



HAL
open science

Supervision distribuée d'un système de production flexible utilisant des robots mobiles collaboratifs pour la tâche de transport

Souleymane Moussa Goumeyer

► To cite this version:

Souleymane Moussa Goumeyer. Supervision distribuée d'un système de production flexible utilisant des robots mobiles collaboratifs pour la tâche de transport. Robotique [cs.RO]. Normandie Université, 2020. Français. NNT : 2020NORMIR19 . tel-03252956

HAL Id: tel-03252956

<https://theses.hal.science/tel-03252956v1>

Submitted on 8 Jun 2021

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.



Normandie Université

THESE

Pour obtenir le diplôme de doctorat

Spécialité Informatique

Préparée au sein de l'INSA de Rouen

Supervision distribuée d'un système de production flexible utilisant des robots mobiles collaboratifs pour la tâche de transport

**Présentée et soutenue par
Souleymane MOUSSA GOUMEYE**

**Thèse soutenue publiquement le 12 novembre 2020
devant le jury composé de**

Pr. Claude YUGMA	Professeur à l'école des Mines de Saint-Etienne	Rapporteur
Dr. Ahmed NAIT-SIDI-MOH	Maître de conférences, HDR à l'Université de Picardie Jules Verne	Rapporteur
Pr. Samia BOUCHAFA	Professeure à l'Université d'Evry-Val -d'Essonne	Examineur
Pr. Abderraouf BENALI	Professeur à l'Université de Versailles Saint-Quentin	Examineur
Pr. Abdelaziz BENSRAIR	Professeur à l'INSA de Rouen	Directeur de thèse
Dr. Fabrice DUVAL	Enseignant-chercheur, HDR à CESI de Rouen	Codirecteur de thèse
Dr. Sahnoun M'hammed	Enseignant-chercheur, HDR à CESI de Rouen	Encadrant de thèse

Thèse dirigée par Abdelaziz BENSRAIR, laboratoire LITIS et Fabrice DUVAL, laboratoire LINEACT

INSA INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
ROUEN



litis

LINEACT cesi

Table des matières

Remerciements	xvii
Résumé	1
Abstract	3
1 Introduction générale	5
1.1 Introduction	5
1.2 Contexte	5
1.2.1 Contexte socio-économique	5
1.2.2 Contexte industriel	6
1.2.3 Projet CoRoT	7
1.3 Problématique scientifique	8
1.4 Contributions de la thèse	9
1.5 Organisation du manuscrit	10
2 Revue littéraire	13
2.1 Introduction	13
2.2 Les systèmes de production flexibles	13
2.2.1 Types de flexibilité	14
2.2.1.1 Flexibilité des outils de manutention	14
2.2.1.2 Flexibilité de routage	14
2.2.1.3 Flexibilité des machines	15
2.2.1.4 Flexibilité des opérations	15
2.2.1.5 Flexibilité du produit	16
2.2.2 Niveaux de décision dans les FMS	16
2.3 Industrie 4.0	18
2.3.1 Nouvelle révolution industrielle	18
2.3.2 Principes et outils de l'I4.0	18
2.4 Architectures de contrôle	19
2.4.1 Nuance entre Contrôle et Supervision	19
2.4.2 Types d'architectures de contrôle	20
2.4.2.1 Centralisée	21
2.4.2.2 Hiérarchique	22
2.4.2.3 Hétérarchique	22
2.4.2.4 Hybride	23
2.4.3 Architectures de supervision notables	23
2.4.4 Synthèse sur les architectures de supervision	25
2.5 Ordonnancement de tâches	26
2.5.1 Types d'ordonnancement	28
2.5.2 Méthodes d'ordonnancement de tâches	29

2.5.2.1	Champs potentiels	29
2.5.2.2	Contract Net	31
2.5.3	Problèmes de Job-Shop avec et sans flexibilité de routage	34
2.5.4	Ordonnancement des tâches de transport	34
2.5.4.1	Tâche de transport	34
2.5.4.2	Ordonnancement des tâches de transport indépendamment des tâches de production	35
2.5.4.3	Ordonnancement tâches de transport et de production	37
2.5.5	Mécanisme d'ordonnancement distribué	38
2.6	Entités autonomes de transport	40
2.6.1	Les véhicules à guidage automatique	40
2.6.2	Systèmes de navigation	40
2.6.2.1	Navigation semi-autonome	41
2.6.2.2	Navigation autonome	41
2.6.3	Les bras manipulateurs	42
2.6.4	Les manipulateurs mobiles	43
2.6.4.1	Modularité des manipulateurs mobiles	43
2.6.5	Les robots collaboratifs	45
2.6.5.1	Chronologie	45
2.6.5.2	Atouts et différences avec la robotique industrielle classique	46
2.6.6	Les opérateurs humains	46
2.6.6.1	Exemple d'action nécessitant une intervention humaine : Démonstrateur du futur	47
2.6.6.2	Moyens de communication des opérateurs	49
2.7	Coalitions	50
2.7.1	Coopération	50
2.7.1.1	Différence entre coopération et collaboration	50
2.7.2	Coalitions robotiques	51
2.7.3	Coalitions humains robots	53
2.8	Conclusion	54
3	Architecture générique de supervision	57
3.1	Introduction	57
3.2	Architecture proposée	57
3.2.1	Évolution des architectures	57
3.2.2	Architecture des agents dans DIMS	58
3.2.2.1	Unité de Contrôle	58
3.2.2.2	Environnement commun	59
3.2.3	Application de DIMS à la robotique	59
3.2.3.1	Module d'interaction	60
3.2.3.2	Base de données	61
3.2.3.3	Module de décision	61
3.2.3.4	Unité de surveillance	61
3.2.3.5	Module de simulation	62
3.2.3.6	Gestionnaire d'agent	62
3.2.3.7	Contrôleur	62
3.2.3.8	Module de stockage de données	62

	v
3.2.4	Informations échangées entre les modules 62
3.3	Ordonnancement de tâches basé sur DIMS 64
3.3.1	Particularité de l'approche 64
3.3.2	Limites de l'approche d'assignation de DIMS 65
3.3.3	Amélioration de la méthode 65
3.3.3.1	Notations 65
3.3.3.2	Mécanisme d'enchères et d'optimisation 66
3.4	Cas d'étude 71
3.4.1	Description du cas d'étude 71
3.4.2	Résultats 72
3.5	Conclusion 74
4	Ordonnancement de tâches par formation de coalitions 75
4.1	Introduction 75
4.2	Les tâches industrielles 75
4.2.1	Ordre 76
4.2.2	Job 76
4.2.3	Tâches 77
4.2.4	Opérations 77
4.3	Formalisme général de l'assignation des tâches 77
4.4	Approche IT2P 77
4.4.1	Intérêt pour notre étude 78
4.4.2	Différence entre les deux approches 79
4.4.3	Mécanisme d'ordonnancement de l'approche IT2P 79
4.4.3.1	Notations 79
4.4.3.2	Algorithme d'allocation 80
4.5	Approche DISCOFT 83
4.5.1	Caractéristiques générales et originalité 83
4.5.2	Principe d'assignation avec DISCOFT 84
4.5.3	Les opérations possibles d'une tâche de transport 84
4.5.3.1	Opérations secondaires au point de chargement 85
4.5.3.2	Opération secondaire au point de déchargement 86
4.5.3.3	Choix effectué pour notre modèle 86
4.5.4	Conditions d'appartenance aux coalitions 87
4.5.5	Notations 90
4.5.6	Algorithme de Décision Locale : ADL 91
4.5.6.1	Complexité du problème 91
4.5.6.2	Hypothèses 93
4.5.6.3	Fonction objectif 94
4.5.6.4	Contraintes 95
4.5.6.5	Soumission d'enchère 96
4.5.7	Algorithme de Décision Globale : ADG 96
4.5.7.1	Ajustement des Buffers des gagnants 97
4.6	Conclusion 99

