adapté la méthode d'optimisation dynamique développée par Xu Yiyi *et al.* [18]. Cette méthode, conçue pour l'optimisation multi-objectifs, s'inscrit parmi les approches Pareto, permettant d'améliorer simultanément tous les objectifs grâce à un système de classement lexicographique dynamique. Cependant, ce système exige des décideurs (les transporteurs) de déterminer l'ordre d'importance des objectifs avant le début du processus. En effet, il peut être difficile d'établir un classement définitif de tous les objectifs. De plus, le classement lexicographique ne permet pas de compenser de petits compromis sur un objectif plus important par de plus grands gains sur un objectif moins important.

Pour surmonter ces limitations, le système de classement lexicographique dynamique est combiné avec l'entropie et la programmation par objectifs. Lors de l'affectation des tâches de transport, le système prend en compte le degré d'incertitude, c'est-à-dire le niveau de hasard reflété dans la valeur d'entropie, dans la réalisation des différents objectifs. Les valeurs d'entropie sont mises à jour en temps réel en fonction des données disponibles, telles que les étapes de traitement des jobs, les emplacements des véhicules de transport ou l'accumulation de jobs en attente dans les buffers des machines. Ensuite, les décideurs définissent en amont des niveaux de réalisation souhaités pour chaque objectif. Le système trouve alors une solution qui se rapproche le plus de ces objectifs en utilisant la programmation par objectifs.

Notre approche diffère de la méthode originale de Xu Yiyi *et al.* de deux façons. Premièrement, elle

5.5. EXPÉRIMENTATIONS

se concentre uniquement sur la minimisation du makespan. Deuxièmement, le classement des objectifs et le calcul de leurs valeurs d'entropie ont été adaptés, en tenant compte des dispositions physiques de l'atelier (layouts) et des étapes de traitement de chaque job. Ce qui nous permet de réaliser des comparaisons sur les mêmes instances de référence (Bilge et Ulusoy, 1995) [1]. Toutefois, la structure de la méthode (décrit sur la figure 5.2) et l'algorithme principal restent inchangés.

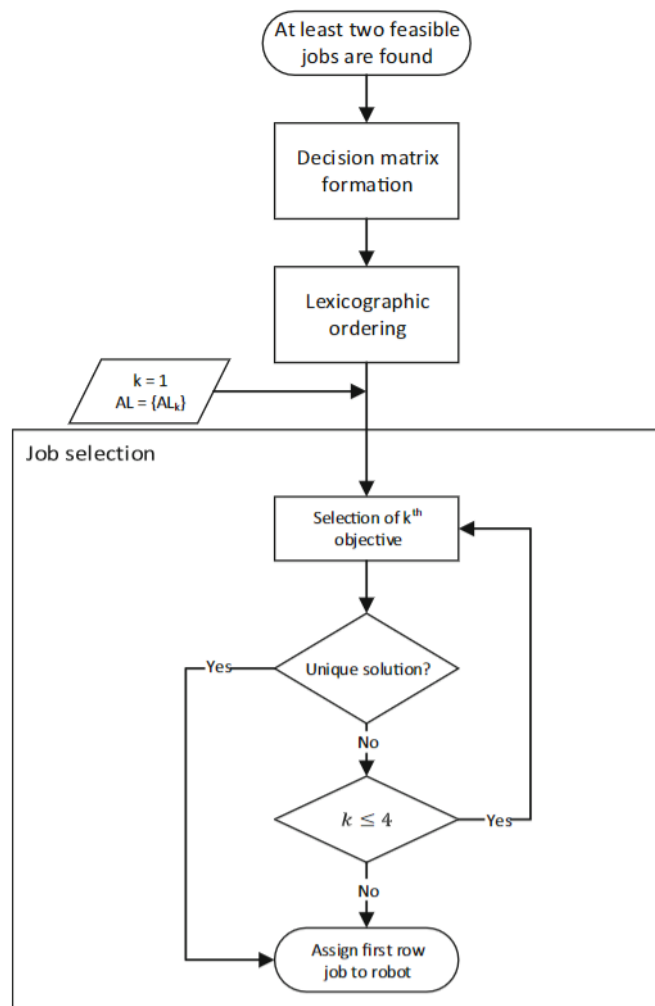


FIGURE 5.2 – Logigramme de l'algorithme d'ordonnancement dynamique par entropie (Xu Yiyi *et al.*) [18]

5.5.2 Expérience n°1

La première série d'expérimentations vise à évaluer les performances de notre algorithme d'ordonnancement dans un environnement de simulation sans perturbations externes et en l'absence d'in-

teractions humaines. Étant donné que l'environnement est statique, notre algorithme va se contenter de rechercher le job qui minimise sa contribution marginale (u_p^q) dans l'augmentation du makespan. L'objectif est d'évaluer la capacité de l'algorithme à optimiser l'allocation des tâches dans un contexte déterministe. Pour ce faire, nous avons réalisé des simulations avec les AIV, en faisant varier la taille de la flotte de deux (02) à cinq (05) robots. Les résultats obtenus ont été comparés à ceux issus de l'algorithme de VNS [14] et de l'algorithme de l'entropie [18].

5.5.2.1 Démarche expérimentale

Afin d'évaluer les performances de notre algorithme d'ordonnancement basé sur la théorie des jeux, nous avons mené une série de tests en simulation. chaque instance de référence de Bilge et Ulusoy [1] a été traitée comme suit :

1. Dans un premier temps, nous avons utilisé l'algorithme de VNS pour générer des ordonnancements de référence. Le VNS, en tant que méthode d'optimisation statique, nous permet d'obtenir une solution de qualité pour un environnement de production déterministe, c'est-à-dire sans perturbations aléatoires ni événements imprévus. Les ordonnancements générés par le VNS ont ensuite été simulés avec des AIV, dans un environnement sans présence humaine afin d'évaluer leurs performances en termes de makespan. Cette étape nous fournit une base de comparaison pour évaluer les performances de notre algorithme dans les environnements déterministes.
2. Deuxièmement, nous avons utilisé l'algorithme de l'entropie pour l'optimisation-simulation dynamique de l'instance, dans un environnement déterministe. Cet algorithme, étant aussi un algorithme d'ordonnancement dynamique, nous permet de comparer notre méthode avec à un algorithme similaire.
3. Pour terminer, nous avons simulé l'instance en utilisant notre algorithme d'ordonnancement basé sur la théorie des jeux. Les conditions expérimentales étaient identiques à celles des étapes précédentes, afin de permettre une comparaison équitable entre les différentes méthodes.

Par ailleurs, compte tenu de la variabilité inhérente aux algorithmes d'ordonnancement dynamiques, chaque instance a été simulée 30 fois et nous avons retenu la moyenne du makespan et son écart type.

5.5.2.2 Résultats

Le tableau 5.3 présente les résultats des simulations réalisées à partir des ordonnancements obtenus avec les trois algorithmes d'optimisation utilisés. La colonne *Instance* identifie chaque instance de test. Les colonnes *VNS*, *Entropie* et *TDJ* indiquent respectivement les résultats obtenus avec les algorithmes de VNS, d'optimisation basée sur l'entropie et notre algorithme basé sur la théorie des jeux. Les valeurs présentées correspondent au makespan moyen (et son écart-type) sur 30 simulations.

À première vue, le tableau 5.3 révèle une faible variabilité des résultats obtenus avec l'algorithme VNS, caractérisée par des écarts-types inférieurs à 1. Cette faible dispersion s'explique par le caractère déterministe de l'algorithme VNS qui génère un seul ordonnancement prévisionnel, reproduit identiquement à chaque simulation. En revanche, les algorithmes d'ordonnement dynamiques, tels que ceux basés sur l'entropie et la théorie des jeux, génèrent des ordonnancements différents à chaque répétition, en fonction des informations des robots. Cette adaptabilité entraîne une plus grande variabilité des résultats, comme en témoignent les écarts-types plus élevés.

5.5.2.3 Analyse des résultats

L'analyse du tableau 5.3 nous permet de faire les observations suivantes.

1. **L'ordonnement dynamique produit des performances similaires, voire meilleures, que l'ordonnement statique lorsque plusieurs jobs partagent des étapes de traitement identiques** : Même si, globalement, l'ordonnement statique surpasse l'ordonnement dynamique (entropie et théorie des jeux (TDJ)) dans les environnements déterministes, certaines instances font exception à la règle. Il s'agit en l'occurrence des ensembles de jobs 6 et 8 (voir annexe A) qui présentent une caractéristique frappante : une grande partie des jobs partagent des séquences de traitement identiques. Dans l'ensemble 8 par exemple, tous les jobs suivent le processus de fabrication $M_2 \rightarrow M_3 \rightarrow M_4$. Ce phénomène favorise la formation de goulots d'étranglement et la multiplication des rencontres de robots lors de la simulation de l'ordonnement prévisionnel. En revanche, les simulations réalisées avec l'ordonnement dynamique démontrent un risque plus faible de goulots d'étranglement, ce qui se traduit par une efficacité dans 75% des simulations effectuées avec les ensembles 6 et 8, comparé à l'ordonnement prévisionnel issu de l'optimisation statique.

2. **Notre algorithme d’ordonnement (TDJ) affiche, en moyenne, une performance relative inférieure d’au plus 7% par rapport à l’ordonnement statique (VNS) :** Afin de quantifier l’amélioration (ou la perte) apportée par notre algorithme, nous avons calculé pour chaque instance de test la différence relative (le « gap ») entre le makespan obtenu avec notre algorithme et celui obtenu avec l’algorithme VNS. Puis, nous avons calculé le gap moyen en fonction de la taille de la flotte d’AIV. Les résultats montrent que, pour des flottes de 2, 3, 4 et 5 AIV, les gaps moyens sont respectivement de 7%, 6%, 5% et 6%. Une analyse plus fine, réalisée en considérant les différents layouts, révèle des variations plus importantes. Par exemple, pour une flotte de 2 AIV, les gaps moyens pour les layouts 1, 2, 3 et 4 sont respectivement de 9,4%, 10%, 4,2% et 3,7%. Par conséquent, malgré son approche gloutonne et décentralisée, notre algorithme démontre une capacité à générer des ordonnancements compétitifs, comme le montre la figure 5.3. Il offre ainsi un bon compromis entre efficacité et complexité algorithmique.

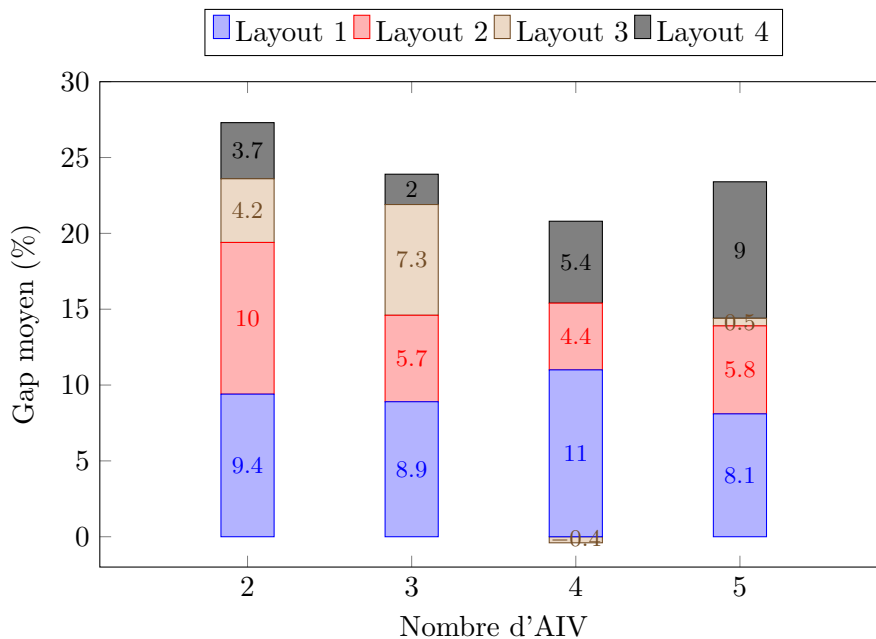


FIGURE 5.3 – Représentation du gap moyen entre l’algorithme du VNS [14] et celui de la TDJ.

3. **En moyenne, notre algorithme (TDJ) montre une efficacité relative similaire à celle de l’algorithme basé sur l’entropie :** Une analyse similaire à celle effectuée précédemment nous permet de calculer le gap entre les résultats obtenus avec notre algorithme et ceux obtenus avec l’algorithme basé sur l’entropie. Les résultats révèlent que, en moyenne, pour une flotte

5.5. EXPÉRIMENTATIONS

de 2, 3, 4 et 5 AIV, notre algorithme montre une efficacité relative de 1%, 0%, 0% et 0% respectivement. En analysant les résultats par layout, on constate que les performances sont mitigées, variant entre -5% et 5% d'efficacité relative, comme le montre la figure 5.4.

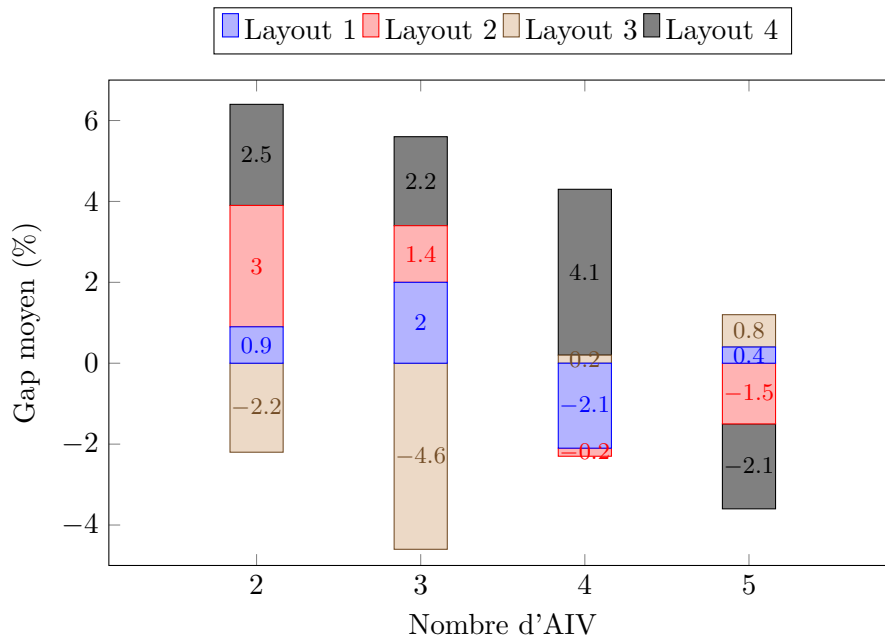


FIGURE 5.4 – Représentation du gap moyen entre l'algorithme basé sur l'entropie [18] et celui de la TDJ.

4. **En moyenne, sur l'ensemble des simulations, les performances de notre algorithme se situent entre ceux du VNS et de l'entropie** : Même si l'algorithme de la TDJ et celui basé sur l'entropie présentent des performances relatives similaires, une analyse instance par instance permet de constater que, dans la majorité des cas, le makespan obtenu avec la TDJ est inférieur à celui obtenu par l'entropie, symbolisant une supériorité globale de notre algorithme. En l'occurrence, pour des flottes de 2, 3, 4 et 5 AIV, notre algorithme offre respectivement de meilleurs résultats dans 58%, 65%, 63% et 63% des instances simulées. Cependant, comparé au VNS, ce dernier reste meilleur que notre algorithme dans la grande majorité des cas, comme le témoigne la figure 5.5. Cette observation confirme l'efficacité des méthodes d'ordonnement statiques comparé aux méthodes d'ordonnement dynamiques dans les environnements déterministes. Toutefois, il est important de préciser que les makespans du VNS sont les meilleurs qui ont été obtenus après la phase d'optimisation, tandis qu'avec la TDJ (aussi valable pour l'algorithme d'entropie) nous n'avons conservé que le makespan moyen. Des analyses complé-

5.5. EXPÉRIMENTATIONS

mentaires, notamment sur les meilleurs ordonnancements individuels, seraient donc nécessaires pour affiner ces résultats.

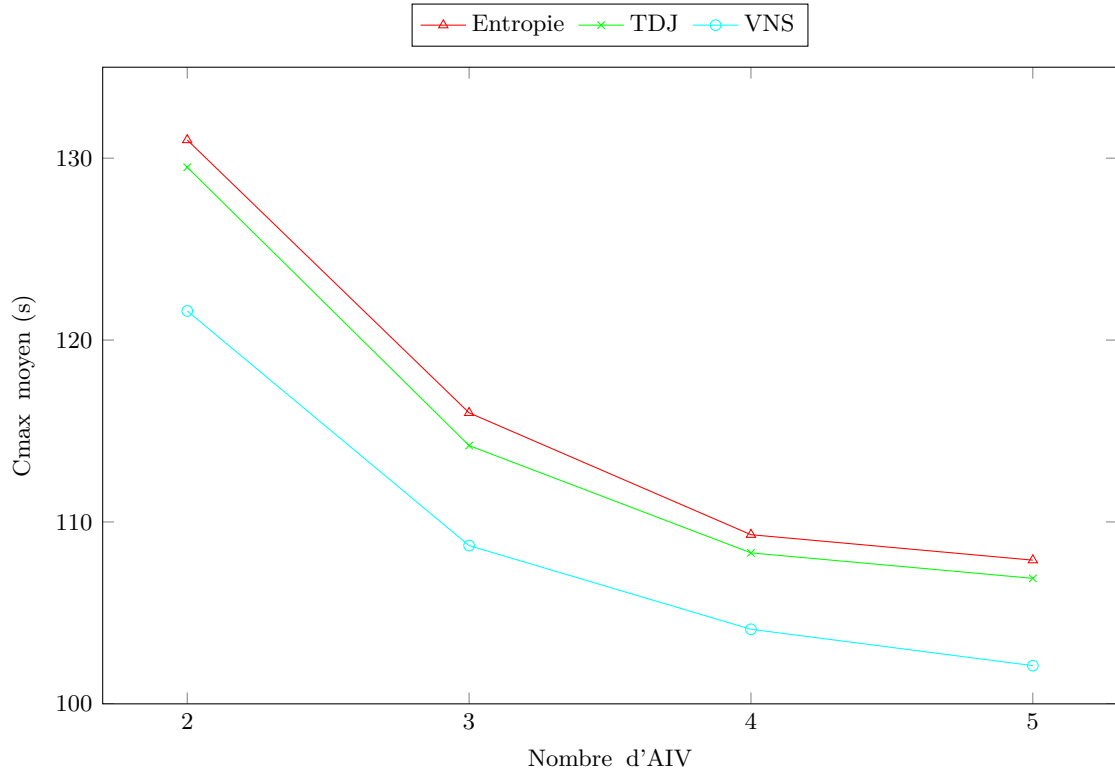


FIGURE 5.5 – Évolution du makespan moyen en fonction de l’algorithme d’ordonnancement et du nombre d’AIV : en l’absence d’humains.

En complément de l’analyse numérique, il est important de préciser que, même si les résultats présentés sont prometteurs, ils doivent être interprétés avec réserve. En effet, le nombre limité de répétitions des simulations réalisées (30) ne permet pas d’établir des conclusions définitives, en l’état, sur la performance relative des différents algorithmes d’ordonnancement dynamiques (Entropie vs. TDJ). Pour obtenir des résultats plus robustes et plus précis, il serait nécessaire de mener des simulations sur un nombre de répétitions plus important, permettant ainsi d’appliquer la loi des grands nombres et de converger vers des valeurs moyennes stables. De plus, une analyse plus approfondie des meilleurs ordonnancements individuels pourrait apporter des éclairages supplémentaires sur les forces et les faiblesses de chaque algorithme.