5 Cas d'étude : Démonstrateur du futur	101
5.1 Introduction	101
5.2 Choix du solveur	101
5.3 Description du cas d'étude	103
5.3.1 Ressources de production	104
5.3.1.1 Poste automatique	104
5.3.1.2 Postes manuels	104
5.3.1.3 Poste de stockage	105
5.3.1.4 Tâches de production	105
5.3.2 Produits	107
5.3.3 Ressources de transport	107
5.3.3.1 Robots et opérateurs	108
5.3.3.2 Ressource virtuelle	108
5.3.3.3 Aspect sécuritaire	108
5.4 Expériences et Discussion	110
5.4.1 Expérience 1	110
5.4.1.1 Les enchères et la sélection des gagnants	110
5.4.1.2 Longueur des coalitions	113
5.4.1.3 Activité des ressources	114
5.4.1.4 Tâches de production et de transport	119
5.4.2 Expérience 2	119
5.5 Conclusion	124
Conclusion générale	125
Productions scientifiques	129
A Implémentation réelle de DIMS sur un atelier robotique	131
A.1 Modélisation du cas d'étude	131
A.2 Enregistrement des coordonnées cartésiennes via ROS	133
A.2.1 Installation de Moveit!	133
A.3 Enregistrement des points sur la maquette	136
A.4 Changement de repère pour l'UR5	138
A.5 Communication entre entités	138
A.5.1 ROS multi-Machine	139
B Modèle de l'approche de plans de transport et de production intégrés : IT2P	141
B.1 Modèle mathématique de l'approche IT2P	141
B.1.1 Notations	141
B.1.2 Fonction objectif et contraintes	142
B.2 Codage du chromosome	142

Table des figures

1.1	Toupie à main	6
1.2	Produits convoités au début de l'année 2020, de gauche à droite : Gel hydro-alcoolique, Masque chirurgical	6
1.3	Solution du projet CoRoT	8
1.4	Organigramme général du manuscrit de thèse	12
2.1	Illustration flexibilité de routage : Spécification des différents chemins possibles en pointillés pour la fabrication d'un produit devant passé par une opération de fraisage puis de lavage	15
2.2	Flexibilité d'une machine pouvant fabriquer trois types de produits	16
2.3	Illustration de la flexibilité des produits	16
2.4	Les trois niveaux de décisions	17
2.5	Les quatre révolutions industrielles ¹	18
2.6	Modules de la supervision	21
2.7	Types d'architecture selon Dilts et al. [39]	22
2.8	Classes d'architecture selon Trentesaux [170]	22
2.9	Dilemme entre niveaux de flexibilité et contrôle	27
2.10	Variation du champs potentiel par rapport à la distance	31
2.11	Image ² illustrative d'une vente aux enchères	32
2.12	Mécanisme d'enchères	33
2.13	Opérations d'une tâche de transport	35
2.14	Exemple d'AGV (Mir100) adapté et utilisé pour le transport de tricycle	40
2.15	AGV Robotnik	42
2.16	Trois bras manipulateurs Niryo One du laboratoire Lineact- Cesi de Rouen	43
2.17	Manipulateur mobile modulable	44
2.18	Robots collaboratifs : 1 CR35ia de FANUC ³ , 2 KMR iiwa de KUKA ⁴ et 3 YuMi de ABB ⁵	45
2.19	Robots collaboratifs travaillant à proximité d'un humain au la- boratoire Lineact-CESI de Rouen	47
2.20	Opérateur utilisant la RA pour des tâches de maintenance à l'aide d'une lunette (Image de gauche) et d'une tablette (Image de droite)	47
2.21	Processus de chargement des produits finis du démonstrateur du futur au Laboratoire LINEACT-CESI	48
2.22	Quelques outils d'interaction, de gauche à droite : Un Joys- tick ⁶ connecté en Bluetooth, un opérateur communiquant grâce un ordinateur, une tablette connectée en Wifi et un opérateur avec une tablette reliée à sa main, connectée en Wifi	49

2.23	Exemple de formation de coalitions pour la réalisation de tâches complexes	52
3.1	Architecture DIMS proposée par Zhang	60
3.2	Adaptation de DIMS	61
3.3	Mécanisme global d'ordonnancement	67
3.4	Mini-atelier de production	72
3.5	Sequences des produits	72
3.6	Diagramme de gantt de l'activité des trois robots	73
3.7	Évolution des coûts et temps totaux par génération	73
4.1	Organisation hiérarchique des activités de production	76
4.2	Diagramme global de l'assignation	78
4.3	Positionnement de IT2P par rapport aux autres approches	82
4.4	Mécanisme d'ordonnancement des tâches	84
4.5	Modèle éclaté d'une tâche de transport	85
4.6	Modèle compact d'une tâche de transport	87
4.7	Structure de l'approche DISCOFT	97
5.1	Positionnement des solveurs utilisés	102
5.2	Instanciation des sovs sous Visual studio 2019	103
5.3	Vue d'ensemble du cas d'étude	103
5.4	La chaîne automatisée et ses modules	105
5.5	Poste manuel	106
5.6	Partie inférieure boîte	107
5.7	Partie supérieure boîte	107
5.8	Boîte : Produit 2	107
5.9	Partie inférieure téléphone	107
5.10	Composants internes téléphone	107
5.11	Téléphone emboîté	107
5.12	Manipulateur mobile : Mir100+UR5	109
5.13	Bras manipulateur : UR10	109
5.14	Vision 3D de l'atelier	109
5.15	Enchères reçus pour l'assemblage de deux produits	110
5.16	Exemple graphique de formation de coalition	111
5.17	Diagramme gantt des coalitions gagnantes	114
5.18	Nombre de ressources par tâche	115
5.19	Activités avec toutes les ressources	116
5.20	Variation des POG au test 1	117
5.21	Variation des PTO au test 1	117
5.22	Variation des POG au test 2	117
5.23	Variation des PTO au test 2	117
5.24	Variation des POG au test 3	118
5.25	Variation des PTO au test 3	118
5.26	Comparaison entre DISCOFT et IT2P	120
5.27	Comparaison incluant le max des temps de chargement et déchargement	122
5.28	Variation des temps de livraison du Job_0 sans temps de chargement et déchargement	123

5.29	Variation des temps de livraison du Job_1 sans temps de chargement et déchargement	123
5.30	Variation des temps de livraison du Job_0	123
5.31	Variation des temps de livraison du Job_1	123
A.1	Enveloppes de travail des robots	131
A.2	Modélisation atelier sous Rviz	132
A.3	Application de l'étude ergonomique sur l'atelier réel	133
A.4	Points à enregistrer sur la maquette	137
A.5	Architecture implémentation via ROS	140
B.1	Exemple de chromosome	143

Liste des tableaux

2.1	Récapitulatifs sur les caractéristiques des architectures de contrôle	26
2.2	Approches d'ordonnement	30
4.2	Opérations secondaires possibles avant le chargement	85
4.3	Critères des coalitions	88
5.1	Tâches effectuées dans l'atelier	105
5.2	Temps de processus en seconde des postes par type de tâche	107
5.3	Séquence des produits assemblés	107
5.4	Caractéristiques des ressources de transport.	110
5.5	Récapitulatifs de l'assignation des tâches de transport de deux produits.	113
5.6	Synthèse des tests effectués	119
A.1	Coordonnées de M en mètre dans les deux repères	138

Liste des abréviations

AARIA	Autonomous Agents pour Rock Island Arsenal
ABAS	Actor-Based Assembly Systems
ADACOR	ADApTive holonic COntrol aRchitecture
ADS2	Anytime Distributed Supervision of Distributed Systems
AGV	Automatic Guided Vehicle
ARTI	Activity-Resource-Type-Instance
BMS	Bionic Manufacturing Systems
CNET	Contract NET
CPS	Systèmes Cyber Physiques
DIMS	Dynamically Integrated Manufacturing System
FJSP	Flexible Job Shop Scheduling Problem
FrMS	FRactal Manufacturing Systems
GPS	Global Positioning System
HMS	Holonic Manufacturing Systems
IHM	Interface Homme Machine
IHM	Interfaces Homme-Machine
ILP	Integer Linear Programming
IoT	Internet des Objets
JSP	Job-Shop Scheduling Problem
KB	Knowledge Base
kg	Kilogramme
MES	Manufacturing Execution System
MILP	Mixed-Integer Linear Programming
m	mètre
NASA	National Aeronautics and Space Administration

ORCA	Optimized and Reactive Control Architecture
PAC	Production Activity Control
PD	Product Driven
plc	Présence lieu chargement
PME	Petite et Moyenne Entreprise
PN	Petri Net
PROSA	Product Resource Order Staff Architecture
PROSIS	Product, Resource, Order and Simulation Isoarchic Structure
RA	Réalité augmentée
RSA	Robotic Supervision Architecture
SCADA	System Control And Data Acquisition
SFP	Système de Production Flexible
SHFMS	Semi-Heterarchique Flexible Manufacturing Systems
Slam	Simultaneous Localization And Mapping

Je dédie ce travail à :

*Mon très cher ami et grand père sa majesté **Zakou***

***Chaïbou**, parti en août 2020*

*Mon ami, le capitaine **Zakaria Habiboulaye** tombé*

lors de l'attaque terroriste contre le détachement

militaire d'Inatès en décembre 2019

Mon père, ma mère ainsi qu'à tous mes proches

...

Remerciements

Hormis un acharnement personnel, la réussite de ce travail de thèse a été rendue possible grâce à une équipe d'encadrement motivée. Mes vifs remerciements vont à leur égard. Notamment au Dr. **Sahnoun M'hammed**, grâce à qui j'ai pu décrocher cette thèse. Son encadrement par le suivi régulier de mes travaux, sa disponibilité et sa minutie scientifique m'ont permis de développer une rigueur relative aux travaux de recherche. Dr. **Duval Fabrice** dans la co-direction de cette thèse n'a pas hésité à partager tout le savoir technique, nécessaire à la bonne mise en application des théories développées et à la résolution des problèmes rencontrés. La direction de la thèse est partagée avec Pr. **Benshair Abdelaziz**, qui a su me faire profiter de sa longue expérience dans le domaine, bien orienter et conseiller.

Par ailleurs, j'adresse ma reconnaissance à l'ensemble des membres du jury qui ont bien voulu évaluer ce travail de recherche, notamment aux rapporteurs. Leurs appréciations du travail effectué, fut non seulement encourageant mais aussi d'une grande utilité pour ce qui est des points à améliorer.

Le soutien sans cesse des parents et des amis a été déterminant dans l'accomplissement de ce travail de trois ans. Je porte une mention particulière à mon père qui m'a motivé et rassuré sur la nécessité de faire un doctorat. L'encouragement et les prières quasi-quotidiens de ma mère à mon attention, ont apporté inspiration et chance dans ma quête.

L'environnement de travail fut d'un appui sans équivoque durant ces années de thèse. Ainsi je remercie chaleureusement, tous les membres du laboratoire LINEACT de Rouen, qui m'ont accompagné et soutenu jusqu'au bout. Cette proximité a permis d'effectuer la thèse dans de bonnes conditions humaines et professionnelles. J'adresse une reconnaissance particulière au Dr. **Bettayeb Belgacem** qui fut comme un second encadrant pour moi ; Au Dr. **Louis Anne**, directrice de recherche LINEACT, pour sa disponibilité et au Pr. **Belhacène Mazari** notre directeur national, pour ses judicieux conseils et son suivi des travaux.

Je remercie aussi tout le personnel du laboratoire Litis et de l'INSA de Rouen, particulièrement Mme. **Diarra Brigitte** pour son implication et sa bonne gestion des démarches administratives.

J'exprime ma gratitude au Pr. **Benali Abderraouf** et au Dr. **Kotowicz Jean-philippe**, membres de mon comité de suivi de thèse. Grâce, à nos rencontres annuelles et à leur disponibilité, ils ont apporté une vision extérieure, tout en s'assurant de la bonne marche de la thèse.

Ce travail de thèse a été rendu possible grâce au soutien financier de 2.6 millions d'euros du Fonds européen de développement régional, fourni par le programme Interreg V France Manche Angleterre dans le cadre du projet CoRoT. Mes sincères remerciements vont également à l'ensemble des collaborateurs académiques et industriels du projet CoRoT.

Résumé

Les manipulateurs mobiles composés d'une base mobile (AGV : Véhicule à guidage automatique) et d'un bras manipulateur ont été développés et introduits dans les ateliers de production flexibles (FMS) pour assurer l'exécution complète et autonome des tâches de transport. La supervision de ces entités intelligentes, leur architecture de contrôle, leur reconfigurabilité ainsi que l'ordonnancement de leurs tâches représentent des problèmes scientifiques et technologiques de taille ayant attiré plusieurs chercheurs.

Par contre, peu de travaux ont étudié ces problématiques avec des systèmes de transport hétérogènes (constitués de plusieurs types de ressources : Manipulateurs mobiles, AGVs et bras manipulateurs) et/ou modulaires.

Ce travail de thèse a permis de mettre en oeuvre, de tester et de valider une architecture de supervision distribuée incluant une approche d'ordonnancement de tâches. Cette approche considère à la fois la modularité et l'intelligence des manipulateurs mobiles, de même que l'hétérogénéité des systèmes de transport. La méthode d'ordonnancement des tâches, se fonde sur un mécanisme hybride combinant un algorithme de décision globale basé sur les enchères et un modèle de décision locale basé sur une programmation linéaire en nombres entiers (ILP). Ceci permet de profiter pleinement de l'intelligence embarquée dans les robots en leur donnant la possibilité d'exécuter localement des ordres de haut niveau tout en respectant les objectifs globaux au travers d'un principe de coalition.

Deux cas d'étude sur des ateliers de production flexibles ont été menés afin de valider la méthode proposée.

Mots clés : FMS, Atelier reconfigurable, Industrie 4.0, Architectures de contrôle, Manipulateurs mobiles, Robots collaboratifs, Modularité, Supervision distribuée, Ordonnancement, Formation de coalitions, Enchères, Optimisation

Abstract

Mobile manipulators composed of a mobile base (AGV : Automated Guided Vehicle) and a robotic arm have been developed and introduced in Flexible Manufacturing Systems (FMS) to ensure the complete and autonomous execution of transportation tasks. The supervision of these smart entities, their control architecture, their reconfigurability and their tasks scheduling are quite challenging and attract several researchers.

On the other hand, few works have studied these problems with heterogeneous (made up of several types of resources : mobile manipulators, AGVs, manipulator arms) and/or modular transport systems. This thesis allowed the development, the experimentation and the validation of a distributed supervision system including task scheduling approach. This approach considers both the modularity and intelligence of mobile manipulators, as well as the heterogeneity of transport systems. The task scheduling method is based on a hybrid mechanism combining a global decision algorithm based on auctions and a local decision model based on Integer Linear Programming (ILP). This allows taking full advantage of the intelligence embedded in the robots by giving them the possibility to execute locally high-level orders while respecting global objectives through a coalition principle. Two real case studies on FMS were carried out in order to validate the proposed method.

Keywords : *FMS, Reconfigurable manufacturing systems, Control architectures, Industry 4.0, Mobile manipulators, Collaborative robots, Modularity, Distributed supervision, Scheduling, Coalitions formation, Auctions, Optimization*

Chapitre 1

Introduction générale

1.1 Introduction

Ce chapitre donne un aperçu général de la thèse en situant d’abord le contexte ayant motivé la recherche. Par la suite, les objectifs du projet européen CoRoT qui a financé ce travail de thèse sont donnés dans le contexte de la recherche. En outre, le positionnement et l’originalité du travail effectué au cours de ces trois années de doctorat sont montrés par le dégagement d’une problématique scientifique et l’exaltation de la contribution apportée. Enfin, un plan détaillé comprenant les objectifs de chaque chapitre est exposé.

1.2 Contexte

Les ateliers de production sont soumis à beaucoup de changements, causés par des facteurs sociaux, économiques, technologiques et industriels.

1.2.1 Contexte socio-économique

Au cours de ces dernières années, nous assistons à une forte imprévisibilité dans les demandes du marché [109], [163]. Ainsi des changements significatifs dans les demandes des clients vis-à-vis des produits manufacturiers sont observés [133]. Si auparavant, la sensibilité des prix conduisait à des produits plus uniformes, de nos jours, de nombreux produits de haute qualité et souvent personnalisés, à prix raisonnables et une livraison rapide sont exigés [96],[128].