5.5. EXPÉRIMENTATIONS

Instance	2 AIV			3 AIV			4 AIV			5 AIV		
	VNS	Entropie	TDJ	VNS	Entropie	TDJ	VNS	Entropie	TDJ	VNS	Entropie	TDJ
EX11	118.5 (0.5)	131.1 (5.1)	114.7 (6.2)	90.3 (0.2)	109.7 (4.5)	101.6 (5.7)	84.7 (0.1)	104.1 (7.7)	95.4 (7.3)	84.2 (0.1)	102.3 (4.6)	96.7 (8.0)
EX21	119.1 (0.2)	135.1 (8.5)	133.2 (4.2)	95.0 (0.1)	105.4 (7.7)	103.0 (3.7)	92.4 (0.1)	102.0 (6.2)	98.5 (9.2)	92.9 (0.1)	95.6 (5.4)	100.4 (7.5)
EX31	122.2 (0.3)	135.1 (4.7)	143.0 (9.9)	104.1 (0.1)	123.3 (6.2)	125.2 (8.5)	94.5 (0.1)	108.1 (5.9)	99.9 (14.8)	94.8 (0.1)	115.7 (10.9)	100.6 (13.9)
EX41	140.1 (0.2)	140.4 (9.9)	152.1 (4.4)	114.5 (0.2)	122.1 (5.4)	116.5 (5.2)	101.3 (0.2)	113.7 (7.4)	110.8 (11.2)	95.5 (0.1)	111.8 (7.7)	109.3 (22.6)
EX51	103.1 (0.1)	124.8 (14.4)	113.5 (5.8)	83.2 (0.1)	94.9 (6.9)	90.8 (3.6)	73.4 (0.1)	93.9 (8.1)	87.9 (20.3)	73.8 (0.1)	87.4 (5.9)	85.3 (6.8)
EX61	135.6 (0.4)	147.9 (9.8)	137.1 (5.0)	114.5 (0.1)	118.6 (7.7)	128.2 (4.0)	114.5 (0.1)	106.6 (7.9)	106.6 (11.9)	114.5 (0.1)	107.0 (7.6)	109.9 (8.1)
EX71	144.8 (0.3)	161.0 (15.9)	160.8 (8.0)	112.9 (0.2)	134.2 (14.5)	124.7 (7.9)	95.4 (0.2)	98.9 (4.8)	119.7 (15.9)	91.3 (0.2)	95.1 (4.5)	105.5 (10.7)
EX81	154.8 (0.9)	155.2 (7.1)	179.3 (4.4)	167.7 (0.1)	171.8 (8.1)	166.4 (12.5)	168 (0.1)	171.5 (6.1)	167.8 (5.8)	167.7 (0.1)	168.9 (9.7)	163.4 (14.0)
EX91	123.5 (0.2)	145.7 (9.1)	138.3 (9.2)	115.1 (0.1)	120.3 (5.0)	125.3 (3.2)	111.2 (0.1)	120.7 (7.4)	123.2 (13.3)	111.3 (0.1)	122.8 (5.5)	117.9 (12.6)
EX101	159.7 (0.3)	181.2 (11.8)	175.7 (11.2)	147.8 (0.1)	165.5 (7.8)	157.2 (6.2)	140.6 (0.1)	140.1 (4.5)	155.2 (20.7)	139.7 (0.1)	139.2 (6.4)	149.6 (24.6)
EX12	76.5 (0.1)	106.0 (4.4)	84.6 (6.2)	72.8 (0.0)	89.8 (4.4)	81.7 (5.6)	72.8 (0.1)	90.7 (6.3)	76.3 (5.1)	73.3 (0.1)	85.1 (8.4)	76.3 (8.7)
EX22	82.8 (0.1)	97.8 (6.9)	100.5 (6.7)	81.9 (0.1)	85.7 (7.2)	82.2 (7.7)	82.1 (0.1)	80.2 (4.3)	89.2 (9.4)	82.3 (0.1)	78.8 (4.3)	89.0 (8.2)
EX32	87.7 (0.1)	103.4 (10.6)	116.9 (5.8)	84.6 (0.0)	94.9 (4.9)	92.6 (11.6)	84.9 (0.0)	84.8 (5.2)	89.6 (12.8)	84.9 (0.1)	90.7 (8.4)	100.1 (13.8)
EX42	94.8 (0.1)	97.0 (8.2)	93.3 (4.4)	79.9 (0.1)	87.2 (6.3)	86.4 (13.0)	72.3 (0.1)	77.8 (4.7)	87.1 (10.2)	72.2 (0.1)	82.3 (4.7)	79.0 (6.9)
EX52	68.5 (0.1)	88.3 (3.5)	75.5 (5.4)	64.7 (0.1)	76.8 (4.4)	71.0 (7.0)	64.0 (0.1)	72.3 (5.4)	68.4 (6.0)	63.5 (0.1)	67.1 (4.9)	66.5 (5.9)
EX62	102.8 (0.1)	107.1 (5.9)	109.7 (3.0)	102.5 (0.0)	94.4 (3.5)	107.0 (7.2)	102.8 (0.0)	91.0 (5.0)	99.8 (7.2)	103.3 (0.1)	102.7 (6.8)	104.7 (9.7)
EX72	96.3 (0.1)	108.5 (8.7)	105.6 (5.7)	79.1 (0.1)	85.6 (5.2)	80.5 (11.7)	79.2 (0.1)	82.5 (6.1)	82.3 (6.1)	78.7 (0.0)	79.6 (5.0)	85.0 (10.2)
EX82	155.6 (0.1)	166.4 (7.0)	161.6 (4.4)	155.7 (0.0)	159.9 (4.0)	160.4 (16.1)	156.2 (0.0)	157.8 (5.4)	149.1 (10.8)	156 (0.0)	157.6 (5.6)	147.0 (19.6)
EX92	99.5 (0.1)	103.6 (3.9)	104.4 (4.2)	100.1 (0.0)	105.8 (4.4)	105.3 (15.1)	102.6 (0.0)	109.9 (5.6)	103.3 (8.4)	100.1 (0.1)	102.7 (5.1)	105.8 (12.2)
EX102	143.6 (0.1)	152.3 (8.5)	145.1 (3.5)	135.9 (0.1)	138.0 (5.3)	139.3 (11.4)	131.4 (0.0)	136.2 (7.2)	131.9 (8.7)	132.2 (0.1)	135.9 (5.1)	138.5 (5.7)
EX13	80.7 (0.0)	89.2 (7.8)	82.6 (4.6)	74.6 (0.0)	86.8 (7.2)	80.7 (8.0)	74.7 (0.1)	82.0 (7.0)	76.4 (8.7)	73.8 (0.1)	80.0 (5.8)	76.0 (9.1)
EX23	84.4 (0.0)	83.1 (4.0)	85.7 (4.1)	80.1 (0.0)	78.7 (4.0)	93.8 (9.2)	83.0 (0.1)	78.2 (6.0)	87.3 (3.2)	82.9 (0.0)	79.3 (6.5)	85.5 (6.9)
EX33	89.3 (0.1)	97.4 (5.9)	90.9 (7.1)	82.8 (0.0)	85.7 (3.5)	90.4 (12.6)	83.1 (0.0)	86.4 (5.4)	83.9 (12.9)	84.8 (0.0)	94.0 (10.4)	90.4 (13.7)
EX43	87.8 (0.1)	88.4 (5.3)	96.2 (3.9)	84.9 (0.1)	96.2 (4.2)	90.5 (9.0)	80.8 (0.1)	81.6 (7.4)	81.7 (8.2)	76.1 (0.0)	82.1 (7.6)	80.6 (9.7)
EX53	90.3 (0.0)	80.3 (6.4)	72.8 (3.8)	63.6 (0.0)	73.6 (5.7)	70.1 (6.9)	64.2 (0.1)	70.3 (4.0)	65.9 (7.0)	63.6 (0.0)	69.0 (5.8)	68.8 (6.6)
EX63	100.9 (0.0)	94.5 (7.4)	116.0 (4.0)	100.9 (0.0)	90.1 (6.5)	104.4 (6.8)	130.1 (0.0)	88.5 (5.9)	103.1 (7.3)	101.0 (0.0)	86.3 (7.3)	95.9 (9.6)
EX73	88.9 (0.1)	104.6 (9.4)	106.8 (7.0)	80.9 (0.1)	75.8 (5.0)	96.7 (18.0)	77.7 (0.1)	77.1 (5.1)	87.1 (9.6)	77.6 (0.0)	74.8 (3.9)	79.8 (5.8)
EX83	155.5 (0.0)	155.3 (6.9)	162.0 (0.8)	156.0 (0.0)	157.7 (5.9)	146.2 (9.4)	155.4 (0.0)	155.5 (4.7)	138.6 (16.8)	155.8 (0.0)	156.3 (5.6)	131.5 (12.1)
EX93	103.0 (0.1)	103.8 (5.7)	110.0 (4.5)	98.3 (0.0)	99.5 (3.7)	103.5 (7.4)	97.7 (0.0)	96.4 (6.4)	100.8 (8.2)	97.8 (0.0)	99.2 (4.4)	94.4 (8.2)
EX103	137.9 (0.0)	140.6 (4.1)	138.3 (5.6)	136.3 (0.0)	137.5 (5.8)	136.2 (10.2)	133.8 (0.0)	132.4 (8.1)	133.3 (13.6)	134.0 (0.0)	136.6 (6.2)	133.8 (10.9)
EX14	128.7 (0.1)	146.0 (9.1)	129.8 (10.2)	95.5 (0.1)	114.2 (5.3)	98.4 (12.6)	85.5 (0.1)	103.4 (6.4)	94.6 (8.6)	82.3 (0.0)	100.4 (8.2)	98.2 (6.6)
EX24	138.2 (0.1)	134.5 (10.1)	143.6 (5.9)	112.9 (0.1)	108.7 (7.4)	105.2 (10.4)	98.5 (0.1)	106.3 (9.9)	102.4 (16.5)	100.1 (0.1)	100.6 (7.0)	99.7 (14.2)
EX34	151.6 (0.3)	149.6 (11.8)	158.9 (12.8)	124.6 (0.1)	137.6 (8.0)	127.7 (25.6)	102.1 (0.1)	114.4 (4.8)	103.6 (10.4)	102.7 (0.0)	118.0 (15.8)	108.1 (13.0)
EX44	149.3 (0.1)	150.3 (14.1)	157.8 (9.0)	127.0 (0.1)	132.9 (12.7)	128.6 (21.7)	103.3 (0.1)	125.4 (11.3)	116.3 (18.6)	92.8 (0.1)	125.6 (13.8)	110.1 (17.7)
EX54	112.6 (0.1)	132.9 (12.6)	116.6 (8.8)	86.3 (0.1)	98.3 (9.8)	97.7 (16.7)	83.9 (0.1)	97.2 (6.6)	84.7 (10.8)	69.7 (0.0)	89.9 (6.8)	85.9 (10.5)
EX64	154.4 (0.1)	168.3 (13.0)	158.9 (6.2)	132.8 (0.0)	125.5 (10.2)	138.4 (7.2)	124.3 (0.1)	115.7 (9.5)	138.4 (11.9)	118.5 (0.1)	110.1 (5.4)	120.4 (26.7)
EX74	169.6 (0.1)	176.8 (10.8)	173.5 (7.5)	137.8 (0.1)	149.7 (10.3)	141.4 (15.0)	113.0 (0.1)	138.6 (11.4)	122.4 (14.5)	102.8 (0.1)	128.9 (9.7)	120.7 (14.4)
EX84	166.5 (0.3)	193.2 (11.4)	170.3 (7.2)	175.2 (0.0)	170.9 (12.3)	174.9 (9.6)	175.1 (0.0)	172.9 (8.5)	175.8 (7.8)	180.3 (0.1)	174.0 (10.1)	192.1 (21.1)
EX94	148.4 (0.1)	166.3 (6.1)	159.6 (14.6)	130.3 (0.1)	132.2 (3.8)	133.5 (13.1)	117.1 (0.1)	130.4 (9.4)	124.2 (11.5)	121.7 (0.1)	129.9 (9.1)	122.6 (21.4)
EX104	197.2 (0.1)	195.7 (9.1)	202.9 (7.4)	165.3 (0.0)	166.0 (7.3)	165.9 (10.0)	154.7 (0.0)	158.2 (6.6)	152.3 (16.9)	154.7 (0.0)	152.1 (6.7)	151.3 (20.5)

TABLE 5.3 – Résultats de l'expérience n°1 : simulation dans un environnement déterministe

5.5. EXPÉRIMENTATIONS

Instance	2 AIV			3 AIV			4 AIV			5 AIV		
	VNS	Entropie	TDJ	VNS	Entropie	TDJ	VNS	Entropie	TDJ	VNS	Entropie	TDJ
EX11	170.6 (22.1)	205.7 (33.1)	119.9 (6.3)	137.7 (24.8)	148.8 (15.1)	101.8 (7.7)	128.2 (14.4)	133.3 (16.2)	96.4 (4.8)	126.7 (16.2)	139.1 (16.7)	96.7 (6.3)
EX21	203.4 (23.2)	199.2 (20.6)	140.9 (11.3)	165.2 (23.9)	158.8 (14.1)	109.3 (5.4)	146.6 (20.0)	151.2 (11.7)	109.0 (3.8)	145.5 (17.7)	125.2 (12.3)	104.6 (5.6)
EX31	222.9 (23.0)	223.9 (31.2)	162.5 (16.2)	174.6 (17.9)	185.9 (28.5)	136.7 (8.3)	163.5 (21.9)	169.0 (21.2)	128.4 (8.4)	143.2 (11.2)	167.2 (24.1)	121.7 (8.7)
EX41	244.7 (15.1)	216.3 (30.2)	164.8 (9.4)	199.4 (28.5)	194.8 (31.0)	121.3 (4.3)	192.0 (22.5)	157.2 (20.7)	107.2 (4.6)	164.0 (30.8)	165.8 (27.2)	107.6 (6.9)
EX51	186.1 (32.2)	187.6 (22.6)	150.2 (13.6)	141.8 (12.9)	153.9 (23.0)	90.8 (3.2)	126.3 (16.5)	130.5 (18.4)	83.5 (4.9)	113.7 (22.2)	113.6 (14.3)	85.2 (5.8)
EX61	263.2 (11.2)	236.1 (26.1)	157.0 (11.8)	183.3 (27.7)	177.1 (16.0)	129.3 (5.1)	173.8 (11.9)	157.0 (17.9)	128.1 (5.3)	172.0 (25.9)	152.2 (23.8)	123.3 (2.5)
EX71	256.1 (20.2)	247.5 (18.5)	174.3 (12.5)	213.7 (32.5)	176.4 (21.2)	133.4 (11.9)	168.0 (17.2)	145.2 (18.9)	109.2 (7.3)	153.6 (13.1)	135.8 (15.2)	103.7 (7.7)
EX81	243.3 (18.6)	241.1 (24.7)	207.7 (13.8)	242.6 (20.4)	205.9 (22.0)	171.5 (1.7)	218.1 (22.4)	179.5 (11.5)	170.8 (3.8)	227.7 (12.5)	160.3 (17.1)	170.7 (4.2)
EX91	214.4 (27.2)	224.5 (28.5)	152.7 (5.5)	199.1 (29.0)	184.3 (15.8)	137.5 (12.5)	166.8 (13.8)	153.8 (20.5)	122.8 (6.7)	170.5 (23.0)	166.0 (21.1)	124.5 (4.7)
EX101	251.9 (12.4)	261.2 (23.5)	197.9 (11.2)	225.0 (30.6)	209.8 (23.1)	179.9 (28.4)	217.6 (28.5)	187.4 (18.4)	164.4 (4.7)	193.2 (25.1)	177.4 (17.5)	162.8 (3.9)
EX12	129.2 (14.9)	133.9 (18.2)	123.5 (10.2)	114.5 (6.3)	108.9 (12.4)	84.2 (4.0)	101.9 (10.2)	98.2 (8.4)	83.3 (5.7)	103.8 (10.0)	96.7 (16.6)	84.2 (4.6)
EX22	134.9 (19.1)	146.0 (13.6)	109.3 (10.0)	111.1 (8.5)	114.1 (12.3)	96.8 (6.2)	121.6 (18.7)	105.3 (10.7)	95.3 (4.9)	110.3 (10.0)	98.5 (9.9)	92.0 (3.4)
EX32	151.1 (20.9)	148.0 (14.0)	125.5 (10.9)	125.6 (16.9)	117.1 (10.6)	111.9 (7.7)	130.7 (17.1)	116.1 (15.1)	109.5 (7.0)	111.6 (11.4)	119.4 (11.9)	111.9 (6.8)
EX42	175.4 (21.6)	148.5 (14.3)	149.5 (18.4)	146.5 (17.4)	120.3 (16.0)	109.4 (7.9)	106.6 (15.3)	112.4 (17.3)	81.1 (4.3)	111.6 (10.6)	98.7 (9.6)	80.7 (2.9)
EX52	125.1 (18.7)	139.1 (15.6)	117.4 (17.0)	98.1 (13.5)	93.6 (7.4)	71.7 (7.2)	84.7 (11.8)	91.4 (15.6)	72.2 (4.7)	99.4 (12.4)	86.1 (9.6)	70.9 (3.5)
EX62	159.2 (21.0)	171.9 (26.1)	122.4 (8.8)	139.4 (15.2)	121.1 (13.2)	115.9 (5.3)	123.0 (12.0)	109.4 (10.7)	114.7 (3.6)	146.3 (13.0)	107.8 (7.6)	110.6 (3.3)
EX72	176.2 (10.6)	178.1 (15.3)	127.8 (9.6)	134.3 (14.6)	123.5 (16.5)	105.8 (5.1)	125.2 (14.0)	112.2 (14.0)	95.8 (5.3)	105.2 (9.7)	94.9 (8.6)	90.8 (4.4)
EX82	209.8 (28.9)	166.3 (19.6)	184.1 (12.4)	192.5 (19.2)	156.9 (16.6)	158.7 (1.5)	169.2 (7.3)	159.2 (10.6)	157.2 (2.6)	185.3 (15.4)	115.5 (9.1)	157.7 (2.6)
EX92	151.0 (19.6)	150.0 (14.7)	122.8 (4.8)	116.7 (8.1)	116.3 (10.3)	110.3 (3.6)	143.3 (13.1)	113.8 (13.4)	103.8 (3.2)	130.5 (14.1)	108.8 (11.8)	104.8 (3.4)
EX102	193.4 (22.0)	188.1 (18.1)	162.3 (8.6)	169.3 (19.7)	142.4 (7.2)	151.8 (3.3)	162.3 (11.4)	132.9 (19.1)	149.8 (3.4)	158.5 (9.8)	133.7 (11.4)	150.1 (4.8)
EX13	118.8 (16.4)	124.9 (14.3)	117.9 (14.1)	103.1 (15.2)	107.3 (15.1)	109.5 (10.6)	98.1 (11.9)	98.6 (10.9)	84.3 (4.6)	98.4 (10.7)	98.3 (14.2)	85.0 (3.1)
EX23	130.9 (20.5)	142.2 (20.5)	99.1 (8.5)	111.3 (17.0)	100.2 (10.8)	94.2 (4.2)	120.5 (12.1)	100.6 (8.3)	92.6 (4.8)	119.2 (11.6)	99.1 (8.1)	90.5 (3.1)
EX33	147.3 (9.6)	149.2 (22.9)	121.4 (9.0)	125.0 (7.3)	114.4 (12.6)	109.2 (7.0)	114.6 (16.4)	108.7 (11.8)	104.1 (7.8)	114.4 (8.0)	114.3 (11.3)	104.7 (6.7)
EX43	188.7 (15.9)	148.9 (13.8)	160.8 (18.7)	164.1 (17.3)	135.3 (14.7)	131.5 (15.6)	134.9 (12.8)	123.5 (18.3)	86.9 (3.6)	122.0 (13.3)	115.4 (17.4)	87.0 (3.2)
EX53	142.1 (17.2)	105.8 (12.9)	107.8 (11.6)	93.9 (12.3)	97.5 (14.8)	98.7 (11.2)	98.5 (11.4)	93.3 (14.1)	79.3 (5.9)	93.5 (11.1)	82.1 (11.4)	75.4 (7.2)
EX63	152.0 (11.2)	146.7 (15.0)	119.3 (12.0)	133.7 (15.6)	115.4 (10.9)	110.9 (3.7)	153.1 (8.6)	115.5 (22.0)	113.7 (3.3)	133.2 (9.9)	104.8 (13.4)	111.4 (3.9)
EX73	177.6 (22.8)	146.2 (19.2)	123.1 (11.8)	144.5 (22.8)	114.4 (10.4)	90.9 (4.2)	121.5 (11.2)	100.2 (14.0)	88.7 (3.6)	114.3 (20.2)	93.4 (11.7)	88.2 (6.4)
EX83	236.0 (11.0)	159.1 (17.7)	170.2 (4.6)	226.5 (23.1)	160.8 (12.0)	157.0 (1.2)	205.7 (16.4)	158.1 (9.2)	157.5 (2.1)	181.8 (7.6)	117.4 (11.5)	157.1 (1.2)
EX93	163.8 (26.2)	143.2 (15.5)	118.7 (8.6)	156.7 (26.4)	118.1 (10.4)	106.9 (5.0)	142.6 (19.7)	114.0 (12.4)	104.2 (3.5)	123.9 (7.8)	119.5 (14.4)	106.3 (3.5)
EX103	203.9 (16.2)	170.7 (17.8)	158.2 (7.9)	186.0 (11.5)	145.0 (14.2)	149.5 (3.7)	198.6 (25.2)	142.2 (11.1)	145.4 (4.6)	170.5 (13.1)	131.6 (15.3)	143.1 (5.5)
EX14	218.1 (22.9)	226.0 (25.0)	147.8 (12.0)	153.9 (19.5)	158.1 (20.9)	124.4 (17.4)	133.3 (16.1)	138.6 (18.8)	98.6 (5.8)	121.4 (16.0)	130.2 (20.7)	99.7 (6.2)
EX24	237.7 (16.8)	228.8 (24.6)	160.1 (10.4)	197.5 (25.6)	159.9 (20.5)	123.4 (10.5)	183.4 (17.8)	153.5 (17.0)	113.4 (3.9)	148.7 (25.4)	129.5 (17.2)	109.8 (5.5)
EX34	243.1 (16.7)	239.8 (35.4)	166.4 (13.9)	218.4 (28.3)	166.1 (19.2)	141.9 (12.2)	153.4 (15.2)	155.4 (21.0)	134.5 (9.0)	147.4 (20.6)	152.1 (27.9)	134.0 (9.2)
EX44	248.1 (21.1)	228.9 (26.9)	194.7 (13.4)	200.9 (27.2)	179.8 (17.0)	162.6 (15.5)	172.6 (16.7)	164.3 (26.0)	107.3 (6.0)	155.4 (21.7)	169.2 (25.0)	123.2 (9.4)
EX54	220.7 (19.3)	198.3 (23.4)	180.5 (15.6)	139.0 (23.4)	144.0 (19.7)	114.2 (9.2)	140.6 (32.2)	124.9 (20.5)	95.4 (5.5)	102.1 (11.7)	118.7 (15.8)	82.9 (7.8)
EX64	251.7 (16.7)	244.9 (32.6)	183.9 (9.3)	211.6 (18.8)	182.5 (23.2)	141.0 (6.7)	176.5 (21.1)	163.9 (17.8)	134.0 (5.4)	175.7 (20.7)	142.6 (21.4)	129.2 (3.0)
EX74	256.0 (12.7)	271.2 (22.0)	192.5 (11.2)	230.6 (19.8)	221.5 (23.0)	171.7 (11.0)	215.0 (27.9)	165.6 (21.6)	119.8 (9.2)	173.9 (24.9)	141.3 (21.3)	108.7 (7.7)
EX84	265.8 (7.5)	254.7 (21.8)	218.1 (14.9)	251.1 (11.0)	188.9 (21.2)	182.6 (6.9)	235.3 (18.5)	183.3 (23.9)	177.3 (2.7)	258.2 (16.5)	164.7 (17.0)	177.3 (2.8)
EX94	246.8 (18.7)	234.3 (29.4)	170.2 (7.2)	197.2 (20.9)	169.0 (23.8)	138.4 (5.9)	167.2 (23.1)	158.7 (24.8)	127.8 (6.3)	174.4 (12.3)	159.5 (21.6)	126.9 (4.8)
EX104	263.2 (8.6)	253.4 (26.7)	223.9 (12.6)	245.1 (16.6)	215.0 (19.5)	176.0 (6.4)	230.0 (17.3)	173.9 (13.3)	168.7 (3.9)	215.8 (25.2)	176.4 (12.2)	167.2 (4.8)