La consommation quantitative des produits varie fortement, poussant les entreprises à suivre la tendance pour rester compétitif [14]. La toupie à main, plus connue sous l’appellation *Hand spinner* (voir figure 1.1) permet d’illustrer ce constat [179]. Le *Hand spinner* est devenu assez rapidement une tendance mondiale au cours du printemps 2017. Le site d’analyse de ventes industrielles, *NDP group*¹ estime à plus de 150.000 exemplaires (plus d’un 1 million d’euros) le nombre de *Hand spinners* vendu en seulement trois semaines rien qu’en France. Les ventes en Angleterre dans la même période sont assez identiques à celles de la France. Durant les semaines suivantes, les

1. <https://www.npdgroup.fr/wps/portal/npd/fr/actu/communiqués-de-presse/la-folie-des-toupies-les-ventes-des-hand-spinners-passent-la-barre-des-1-million-d-euros-en-france-en-moins-de-3-semaines-ndp-decrypte-les-rouages-du-phenomene-qui-a-pris-le-ma/>, consulté le 03 août 2020



FIGURE 1.1 – Toupie à main

ventes n'ont fait qu'accroître.

Cette tendance a fortement baissé à la fin de l'été, pour presque disparaître quelques mois après.

Le premier semestre de l'année 2020, a mis à rude épreuve la capacité de production des usines de fabrication d'équipements médicaux [105], notamment les masques et les solutions hydro-alcooliques (Voir Figure 1.2) ou même les respirateurs artificiels. En effet, la crise de la COVID-19 a causé une pénurie généralisée de ces produits, devenus la convoitise de plusieurs pays bien que fortement industrialisés [20]. Cette situation montre, la nécessité d'apporter de meilleures solutions aux usines afin de leur permettre de répondre quantitativement, qualitativement et rapidement aux besoins du marché assujettis à des changements imprévisibles [105].



FIGURE 1.2 – Produits convoités au début de l'année 2020, de gauche à droite : Gel hydro-alcoolique, Masque chirurgical

1.2.2 Contexte industriel

Les lignes de production dédiées sont des installations industrielles rentables mise en place généralement pour la fabrication ou l'assemblage d'un

seul type de produit. Le paradigme de la production en masse dans les usines a été initiée par Henry Ford au début du 20^{ème} siècle sur une ligne d'assemblage. Ces lignes ont de bons rendements, lorsqu'il s'agit de produire une même pièce et lorsque la demande du marché par rapport à cette pièce ou produit est élevée [87]. Cependant, pour la fabrication de produits variés et personnalisés en masse qui sont pourtant les tendances actuelles du marché [93], ces lignes traditionnelles voient leurs limites. L'intégration des ateliers ou systèmes de production flexibles (en anglais : Flexible Manufacturing Systems : FMS) est une solution pour faire face à ces changements. Les lignes de production dédiées ont alors évolué vers des cellules de production plus flexibles. La flexibilité étant la capacité et la vitesse d'adaptation à de nouvelles situations [136]. Par contre, ces usines ont des coûts d'installation et de production élevés. En plus, paradoxalement l'un des grands handicaps rencontré dans les FMS, est le manque ou l'absence de flexibilité [120]. L'avènement de l'Industrie 4.0 permet de palier aux problèmes des FMS en permettant la production en masse des produits personnalisés avec une meilleure qualité et dans un court délai de livraison. Cette nouvelle révolution industrielle se base sur l'utilisation de diverses technologies (Systèmes Cybers Physiques, Cloud computing, Internet des objets,...) pour permettre une gestion intelligente et optimisée de l'usine et ses éléments [193]. Ce qui a pour conséquence un accroissement de la flexibilité.

Dans cette optique de flexibilité, les systèmes utilisant des convoyeurs pour assurer les opérations de transport et transfert ont été jugés comme un frein car définissant un chemin et une séquence non modifiable au produit. Ainsi dans les systèmes de transport, nous assistons à l'emploi des systèmes robotisés et autonomes tels que les bases mobiles ou les manipulateurs mobiles pour les tâches de transport, répondant bien aux besoins en flexibilité des usines [147].

1.2.3 Projet CoRoT

Les manipulateurs mobiles sont des équipements industriels assez coûteux et difficilement accessibles aux PME (Petite et Moyenne Entreprises). Le projet européen CoRoT² vit le jour dans le but de rendre accessible l'utilisation des solutions robotisées de transport aux PME. Ainsi, ce projet de 3.9 millions d'euros vise à développer des manipulateurs mobiles à des prix avantageux pour garantir leur utilisation même dans les structures les plus modestes. Les objectifs du projet CoRoT peuvent alors être synthétisés en trois points :

- (1) Rendre accessible aux PME l'utilisation de systèmes automatisés de transport, notamment avec des robots manipulateurs autonomes, intelligents et collaboratifs pour les ateliers de production.
- (2) Développer les outils numériques de supervision distribuée permettant la gestion des systèmes de production flexible utilisant ces robots.

2. Le nom CoRoT vient de l'anglais Collaborative ROboT, site web du projet : <http://www.corot-project.org/fr>, consulté le 27 Août 2020



FIGURE 1.3 – Solution du projet CoRoT

- (3) Sensibiliser et accompagner les entreprises sur l'utilisation de ces nouvelles technologies.

Ce travail de thèse s'inscrit dans le cadre du projet CoRoT, et intervient dans le deuxième point.

1.3 Problématique scientifique

Les manipulateurs mobiles sont des entités mobiles. Une entité peut se définir comme toute unité autonome capable de prendre des décisions, de communiquer et d'agir [121], [170]. Deux entités constituent les manipulateurs mobiles, à savoir un bras manipulateur pour le chargement et le déchargement des pièces, et une base mobile ou AGV³ (Automatic Guided Vehicle). Ces deux éléments constitutifs du manipulateur mobile sont des éléments intelligents et peuvent être assemblés ou désassemblés dans certains cas, à l'instar de la solution proposée par le projet CoRoT. Cette configuration confère une certaine modularité aux manipulateurs mobiles [61]. Ainsi la modularité d'un manipulateur mobile fait qu'on peut le considérer sous deux axes :

- (1) Un AGV et un bras manipulateur joint physiquement et disposant d'une seule unité décisionnelle.
- (2) Deux entités séparées, chacune étant dotée de son propre module de décision et de contrôle.

La modularité complexifie aussi l'assignation de tâches, car le système devient fortement hétérogène. On peut ainsi avoir un système composé de manipulateurs mobiles (AGV et bras joints), d'AGVs et/ou de bras.

Plusieurs travaux se sont intéressés à l'emploi de ces manipulateurs dans l'industrie. Ces travaux traitent principalement sur l'optimisation du nombre d'AGVs à utiliser et les possibilités de gérer plusieurs produits à la fois [94]; Sur l'itinéraire des véhicules et la résolution des conflits pour garantir des

3. L'acronyme VGA pour Véhicule à Guidage Automatique est utilisé en français

déplacements sûrs et efficaces [69], [147]; Sur le positionnement des véhicules vides et la gestion des batteries [94]. D'autres travaux ont porté sur l'ordonnancement des tâches, qui consiste à séquencer et assigner des tâches de transport aux manipulateurs mobiles.

Cependant, la plupart des travaux existants considèrent les manipulateurs mobiles comme une seule entité lors de l'assignation des tâches de transport [75], [178], [79]. Cela a pour conséquence, une considération partielle des opérations qui constitue la tâche. En général seule l'opération de déplacement du produit est considérée, ce qui conduit à une imprécision dans l'ordonnancement et à la négligence du rôle du bras qui est de charger ou décharger des produits.

D'autre part la modularité de ces manipulateurs, fait en sorte que les deux entités (Bras et AGV) peuvent travailler séparément, rendant le système hétérogène. Les travaux d'ordonnancement de tâches de transport traitent souvent des systèmes homogènes constitués que d'AGVs, de bras manipulateurs uniquement ou que de manipulateurs mobiles pour réduire la complexité [35], [181].

Enfin, l'intelligence des manipulateurs modulables souffre d'une mauvaise mise à profit dans les travaux dédiés à leur intégration. En effet, grâce à leur intelligence ils peuvent exécuter des ordres de haut niveau comme effectuer localement l'assignation d'une tâche macroscopique (tâches constituées de plusieurs opérations) tout en faisant attention aux objectifs globaux de l'atelier en terme de temps de livraison, qualité de produit, exploitation de ressources, etc.