TABLE 5.4 – Résultats de l'expérience n°2 : simulation dans un environnement dynamique

5.5.3 Expérience n°2

La seconde expérience vise à évaluer la capacité de notre algorithme à prendre des décisions d'ordonnancement en temps réel dans un environnement dynamique où cohabitent des robots et des opérateurs humains. Cette fois, notre l'optimisation consiste à choisir le meilleur couple $(job, chemin)$. Ainsi, pour chaque job, l'algorithme explore l'ensemble des chemins possibles entre l'emplacement actuel du job et la destination correspondante grâce à un algorithme de parcours en profondeur (*deep first search - DFS*, en anglais), utilisé en théorie des graphes pour trouver tous les chemins possibles entre deux sommets dans un graphe connexe non orienté. Après cela, l'algorithme d'ordonnancement évalue le nombre d'humains actuellement présent sur chaque chemin, ce qui lui permet d'estimer le potentiel retard engendré par les interactions avec les humains. Enfin, l'algorithme calcule la réponse du joueur (R_{pi}^q) pour chaque chemin possible en prenant en compte à la fois le temps de traitement, le temps de trajet et le potentiel retard lié aux humains (voir algorithme 3). Il répète ce processus pour tous les jobs à traiter et finit par déterminer le couple $(job, chemin)$ qui minimise sa contribution marginale (u_p^q). Cette démarche exploratoire permet d'intégrer directement la présence humaine au moment de la prise de décision.

En outre, comme pour la première expérience, les simulations sont réalisées avec les AIV, en faisant varier la taille de la flotte de deux (02) à cinq (05) robots. Les résultats obtenus sont comparés à ceux issus de la méthode de VNS [14] et de l'entropie [18].

5.5.3.1 Démarche expérimentale

Dans la réalisation de cette seconde expérience, nous avons conservé la même démarche expérimentale que celle décrite pour la première expérience, mais en réintégrant les travailleurs humains dans l'atelier. Les interactions homme-robot, source d'aléas et d'imprévus, rendent donc l'environnement dynamique. Toutefois, nous avons limité le nombre d'humains à 12 repartis comme suit : 8 techniciens (2 par machines), 2 superviseurs (1 par paire de machine) et 2 membres administratifs. Ici aussi, les simulations ont été répétées 30 fois pour chaque instance, afin d'évaluer l'impact de la présence humaine sur les performances des algorithmes en termes de makespan moyen et d'écart-type.

5.5.3.2 Résultats

Les résultats des simulations réalisées dans un environnement dynamique, en présence des travailleurs humains, sont présentés dans le tableau 5.4. Ces résultats correspondent au makespan moyen (et son écart-type) sur 30 répétitions des simulations.

Le tableau 5.4 révèle que la présence de travailleurs humains dans l'environnement de simulation induit une forte variabilité des résultats, particulièrement pour l'algorithme VNS. Cette observation s'explique notamment par la nature statique de l'ordonnancement généré par le VNS, qui est peu robuste face aux perturbations aléatoires. En comparaison, les algorithmes d'ordonnancement dynamiques, bien que soumis aux mêmes perturbations, présentent une variabilité moindre, témoignant de leur capacité à mieux s'adapter à un environnement changeant. Cette affirmation est renforcée par l'analyse de l'écart-type de notre algorithme, qui est généralement inférieur à 10, témoignant ainsi de sa stabilité relative. Cela peut s'expliquer par la capacité de notre algorithme à directement intégrer la présence humaine dans ses prises de décisions. Néanmoins, tous les algorithmes sont affectés par la présence humaine, se traduisant par une augmentation systématique du makespan.

5.5.3.3 Analyse des résultats

Une lecture attentive du tableau 5.4 nous permet de faire les observations ci-après.

1. **Dans l'ensemble des simulations, l'ordonnancement dynamique se révèle nettement supérieur à l'ordonnancement statique** : L'introduction d'imprévus liés à la présence humaine a radicalement modifié la hiérarchie des algorithmes. Alors que le VNS dominait dans un environnement déterministe, il est désormais distancé par les méthodes dynamiques, plus adaptées à la gestion des aléas. Cette inversion de tendance est particulièrement marquante sur la figure 5.6, qui illustre l'évolution du makespan moyen en fonction de l'algorithme utilisé, de l'état de l'atelier (avec ou sans présence humaine) et de la taille de la flotte d'AIV. Par ailleurs, sur l'ensemble des simulations effectuées, l'algorithme du VNS n'obtient de meilleurs résultats que dans seulement 2% des cas (soit 3 cas sur 160), comparé aux deux méthodes d'ordonnancement dynamiques combinées.
2. **En moyenne, notre algorithme est le plus robuste en présence des humains** : Pour évaluer les perturbations causées par la présence humaine, nous avons calculé la différence entre

5.5. EXPÉRIMENTATIONS

les makespans moyens obtenus avec et sans humains dans l'environnement, pour chaque algorithme d'optimisation. Les résultats montrent que, dans les simulations réalisées avec 2 AIV par exemple, la présence humaine a entraîné une augmentation moyenne de 78 secondes pour le VNS, 60 secondes pour l'entropie, et 23 secondes pour notre algorithme. Ces observations suggèrent que notre approche, basée sur la théorie des jeux, tolère mieux la présence des humains et démontre son efficacité dans la résolution de problèmes d'ordonnancement en conditions dynamiques, comme le témoigne la figure 5.6.

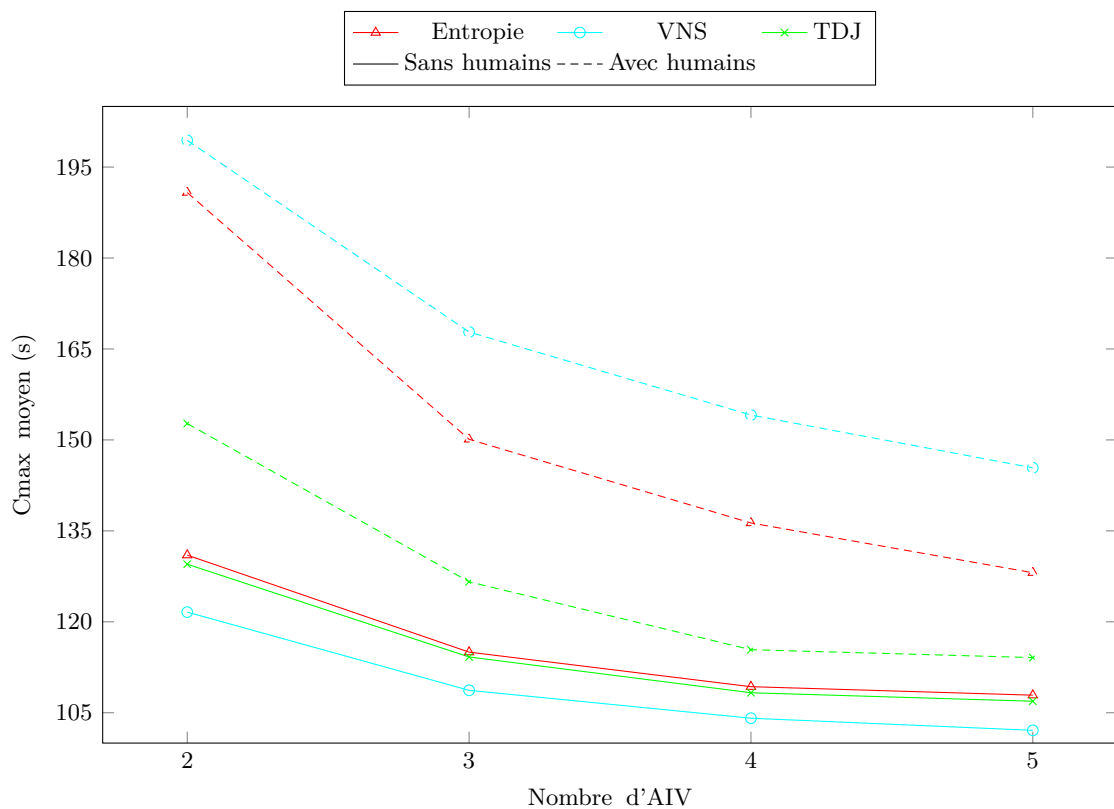


FIGURE 5.6 – Comparaison des makespans moyens avec ou sans présence humaine dans l'atelier

3. Les layouts influencent l'efficacité de notre algorithme : Une analyse approfondie des résultats en fonction des différents layouts met en évidence l'influence de la configuration de l'atelier sur les performances relatives des algorithmes. Bien que notre algorithme présente une efficacité moyenne supérieure, les gains observés varient considérablement d'un layout à l'autre. Par exemple, pour les simulations avec 2 AIV, les gains relatifs par rapport au VNS sont de 27,7%, 15,9%, 21,2% et 25,2% respectivement sur les layouts 1, 2, 3 et 4. Le constat est le

5.5. EXPÉRIMENTATIONS

même en comparant notre algorithme à celui basé sur l'entropie. Toujours sur les simulations avec 2 AIV, notre algorithme affiche une supériorité relative de 27,5% sur le layout 1, 14,1% sur le layout 2, 9,7% sur le layout 3 et 22,6% sur le layout 4, comparé à l'algorithme basé sur l'entropie. Cette observation confirme encore une fois l'influence des layouts d'atelier sur les performances des algorithmes d'ordonnancement. En l'occurrence, la flexibilité de routage des layouts 1 et 4 est un avantage pour notre algorithme, vu qu'il recherche la meilleure combinaison du job à transporter et du chemin à prendre selon le niveau de congestion. Les figures 5.7 et 5.8 présentent, les comparaisons de notre algorithme avec les autres méthodes utilisées (VNS et entropie).

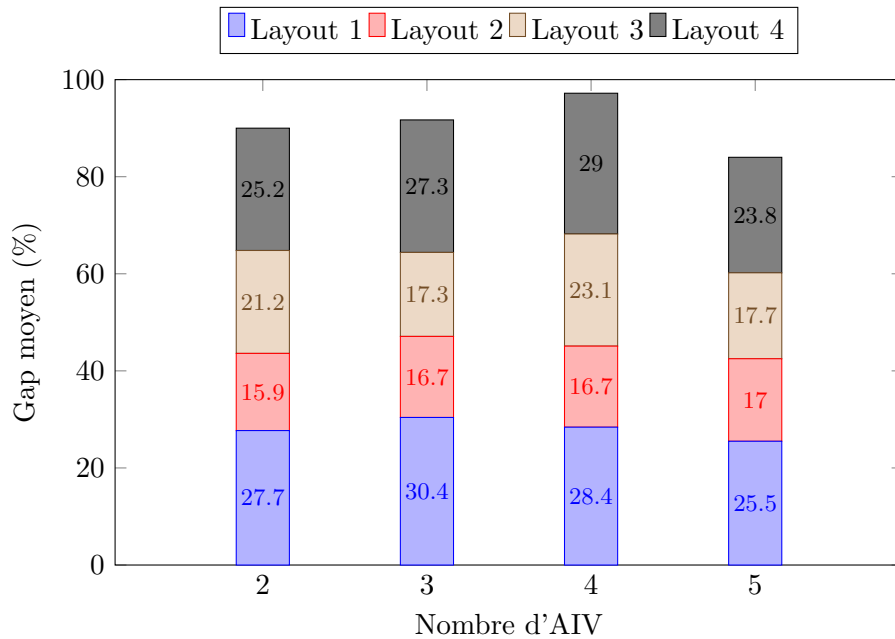


FIGURE 5.7 – Représentation du gap moyen entre notre algorithme et celui du VNS [14].

- 4. Plus la flotte d'AIV est grande, moins les gains d'efficacité supplémentaires sont importants.** Bien que l'ajout de robots à la flotte entraîne une diminution du makespan moyen, les gains en termes d'efficacité diminuent de manière significative au-delà d'une certaine taille de flotte. Nos résultats montrent que le passage de 2 à 3 robots génère des gains d'efficacité de 16% avec le VNS, 21% avec l'algorithme basé sur l'entropie et 17% avec notre algorithme, tandis que l'ajout d'un quatrième robot n'apporte plus que des gains marginaux de l'ordre de 9% pour tous les algorithmes. Cette tendance, observée aussi bien en présence qu'en l'absence d'humains

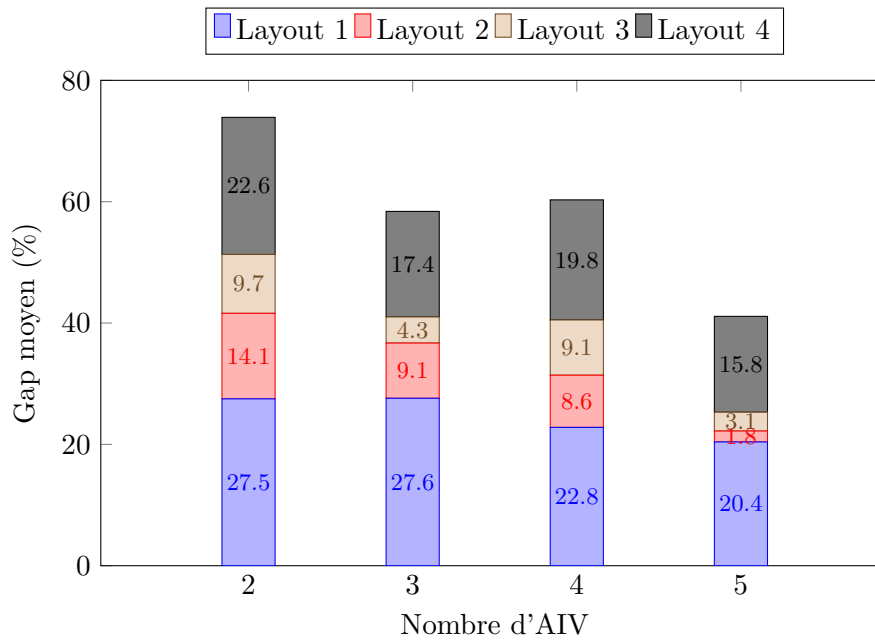


FIGURE 5.8 – Représentation du gap moyen entre notre algorithme et celui basé sur l’entropie [18].

(voir figure 5.9), suggère qu’une flotte de 3 robots pourrait constituer un compromis optimal entre coût et performance.

5. **Les méthodes d’ordonnancement dynamiques permettent de réduire le nombre d’interactions humain-robot** : L’analyse des simulations montre que les méthodes d’ordonnancement dynamiques réduisent de manière significative les interactions entre humains et robots dans les couloirs de déplacement partagés. Bien que ces interactions soient inévitables dans un environnement de production collaboratif, elles peuvent causer des ralentissements et augmenter les risques de collisions. Ainsi, en calculant la moyenne des interactions sur l’ensemble des simulations, il apparaît que les méthodes dynamiques réduisent nettement ces rencontres, comme illustré dans la figure 5.10. Cet effet est particulièrement marqué lorsque notre algorithme est comparé à l’algorithme VNS sur une flotte de 5 AIV, avec une réduction des interactions de 84%. Contrairement aux ordonnancements statiques, où les AIV suivent des plannings rigides préétablis, les approches dynamiques réévaluent en continu l’exécution des tâches. Notre algorithme prend spécifiquement en compte la présence humaine dans ses décisions, ce qui permet aux robots d’éviter les chemins occupés et d’optimiser leurs déplacements. Cela améliore la fluidité des opérations, réduit les interruptions et augmente à la fois la sécurité et l’efficacité du système.

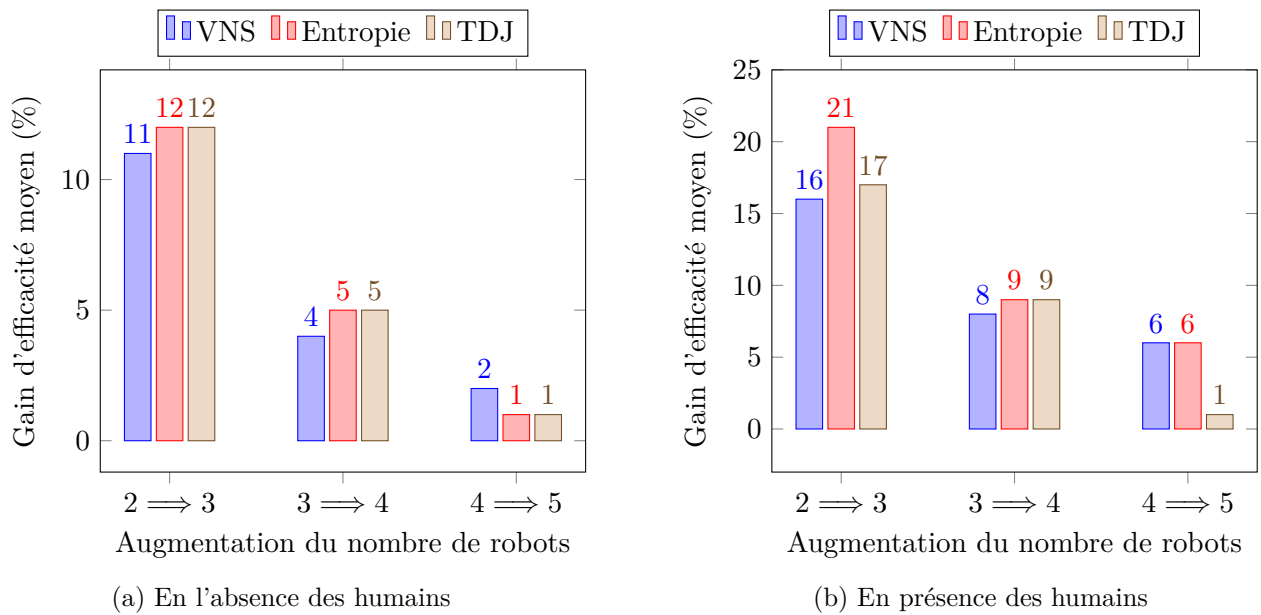


FIGURE 5.9 – Représentation du gain moyen d'efficacité après augmentation du nombre de robot

Par ailleurs, même si l'expérience n'a été réalisée que sur 30 répétitions des simulations, notre algorithme (TDJ) se révèle être la méthode la plus efficace pour gérer un environnement industriel dynamique avec la présence humaine. Ses performances supérieures nettes, tant en termes de makespan que de stabilité, en font un bon candidat pour l'ordonnancement des tâches dans l'Industrie 5.0, où les interactions homme-robot sont de plus en plus fréquentes. De plus, bien que l'algorithme d'entropie offre une alternative, il reste moins performant que la TDJ, notamment dans les scénarios avec 2 ou 3 robots. D'ailleurs, ces résultats devraient être confirmés par une analyse complémentaire réalisée sur un grand nombre de répétitions, permettant à la loi des grands nombres de faire converger les résultats vers une moyenne stable.

5.6 Conclusion

Dans ce dernier chapitre, nous avons apporté des contributions dans l'optimisation de l'ordonnancement des tâches de transport au sein de systèmes de production complexes intégrant des véhicules autonomes intelligents (AIV) et des opérateurs humains. En s'appuyant sur un modèle mathématique de théorie des jeux, ce chapitre a permis de concevoir un algorithme d'ordonnancement dynamique capable de s'adapter en temps réel à des environnements de production fluctuants.

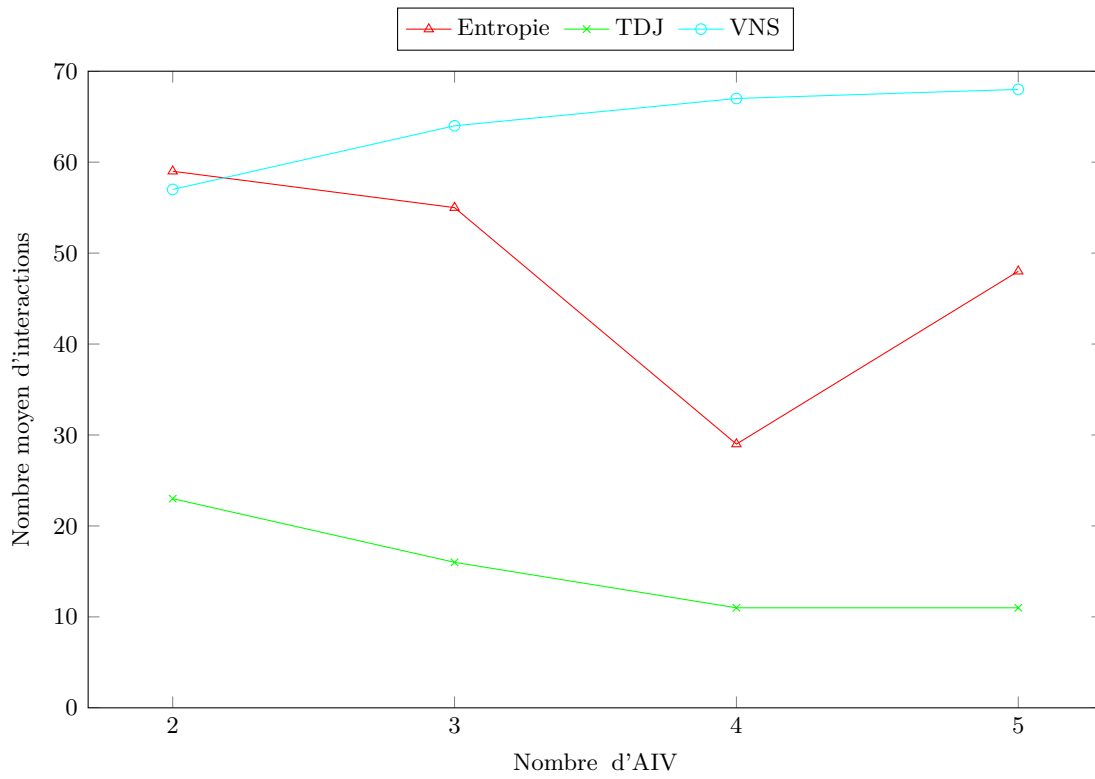


FIGURE 5.10 – Évolution du nombre moyen d’interactions humain-robot en fonction de l’algorithme d’ordonnancement et du nombre d’AIV.

Les résultats expérimentaux confirment la robustesse et l’efficacité de notre approche algorithmique, qui génère des performances compétitives face à des algorithmes d’ordonnancement de référence, à savoir l’algorithme du VNS proposé par Moussa Abderrahim *et al.* [14] et l’algorithme basé sur l’entropie proposé par Xu Yiyi *et al.* [18], dans des environnements déterministes. Notre algorithme se distingue particulièrement dans des contextes marqués par l’imprévisibilité des interactions humaines, où il surpasse les méthodes sus-mentionnées, offrant une solution optimisée pour la gestion des activités industrielles dans des environnements dynamiques.

De plus, cette recherche représente une avancée dans l’optimisation des systèmes de production intégrant des AIV et des travailleurs humains. Notre algorithme d’ordonnancement dynamique offre une solution flexible et robuste pour faire face aux défis posés par la variabilité de l’environnement de production. En améliorant l’efficacité du système de production dans un contexte dynamique, nos travaux ouvrent la voie à des systèmes plus flexibles, résilients et inclusifs, pleinement en phase avec les exigences de l’Industrie 5.0.

Conclusion et perspectives

Conclusion

L'histoire industrielle, marquée par des transformations successives, nous a conduits à l'ère de l'Industrie 5.0 (I5.0), où la collaboration homme-robot est au cœur de la production. Alors que les révolutions précédentes ont mis l'accent sur l'automatisation, l'I5.0 vise à réintégrer l'humain dans des systèmes de production hautement automatisés, en valorisant ses compétences et sa capacité d'adaptation. Cette nouvelle donne industrielle soulève de nombreux défis, notamment celui de l'ordonnancement des tâches dans des environnements où humains et robots cohabitent. L'intégration de véhicules autonomes intelligents (AIV) complexifie encore davantage ces problématiques, nécessitant le développement de nouvelles approches d'ordonnancement capables de gérer la dynamique et l'incertitude inhérentes à ces systèmes collaboratifs. Ainsi, au cours de cette thèse, nous avons exploré en profondeur ces défis, en adoptant une approche essentiellement basée sur la simulation et l'optimisation.

Dans un premier temps (cf. chapitre 1), la thèse a posé les bases théoriques, en retraçant l'évolution des systèmes de production et en précisant la problématique centrale de recherche : optimiser l'ordonnancement conjoint des tâches de transport et de production dans un environnement mixte, où interagissent humains et robots. Le premier chapitre a également été l'occasion de dresser un état de l'art sur la problématique, et de se positionner face à ce qui existe dans la littérature. En l'espèce, ce travail de recherche aborde une problématique nouvelle qui n'a pas encore reçu assez d'attention.

Ensuite, dans le deuxième chapitre (cf. chapitre 2), nous nous sommes attelé au développement d'un simulateur, l'une des contributions principales de cette thèse, offrant un outil efficace pour reproduire et analyser les dynamiques complexes des systèmes de production. Ce simulateur s'est révélé être un levier important dans le troisième chapitre (cf. chapitre 3), afin d'évaluer l'impact de diverses

5.6. CONCLUSION

contraintes, telles que l'évitement des collisions et la gestion des deadlocks, tout en testant différents scénarios d'ordonnancement.

Puis, dans le quatrième chapitre (cf. chapitre 4), l'introduction des AIV a marqué un virage majeur dans l'optimisation de l'ordonnancement, surpassant les véhicules autonomes guidés (AGV) en termes de flexibilité et de performance. L'intégration du facteur humain a également ajouté une nouvelle complexité, affectant les temps de production (makespan) et nécessitant de nouvelles approches d'ordonnancement. À cet égard, dans le cinquième chapitre (cf. chapitre 5), nous avons développé un algorithme basé sur la théorie des jeux, conçu pour s'adapter en temps réel aux variations de l'environnement de production. Il s'est montré particulièrement efficace face aux interactions imprévisibles entre humains et robots.

Les résultats expérimentaux obtenus ont confirmé la robustesse et l'efficacité de l'approche proposée dans cette thèse, qui non seulement optimise l'ordonnancement dans des environnements dynamiques, mais ouvre également la voie à des systèmes de production plus résilients, flexibles et inclusifs. Cette contribution s'inscrit pleinement dans la philosophie de l'I5.0, où la technologie et l'humain coexistent de manière harmonieuse pour améliorer l'efficacité des processus industriels.

Toutefois, cette thèse n'est qu'une étape dans le vaste champ de la recherche sur l'I5.0. Les perspectives futures incluent l'approfondissement de la modélisation des comportements des opérateurs humains dans l'atelier, ainsi que le perfectionnement de notre algorithme d'ordonnancement dynamique. La prise en compte de nouveaux facteurs, tels que les pannes imprévisibles des machines et les contraintes spécifiques à certains environnements de production, pourrait encore améliorer les performances des systèmes. De plus, l'utilisation de technologies telles que le « deep learning » ou le « reinforcement learning » pour améliorer les interactions homme-robot et adapter en temps réel les processus de production représente une voie prometteuse.

En définitive, cette thèse a apporté une contribution notable à la résolution des problématiques d'ordonnancement dans des ateliers de production complexes, tout en jetant les bases pour des systèmes de production de plus en plus adaptatifs et collaboratifs. Ces travaux s'inscrivent dans une dynamique d'innovation qui place l'humain au cœur de l'industrie, tout en exploitant les technologies avancées pour répondre aux défis des systèmes de production du futur.

Perspectives

La recherche présentée dans cette thèse représente une étape majeure vers l'optimisation des processus industriels dans le contexte de l'I5.0. En dépit des avancées significatives réalisées, de nombreuses pistes restent à explorer, tant pour perfectionner les modèles existants que pour envisager de nouvelles applications. L'avenir de la collaboration homme-robot, combinée aux progrès technologiques dans les systèmes d'ordonnancement dynamique, offre un potentiel considérable pour transformer en profondeur les pratiques industrielles. Ainsi, les perspectives envisagées après cette thèse concernent quatre aspects importants : les perspectives immédiates dans la continuité des travaux de la thèse, ainsi que les perspectives à court, moyen et long terme.

Perspectives immédiates (continuité des travaux de la thèse)

Comme nous l'avons rappelé à la fin du chapitre 5, les résultats obtenus méritent d'être validés par une analyse complémentaire réalisée sur un grand nombre d'itérations, afin d'obtenir des résultats beaucoup plus stables et plus précis. On veillera également à implémenter d'autres modèles basés sur la théorie des jeux pour établir une comparaison et trouver le modèle qui fournit les meilleures performances.

Perspectives à court terme (1-3 ans)

À court terme, il serait pertinent de :

- Améliorer les modèles de comportement des humains, en intégrant notamment l'hétérogénéité des individus et les comportements de groupes de travailleurs.
- Développer une approche combinant les systèmes multi-agents, la théorie des jeux et les techniques d'apprentissage automatique pour élaborer une technique beaucoup plus efficace, robuste et évolutive.
- Optimiser les temps de calcul des algorithmes d'ordonnancement en explorant des méthodes de calcul distribuées ou parallèles. Cela permettrait d'améliorer encore la réactivité des systèmes en temps réel.
- Élargir l'application des modèles à d'autres industries (logistique, automobile, aéronautique, etc.) pour évaluer la flexibilité des solutions dans différents contextes.

Perspectives à moyen terme (3-5 ans)

À moyen terme, il serait souhaitable de :

- Poursuivre les expérimentations dans des environnements industriels réels, ou défaut sur des démonstrateurs, afin de valider les modèles proposés. Une implantation sur le terrain permettrait de tester la robustesse des algorithmes face à des situations imprévues non simulées et d'ajuster les paramètres en conséquence.
- Évaluer l'impact environnemental et énergétique des systèmes de production optimisés pour s'assurer que ces solutions répondent aux exigences de durabilité industrielle, réduisant l'empreinte carbone des processus de production.
- Développer des modèles d'ordonnancement qui intègrent la production des produits et le recyclage des rébus et/ou pièces déjà usinées.
- Mesurer et quantifier le bien-être des travailleurs, et l'intégrer dans les critères de performance des industries.

Perspectives à long terme (5-10 ans)

À long terme, les perspectives sont encore plus ambitieuses :

- Développer des usines intelligentes et résilientes, où l'intégration complète des technologies de l'Industrie 5.0 (AIV, IA, IoT, etc.) permettrait d'assurer une gestion optimisée, non seulement des flux de production, mais aussi des ressources humaines, énergétiques et matérielles.
- Mettre en place une normalisation des méthodes d'ordonnancement dans l'industrie afin de garantir une interopérabilité et une application large des solutions proposées, facilitant ainsi la transition de divers secteurs vers l'Industrie 5.0.

Bibliographie

- [1] Ü. Bilge et G. Ulusoy, “A time window approach to simultaneous scheduling of machines and material handling system in an fms,” *Operations Research*, vol. 43, n°. 6, p. 1058–1070, 1995.
- [2] P. K. R. Maddikunta, Q.-V. Pham, P. B, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby et M. Liyanage, “Industry 5.0 : A survey on enabling technologies and potential applications,” *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022.
- [3] I. Chalfoun, “Conception et déploiement des Systèmes de Production Reconfigurables et Agiles (SPRA),” Theses, Université Blaise Pascal - Clermont-Ferrand II, sept. 2014. [En ligne]. Disponible : <https://theses.hal.science/tel-01247600>
- [4] S. Ariaifar et N. Ismail, “An improved algorithm for layout design in cellular manufacturing systems,” *Journal of Manufacturing Systems*, vol. 28, n°. 4, p. 132–139, 2009.
- [5] D. Balderas, A. Ortiz, E. Méndez, P. Ponce et A. Molina, “Empowering digital twin for industry 4.0 using metaheuristic optimization algorithms : case study pcb drilling optimization,” *The International Journal of Advanced Manufacturing Technology*, vol. 113, 03 2021.
- [6] G. C., G. A., J. Mantari, C. Alberto et R. J., “A contemporary approach to the mse paradigm powered by artificial intelligence from a review focused on polymer matrix composites,” *Mechanics of Advanced Materials and Structures*, vol. 29, p. 1–21, 03 2021.
- [7] M. Shyam, K. V. N., G. V. N., S. G. Jangali et K. B. B., “A state-of-the-art review on industry 4.0 and related key technologies,” *AIP Conference Proceedings*, vol. 2358, n°. 1, p. 100012, 07 2021.
- [8] Momenta. (2024) Industry 5.0 | sustainability, human-centric, and resilience. (Consulté le 23/05/2024). [En ligne]. Disponible : <https://www.momenta.one/industry5.0>
- [9] R. Jahanmahin, S. Masoud, J. Rickli et A. Djuric, “Human-robot interactions in manufacturing : A survey of human behavior modeling,” *Robotics and Computer-*

BIBLIOGRAPHIE

- Integrated Manufacturing*, vol. 78, p. 102404, 2022. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0736584522000916>
- [10] J. M. Framinan, R. Leisten et R. R. García, “Manufacturing scheduling systems,” *An integrated view on Models, Methods and Tools*, p. 51–63, 2014.
- [11] A. Group. (2024) Ba systèmes becomes alstef mobile robotics. (Consulté le 17/06/2024). [En ligne]. Disponible : <https://alstefgroup.com/ba-systemes-becomes-alstef-mobile-robotics/>
- [12] Y. Xu, M. Sahnoun, F. Abdelaziz, D. Baudry et A. Louis, “Multi-objective flexible job shop scheduling transportation problem : Simulation approach,” dans *The 12th International Conference on Multiple Objective Programming and Goal Programming*, 10 2017.
- [13] U. Wilensky. (1999) Netlogo home page. (Consulté le 18/07/2024). [En ligne]. Disponible : <http://ccl.northwestern.edu/netlogo/>
- [14] M. Abderrahim, A. Bekrar, D. Trentesaux et et al., “Bi-local search based variable neighborhood search for job-shop scheduling problem with transport constraints,” *Optimization Letters*, 11 2020.
- [15] K. Sanogo, A. Mekhalef Benhafssa, M. Sahnoun, B. Bettayeb, M. Abderrahim et A. Bekrar, “A multi-agent system simulation based approach for collision avoidance in integrated job-shop scheduling problem with transportation tasks,” *Journal of Manufacturing Systems*, vol. 68, p. 209–226, 2023.
- [16] M. Tech. (2024) Industrial mobile amr robot - evobot. (Consulté le 18/08/2024). [En ligne]. Disponible : <https://www.mgtech-group.com/en/mg-tech-moove/agv-aiv-industrial-robots/industrial-mobile-amr-robot-evobot>
- [17] S. Kader, M. B. Abdelkader et S. M’hammed. (2024) A multi-agent system simulation of job shop scheduling with human consideration : A comparative analysis of agvs and aivs. [En ligne]. Disponible : <https://ssrn.com/abstract=4730141>
- [18] Y. Xu, M. Sahnoun, F. B. Abdelaziz et D. Baudry, “A simulated multi-objective model for flexible job shop transportation scheduling,” *Annals of Operations Research*, p. 1–22, 2020.
- [19] M. Akrich, “La construction d’un système socio-technique. esquisse pour une anthropologie des techniques,” *Anthropologie et sociétés*, vol. 13, n^o. 2, p. 31–54, 1989.
- [20] M. Doyle-Kent, “Collaborative robotics in industry 5.0,” Thèse de doctorat, Wien, 2021.

- [21] A. Moussa, “Approach for energy management in industrial systems : use case : scheduling of automated guided vehicles in job-shop manufacturing system with energy constraints,” Thèse de doctorat, Oran University 1, 2021.
- [22] K. Tliba, “Ordonnancement dynamique d’un système de production de type flow-shop hybride à l’aide d’un jumeau numérique,” Thèse de doctorat, CY Cergy Paris Université, 2022.
- [23] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy et H. Van Brussel, “Reconfigurable manufacturing systems,” *CIRP Annals*, vol. 48, n^o. 2, p. 527–540, 1999.
- [24] H. A. ElMaraghy, “Flexible and reconfigurable manufacturing systems paradigms,” *International journal of flexible manufacturing systems*, vol. 17, p. 261–276, 2005.
- [25] N. Singh, “Design of cellular manufacturing systems : An invited review,” *European Journal of Operational Research*, vol. 69, n^o. 3, p. 284–291, 1993, cellular Manufacturing Systems.
- [26] H. ElMaraghy, L. Monostori, G. Schuh et W. ElMaraghy, “Evolution and future of manufacturing systems,” *CIRP Annals*, vol. 70, n^o. 2, p. 635–658, 2021.
- [27] M. Haeffner et K. Panuwatwanich, “Perceived impacts of industry 4.0 on manufacturing industry and its workforce : Case of germany,” dans *8th International Conference on Engineering, Project, and Product Management (EPPM 2017)*, S. Şahin, édit. Cham : Springer International Publishing, 2018, p. 199–208.
- [28] S. Nahavandi, “Industry 5.0—a human-centric solution,” *Sustainability*, vol. 11, n^o. 16, 2019. [En ligne]. Disponible : <https://www.mdpi.com/2071-1050/11/16/4371>
- [29] W. ElMaraghy, H. ElMaraghy, T. Tomiyama et L. Monostori, “Complexity in engineering design and manufacturing,” *CIRP Annals*, vol. 61, n^o. 2, p. 793–814, 2012.
- [30] Y. Koren et M. Shpitalni, “Design of reconfigurable manufacturing systems,” *Journal of Manufacturing Systems*, vol. 29, n^o. 4, p. 130–141, 2010.
- [31] A. Sharma et B. J. Singh, “Evolution of industrial revolutions : A review,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, n^o. 11, p. 66–73, 2020.
- [32] J. J. Browne, D. Dubois, K. Rathmill, S. Sethi et K. Stecke, “Classification of flexible manufacturing systems,” *The FMS magazine*, vol. 2, n^o. 2, p. 114–117, 1984.
- [33] J. L. Burbidge, “Production flow analysis for planning group technology,” *Journal of Operations Management*, vol. 10, n^o. 1, p. 5–27, 1991.

- [34] U. Wemmerlov et D. J. Johnson, “Empirical findings on manufacturing cell design,” *International Journal of Production Research*, vol. 38, n^o. 3, p. 481–507, 2000.
- [35] —, “Cellular manufacturing at 46 user plants : Implementation experiences and performance improvements,” *International Journal of Production Research*, vol. 35, n^o. 1, p. 29–49, 1997.
- [36] Y. Koren, *General RMS Characteristics. Comparison with Dedicated and Flexible Systems*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006, p. 27–45.
- [37] M. G. Mehrabi, A. G. Ulsoy et Y. Koren, “Reconfigurable manufacturing systems : Key to future manufacturing,” *Journal of Intelligent Manufacturing*, vol. 11, p. 403–419, 2000.
- [38] S. E. Hashemi-Petroodi, A. Dolgui, S. Kovalev, M. Y. Kovalyov et S. Thevenin, “Workforce reconfiguration strategies in manufacturing systems : a state of the art,” *International Journal of Production Research*, vol. 59, n^o. 22, p. 6721–6744, 2021.
- [39] P. Zheng, H. Wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarak, S. Yu et X. Xu, “Smart manufacturing systems for industry 4.0 : Conceptual framework, scenarios, and future perspectives,” *Frontiers of Mechanical Engineering*, vol. 13, p. 137–150, 2018.
- [40] J. W. Rittinghouse et J. F. Ransome, *Cloud computing : implementation, management, and security*. CRC press, 2017.
- [41] Y. Zhang, G. Zhang, J. Wang, S. Sun, S. Si et T. Yang, “Real-time information capturing and integration framework of the internet of manufacturing things,” *International Journal of Computer Integrated Manufacturing*, vol. 28, n^o. 8, p. 811–822, 2015.
- [42] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn et K. Ueda, “Cyber-physical systems in manufacturing,” *CIRP Annals*, vol. 65, n^o. 2, p. 621–641, 2016.
- [43] L. Monostori, “Cyber-physical production systems : Roots, expectations and r&d challenges,” *Procedia CIRP*, vol. 17, p. 9–13, 2014, variety Management in Manufacturing.
- [44] S. Dieter, G. Oliver, G. Stefan, H. Moritz, K. Tobias et S. Sebastian, *Produktionsarbeit der Zukunft-Industrie 4.0*. Fraunhofer Verlag Stuttgart, 2013, vol. 150.
- [45] B. Thomas, T. H. Michael et V.-H. Birgit, *Industrie 4.0 in Produktion, Automatisierung und Logistik. Anwendung · Technologien · Migration*. Springer, 2014.
- [46] E. Geisberger et M. Broy, *agendaCPS : Integrierte Forschungsagenda Cyber-Physical Systems*. Springer-Verlag, 2012, vol. 1.