1.4 Contributions de la thèse

L'objectif de cette thèse est de définir une architecture de supervision spécifiant les interactions entre les entités d'un atelier flexible. De plus, elle doit mettre en oeuvre une méthode d'ordonnancement de tâches en prenant en compte :

- **La modularité des manipulateurs mobiles** : La méthode proposée devra tenir compte du rôle de chaque entité (Bras et AGV) qui constitue les manipulateurs mobiles. De même, les tâches de transport seront considérées comme une structure complexe pouvant être composée de plusieurs opérations reliées par des contraintes de parallélisme et de précedence.
- **L'hétérogénéité du système de transport** : Les ressources de transport seront de nature différente avec des caractéristiques et performances propres à chacune. Ce qui suscitera le développement d'un modèle générique représentant les différentes entités.
- **L'intelligence des entités** : Le travail devra permettre de bénéficier de l'intelligence des manipulateurs mobiles en leur offrant la possibilité de prendre localement des décisions et de pouvoir effectuer l'ordonnancement de tâches de transport macroscopiques. Le principe de coalition

permettant la mise en commun des compétences des ressources pour la réalisation d'actions complexes sera utilisé. Par ailleurs l'autonomie accordée à ces entités est réglementée par un superviseur central pour éviter les minimums locaux et la négligence des objectifs globaux de l'atelier.

- **L'intégration des opérateurs** : Le rôle de l'humain dans la réalisation des tâches de transport est souvent indispensable. L'hétérogénéité du système incorporera également des opérateurs humains travaillant à proximité des robots et participant aux coalitions.
- **Le plan de production** : Les tâches de transport seront régies par un plan de production qui définit la séquence et les temps de passage des produits sur les machines. L'ordonnement de ces dernières prend en compte toutes les possibilités de réalisation d'une tâche de production (flexibilité de routage). Ainsi, la méthode proposée dans cette thèse fournira à la fois un plan de transport et de production en tenant en considération la flexibilité de routage des produits.

1.5 Organisation du manuscrit

Ce manuscrit de thèse est constitué de cinq chapitres comme l'indique l'organigramme 1.4, répartis comme suit :

- **Chapitre 2** : Fournit des définitions et une recherche bibliographique sur les travaux et thématiques qui cadrent avec cette thèse. Ainsi les FMS sont abordés, de même que les solutions qu'apportent l'Industrie 4.0 à ces ateliers. Puis, des définitions de la supervision sont tirées de la littérature et analysées pour montrer la nuance avec le contrôle. Les types d'architectures de supervision (Centralisé, hiérarchique, hétérarchique et hybride) sont alors détaillés, en montrant les avantages et les inconvénients de chacun puis en faisant un choix sur le mieux adapté à notre besoin. Plusieurs travaux sur l'ordonnement de tâches à des ressources de transport sont étudiés, tout en faisant ressortir leurs limites. La suite du chapitre, s'intéresse aux entités de transport ainsi que leurs différentes configurations. Le chapitre aborde enfin, les travaux basés sur les coalitions des ressources industrielles.
- **Chapitre 3** : Propose une architecture de supervision basée sur un modèle d'agent générique, découlant d'une méthode existante appelée DIMS (Dynamically Integrated Manufacturing System). L'originalité et les motivations ayant conduit au choix de l'architecture DIMS sont discutées, de même que l'adaptation robotique faite. DIMS propose également un algorithme d'assignation de tâche avec un processus d'optimisation indirecte. Après avoir montré les limites de cet algorithme, l'auteur de cette thèse a apporté des modifications majeures permettant un ordonnancement de tâches plus efficace. L'efficacité de cette méthode d'ordonnement de tâches basé sur un algorithme génétique bi-objectif, est prouvée par un cas d'étude à la fin du chapitre.

- **Chapitre 4** : Cette partie du manuscrit, présente d'abord l'approche d'ordonnancement de tâches (IT2P, Integrated Transportation and Production Plans) basé sur une programmation par contrainte qui permet l'obtention simultanée d'un plan de production et de transport partiel (réduisant la complexité des tâches de transport). L'approche IT2P a été développée par l'université d'Exeter (en Angleterre) et le laboratoire LINEACT de Rouen (en France). L'auteur de cette thèse y a contribué par le biais du second laboratoire. Le plan de production obtenu par l'approche IT2P est utilisé comme support par une seconde approche (DISCOFT, DIStributed COalition Formation for Transportation) pour définir un nouveau plan de transport plus complet (tenant compte de la complexité des tâches de transport) de même que la modularité et l'hétérogénéité du système grâce au principe de coalition. DISCOFT est le coeur du travail de cette thèse, et se base sur un mécanisme hybride combinant une programmation linéaire en nombres entiers pour les décisions locales et un système d'enchères centralisé pour les décisions globales.
- **Chapitre 5** : Ce chapitre présente la modélisation d'un cas d'étude fait sur le démonstrateur du futur du laboratoire LINEACT pour la validation de l'approche DISCOFT. Ainsi, la faisabilité et l'efficacité de l'approche proposée sont démontrées par une série de tests et d'expériences. Une étude comparative avec une approche existante est aussi réalisée.

Le manuscrit se termine par une conclusion générale avec un ensemble de perspectives pour de futurs travaux.

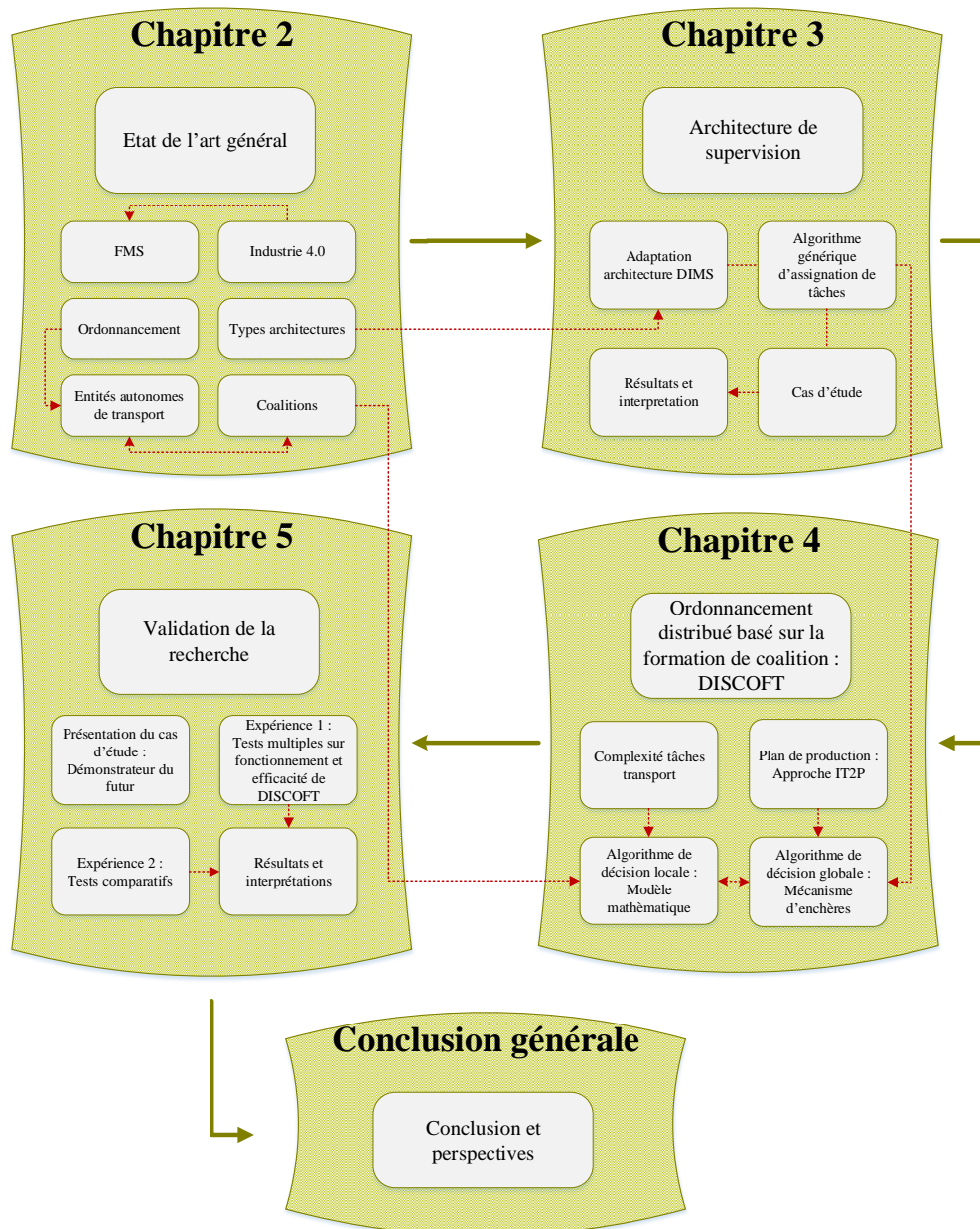


FIGURE 1.4 – Organigramme général du manuscrit de thèse

Chapitre 2

Revue littéraire

2.1 Introduction

Les ateliers de production flexibles sont des structures complexes par leurs composition et gestion, ayant suscité un intérêt particulier de plusieurs travaux de recherches. Ainsi la section 2.2 de ce chapitre est dédié à une recherche bibliographique sur les systèmes flexibles et les types de décision rencontrés dans les ateliers de production. L'apport de l'Industrie 4.0 pour les FMS est montré dans la section 2.3.