- [47] G. Büchi, M. Cugno et R. Castagnoli, “Smart factory performance and industry 4.0,” *Technological Forecasting and Social Change*, vol. 150, p. 119790, 2020.
- [48] R. W. Whitman, *Essays on a Half Century : Ideas, Policies, and Action*. Routledge, 2019.
- [49] K. Henning, H. Johannes, H. Ariane et W. Wolfgang, *Recommendations for implementing the strategic initiative INDUSTRIE 4.0 : Securing the future of German manufacturing industry ; final report of the Industrie 4.0 Working Group*. Forschungsunion, 2013.
- [50] Y. Lu, “Industry 4.0 : A survey on technologies, applications and open research issues,” *Journal of Industrial Information Integration*, vol. 6, p. 1–10, 2017.
- [51] L. Heiner, F. Peter, K. Hans-Georg, F. Thomas et H. Michael, “Industry 4.0,” *Business & information systems engineering*, vol. 6, p. 239–242, 2014.
- [52] K. Zhou, T. Liu et L. Zhou, “Industry 4.0 : Towards future industrial opportunities and challenges,” dans *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, p. 2147–2152.
- [53] T. H.-J. Uhlemann, C. Lehmann et R. Steinhilper, “The digital twin : Realizing the cyber-physical production system for industry 4.0,” *Procedia CIRP*, vol. 61, p. 335–340, 2017, the 24th CIRP Conference on Life Cycle Engineering.
- [54] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel et M. Harnisch, “Industry 4.0 : The future of productivity and growth in manufacturing industries,” *Boston consulting group*, vol. 9, n^o. 1, p. 54–89, 2015.
- [55] N. Derbel, J. Ghommam et Q. Zhu, *New developments and advances in robot control*. Springer, 2019, vol. 175.
- [56] R. Goel et P. Gupta, *Robotics and Industry 4.0*. Cham : Springer International Publishing, 2020, p. 157–169.
- [57] M. Javaid, A. Haleem, R. P. Singh et R. Suman, “Substantial capabilities of robotics in enhancing industry 4.0 implementation,” *Cognitive Robotics*, vol. 1, p. 58–75, 2021.
- [58] J. Banks, J. Carson, B. Nelson et D. Nicol, *Discrete-Event System Simulation*, 5^e éd. Prentice Hall, 2010.
- [59] D. Mourtzis, M. Doukas et D. Bernidaki, “Simulation in manufacturing : Review and challenges,” *Procedia CIRP*, vol. 25, p. 213–229, 2014, 8th International Conference on Digital Enterprise

- Technology - DET 2014 Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution.
- [60] A. Rojko, “Industry 4.0 concept : Background and overview.” *International journal of interactive mobile technologies*, vol. 11, n°. 5, 2017.
- [61] L. Duan et L. Da Xu, “Data analytics in industry 4.0 : A survey,” *Information Systems Frontiers*, p. 1–17, 2021.
- [62] M. S. Reis et G. Gins, “Industrial process monitoring in the big data/industry 4.0 era : From detection, to diagnosis, to prognosis,” *Processes*, vol. 5, n°. 3, p. 35, 2017.
- [63] J. Lee, H.-A. Kao et S. Yang, “Service innovation and smart analytics for industry 4.0 and big data environment,” *Procedia CIRP*, vol. 16, p. 3–8, 2014, product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems.
- [64] K. P. Jan Stentoft, Kent Adsbøll Wickstrøm et A. Haug, “Drivers and barriers for industry 4.0 readiness and practice : empirical evidence from small and medium-sized manufacturers,” *Production Planning & Control*, vol. 32, n°. 10, p. 811–828, 2021.
- [65] S. Jan, J. K. Wickstrøm, P. Kristian et H. Anders, “Drivers and barriers for industry 4.0 readiness and practice : A sme perspective with empirical evidence,” dans *52nd Hawaii International Conference on System Sciences*, 2019. [En ligne]. Disponible : <http://hdl.handle.net/10125/59952>
- [66] T. M. Cătălina, O. Ionica, A. H. Danial, M.-P. Andreea, T. D. Ioan et C. Sorinel, “Drivers and barriers in using industry 4.0 : A perspective of smes in romania,” *Processes*, vol. 7, n°. 3, 2019.
- [67] PwC. (2016) Industry 4.0 : Building the digital enterprise. (Consulté le 23/05/2024). [En ligne]. Disponible : <https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-your-digital-enterprise-april-2016.pdf>
- [68] A. J. Trappey, C. V. Trappey, U. Hareesh Govindarajan, A. C. Chuang et J. J. Sun, “A review of essential standards and patent landscapes for the internet of things : A key enabler for industry 4.0,” *Advanced Engineering Informatics*, vol. 33, p. 208–229, 2017.
- [69] G. Culot, G. Nassimbeni, G. Orzes et M. Sartor, “Behind the definition of industry 4.0 : Analysis and open questions,” *International Journal of Production Economics*, vol. 226, p. 107617, 2020.
- [70] S.-S. Gábor, V. Bence et V. József, “Impact of industry 4.0 and digitization on labor market for 2030-verification of keynes’ prediction,” *Sustainability*, vol. 13, n°. 14, 2021.

- [71] F. Grigoli, Z. Koczan et P. Topalova, “Automation and labor force participation in advanced economies : Macro and micro evidence,” *European Economic Review*, vol. 126, p. 103443, 2020.
- [72] G. Graetz et G. Michaels, “Robots at Work,” *The Review of Economics and Statistics*, vol. 100, n^o. 5, p. 753–768, 12 2018.
- [73] I. Castelo-Branco, F. Cruz-Jesus et T. Oliveira, “Assessing industry 4.0 readiness in manufacturing : Evidence for the european union,” *Computers in Industry*, vol. 107, p. 22–32, 2019.
- [74] W. E. Forum, “The future of jobs report 2018,” *Insight Report, Centre for the New Economy and Society, Geneva/Switzerland*, 2018.
- [75] M. Ziaei Nafchi et H. Mohelská, “Effects of industry 4.0 on the labor markets of iran and japan,” *Economies*, vol. 6, n^o. 3, 2018.
- [76] C. B. Frey et M. A. Osborne, “The future of employment : How susceptible are jobs to computerisation ?” *Technological Forecasting and Social Change*, vol. 114, p. 254–280, 2017.
- [77] Y. J. Kim, K. Kim et S. Lee, “The rise of technological unemployment and its implications on the future macroeconomic landscape,” *Futures*, vol. 87, p. 1–9, 2017.
- [78] M. A. Islam, “Industry 4.0 : Skill set for employability,” *Social Sciences & Humanities Open*, vol. 6, n^o. 1, p. 100280, 2022.
- [79] S. Karadayi-Usta, “An interpretive structural analysis for industry 4.0 adoption challenges,” *IEEE Transactions on Engineering Management*, vol. 67, n^o. 3, p. 973–978, 2020.
- [80] M. Cugno, R. Castagnoli et G. Büchi, “Openness to industry 4.0 and performance : The impact of barriers and incentives,” *Technological Forecasting and Social Change*, vol. 168, p. 120756, 2021.
- [81] R. Morrar, H. Arman et S. Mousa, “The fourth industrial revolution (industry 4.0) : A social innovation perspective,” *Technology innovation management review*, vol. 7, n^o. 11, p. 12–20, 2017.
- [82] J. Lee, B. Bagheri et C. Jin, “Introduction to cyber manufacturing,” *Manufacturing Letters*, vol. 8, p. 11–15, 2016.
- [83] C. Yu, X. Xu et Y. Lu, “Computer-integrated manufacturing, cyber-physical systems and cloud manufacturing – concepts and relationships,” *Manufacturing Letters*, vol. 6, p. 5–9, 2015.
- [84] E. Rauch, C. Linder et P. Dallasega, “Anthropocentric perspective of production before and within industry 4.0,” *Computers & Industrial Engineering*, vol. 139, p. 105644, 2020.

- [85] A. Santana, P. Afonso, A. Zanin et R. Wernke, “Costing models for capacity optimization in industry 4.0 : Trade-off between used capacity and operational efficiency,” *Procedia Manufacturing*, vol. 13, p. 1183–1190, 2017, manufacturing Engineering Society International Conference 2017, MESIC 2017, 28-30 June 2017, Vigo (Pontevedra), Spain.
- [86] K. Schwab, *The fourth industrial revolution*. Crown Currency, 2017.
- [87] E. Brynjolfsson et A. McAfee, *The second machine age : Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company, 2014.
- [88] A. Moraes, A. M. Carvalho et P. Sampaio, “Lean and industry 4.0 : a review of the relationship, its limitations, and the path ahead with industry 5.0,” *Machines*, vol. 11, n^o. 4, p. 443, 2023.
- [89] U. N. D. of Economic et S. Affairs. (2015) The 17 goals | sustainable development. (Consulté le 22/05/2024). [En ligne]. Disponible : <https://sdgs.un.org/goals>
- [90] M. Fukuyama *et al.*, “Society 5.0 : Aiming for a new human-centered society,” *Japan Spotlight*, vol. 27, n^o. 5, p. 47–50, 2018.
- [91] A. Deguchi, C. Hirai, H. Matsuoka, T. Nakano, K. Oshima, M. Tai et S. Tani, “What is society 5.0,” *Society*, vol. 5, n^o. 0, p. 1–24, 2020.
- [92] Y. Shiroishi, K. Uchiyama et N. Suzuki, “Society 5.0 : For human security and well-being,” *Computer*, vol. 51, n^o. 7, p. 91–95, 2018.
- [93] E. Commission. (2021) Industry 5.0. (Consulté le 22/05/2024). [En ligne]. Disponible : https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50_en#what-is-industry-50
- [94] E. Commission, D.-G. for Research, Innovation, M. Breque, L. De Nul et A. Petridis, *Industry 5.0 – Towards a sustainable, human-centric and resilient European industry*. Publications Office of the European Union, 2021.
- [95] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis et L. Wang, “Industry 5.0 : Prospect and retrospect,” *Journal of Manufacturing Systems*, vol. 65, p. 279–295, 2022.
- [96] X. Xu, Y. Lu, B. Vogel-Heuser et L. Wang, “Industry 4.0 and industry 5.0—inception, conception and perception,” *Journal of Manufacturing Systems*, vol. 61, p. 530–535, 2021.
- [97] P. Johri, J. Singh, A. Sharma et D. Rastogi, “Sustainability of coexistence of humans and machines : An evolution of industry 5.0 from industry 4.0,” dans *2021 10th International Conference*

- on System Modeling & Advancement in Research Trends (SMART)*, 2021, p. 410–414.
- [98] P. Kasinathan, R. Pugazhendhi, R. M. Elavarasan, V. K. Ramachandaramurthy, V. Ramana-
than, S. Subramanian, S. Kumar, K. Nandhagopal, R. R. V. Raghavan, S. Rangasamy, R. De-
vendiran et M. H. Alsharif, “Realization of sustainable development goals with disruptive tech-
nologies by integrating industry 5.0, society 5.0, smart cities and villages,” *Sustainability*, vol. 14,
n°. 22, 2022.
- [99] M. Rada. Industry 5.0 definition. (Consulté le 23/05/2024). [En ligne]. Disponible :
<https://michael-rada.medium.com/industry-5-0-definition-6a2f9922dc48>
- [100] F. Longo, A. Padovano et S. Umbrello, “Value-oriented and ethical technology engineering
in industry 5.0 : A human-centric perspective for the design of the factory of
the future,” *Applied Sciences*, vol. 10, n°. 12, 2020. [En ligne]. Disponible : <https://www.mdpi.com/2076-3417/10/12/4182>
- [101] Y. K. Leong, J. H. Tan, K. W. Chew et P. L. Show, “Simulation optimization using metamodels,”
dans *The Prospect of Industry 5.0 in Biomanufacturing*, 2021, p. 1–20.
- [102] H. D. Nguyen et K. P. Tran, *Artificial Intelligence for Smart Manufacturing in Industry 5.0 : Me-
thods, Applications, and Challenges*, ser. Artificial Intelligence for Smart Manufacturing. Sprin-
ger International Publishing, 2023, vol. Part F4.
- [103] B. Wang, H. Zhou, X. Li, G. Yang, P. Zheng, C. Song, Y. Yuan, T. Wuest, H. Yang et L. Wang,
“Human digital twin in the context of industry 5.0,” *Robotics and Computer-Integrated Manu-
facturing*, vol. 85, p. 102626, 2024.
- [104] C. Zhang, Z. Wang, G. Zhou, F. Chang, D. Ma, Y. Jing, W. Cheng, K. Ding et D. Zhao,
“Towards new-generation human-centric smart manufacturing in industry 5.0 : A systematic
review,” *Advanced Engineering Informatics*, vol. 57, p. 102121, 2023.
- [105] L. J. et B. Alenljung, *Socially Embodied Human-Robot Interaction : Addressing Human Emotions
with Theories of Embodied Cognition*. Handbook of Research on Synthesizing Human Emotion
in Intelligent Systems and Robotics, 2015.
- [106] B. M. Kopp T. et K. S., “Success factors for introducing industrial human-robot interaction in
practice : an empirically driven framework,” *The International Journal of Advanced Manufac-
turing Technology*, n°. 112, p. 685–704, 2021.

- [107] B. Dieber, A. Schlotzhauer et M. Brandstötter, “Safety and security – erfolgskfaktoren von sensitiven robotertechnologien,” *e & i Elektrotechnik und Informationstechnik*, n°. 134, p. 299–303, 2017.
- [108] I. Aaltonen, T. Salmi et I. Marstio, “Refining levels of collaboration to support the design and evaluation of human-robot interaction in the manufacturing industry,” *Procedia CIRP*, vol. 72, p. 93–98, 2018, 51st CIRP Conference on Manufacturing Systems. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S2212827118303743>
- [109] D. Ouelhadj et S. Petrovic, “A survey of dynamic scheduling in manufacturing systems,” *Journal of scheduling*, vol. 12, p. 417–431, 2009.
- [110] M. Mazar, “Simulation et optimisation de la gestion dynamique de tâches évolutives sur des robots mobiles autonomes. Application sur le traitement UV-C robotisé dans l’horticulture.” Theses, HESAM Université, janv. 2022. [En ligne]. Disponible : <https://pastel.hal.science/tel-03624997>
- [111] J. Zhang, G. Ding, Y. Zou et *et al.*, “Review of job shop scheduling research and its new perspectives under industry 4.0,” *J Intell Manuf*, vol. 30, 2019.
- [112] B. Cunha, A. M. Madureira, B. Fonseca et D. Coelho, “Deep reinforcement learning as a job shop scheduling solver : A literature review,” dans *Hybrid Intelligent Systems*, A. M. Madureira, A. Abraham, N. Gandhi et M. L. Varela, édit. Springer International Publishing, 2020, p. 350–359.
- [113] J. Mohan, K. Lanka et A. N. Rao, “A review of dynamic job shop scheduling techniques,” *Procedia Manufacturing*, vol. 30, p. 34–39, 2019, digital Manufacturing Transforming Industry Towards Sustainable Growth.
- [114] O. H. Owiti, “Game theoretic multi-agent algorithms for the job shop scheduling problem,” Mémoire de maîtrise, UNIVERSITY OF NAIROBI, 2014.
- [115] H. Abedinnia, C. H. Glock et M. D. Schneider, “Machine scheduling in production : A content analysis,” *Applied Mathematical Modelling*, vol. 50, p. 279–299, 2017.
- [116] D. W. Gage, *UGV history 101 : A brief history of Unmanned Ground Vehicle (UGV) development efforts*. Naval Ocean Systems Center San Diego, CA, USA, 1995.
- [117] C. Cronin, A. Conway et J. Walsh, “State-of-the-art review of autonomous intelligent vehicles (aiv) technologies for the automotive and manufacturing industry,” dans *2019 30th Irish Signals*

- and Systems Conference (ISSC)*, 2019, p. 1–6.
- [118] H. Hu, X. Jia, Q. He, S. Fu et K. Liu, “Deep reinforcement learning based agvs real-time scheduling with mixed rule for flexible shop floor in industry 4.0,” *Computers & Industrial Engineering*, vol. 149, p. 106749, 2020.
- [119] L. Martin, M. González-Romo, M. Sahnoun, B. Bettayeb, N. He et J. Gao, “Effect of human-robot interaction on the fleet size of aiv transporters in fms,” dans *2021 1st International Conference On Cyber Management And Engineering (CyMaEn)*, 2021, p. 1–5.
- [120] M. Aizat, A. Azmin et W. Rahiman, “A survey on navigation approaches for automated guided vehicle robots in dynamic surrounding,” *IEEE Access*, vol. 11, p. 33 934–33 955, 2023.
- [121] C. Destouet, H. Tlahig, B. Bettayeb et B. Mazari, “Flexible job shop scheduling problem under industry 5.0 : A survey on human reintegration, environmental consideration and resilience improvement,” *Journal of Manufacturing Systems*, vol. 67, p. 155–173, 2023. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S027861252300016X>
- [122] M. Sahnoun, Y. Xu, F. B. Abdelaziz et D. Baudry, “Optimization of transportation collaborative robots fleet size in flexible manufacturing systems,” dans *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*. IEEE, 2019, p. 1–5.
- [123] L. Martin, M. González-Romo, M. Sahnoun, B. Bettayeb, N. He et J. Gao, “Effect of human-robot interaction on the fleet size of aiv transporters in fms,” dans *2021 1st International Conference On Cyber Management And Engineering (CyMaEn)*. IEEE, 2021, p. 1–5.
- [124] S. Kader, S. M’hammed et M. B. Abdelkader, “Multi-agent simulation for scheduling and path planning of autonomous intelligent vehicles,” dans *Simulation Tools and Techniques*, G.-L. José-Luis, R.-N. Agustín, M.-F. María-José et W. Gabriel, édit. Cham : Springer Nature Switzerland, 2024, p. 195–205.
- [125] S. Panagou, F. Fruggiero, W. P. Neumann et A. Lambiase, “The entropic complexity of human factor in collaborative technologies,” dans *Congress of the International Ergonomics Association*. Springer, 2021, p. 503–510.
- [126] P. O. Skobelev et S. Y. Borovik, “On the way from industry 4.0 to industry 5.0 : From digital manufacturing to digital society,” *Industry 4.0*, vol. 2, n^o. 6, p. 307–311, 2017.
- [127] D. B. Fontes, S. M. Homayouni et J. F. Gonçalves, “A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources,”

- European Journal of Operational Research*, vol. 306, n^o. 3, p. 1140–1157, 2023.
- [128] S. Fabio, G. Eric, N. W., B. Daria et G. Christoph, “Human factors in production and logistics systems of the future,” *Annual Reviews in Control*, vol. 49, 05 2020.
- [129] J. Petronijevic, A. Etienne et J.-Y. Dantan, “Human factors under uncertainty : A manufacturing systems design using simulation-optimisation approach,” *Computers & Industrial Engineering*, vol. 127, p. 665–676, 2019.
- [130] A. Ahmadi-Javid et P. Hooshangi-Tabrizi, “Integrating employee timetabling with scheduling of machines and transporters in a job-shop environment : A mathematical formulation and an anarchic society optimization algorithm,” *Computers & Operations Research*, vol. 84, p. 73–91, 2017.
- [131] L. Gualtieri, E. Rauch et R. Vidoni, “Development and validation of guidelines for safety in human-robot collaborative assembly systems,” *Computers& Industrial Engineering*, vol. 163, p. 107801, 2022.
- [132] J. G. Martin, F. J. Muros, J. M. Maestre et E. F. Camacho, “Multi-robot task allocation clustering based on game theory,” *Robotics and Autonomous Systems*, vol. 161, p. 104314, 2023.
- [133] L. Nie, X. Wang et F. Pan, “A game-theory approach based on genetic algorithm for flexible job shop scheduling problem,” *Journal of Physics : Conference Series*, vol. 1187, n^o. 3, p. 032095, apr 2019.
- [134] P. Burggräf, J. Wagner, T. Saßmannshausen, D. Ohrndorf et K. Subramani, “Multi-agent-based deep reinforcement learning for dynamic flexible job shop scheduling,” *Procedia CIRP*, vol. 112, p. 57–62, 2022, 15th CIRP Conference on Intelligent Computation in ManufacturingEngineering, 14-16 July 2021.
- [135] P. Renna, M. Thürer et M. Stevenson, “A game theory model based on gale-shapley for dual-resource constrained (drc) flexible job shop scheduling,” *International Journal of Industrial Engineering Computations*, vol. 11, n^o. 2, p. 173–184, 2020.
- [136] C. Hofmann, X. Liu, M. May et G. Lanza, “Hybrid monte carlo tree search based multi-objective scheduling,” *Production Engineering*, vol. 17, n^o. 1, p. 133–144, 2023.
- [137] J. Wang, J. Yang, Y. Zhang, S. Ren et Y. Liu, “Infinitely repeated game based real-time scheduling for low-carbon flexible job shop considering multi-time periods,” *Journal of Cleaner Production*, vol. 247, p. 119093, 2020.

- [138] J. Liu, B. Sun, G. Li et Y. Chen, “An integrated scheduling approach considering dispatching strategy and conflict-free route of amrs in flexible job shop,” *The International Journal of Advanced Manufacturing Technology*, vol. 127, n^o. 3, p. 1979–2002, 2023.
- [139] H. Zou et Y. Xi, “Decentralised task allocation using gdl negotiations in multi-agent system,” *Cognitive Robotics*, vol. 1, p. 197–204, 2021.
- [140] H. Wu et H. Shang, “Potential game for dynamic task allocation in multi-agent system,” *ISA Transactions*, vol. 102, p. 208–220, 2020.
- [141] Z. Xiaofei, X. Jiazhong, G. Jianghua, W. Yaping et X. Zhiqiang, “Multi-task multi-agent reinforcement learning for real-time scheduling of a dual-resource flexible job shop with robots,” *Processes*, vol. 11, n^o. 1, 2023. [En ligne]. Disponible : <https://www.mdpi.com/2227-9717/11/1/267>
- [142] A.-A. M. Sayed, A. A. Fayez et M. S. M, “Developing a multi-objective flexible job shop scheduling optimization model using lexicographic procedure considering transportation time,” *Journal of Umm Al-Qura University for Engineering and Architecture*, vol. 14, n^o. 1, p. 57–70, 2023.
- [143] M. Petrović, A. Jokić, Z. Miljković et Z. Kulesza, “Multi-objective scheduling of a single mobile robot based on the grey wolf optimization algorithm,” *Applied Soft Computing*, vol. 131, p. 109784, 2022.
- [144] J. Yang et H. Xu, “Hybrid memetic algorithm to solve multiobjective distributed fuzzy flexible job shop scheduling problem with transfer,” *Processes*, vol. 10, n^o. 8, 2022.
- [145] F. Zhang, R. Li et W. Gong, “Deep reinforcement learning-based memetic algorithm for energy-aware flexible job shop scheduling with multi-agv,” *Computers & Industrial Engineering*, vol. 189, p. 109917, 2024.
- [146] L. Zaidi, B. Bettayeb et M. Sahnoun, “Optimisation and simulation of transportation tasks in flexible job shop with muti-robot systems,” dans *2021 1st International Conference On Cyber Management And Engineering (CyMaEn)*. IEEE, 2021, p. 1–6.
- [147] B. Marzouki, O. B. Driss et K. Ghedira, “Improved chemical reaction optimization for distributed flexible job shop problem with transportation times,” *IFAC-PapersOnLine*, vol. 55, n^o. 10, p. 1249–1254, 2022, 10th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2022.

- [148] C. Peng, Z. Li, H. Zhong, X. Li, A. Lin et Y. Liao, “Research on multi-objective scheduling algorithm of job shop considering limited storage and transportation capacity,” *IEEE Access*, vol. 11, p. 94 252–94 280, 2023.
- [149] S. M. H. Dalila B.M.M. Fontes et J. C. Fernandes, “Energy-efficient job shop scheduling problem with transport resources considering speed adjustable resources,” *International Journal of Production Research*, vol. 62, n^o. 3, p. 867–890, 2024.
- [150] J.-P. Huang, L. Gao et X.-Y. Li, “A hierarchical multi-action deep reinforcement learning method for dynamic distributed job-shop scheduling problem with job arrivals,” *IEEE Transactions on Automation Science and Engineering*, p. 1–13, 2024.
- [151] C. H. Lim et S. K. Moon, “A two-phase iterative mathematical programming-based heuristic for a flexible job shop scheduling problem with transportation,” *Applied Sciences*, vol. 13, n^o. 8, 2023.
- [152] A. Sobhani, “Integrating human factors aspects into performance optimization models of a serial assembly system,” Thèse de doctorat, Toronto Metropolitan University, 1 2014. [En ligne]. Disponible : https://rshare.library.torontomu.ca/articles/thesis/Integrating_human_factors_aspects_into_performance_optimization_models_of_a_serial_assembly_system/14655669
- [153] I. E. Mouayni, G. Demesure, H. B.-E. Haouzi, P. Charpentier et A. Siadat, “Jobs scheduling within industry 4.0 with consideration of worker’s fatigue and reliability using greedy randomized adaptive search procedure,” *IFAC-PapersOnLine*, vol. 52, n^o. 19, p. 85–90, 2019, 14th IFAC Symposium on Analysis, Design, and Evaluation of Human Machine Systems HMS 2019.
- [154] Z. H. Behdin Vahedi-Nouri, Reza Tavakkoli-Moghaddam et A. Dolgui, “Production scheduling in a reconfigurable manufacturing system benefiting from human-robot collaboration,” *International Journal of Production Research*, vol. 62, n^o. 3, p. 767–783, 2024.
- [155] Y. L. Lei Cai, Wenfeng Li et L. He, “Real-time scheduling simulation optimisation of job shop in a production-logistics collaborative environment,” *International Journal of Production Research*, vol. 61, n^o. 5, p. 1373–1393, 2023.
- [156] C. Destouet, H. Tlahig, B. Bettayeb et B. Mazari, “Nsga-ii for solving a multi-objective, sustainable and flexible job shop scheduling problem,” dans *Advances in Production Management Systems. Production Management Systems for Responsible Manufacturing, Service, and Logis-*

- tics Futures*, E. Alfnes, A. Romsdal, J. O. Strandhagen, G. von Cieminski et D. Romero, édit. Cham : Springer Nature Switzerland, 2023, p. 548–562.
- [157] R. Bridger, *Introduction to human factors and ergonomics*. CRC press, 2017.
- [158] M. Yuan, L. Zheng, H. Huang, K. Zhou, F. Pei et W. Gu, “Research on flexible job shop scheduling problem with agv using double dqn,” *Journal of Intelligent Manufacturing*, p. 1–27, 2023.
- [159] Y. Wang, J.-Q. Wang et Y. Yin, “Multitasking scheduling and due date assignment with deterioration effect and efficiency promotion,” *Computers & Industrial Engineering*, vol. 146, p. 106569, 2020.
- [160] E. Şenyiğit, U. Atici et M. B. Şenol, “Effects of ocr parameters and learning rate on machine scheduling,” *Central European Journal of Operations Research*, vol. 30, n^o. 3, p. 941–959, 2022.
- [161] J. Soriano, E. Jalao et I. Martinez, “Integrated employee scheduling with known employee demand, including breaks, overtime, and employee preferences,” *Journal of Industrial Engineering and Management*, vol. 13, p. 451, 09 2020.
- [162] A. Khaitan, M. Kumar Mehlawat, P. Gupta et W. Pedrycz, “Socially aware fuzzy vehicle routing problem : A topic modeling based approach for driver well-being,” *Expert Systems with Applications*, vol. 205, p. 117655, 2022.
- [163] V. B.-G. M. P. Doan, J. Fondrevelle et J. F. F. Ribeiro, “Impact of flexible work contracts and multi-skilled agents on a multi-objective workforce planning problem,” *International Journal of Production Research*, vol. 60, n^o. 5, p. 1650–1665, 2022.
- [164] Z. Peng, H. Zhang, H. Tang, Y. Feng et W. Yin, “Research on flexible job-shop scheduling problem in green sustainable manufacturing based on learning effect,” *Journal of Intelligent Manufacturing*, p. 1–22, 2022.
- [165] L. Jian, L. Huankun, H. Pengbo, X. Liping, H. Kui et L. Shanhui, “Flexible job shop scheduling optimization for green manufacturing based on improved multi-objective wolf pack algorithm,” *Applied Sciences*, vol. 13, n^o. 14, 2023.
- [166] F. Yanming, L. Zhuohang, C. Ningjiang et Q. Chiwen, “A discrete multi-objective rider optimization algorithm for hybrid flowshop scheduling problem considering makespan, noise and dust pollution,” *IEEE Access*, vol. 8, p. 88 527–88 546, 2020.

- [167] F. Pistolesi et B. Lazzerini, “Tema : A tensorial memetic algorithm for many-objective parallel disassembly sequence planning in product refurbishment,” *IEEE Transactions on Industrial Informatics*, vol. 15, n^o. 6, p. 3743–3753, 2019.
- [168] C. H. Glock, E. H. Grosse, H. Abedinnia et S. Emde, “An integrated model to improve ergonomic and economic performance in order picking by rotating pallets,” *European Journal of Operational Research*, vol. 273, n^o. 2, p. 516–534, 2019.
- [169] Z. Shoujing, D. Haotian, B. Sebastian, J. Shoufeng, H. Tiantian et L. Zhixiong, “Dual resource constrained flexible job shop scheduling based on improved quantum genetic algorithm,” *Machines*, vol. 9, n^o. 6, 2021.
- [170] L. Hongyu et W. Xiuli, “A survival duration-guided nsga-iii for sustainable flexible job shop scheduling problem considering dual resources,” *IET Collaborative Intelligent Manufacturing*, vol. 3, n^o. 2, p. 119–130, 2021.
- [171] M. Dai, D. Tang, A. Giret et M. A. Salido, “Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints,” *Robotics and Computer-Integrated Manufacturing*, vol. 59, p. 143–157, 2019.
- [172] W. Weng, J. Chen, M. Zheng et S. Fujimura, “Realtime scheduling heuristics for just-in-time production in large-scale flexible job shops,” *Journal of Manufacturing Systems*, vol. 63, p. 64–77, 2022.
- [173] M. Rinaldi, M. Fera, E. Bottani et E. H. Grosse, “Workforce scheduling incorporating worker skills and ergonomic constraints,” *Computers & Industrial Engineering*, vol. 168, p. 108107, 2022.
- [174] Y. Liu, L. Zhang et T. Sun, “An improved nondominated sorting genetic algorithm-ii for multi-objective flexible job-shop scheduling problem considering worker assignments,” dans *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2021, p. 837–843.
- [175] W. Tan, X. Yuan, J. Wang et X. Zhang, “A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced nsga-ii : An application from casting workshop,” *Computers & Industrial Engineering*, vol. 160, p. 107557, 2021.
- [176] S. Xiao, Z. Wu et S. Yu, “A two-stage assignment strategy for the robust scheduling of dual-resource constrained stochastic job shop scheduling problems,” *IFAC-PapersOnLine*, vol. 52,

- n°. 13, p. 421–426, 2019, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.
- [177] A. Gnanavelbabu, R. H. Caldeira et T. Vaidyanathan, “A simulation-based modified backtracking search algorithm for multi-objective stochastic flexible job shop scheduling problem with worker flexibility,” *Applied Soft Computing*, vol. 113, p. 107960, 2021.
- [178] Q. D. Guiliang Gong, Raymond Chiong et X. Gong, “A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility,” *International Journal of Production Research*, vol. 58, n°. 14, p. 4406–4420, 2020.
- [179] D. Müller et D. Kress, “Filter-and-fan approaches for scheduling flexible job shops under workforce constraints,” *International Journal of Production Research*, vol. 60, n°. 15, p. 4743–4765, 2022.
- [180] F. Farah, N. H. Eddine et B. D. Olfa, “Multi-start tabu agents-based model for the dual-resource constrained flexible job shop scheduling problem,” dans *Computational Collective Intelligence*, N. T. Nguyen, Y. Manolopoulos, R. Chbeir, A. Kozierekiewicz et B. Trawiński, édit. Cham : Springer International Publishing, 2022, p. 674–686.
- [181] D. Kress, D. Müller et J. Nossack, “A worker constrained flexible job shop scheduling problem with sequence-dependent setup times,” *OR spectrum*, vol. 41, p. 179–217, 2019.
- [182] P. Stączek, J. Pizoń, W. Danilczuk et A. Gola, “A digital twin approach for the improvement of an autonomous mobile robots (amr’s) operating environment—a case study,” *Sensors*, vol. 21, n°. 23, 2021.
- [183] A. Vital-Soto, M. F. Baki et A. Azab, “A multi-objective mathematical model and evolutionary algorithm for the dual-resource flexible job-shop scheduling problem with sequencing flexibility,” *Flexible Services and Manufacturing Journal*, vol. 35, n°. 3, p. 626–668, September 2023.
- [184] Q. Luo, Q. Deng, G. Gong, X. Guo et X. Liu, “A distributed flexible job shop scheduling problem considering worker arrangement using an improved memetic algorithm,” *Expert Systems with Applications*, vol. 207, p. 117984, 2022.
- [185] A. Mahdavi et M. Carvalho, “Optimal trajectory and schedule planning for autonomous guided vehicles in flexible manufacturing system,” dans *2018 Second IEEE International Conference on Robotic Computing (IRC)*, 2018, p. 167–172.

- [186] L. Berterrotière, S. Dauzère-Pérès et C. Yugma, “Flexible job-shop scheduling with transportation resources,” *European Journal of Operational Research*, vol. 312, n^o. 3, p. 890–909, 2024.
- [187] W. Li, L. He et Y. Cao, “Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers,” *IEEE Transactions on Cybernetics*, vol. 52, n^o. 10, p. 10 721–10 734, 2022.
- [188] M. Čech, P. Wicher, R. Lenort, T. Malčic, J. David, D. Holman, D. Staš et J. Záruba, “Autonomous mobile robot technology for supplying assembly lines in the automotive industry,” *Acta logistica*, vol. 7, n^o. 2, p. 103–109, 2020.
- [189] S. Bakshi, T. Feng, Z. Yan et D. Chen, “Fast Scheduling of Autonomous Mobile Robots Under Task Space Constraints With Priorities,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, n^o. 7, p. 071009, 04 2019.
- [190] B. Chaima, Z. Youcef, P. Denis, I. Fares et L. Yiliu, “Distributed vs. hybrid control architecture using stpa and ahp-application to an autonomous mobile multi-robot system,” *International Journal of Safety and Security Engineering*, vol. 11, p. 1–12, 2021.
- [191] Y.-J. Yao, Q.-H. Liu, X.-Y. Li et L. Gao, “A novel milp model for job shop scheduling problem with mobile robots,” *Robotics and Computer-Integrated Manufacturing*, vol. 81, p. 102506, 2023.
- [192] B. Khadidja, G. Latéfa et M. Fouad, “Mathematical formulation of the permutation flow-shop scheduling problem considering transport activity,” dans *2023 International Conference on Decision Aid Sciences and Applications (DASA)*, 2023, p. 296–300.
- [193] C. Zhengying, F. Yihang, Y. Shanshan et Y. Jia, “A state transition diagram and an artificial physarum polycephalum colony algorithm for the flexible job shop scheduling problem with transportation constraints,” *Processes*, vol. 11, n^o. 9, 2023.
- [194] S. Gaiardelli, D. Carra, S. Spellini et F. Fummi, “On the impact of transport times in flexible job shop scheduling problems,” dans *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, p. 1–8.
- [195] Q. Liu, N. Wang, J. Li, T. Ma, F. Li et Z. Gao, “Research on flexible job shop scheduling optimization based on segmented agv,” *Computer Modeling in Engineering & Sciences*, vol. 134, n^o. 3, p. 2073–2091, 2023.
- [196] Q. Liu, C. Wang, X. Li et L. Gao, “An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated

- guided vehicle transportation task,” *Journal of Manufacturing Systems*, vol. 70, p. 127–136, 2023.
- [197] H. Mahdi, F. Dalila et G. José, “A multistart biased random key genetic algorithm for the flexible job shop scheduling problem with transportation,” *International Transactions in Operational Research*, vol. 30, p. 1–29, 10 2023.
- [198] Q. Gao, F. GU, L. Li et J. Guo, “A framework of cloud-edge collaborated digital twin for flexible job shop scheduling with conflict-free routing,” *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102672, 2024.
- [199] Z.-Q. Zhang, F.-C. Wu, B. Qian, R. Hu, L. Wang et H.-P. Jin, “A q-learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation,” *Expert Systems with Applications*, vol. 234, p. 121050, 2023.
- [200] Y. Yao, L. Gui, X. Li et L. Gao, “Tabu search based on novel neighborhood structures for solving job shop scheduling problem integrating finite transportation resources,” *Robotics and Computer-Integrated Manufacturing*, vol. 89, p. 102782, 2024.
- [201] G. Zhang, S. Yan, X. Song, D. Zhang et S. Guo, “Evolutionary algorithm incorporating reinforcement learning for energy-conscious flexible job-shop scheduling problem with transportation and setup times,” *Engineering Applications of Artificial Intelligence*, vol. 133, p. 107974, 2024.
- [202] Y. Youjie, L. Qihao, F. Ling, L. Xinyu, Y. Yanbin, G. Liang et Z. Wei, “A novel mathematical model for the flexible job-shop scheduling problem with limited automated guided vehicles,” *IEEE Transactions on Automation Science and Engineering*, p. 1–14, 2024.
- [203] H. Lou, X. Wang, Z. Dong et Y. Yang, “Memetic algorithm based on learning and decomposition for multiobjective flexible job shop scheduling considering human factors,” *Swarm and Evolutionary Computation*, vol. 75, p. 101204, 2022.
- [204] J. Liu, B. Sun, G. Li et Y. Chen, “Multi-objective adaptive large neighbourhood search algorithm for dynamic flexible job shop schedule problem with transportation resource,” *Engineering Applications of Artificial Intelligence*, vol. 132, p. 107917, 2024.
- [205] Y. Yao, X. Li et L. Gao, “A dqn-based memetic algorithm for energy-efficient job shop scheduling problem with integrated limited agvs,” *Swarm and Evolutionary Computation*, vol. 87, p. 101544, 2024.

BIBLIOGRAPHIE

- [206] R. Lehman, *Computer Simulation and Modeling : An Introduction*. Lawrence Erlbaum Associates, 1977. [En ligne]. Disponible : <https://books.google.fr/books?id=IAEnAAAAMAAJ>
- [207] A. A. B. Pritsker, “Compilation of definitions of simulation,” *SIMULATION*, vol. 33, n^o. 2, p. 61–63, 1979.
- [208] A. Pritsker et C. Pegden, *Introduction to Simulation and SLAM*, ser. A Halsted Press book. Wiley, 1979. [En ligne]. Disponible : <https://books.google.fr/books?id=atomAAAAMAAJ>
- [209] C. Ruiz-Martin, “Theory and foundations of modeling and simulation,” *SIMULATION*, vol. 99, n^o. 5, p. 431–432, 2023.
- [210] G. Zambrano Rey, T. Bonte, V. Prabhu et D. Trentesaux, “Reducing myopic behavior in fms control : A semi-heterarchical simulation–optimization approach,” *Simulation Modelling Practice and Theory*, vol. 46, p. 53–75, 2014, simulation-Optimization of Complex Systems : Methods and Applications. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1569190X14000069>
- [211] F. Dunke et S. Nickel, “Day-ahead and online decision-making for collaborative on-site logistics,” *Journal of Simulation*, vol. 13, n^o. 2, p. 138–151, 2019.
- [212] J. de Mooij, P. Bhattacharya, D. Dell’Anna, M. Dastani, B. Logan et S. Swarup, “A framework for modeling human behavior in large-scale agent-based epidemic simulations,” *SIMULATION*, vol. 99, n^o. 12, p. 1183–1211, 2023.
- [213] C. Adam et H. Arduin, “Agent-based epidemics simulation to compare and explain screening and vaccination prioritization strategies,” *SIMULATION*, vol. 100, n^o. 4, p. 335–355, 2024.
- [214] Y. Ibrahim et R. Hassan, “A revolutionary crowd model : Implemented to contrast oscillating to consistent media influence on crowd behavior,” *SIMULATION*, vol. 93, n^o. 11, p. 951–971, 2017.
- [215] A. Drogoul, “De la simulation multi-agents à la résolution collective de problèmes,” Thèse de doctorat, Thesis at University of Paris IV, 1993.
- [216] G. P. O’Driscoll et M. J. Rizzo, “The economics of time and ignorance,” (*No Title*), 1996.
- [217] G. Diaz, *Numerical Simulations*. John Wiley & Sons, Ltd, 2022, ch. 6, p. 115–132.
- [218] B. Lv, B. Lin, Z. Cao, W. Liu et G. Wang, “Numerical simulation and experimental investigation of structured surface generated by 3d vibration-assisted milling,” *Journal of Manufacturing Processes*, vol. 89, p. 371–383, 2023.