Les catégories d'architectures de supervision utilisées dans les ateliers de production flexibles sont abordées dans la section 2.4, en fournissant plusieurs exemples d'architectures de même que les avantages et inconvénients de chaque catégorie. Cette partie traite aussi des différents modules qui constituent un système de supervision. L'ordonnancement des tâches faisant parti des constituants d'un système de supervision est traité dans la section 2.5.

Une attention est portée par la suite aux entités autonomes, plus particulièrement à celles utilisées pour le transport dans les usines intelligentes en section 2.6. Des travaux évoquant le fonctionnement en équipe de plusieurs entités intelligentes sont analysés dans la section 2.7.

Le chapitre se termine par une conclusion, synthétisant les travaux de recherches en relation avec cette thèse.

2.2 Les systèmes de production flexibles

La production industrielle est la transformation de matière première en produit fini demandé par le marché. Des entrées supplémentaires sont nécessaires pour le bon fonctionnement telles que : l'énergie, les équipements, les informations du marché, le Design du produit [140].

La flexibilité est la capacité et la vitesse d'adaptation à de nouvelles situations [136], et a été introduite à divers niveaux des systèmes de production pour suivre les tendances et la variabilité des demandes du marché assez rapidement tout en minimisant les coûts. Ainsi, un système de production flexible ou FMS¹ se caractérise par sa faculté à réagir aux changements prévisibles et

1. L'acronyme **SFP** pour Système de Production Flexible est utilisé pour caractériser ce type de système. En anglais ça se traduit par "*Flexible Manufacturing System*" d'où l'acronyme **FMS**. L'auteur de cette thèse a fait le choix d'utiliser ce dernier (FMS), dans tout le manuscrit.

imprévisibles [162]. Les FMS sont des systèmes complexes constitués d'éléments comme les postes ou stations de travail, d'aires de stockages, de matériels de transport tels que les convoyeurs, robots et des opérateurs humains [17], [139].

2.2.1 Types de flexibilité

La flexibilité est un concept ancien utilisé dans divers domaines. Elle voit ses débuts dans le monde économique dans les années 1920 [149], pour ensuite être utilisé dans le milieu manufacturier au début des années 1950 sur les machines [78].

La classification et la définition des types de flexibilité des systèmes manufacturiers sont des thématiques qui ont fortement été abordées dans les travaux de recherches bien avant les années 1990. Malgré cette ancienneté, de récents travaux évoquent des difficultés à trouver des appellations standards sur les termes de types de flexibilité, ce qui fait en sorte que différents termes peuvent être utilisés pour caractériser une même flexibilité [91]. Ainsi plus de 70 termes relatifs aux types de flexibilité ainsi que leurs mesures ont été recensés dans la littérature [153], [149]. Sachant que plusieurs termes peuvent renvoyer à un même type de flexibilité ou juste avec des différences mineures dans la définition.

L'objectif de cette partie n'est pas de tous les énumérer mais plutôt de citer quelques-uns en lien direct avec la thématique de recherche de cette thèse comme la flexibilité : Des outils de manutention, de routage, des opérations, des machines et du produit.

2.2.1.1 Flexibilité des outils de manutention

Les outils de manutention constituent l'ensemble des équipements et matériels (Chariots élévateurs, convoyeurs, robots mobiles, bras manipulateurs, etc.) utilisés pour le transport des produits y compris ceux servant à ranger, charger ou décharger ces produits. La flexibilité de ces outils concerne leur capacité à déplacer efficacement les produits des systèmes de production qu'ils servent [149]. Cette flexibilité permet aussi aux ressources de manutention de prendre en charge une grande variété de produits.

La modularité permettant aux ressources de transport de se scinder en plusieurs autres, est également une forme de flexibilité voire de reconfigurabilité. En effet, cette transformation leur permet de répondre efficacement et de s'adapter aux besoins de transport [61].

2.2.1.2 Flexibilité de routage

Concerne l'aptitude de pouvoir utiliser plusieurs machines pour réaliser le même type d'opération sur un produit. Dans la flexibilité de routage, chaque opération dans la séquence de fabrication d'un produit dispose de plusieurs machines capables de la réaliser avec des performances et qualité similaires ou différentes [161]. C'est ce type de flexibilité qui est intégré dans les problèmes de Job-Shop flexible (Voir plus de détails dans la section 2.5.3).

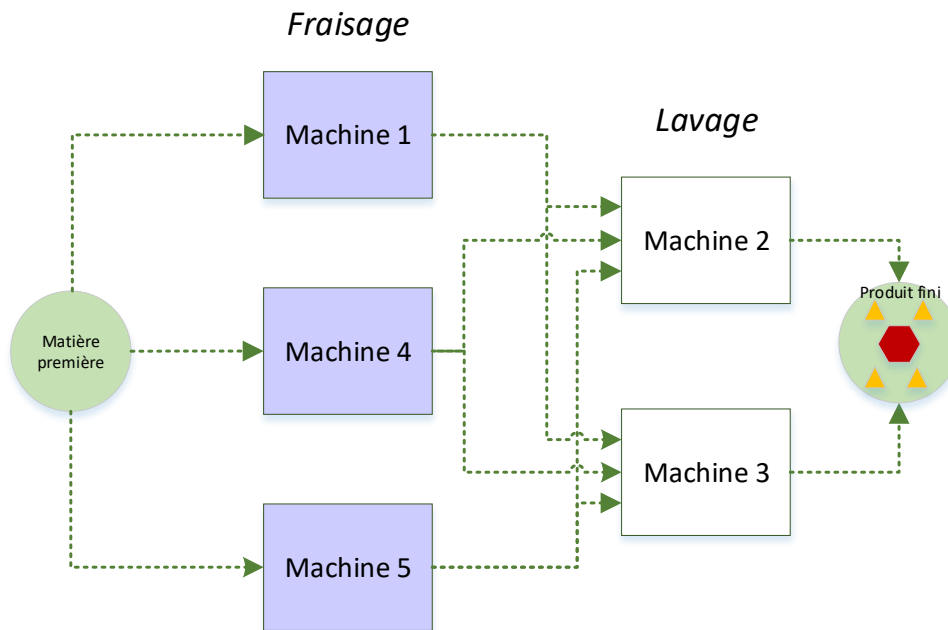


FIGURE 2.1 – Illustration flexibilité de routage : Spécification des différents chemins possibles en pointillés pour la fabrication d'un produit devant passer par une opération de fraisage puis de lavage

2.2.1.3 Flexibilité des machines

La variété des types d'opérations que peut réaliser une machine constitue sa flexibilité. Cet atout au niveau des machines permet entre autre la réduction du temps d'intégrer de nouveaux produits dans le système. Cette flexibilité s'applique à l'ensemble des ressources de production (ressources utilisées pour fabriquer ou assembler des produits). Grâce à la flexibilité des machines, une ressource en panne peut être facilement remplacée par une autre [128].

2.2.1.4 Flexibilité des opérations

Ce type de flexibilité est relatif aux produits, notamment sur les différentes possibilités que disposent un produit à être fabriqué. Ainsi lorsque plusieurs séquences différentes conduisent à l'obtention d'un même produit on parle de flexibilité d'opération. La séquence d'un produit étant l'ensemble des étapes décrivant sa fabrication [23]. La flexibilité des opérations se distingue de celle de routage car la première concerne l'ordre des processus et les types d'opérations (Exemple : (Perçage puis soudure) ou (soudure puis perçage) conduisent à l'obtention du même produit) et la seconde (routage) est relative à l'ensemble des machines capables de réaliser un type d'opération (Exemple : Perçage (M_1 , M_2 ou M_4) puis soudure (M_3 ou M_1)).