- [219] N. Roussel, J. Spangenberg, J. Wallevik et R. Wolfs, “Numerical simulations of concrete processing : From standard formative casting to additive manufacturing,” *Cement and Concrete Research*, vol. 135, p. 106075, 09 2020.
- [220] E. Ouda, A. Sleptchenko et M. C. E. Simsekler, “Comprehensive review and future research agenda on discrete-event simulation and agent-based simulation of emergency departments,” *Simulation Modelling Practice and Theory*, vol. 129, p. 102823, 2023.
- [221] B. P. Zeigler, H. Praehofer et T. G. Kim, *Theory of modeling and simulation*. Academic press, 2000.
- [222] G. A. Wainer, *Discrete-event modeling and simulation : a practitioner’s approach*. CRC press, 2017.
- [223] A. Allal, M. Sahnoun, R. Adjoudj, S. Benslimane et M. Mazar, “Multi-agent based simulation-optimization of maintenance routing in offshore wind farms,” *Computers & Industrial Engineering*, vol. 157, p. 107342, 2021.
- [224] M. Merdan, T. Moser, W. Sunindyo, S. Biffi et P. Vrba, “Workflow scheduling using multi-agent systems in a dynamically changing environment,” *Journal of Simulation*, vol. 7, n^o. 3, p. 144–158, 2013.
- [225] W. Jia-Min, L. Qiu-Ping, G. Chao, Z. Ming-Lu, G. Pei, T. Xun et H. Yong-Hua, “Application of discrete event simulation model in analysis on cost-effectiveness of epidemiology screening,” *Zhonghua liu Xing Bing xue za zhi= Zhonghua Liuxingbingxue Zazhi*, vol. 44, n^o. 3, p. 463–469, 2023.
- [226] G. Datseris, A. R. Vahdati et T. C. DuBois, “Agents.jl : a performant and feature-full agent-based modeling software of minimal code complexity,” *SIMULATION*, p. 00375497211068820, 2022.
- [227] C. M. Macal, “Everything you need to know about agent-based modelling and simulation,” *Journal of Simulation*, vol. 10, n^o. 2, p. 144–156, 2016.
- [228] A. J. Collins et E. Frydenlund, “Strategic group formation in agent-based simulation,” *SIMULATION*, vol. 94, n^o. 3, p. 179–193, 2018.
- [229] G. Grigoryan, S. Etemadidavan et A. J. Collins, “Computerized agents versus human agents in finding core coalition in glove games,” *SIMULATION*, vol. 98, n^o. 9, p. 807–821, 2022.

- [230] B. Yao, Z. Wang, Q. Cao, L. Jin et M. Zhang, “Express bus fare optimization based on passenger choice behavior,” *SIMULATION*, vol. 92, n^o. 7, p. 617–625, 2016.
- [231] A. García-Suárez, J.-L. Guisado-Lizar, F. Diaz-del Rio et F. Jiménez-Morales, “A cellular automata agent-based hybrid simulation tool to analyze the deployment of electric vehicle charging stations,” *Sustainability*, vol. 13, n^o. 10, 2021. [En ligne]. Disponible : <https://www.mdpi.com/2071-1050/13/10/5421>
- [232] F. Riaz et M. A. Niazi, “Enhanced emotion enabled cognitive agent-based rear-end collision avoidance controller for autonomous vehicles,” *SIMULATION*, vol. 94, n^o. 11, p. 957–977, 2018.
- [233] A. M. Benhafssa, M. Sahnoun, B. Bettayeb et A. Bekrar, “Optimizing energy-conscious dynamic flexible job shop scheduling : Multi-agent simulation approach,” dans *2021 1st International Conference On Cyber Management And Engineering (CyMaEn)*, 2021, p. 1–6.
- [234] N. Bouaziz, B. Bettayeb, M. Sahnoun, A. Yassine et A. Latreche, “Modeling and simulation of human behavior impact on production throughput,” *IFAC-PapersOnLine*, vol. 55, n^o. 10, p. 1740–1745, 2022, 10th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2022.
- [235] P. Leitão, “Agent-based distributed manufacturing control : A state-of-the-art survey,” *Engineering Applications of Artificial Intelligence*, vol. 22, n^o. 7, p. 979–991, 2009, distributed Control of Production Systems.
- [236] S. Hosseini, M. A. Azgomi et A. R. Torkaman, “Agent-based simulation of the dynamics of malware propagation in scale-free networks,” *SIMULATION*, vol. 92, n^o. 7, p. 709–722, 2016.
- [237] V. Sanz et A. Urquia, “Combining pdevs and modelica for describing agent-based models,” *SIMULATION*, vol. 99, n^o. 5, p. 455–474, 2023.
- [238] S. Tisue et U. Wilensky, “Netlogo : A simple environment for modeling complexity,” dans *International conference on complex systems*, vol. 21. Citeseer, 2004, p. 16–21.
- [239] S. F. Railsback, S. L. Lytinen et S. K. Jackson, “Agent-based simulation platforms : Review and development recommendations,” *SIMULATION*, vol. 82, n^o. 9, p. 609–623, 2006.
- [240] M. De Ryck, M. Versteyhe et F. Debrouwere, “Automated guided vehicle systems, state-of-the-art control algorithms and techniques,” *Journal of Manufacturing Systems*, vol. 54, p. 152–173, 2020.

- [241] P. Malega, V. Gazda et V. Rudy, “Optimization of production system in plant simulation,” *SIMULATION*, vol. 98, n^o. 4, p. 295–306, 2022.
- [242] X. Li et L. Gao, “An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem,” *International Journal of Production Economics*, vol. 174, p. 93–110, 2016.
- [243] K. L. Hoffman et T. K. Ralphs, “Integer and combinatorial optimization,” *Encyclopedia of Operations Research and Management Science*, p. 771–783, 2013.
- [244] G. I. Zobolas, C. D. Tarantilis et G. Ioannou, *Exact, Heuristic and Meta-heuristic Algorithms for Solving Shop Scheduling Problems*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008, p. 1–40.
- [245] A. Ham, “Transfer-robot task scheduling in job shop,” *International Journal of Production Research*, vol. 59, n^o. 3, p. 813–823, 2021.
- [246] D. B. M. Fontes et S. M. Homayouni, “Joint production and transportation scheduling in flexible manufacturing systems,” *Journal of Global Optimization*, vol. 74, n^o. 4, p. 879–908, 2019.
- [247] A. H. Gandomi, X.-S. Yang, S. Talatahari et A. H. Alavi, “Metaheuristic algorithms in modeling and optimization,” *Metaheuristic applications in structures and infrastructures*, vol. 1, p. 1–24, 2013.
- [248] A. Lodi, S. Martello et D. Vigo, “Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems,” *INFORMS journal on computing*, vol. 11, n^o. 4, p. 345–357, 1999.
- [249] S. Desale, A. Rasool, S. Andhale et P. Rane, “Heuristic and meta-heuristic algorithms and their relevance to the real world : a survey,” *Int. J. Comput. Eng. Res. Trends*, vol. 351, n^o. 5, p. 2349–7084, 2015.
- [250] E. Pastore et A. Alfieri, *Integrating Machine Scheduling and Transportation Resource Allocation in a Job Shop : A Simulation Approach*. Cham : Springer International Publishing, 2021, p. 13–26.
- [251] R. K. Phanden, L. K. Saharan et J. A. Erkoyuncu, “Simulation based cuckoo search optimization algorithm for flexible job shop scheduling problem,” dans *Proceedings of the International Conference on Intelligent Science and Technology*, ser. ICIST ’18. New York, NY, USA : Association for Computing Machinery, 2018, p. 50–55. [En ligne]. Disponible : <https://doi.org/10.1145/3233740.3233752>

- [252] F. J. Perez-Grau, J. R. Martinez-de Dios, J. L. Paneque, J. J. Acevedo, A. Torres-González, A. Viguria, J. R. Astorga et A. Ollero, “Introducing autonomous aerial robots in industrial manufacturing,” *Journal of Manufacturing Systems*, vol. 60, p. 312–324, 2021.
- [253] M. Zhong, Y. Yang, Y. Dessouky et O. Postolache, “Multi-agv scheduling for conflict-free path planning in automated container terminals,” *Computers & Industrial Engineering*, vol. 142, p. 106371, 2020.
- [254] H. Bae, K. Gidong, J. Kim, D. Qian et S. Lee, “Multi-robot path planning method using reinforcement learning,” *Applied Sciences*, vol. 9, p. 3057, 07 2019.
- [255] Y. F. Chen, M. Liu, M. Everett et J. P. How, “Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning,” dans *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, p. 285–292.
- [256] A. Salimi Lafmejani et S. Berman, “Nonlinear mpc for collision-free and deadlock-free navigation of multiple nonholonomic mobile robots,” *Robotics and Autonomous Systems*, vol. 141, p. 103774, 2021.
- [257] M. Drótos, P. Györgyi, M. Horváth et T. Kis, “Suboptimal and conflict-free control of a fleet of agvs to serve online requests,” *Computers & Industrial Engineering*, vol. 152, p. 106999, 2021.
- [258] K.-B. Reith, S. Rank et T. Schmidt, “Conflict-minimal routing for free-ranging transportation vehicles in in-house logistics based on an a-priori lane design,” *Journal of Manufacturing Systems*, vol. 61, p. 97–111, 2021.
- [259] J. Zajac et W. Malopolski, “Structural on-line control policy for collision and deadlock resolution in multi-agv systems,” *Journal of Manufacturing Systems*, vol. 60, p. 80–92, 2021.
- [260] E. G. Coffman, M. Elphick et A. Shoshani, “System deadlocks,” *ACM Computing Surveys (CSUR)*, vol. 3, n^o. 2, p. 67–78, 1971.
- [261] M. Chen, Y. Lu et C. Zhang, “Deadlock-solving traffic control methods for automated guided vehicle systems,” dans *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2021, p. 51–57.
- [262] Y. Zhou, H. Hu, Y. Liu et Z. Ding, “Collision and deadlock avoidance in multirobot systems : A distributed approach,” *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, vol. 47, n^o. 7, p. 1712–1726, 2017.

- [263] J. Zajac, “A deadlock handling method for automated manufacturing systems,” *CIRP Annals*, vol. 53, n^o. 1, p. 367–370, 2004. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0007850607607185>
- [264] G. Sebastian, G. Wolfgang et P. Peter, “Agent-based, hybrid control architecture for optimized and flexible production scheduling and control in remanufacturing,” *Journal of Remanufacturing*, vol. 53, 5 2020.
- [265] P. Lacomme, M. Larabi et N. Tchernev, “Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles,” *International Journal of Production Economics*, vol. 143, n^o. 1, p. 24–34, 2013.
- [266] N. Mladenović et P. Hansen, “Variable neighborhood search,” *Computers & operations research*, vol. 24, n^o. 11, p. 1097–1100, 1997.
- [267] J. M. R. et al., “Human-centric artificial intelligence architecture for industry 5.0 applications,” *International Journal of Production Research*, vol. 61, n^o. 20, p. 6847–6872, 2023.
- [268] A. Löcklin, F. Dettinger, M. Artelt, N. Jazdi et M. Weyrich, “Trajectory prediction of workers to improve agv and amr operation based on the manufacturing schedule,” *Procedia CIRP*, vol. 107, p. 283–288, 2022.
- [269] L. M. Hiatt, C. Narber, E. Bekele, S. S. Khemlani et J. G. Trafton, “Human modeling for human–robot collaboration,” *The International Journal of Robotics Research*, vol. 36, n^o. 5-7, p. 580–596, 2017.
- [270] T. Smith, Y. Chen, N. Hewitt, B. Hu et Y. Gu, “Socially aware robot obstacle avoidance considering human intention and preferences,” *International Journal of Social Robotics*, vol. 15, p. pages661–678, 2023.
- [271] H.-I. Lin et W.-K. Chen, “Human intention recognition using markov decision processes,” dans *2014 CACS International Automatic Control Conference (CACS 2014)*, 2014, p. 340–343.
- [272] T. Petković, D. Puljiz, I. Marković et B. Hein, “Human intention estimation based on hidden markov model motion validation for safe flexible robotized warehouses,” *Robotics and Computer-Integrated Manufacturing*, vol. 57, p. 182–196, 2019. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0736584518302965>
- [273] T. Petković, I. Marković et I. Petrović, “Human intention recognition in flexible robotized warehouses based on markov decision processes,” dans *ROBOT 2017 : Third Iberian Robotics*

BIBLIOGRAPHIE

- Conference*, A. Ollero, A. Sanfeliu, L. Montano, N. Lau et C. Cardeira, édit. Cham : Springer International Publishing, 2018, p. 629–640.
- [274] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee et D. Rus, “Intention-aware motion planning,” dans *Algorithmic Foundations of Robotics X*, E. Frazzoli, T. Lozano-Perez, N. Roy et D. Rus, édit. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013, p. 475–491.
- [275] S. Thompson, T. Horiuchi et S. Kagami, “A probabilistic model of human motion and navigation intent for mobile robot path planning,” dans *2009 4th International Conference on Autonomous Robots and Agents*, 2009, p. 663–668.
- [276] F. Ansari, P. Hold et M. Khobreh, “A knowledge-based approach for representing jobholder profile toward optimal human–machine collaboration in cyber physical production systems,” *CIRP Journal of Manufacturing Science and Technology*, vol. 28, p. 87–106, 2020. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1755581719300719>
- [277] M. M. Cummings, “Informing autonomous system design through the lens of skill-, rule-, and knowledge-based behaviors,” *Journal of Cognitive Engineering and Decision Making*, vol. 12, n^o. 1, p. 58–61, 2018.
- [278] C. Breazeal, J. Gray et M. Berlin, “An embodied cognition approach to mindreading skills for socially intelligent robots,” *The International Journal of Robotics Research*, vol. 28, n^o. 5, p. 656–680, 2009.
- [279] M. Urgo, M. Tarabini et T. Tolio, “A human modelling and monitoring approach to support the execution of manufacturing operations,” *CIRP Annals*, vol. 68, n^o. 1, p. 5–8, 2019.
- [280] D. Papanagiotou, G. Senterri et S. Manitsaris, “Egocentric gesture recognition using 3d convolutional neural networks for the spatiotemporal adaptation of collaborative robots,” *Frontiers in neurorobotics*, vol. 15, n^o. 703545, 2021.
- [281] A.-Y. Ali, F. Myles, B. Achim, B. Thomas, F. Pedro, H. Ella-Mae et L. Niels, “Data-driven modelling of human-human co-manipulation using force and muscle surface electromyogram activities,” *Electronics*, vol. 10, n^o. 13, 2021.
- [282] M. Yoichi, S. Satoru, H. Rajibul, G. Dylan, K. Takayuki et H. Norihiro, “How do people walk side-by-side? — using a computational model of human behavior for a social robot,” dans *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012, p. 301–308.

- [283] B. M., B. W. et T. S., “Using em to learn motion behaviors of persons with mobile robots,” dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, p. 502–507 vol.1.
- [284] J.-P. Herrmann, A. Atanasyan, F. Casser et S. Tackenberg, “A petri net architecture for real-time human activity recognition in work systems,” *Procedia Computer Science*, vol. 217, p. 1188–1199, 2023, 4th International Conference on Industry 4.0 and Smart Manufacturing. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1877050922024024>
- [285] L. Gal, R. Michael, R. Ehud et B. Artyom, “Video event modeling and recognition in generalized stochastic petri nets,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, n^o. 1, p. 102–118, 2010.
- [286] J. Qiu, L. Wang, Y. Wang et Y. H. Hu, “Multi-event modeling and recognition using extended petri nets,” *IEEE Access*, vol. 8, p. 37 879–37 890, 2020.
- [287] Z. Xiao, J. Jiang et Z. Ming, “High-level video event modeling, recognition, and reasoning via petri net,” *IEEE Access*, vol. 7, p. 129 376–129 386, 2019.
- [288] M. B. Alatise et G. P. Hancke, “A review on challenges of autonomous mobile robot and sensor fusion methods,” *IEEE Access*, vol. 8, p. 39 830–39 846, 2020.
- [289] R. Singh et M. Mathirajan, “Simulation modelling techniques for managing epidemic outbreak : A review, classification schemes, and meta-analysis,” *Journal of Simulation*, vol. 17, n^o. 6, p. 709–728, 2023.
- [290] K.-H. Bae et H.-J. Ko, “Installation planning for an offshore wind farm : a hybrid modelling framework of integrating simulation and optimisation with a markov chain,” *Journal of Simulation*, vol. 0, n^o. 0, p. 1–18, 2023.
- [291] N. Bouaziz, B. Bettayeb, M. Sahnoun et A. Yassine, “Incorporating uncertain human behavior in production scheduling for enhanced productivity in industry 5.0 context,” *International Journal of Production Economics*, vol. 274, p. 109311, 2024.
- [292] A. R. R Ewald, C Maus et A. Uhrmacher, “Discrete event modelling and simulation in systems biology,” *Journal of Simulation*, vol. 1, n^o. 2, p. 81–96, 2007.
- [293] A. Ghassoul et M. Messaadia, “Analyzing the required skills and competencies in industrial revolution 4.0 and 5.0 : A literature review,” dans *2023 International Conference On Cyber Management And Engineering (CyMaEn)*. IEEE, 2023, p. 39–44.

- [294] L. Martin, J. Gao, A. Wilson, C. H. Le et M. de Ree, *Autonomous Mobile Robots in High Occupancy Aerospace Manufacturing*. Advances in Manufacturing Technology XXXIV, 08 2021.
- [295] Z.-J. M. Shen, J. Xie, Z. Zheng et H. Zhou, “Dynamic scheduling with uncertain job types,” *European Journal of Operational Research*, vol. 309, n^o. 3, p. 1047–1060, 2023.
- [296] Z. Qin, D. Johnson et Y. Lu, “Dynamic production scheduling towards self-organizing mass personalization : A multi-agent dueling deep reinforcement learning approach,” *Journal of Manufacturing Systems*, vol. 68, p. 242–257, 2023.
- [297] Y. Gui, D. Tang, H. Zhu, Y. Zhang et Z. Zhang, “Dynamic scheduling for flexible job shop using a deep reinforcement learning approach,” *Computers & Industrial Engineering*, vol. 180, p. 109255, 2023.
- [298] Y. Du, J. Li, C. Li et P. Duan, “A reinforcement learning approach for flexible job shop scheduling problem with crane transportation and setup times,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, n^o. 4, p. 5695–5709, 2024.
- [299] P. Priore, A. Gómez, R. Pino et R. Rosillo, “Dynamic scheduling of manufacturing systems using machine learning : An updated review,” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 28, n^o. 1, p. 83–97, 2014.
- [300] H. Chakraa, F. Guérin, E. Leclercq et D. Lefebvre, “Optimization techniques for multi-robot task allocation problems : Review on the state-of-the-art,” *Robotics and Autonomous Systems*, vol. 168, p. 104492, 2023.
- [301] C. Ju, J. Kim, J. Seol et H. I. Son, “A review on multirobot systems in agriculture,” *Computers and Electronics in Agriculture*, vol. 202, p. 107336, 2022.
- [302] E. M. Valera, G. Raphael, D. J. Deon, R. Paolo, I. Luca, M. Luca, N. Daniele, M. Dorothy, N. Mircea et K. Christopher, *Multi-robot Teams for Environmental Monitoring*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011, p. 183–209.
- [303] H. G. Tanner, “Switched uav-ugv cooperation scheme for target detection,” dans *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, p. 3457–3462.
- [304] G. P. Das, T. M. McGinnity, S. A. Coleman et L. Behera, “A distributed task allocation algorithm for a multi-robot system in healthcare facilities,” *Journal of Intelligent & Robotic Systems*, vol. 80, p. 33–58, 2015.

- [305] K. Yakovlev., A. Andreychuk., T. Rybecký. et M. Kulich., “On the application of safe-interval path planning to a variant of the pickup and delivery problem,” dans *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics - ICINCO*, INSTICC. SciTePress, 2020, p. 521–528.
- [306] S. G. Marios, S. Hyo-Sang et T. Antonios, “A survey of task allocation techniques in mas,” dans *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, p. 488–497.
- [307] A. Whitbrook, Q. Meng et P. W. H. Chung, “Reliable, distributed scheduling and rescheduling for time-critical, multiagent systems,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, n^o. 2, p. 732–747, 2018.
- [308] K.-S. Kim, H.-Y. Kim et H.-L. Choi, “A bid-based grouping method for communication-efficient decentralized multi-uav task allocation,” *International Journal of Aeronautical and Space Sciences*, vol. 21, p. 290–302, 2020.
- [309] W. L. Yeung, “Efficiency of task allocation based on contract net protocol with audience restriction in a manufacturing control application,” *International Journal of Computer Integrated Manufacturing*, vol. 31, n^o. 10, p. 1005–1017, 2018.
- [310] B. Du et S. Li, “A new multi-satellite autonomous mission allocation and planning method,” *Acta Astronautica*, vol. 163, p. 287–298, 2019, fourth IAA Conference on Dynamics and Control of Space Systems (DYCOSS2018). [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0094576518312475>
- [311] A. Liekna, E. Lavendelis et A. Grabovskis, “Experimental analysis of contract net protocol in multi-robot task allocation,” *Applied Computer Systems*, vol. 13, n^o. 1, p. 6–14, 2012.
- [312] A. Chapman, R. A. Micillo, R. Kota et N. Jennings, “Decentralised dynamic task allocation : A practical game theoretic approach,” dans *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09) (09/05/09 - 14/05/09)*, 2009, p. 915–922. [En ligne]. Disponible : <https://eprints.soton.ac.uk/267066/>
- [313] E. Bakolas et Y. Lee, “Decentralized game-theoretic control for dynamic task allocation problems for multi-agent systems,” dans *2021 American Control Conference (ACC)*, 2021, p. 3228–3233.
- [314] P. Bumjin, K. Cheongwoong et C. Jaesik, “Cooperative multi-robot task allocation with reinforcement learning,” *Applied Sciences*, vol. 12, n^o. 1, 2022. [En ligne]. Disponible : <https://www.mdpi.com/2076-3417/12/1/272>

- [315] A. Aakriti, B. A. Singh et M. Dinesh, “Rtaw : An attention inspired reinforcement learning method for multi-robot task allocation in warehouse environments,” dans *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, p. 1393–1399.
- [316] A. Aakriti, H. Senthil, B. A. Singh et M. Dinesh, “Dc-mrta : Decentralized multi-robot task allocation and navigation in complex environments,” dans *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, p. 11 711–11 718.
- [317] Z. Farouq, M. Ramdane et H. Saad, “Fa-qabc-mrta : a solution for solving the multi-robot task allocation problem,” *Intelligent Service Robotics*, vol. 12, n^o. 4, p. 407–418, 2019.
- [318] S. Kim et H. Lee, “Multi-robot task scheduling with ant colony optimization in antarctic environments,” *Sensors*, vol. 23, n^o. 2, 2023. [En ligne]. Disponible : <https://www.mdpi.com/1424-8220/23/2/751>
- [319] B. Zhou, R. Zhou, Y. Gan, F. Fang et Y. Mao, “Multi-robot multi-station cooperative spot welding task allocation based on stepwise optimization : An industrial case study,” *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102197, 2022. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0736584521000806>
- [320] K. Xiangjun, G. Yunpeng, W. Tianyi, L. Jihong et X. Wenting, “Multi-robot task allocation strategy based on particle swarm optimization and greedy algorithm,” dans *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, p. 1643–1646.
- [321] Z. Farouq, H. Saad et M. Ramdane, “A distributed solution to the multi-robot task allocation problem using ant colony optimization and bat algorithm,” dans *Advances in Machine Learning and Computational Intelligence*. Singapore : Springer Singapore, 2021, p. 477–490.
- [322] W. Dai, H. Lu, J. Xiao, Z. Zeng et Z. Zheng, “Multi-robot dynamic task allocation for exploration and destruction,” *Journal of Intelligent & Robotic Systems*, vol. 98, p. 455–479, 2020.
- [323] Y. Jiang, “A survey of task allocation and load balancing in distributed systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, n^o. 2, p. 585–599, 2016.
- [324] Y. Lu et K. Yan, “Algorithms in multi-agent systems : a holistic perspective from reinforcement learning and game theory,” *arXiv preprint arXiv :2001.06487*, 2020.
- [325] M. J. Osborne et A. Rubinstein, *A course in game theory*. MIT press, 1994.

- [326] J. S. Shamma, “Game theory, learning, and control systems,” *National Science Review*, vol. 7, n^o. 7, p. 1118–1119, 11 2019.
- [327] H. Jaleel et J. S. Shamma, “Distributed optimization for robot networks : From real-time convex optimization to game-theoretic self-organization,” *Proceedings of the IEEE*, vol. 108, n^o. 11, p. 1953–1967, 2020.
- [328] D. Monderer et L. S. Shapley, “Potential games,” *Games and economic behavior*, vol. 14, n^o. 1, p. 124–143, 1996.
- [329] D. Abreu et A. Rubinstein, “The structure of nash equilibrium in repeated games with finite automata,” *Econometrica : Journal of the Econometric Society*, p. 1259–1281, 1988.
- [330] J.-P. Benoit et V. Krishna, “Finitely repeated games,” *Econometrica*, vol. 53, n^o. 4, p. 905–922, 1985.
- [331] A. Chapman, R. A. Micillo, R. Kota et N. Jennings, “Decentralised dynamic task allocation : A practical game-theoretic approach,” dans *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09) (10/05/09 - 15/05/09)*, May 2009, p. 915–922, event Dates : 10-15 May, 2009. [En ligne]. Disponible : <https://eprints.soton.ac.uk/267066/>
- [332] A. Vetta, “Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions,” dans *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, 2002, p. 416–425.
- [333] R. Gopalakrishnan, J. R. Marden et A. Wierman, “Potential games are necessary to ensure pure nash equilibria in cost sharing games,” *Mathematics of Operations Research*, vol. 39, n^o. 4, p. 1252–1296, 2014. [En ligne]. Disponible : <http://www.jstor.org/stable/24541009>
- [334] A. Schrijver, *Combinatorial optimization : polyhedra and efficiency*. Springer, 2003.
- [335] L. Lovász, *Submodular functions and convexity*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1983, p. 235–257.
- [336] J. Edmonds, *Submodular Functions, Matroids, and Certain Polyhedra*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2003, p. 11–26.

BIBLIOGRAPHIE

Annexe A

Les différents ensemble de jobs de l'atelier

Ensemble de jobs n°1

Job 1 : M1(8) ; M2(16) ; M4(12)
 Job 2 : M1(20) ; M3(10) ; M2(18)
 Job 3 : M3(12) ; M4(8) ; M1(15)
 Job 4 : M4(14) ; M2(18)
 Job 5 : M3(10) ; M1(15)

Ensemble de jobs n°3

Job 1 : M1(16) ; M3(15)
 Job 2 : M2(18) ; M4(15)
 Job 3 : M1(20) ; M2(10)
 Job 4 : M3(15) ; M4(10)
 Job 5 : M1(8) ; M2(10) ; M3(15) ; M4(17)
 Job 6 : M2(10) ; M3(15) ; M4(8) ; M1(15)

Ensemble de jobs n°5

Job 1 : M1(6) ; M2(12) ; M4(9)
 Job 2 : M1(18) ; M3(6) ; M2(15)
 Job 3 : M3(9) ; M4(3) ; M1(12)
 Job 4 : M4(6) ; M2(15)
 Job 5 : M3(3) ; M1(9)

Ensemble de jobs n°7

Job 1 : M1(6) ; M4(6)
 Job 2 : M2(11) ; M4(9)
 Job 3 : M2(9) ; M4(7)
 Job 4 : M3(16) ; M4(7)
 Job 5 : M1(9) ; M3(18)
 Job 6 : M2(13) ; M3(19) ; M4(6)
 Job 7 : M1(10) ; M2(9) ; M3(13)
 Job 8 : M1(11) ; M2(9) ; M4(8)

Ensemble de jobs n°9

Job 1 : M3(9) ; M1(12) ; M2(9) ; M4(6)
 Job 2 : M3(16) ; M2(11) ; M4(9)
 Job 3 : M1(21) ; M2(18) ; M4(7)
 Job 4 : M2(20) ; M3(22) ; M4(11)
 Job 5 : M3(14) ; M1(16) ; M2(13) ; M4(9)

Ensemble de jobs n°2

Job 1 : M1(10) ; M4(18)
 Job 2 : M2(10) ; M4(18)
 Job 3 : M1(10) ; M3(20)
 Job 4 : M2(10) ; M3(15) ; M4(12)
 Job 5 : M1(10) ; M2(15) ; M4(12)
 Job 6 : M1(10) ; M2(15) ; M3(12)

Ensemble de jobs n°4

Job 1 : M4(11) ; M1(10) ; M2(7)
 Job 2 : M3(12) ; M2(10) ; M4(8)
 Job 3 : M2(7) ; M3(10) ; M1(9) ; M3(8)
 Job 4 : M2(7) ; M4(8) ; M1(12) ; M2(6)
 Job 5 : M1(9) ; M2(7) ; M4(8) ; M2(10) ; M3(8)

Ensemble de jobs n°6

Job 1 : M1(9) ; M2(11) ; M4(7)
 Job 2 : M1(19) ; M2(20) ; M4(13)
 Job 3 : M2(14) ; M3(20) ; M4(9)
 Job 4 : M2(14) ; M3(20) ; M4(9)
 Job 5 : M1(11) ; M3(16) ; M4(8)
 Job 6 : M1(10) ; M3(12) ; M4(10)

Ensemble de jobs n°8

Job 1 : M2(12) ; M3(21) ; M4(11)
 Job 2 : M2(12) ; M3(21) ; M4(11)
 Job 3 : M2(12) ; M3(21) ; M4(11)
 Job 4 : M2(12) ; M3(21) ; M4(11)
 Job 5 : M1(10) ; M2(14) ; M3(18) ; M4(9)
 Job 6 : M1(10) ; M2(14) ; M3(18) ; M4(9)

Ensemble de jobs n°10

Job 1 : M1(11) ; M3(19) ; M2(16) ; M4(13)
 Job 2 : M2(21) ; M3(16) ; M4(14)
 Job 3 : M3(8) ; M2(10) ; M1(14) ; M4(9)
 Job 4 : M2(13) ; M3(20) ; M4(10)
 Job 5 : M1(9) ; M3(16) ; M4(18)
 Job 6 : M2(19) ; M1(21) ; M3(11) ; M4(15)

TABLE A.1 – Les ensembles de job proposés par Ü. Bilge et G. Ulusoy [1]

Annexe B

Les algorithmes du programme principal de la simulation

Algorithme 4 Pseudo-code de la procédure *go*

Requiert : les jobs, machines, stocks et robots disponibles

```
1: procédure GO :  
2:   TRANSPORTER-JOBS  
3:   TRAITER-JOBS  
4:   Si tous les jobs ont été traités alors  
5:     afficher makespan                                ▷ ou toute autre information pertinente  
6:     arrêter la simulation  
7:   Fin si  
8: Fin procédure
```

Algorithme 5 Pseudo-code de l'algorithme de transport

Requiert : tous les transporteurs disponibles

```

1: procédure TRANSPORTER-JOBS :
2:   Pour chaque transporteur  $t_p$  disponible, faire :
3:     Si  $job\_found? = faux$  alors
4:       Si  $simulation\_type = "advanced"$  alors
5:         Si  $orders\_list = []$  alors ▷ une liste vide
6:            $orders\_list \leftarrow$  RECUPERER_TACHES_DE_TRANSPORT( $schedule,$ 
7:  $t_p$ )
8:           Sinon
9:             Si  $current\_order = []$  alors
10:               $current\_order \leftarrow orders\_list[0]$  ▷ accomplir la première tâche de la liste
11:            Fin si
12:          Fin si
13:        Sinon
14:          Si  $current\_order = []$  alors
15:             $current\_order \leftarrow$  TROUVER_PROCHAINE_TACHE( $t_p$ ) ▷ lancer
16:            l'algorithme d'ordonnancement dynamique
17:          Fin si
18:        Fin si
19:      Si  $current\_order = []$  alors
20:         $job\_found? \leftarrow faux$ 
21:      Sinon
22:         $job\_found? \leftarrow vrai$ 
23:      PLANIFIER_TRAJET( $t_p$ )
24:    Fin si
25:  Fin pour
26: Fin procédure

```

Algorithme 6 Pseudo-code de l'algorithme de planification de trajet

Requiert : le transporteur t_p

```

1: procédure PLANIFIER_TRAJET( $t_p$ ) :
2:   analyser  $current\_order$  ▷ localiser le job et sa destination
3:   Si  $job\_picked? = faux$  alors
4:     RECUPERER_JOBS( $t_p, j_i$ )
5:   Sinon
6:     LIVRER_JOBS( $t_p, j_i, d$ )
7:   Fin si
8: Fin procédure

```

Algorithme 7 Pseudo-code de l'algorithme de récupération de jobs

Requiert : le transporteur t_p et le job j_i

```
1: procédure RECUPERER_JOB( $t_p, j_i$ ) :
2:   Si  $t_p.location = j_i.location$  alors                                ▷ le transporteur a retrouvé le job
3:     Si  $j_i.status = \text{“en traitement”}$  alors
4:       patienter jusqu'à la fin du traitement
5:     Sinon
6:       prendre  $j_i$ 
7:        $job\_picked? \leftarrow vrai$ 
8:     Fin si
9:   Sinon
10:     $job\_picked? \leftarrow faux$ 
11:    Si  $without\_collisions? = vrai$  alors
12:      EVITER_COLLISIONS( $t_p$ )
13:    Sinon
14:      DEPLACEMENT( $t_p$ )
15:    Fin si
16:  Fin si
17: Fin procédure
```

Algorithme 8 Pseudo-code de l'algorithme de livraison de jobs

Requiert : le transporteur t_p , le job j_i et la destination d

```
1: procédure LIVRER_JOBS( $t_p, j_i, d$ ) :
2:   Si  $t_p.location = d$  alors                                          ▷ destination atteinte
3:     deposer  $j_i$ 
4:      $job\_picked? \leftarrow faux$ 
5:      $current\_order \leftarrow []$ 
6:      $job\_found? \leftarrow faux$ 
7:   Sinon
8:     Si  $without\_collisions? = vrai$  alors
9:       EVITER_COLLISIONS( $t_p$ )
10:    Sinon
11:      DEPLACEMENT( $t_p$ )
12:    Fin si
13:     $job\_picked? \leftarrow vrai$ 
14:  Fin si
15: Fin procédure
```

Algorithme 9 Pseudo-code de l'algorithme de déplacement des transporteurs

Requiert : le transporteur t_p

- 1: **procédure** DEPLACEMENT(t_p) :
 - 2: **Si** $t_p.patch_ahead.color = \text{“blanc”}$ **alors** \triangleright le patch devant le transporteur est blanc
 - 3: **avancer**
 - 4: **Sinon**
 - 5: **s’orienter** dans le sens du trajet
 - 6: **Fin si**
 - 7: **Fin procédure**
-

Algorithme 10 Pseudo-code de l'algorithme d'évitement des collisions

Requiert : n'importe quel transporteur t_p de la flotte

```

1: procédure EVITER_COLLISIONS( $t_p$ ) :
2:   Si  $robot = "agv"$  alors                                     ▷ transport réalisé par des AGV
3:     Si  $t_p.location = "checkpoint"$  alors
4:       VERIFIER_ET_RESERVER_CHEMIN( $t_p$ )
5:     Sinon
6:       Si un autre agv  $t_q$  pénètre dans le rayon de détection alors
7:         réduire la vitesse de moitié
8:         Si  $t_q$  pénètre dans le champ de vision alors
9:           arrêter la marche
10:        Sinon
11:          DEPLACEMENT( $t_p$ )
12:        Fin si
13:      Sinon
14:        DEPLACEMENT( $t_p$ )
15:      Fin si
16:    Fin si
17:  Sinon                                                         ▷ transport réalisé par des AIV
18:    Si un autre aiv  $t_q$  pénètre dans le rayon de détection alors
19:      réduire la vitesse de moitié
20:      DEPLACEMENT( $t_p$ )
21:    Si  $t_q$  pénètre dans le champ de vision alors
22:      détection le champ de vision de  $t_q$ 
23:      Si  $detection = vrai$  alors                                 ▷ ils sont face-à-face
24:        lancer le système de priorité
25:        Si  $priorite = vrai$  alors
26:          réduire la vitesse à 0,1m/s
27:          contourner  $t_q$ 
28:        Sinon
29:          arrêter la marche
30:        Fin si
31:      Sinon
32:        arrêter la marche
33:      Fin si
34:    Fin si
35:  Sinon
36:    DEPLACEMENT( $t_p$ )
37:  Fin si
38: Fin si
39: Fin procédure

```

Algorithme 11 Pseudo-code de l'algorithme de réservation de portions de chemin

Requiert : n'importe quel AGV au sein de la flotte

- 1: **procédure** VERIFIER_ET_RESERVER_CHEMIN(agv_0) :
 - 2: **soit** $path$ la portion de chemin suivante
 - 3: **vérifier** l'état de $path$
 - 4: **Si** $path$ est réservé **alors**
 - 5: **arrêter** la marche
 - 6: **Sinon**
 - 7: **réserver** $path$
 - 8: DEPLACEMENT(agv_0)
 - 9: **Fin si**
 - 10: **Fin procédure**
-

Algorithme 12 Pseudo-code de l'algorithme de traitement de jobs

Requiert : toutes les machines disponibles

```
1: procédure TRAITER-JOBS :
2:   Pour chaque machine  $m_k$ , faire :
3:     Si  $m_k.input\_buffer \neq []$  alors ▷ le buffer d'entrée de la machine n'est pas vide
4:       Si  $m_k.status = \text{"libre"}$  alors
5:          $m_k.working\_time \leftarrow 0$ 
6:         Si  $simulation - type = \text{"advanced"}$  alors
7:           Si  $m_k.processing\_list[0] \in m_k.input\_stock$  alors ▷ traiter les jobs
           conformément à l'ordonnancement
8:             soit  $j_i = m_k.processing\_list[0]$ 
9:             prendre  $j_i$  dans le buffer d'entrée
10:            traiter  $j_i$ 
11:             $m_k.status \leftarrow \text{"occupe"}$ 
12:             $j_i.status \leftarrow \text{"en traitement"}$ 
13:          Sinon
14:             $m_k.status \leftarrow \text{"libre"}$ 
15:          Fin si
16:        Sinon
17:          soit  $j_i = m_k.input\_stock[0]$  ▷ traiter les jobs en mode FIFO
18:          prendre  $j_i$  dans le buffer d'entrée
19:          traiter  $j_i$ 
20:           $m_k.status \leftarrow \text{"occupe"}$ 
21:           $j_i.status \leftarrow \text{"en traitement"}$ 
22:        Fin si
23:      Fin si
24:    Fin si
25:    Si  $m_k.status = \text{"occupe"}$  alors
26:       $m_k.working\_time \leftarrow m_k.working\_time + 1$ 
27:      Si  $m_k.working\_time = j_i.processing\_time$  alors ▷ le traitement est terminé
28:        deposer  $j_i$  dans  $m_k.output\_stock$ 
29:         $m_k.status \leftarrow \text{"libre"}$ 
30:         $j_i.status \leftarrow \text{"disponible"}$ 
31:      Fin si
32:    Fin si
33:  Fin pour
34: Fin procédure
```

Résumé : Cette thèse s'intéresse à l'optimisation de l'ordonnancement des tâches dans des environnements industriels complexes où humains et robots travaillent en collaboration. Les principaux défis à relever sont l'évitement des collisions, la gestion des blocages et la minimisation des temps de production. Elle développe une approche par simulation, utilisant des systèmes multi-agents et les chaînes de Markov, pour modéliser les interactions complexes entre les acteurs de l'atelier. Cela permet d'évaluer les performances des robots mobiles et d'explorer les avantages des Véhicules Autonomes Intelligents (AIV) par rapport aux Véhicules Autonomes Guidés (AGV). Enfin, un algorithme de d'ordonnancement dynamique, basé sur la théorie des jeux, est développé pour optimiser la gestion des tâches en temps réel, en tenant compte de la présence d'opérateurs humains. L'objectif est de créer des ordonnancements efficaces qui garantissent une production résiliente.

Mots clés : Industrie 5.0, JSSPT, ordonnancement dynamique, simulation, système multi-agents, chaîne de Markov, théorie des jeux, AIV, AGV, humains.

Abstract : This research focuses on optimizing task scheduling in human-robot collaborative industrial environments. Key challenges include collision avoidance, deadlock management, and minimizing production time. A simulation-based approach, using multi-agent systems and Markov chains, models complex interactions within the workshop. This allows for evaluation of mobile robot performance and exploration of the benefits of Autonomous Intelligent Vehicles (AIV) over Automated Guided Vehicles (AGV). Finally, a dynamic scheduling algorithm, based on game theory, is developed to optimize task allocation in real-time, considering human workers presence. The goal is to create responsive and efficient schedules that ensure resilient production.

Keywords : Industry 5.0, JSSPT, dynamic scheduling, simulation, multi-agent systems, Markov chain, game theory, AIV, AGV, humans.