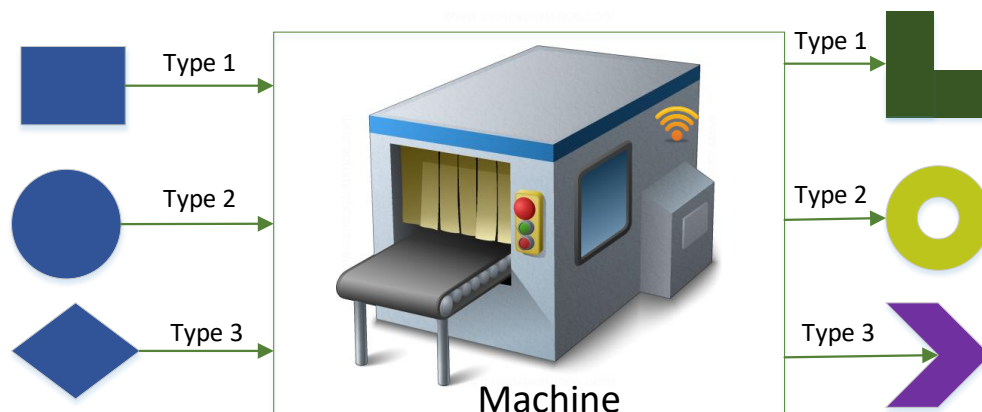


FIGURE 2.2 – Flexibilité d’une machine pouvant fabriquer trois types de produits

2.2.1.5 Flexibilité du produit

C’est la facilité qu’a un produit à être modifié. La flexibilité d’un produit lui garantit le remplacement, la suppression ou l’ajout de nouveaux éléments. C’est donc la rapidité à laquelle un produit en fabrication peut être changé. Cette flexibilité permet aux entreprises d’être plus réactives et compétitives par la modification rapide des produits existants [149]. Les types de produits qu’un système de production peut fabriquer sans qu’on lui ajoute des équipements majeurs est traduite par la flexibilité de production [57].

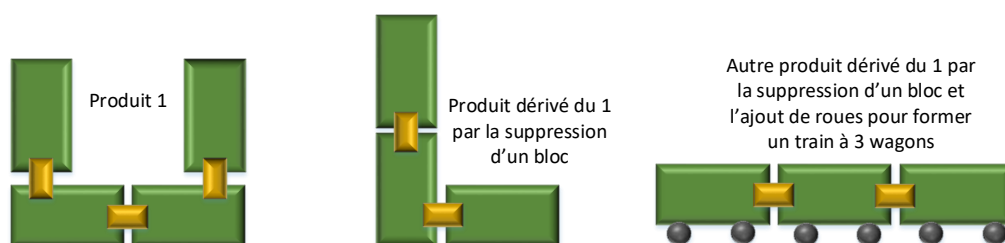


FIGURE 2.3 – Illustration de la flexibilité des produits

2.2.2 Niveaux de décision dans les FMS

Les décisions dans les FMS permettent de définir les paramètres et sorties de l’atelier comme les types de produits à fabriquer, le type de matériels à utiliser, leurs configurations incluant les emplacements, le contrôle et toute l’organisation de la production à savoir l’ordonnancement des tâches entre autre [2]. Ces décisions ont un impact majeur sur la vie d’une usine ou entreprise et peuvent être prises pour une longue durée ou à intervalle régulier en fonction du type de décision et de la flexibilité du système. Trois niveaux sont invoqués dans la littérature pour les différentes décisions : Stratégique, Tactique et Opérationnelle [151].

La figure 2.4, montre la hiérarchie entre les trois niveaux décisions. Le pre-

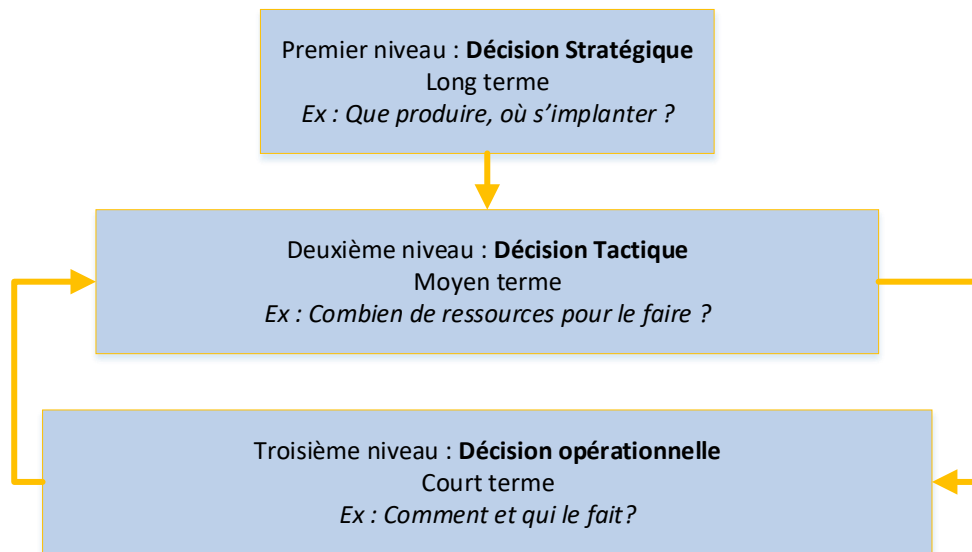


FIGURE 2.4 – Les trois niveaux de décisions

mier niveau concerne les décisions **stratégiques**. A ce niveau, se définit les objectifs fondamentaux et les décisions qui cautionnent la production de l’atelier. Les questions qui concernent les produits à fabriquer, l’emplacement de l’atelier et sa structure générale se posent à ce niveau. Le type de ressources à utiliser (Machines, robots ou/et opérateurs) est aussi évalué. Tuan Le-Anh et al. [94], considèrent que la définition des chemins que doivent ou peuvent suivre les AGVs dans un atelier est une décision d’ordre stratégique. Les choix faits à ce niveau sont généralement pour une longue durée.

Les choix effectués sur le moyen terme, sont effectués au deuxième niveau. Les décisions **tactiques**, reflètent ce second niveau, et fournissent des réponses à des questions plus spécifiques que le premier niveau. Le dimensionnement (nombre de ressources à utiliser et leur emplacement) d’un atelier de production est un choix tactique. Les décisions prises au sommet (premier niveau) servent d’entrée à ce niveau [7].

Les décisions **opérationnelles** régissent l’ensemble des problèmes instantanés ou des choix à appliquer sur le court terme. Ainsi, l’ordonnancement des tâches, le choix des itinéraires, la résolution des problèmes de navigation (comportements et règles à adopter en cas de croisement entre deux véhicules) sont des décisions d’ordre opérationnel. Les décisions prises au niveau tactique sont détaillées et appliquées à ce niveau [59].

Une forte interaction existe entre les niveaux tactique et opérationnel. Par exemple, suite aux recommandations du niveau opérationnel, le tactique peut réduire le nombre de ressource en utilisation dans la production. Le travail effectué dans cette thèse est essentiellement basé sur des décisions opérationnelles mais permet également de mettre en lumière les interactions entre les niveaux tactique et opérationnel grâce à une étude sur les activités et la productivité des ressources de transport.

2.3 Industrie 4.0

Outre la complexité des FMS, on croise également un grand souci qui est paradoxalement le problème de flexibilité [120]. Les coûts d'installation et d'exploitation des FMS sont aussi des paramètres constituant des freins car assez souvent élevés. L'Industrie 4.0 aussi appelée industrie du futur, permet d'apporter des solutions aux problèmes de flexibilité et de gestion des FMS par la mise en place d'usines intelligentes capables d'une plus grande adaptabilité.

2.3.1 Nouvelle révolution industrielle

L'avènement de l'Industrie 4.0 est considéré comme une nouvelle forme de révolution industrielle à l'instar du machinisme au 18^{ème} siècle, la production en masse au 19^{ème} siècle et la production automatisée au 20^{ème} siècle [150], ayant tous pour objectif premier l'accroissement de la productivité. C'est à ce titre que l'Industrie 4.0 ou I4.0 est considérée comme la quatrième révolution industrielle [18]. La figure 2.5 retrace la chronologie des quatre révolutions industrielles.

Le concept de l'I4.0 a été mis en lumière pour la première fois en 2011, par le gouvernement allemand [194].

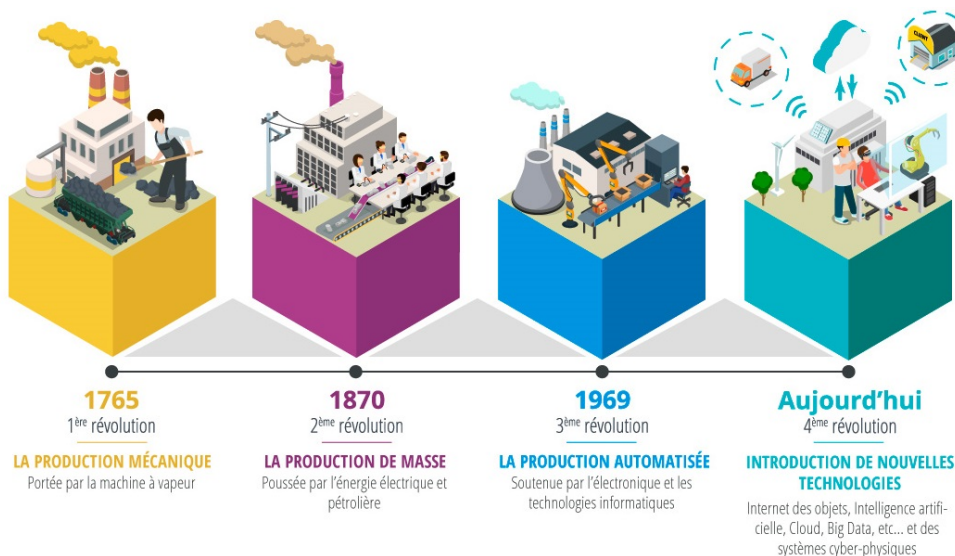


FIGURE 2.5 – Les quatre révolutions industrielles²

2.3.2 Principes et outils de l'I4.0

Les principes fondamentaux de l'I4.0 sont la distribution des prises de décision, la transparence de l'information et l'assistance technique des opérateurs. Une interconnectivité existe entre tous les éléments de l'usine, favorisant des offres de services mutuels [6].

Ce nouveau paradigme industriel englobe un ensemble de développements

2. Source image : <https://www.visiativ-solutions.fr/industrie-4-0/>, consulté le 03 Août 2020

technologiques tel que la robotique, les Systèmes Cyber Physiques (CPS), l'Internet des Objets (IoT), le Cloud Computing, la réalité augmentée, la Big Data et la réalité augmentée [127]. L'adoption de ces outils a un impact positif sur les systèmes de production par l'amélioration des processus de fabrication, une allocation plus efficace des ressources, l'obtention des produits de meilleures qualités et personnalisées tout en minimisant les coûts de production [127], [193].

2.4 Architectures de contrôle

Le développement des architectures permettant le contrôle des systèmes manufacturiers en général et flexibles en particulier est un domaine qui attire autant les recherches académiques qu'industrielles [120]. Leitaó [96] considère qu'un système de production n'a pas un grand intérêt ou utilité sans un système de supervision approprié. Il précise que la fiabilité et la flexibilité d'un système de production ne dépendent pas seulement de ses composants (robots, convoyeurs ou postes de travail) mais également d'une bonne architecture de contrôle.

L'architecture dans le domaine manufacturier est une structure matérielle ou logicielle, définissant l'interaction et les liens qui peuvent exister entre les constituants internes d'un élément ou cet élément avec les entités externes. Ainsi l'architecture de contrôle définit le schéma directeur pour la conception et la construction du contrôle des FMS [157]. Selon la structure, l'architecture de contrôle attribue des responsabilités de contrôle à une ou plusieurs entités décisionnelles [140].

2.4.1 Nuance entre Contrôle et Supervision

Il n'existe pas de consensus ou de standard sur les définitions de contrôle et de supervision [34]. Leurs terminologies varient en fonction du domaine ou de l'application faite. Néanmoins, on peut retrouver des similarités entre les définitions dans plusieurs travaux.

Certaines études, considèrent la supervision comme un élément d'une architecture de contrôle. Dans [19], [175], [130], [98] le superviseur est considéré comme étant l'entité en charge de la coordination et de l'optimisation des tâches des éléments qui dépendent de lui. Dans ce cas de figure, il cohabite avec d'autres modules (Surveillance, base de données, etc) avec qui il forme l'architecture de contrôle (Supervision \subset Contrôle).

Le terme de supervision est quelques fois restreint à la surveillance (Monitoring en anglais), le suivi et le pilotage des processus industriels. C'est dans ces cas où il est possible de parler de poste de supervision en industrie. Son rôle se limite au visionnage des processus en cours, à leur lancement et arrêt en cas de problème ou au besoin. En nous basant sur la thèse de Herpson [70], on peut définir la supervision d'un système comme étant l'ensemble des méthodes permettant la détection, l'isolation, l'identification et la réparation des fautes pouvant subvenir dans ce système.

Dans [28], le superviseur est l'entité qui détermine toutes les issues possibles des tâches et actions envisageables en vue de prévoir le traitement à réaliser

dans chacun des cas. En d'autres termes, c'est le système devant s'assurer de l'exécution des tâches, de la communication qui peut intervenir autour de cette exécution, et aussi l'affinement de la tâche en sous-tâches. Cette définition permet de voir le système de supervision comme un "tout" et le système de contrôle en charge de l'exécution des tâches comme une "partie" de ce "tout". Cette inclusion du système de contrôle dans la supervision est aussi partagée par Couvoisier et al. [31]. Ainsi le système de supervision dans leur travail, prend non seulement en compte le processus (liste des tâches) mais aussi son contrôle. Ce dernier suppose la coordination des différentes ressources affectées par le processus.

Marcos et Combacau [34] se sont intéressés à la terminologie de contrôle, en la définissant comme l'organe déclencheur de l'exécution d'un ensemble d'opérations tout en faisant respecter les différentes séquences. Cette définition de contrôle reste technique et permet de retrouver par exemple le rôle des contrôleurs de certaines ressources comme les robots. Par ailleurs la supervision est différente de la surveillance dans leur travail. Le superviseur est l'entité en charge de la prise de décision (Optimisation, ordonnancement des opérations), et le Monitoring est en charge de la collecte des informations depuis le système de contrôle pour garder un historique d'éventuels diagnostics.

En regardant l'aspect logiciel de la supervision, qui est un ensemble de bibliothèques développées pour remplir des fonctionnalités par rapport à un besoin spécifique et les différentes définitions fournies dans la littérature, on peut schématiser la supervision dans la figure 2.6.

Ainsi elle englobera les modules de : décision, surveillance, communication et reconfiguration, et sera considérée comme le contrôle. De ce fait, nous ne ferons aucune nuance entre la supervision et le contrôle, ce qui permet d'affirmer que : *Architecture de contrôle = Architecture de supervision*.

2.4.2 Types d'architectures de contrôle

Deux grandes familles d'architectures ont été développées et étudiées dans la littérature : Les centralisées ou traditionnelles dans lesquelles une seule entité est responsable de la prise de décision, et les décentralisées ou distribuées accordant plus de l'autonomie à plusieurs entités du système [160]. Cette dernière famille est constituée de plusieurs groupes dont la composition varie légèrement en fonction des travaux de recherches.

Le travail de Dilts et al. [39] considéré comme une référence dans cette catégorie, répartit les architectures distribuées en trois grandes sous-familles : Hiérarchique, Hiérarchique modifiée et hétérarchique. Par ailleurs la répartition de Trentesaux [170], une autre référence également, se distingue par l'introduction la sous-famille hybride. Cette nouvelle classe combine les avantages des classes hiérarchique et hétérarchique, contrairement à la catégorie hiérarchique modifiée qui est une forme d'extension de la classe hiérarchique. Les autres groupes restent identiques pour les deux représentations comme l'illustrent les figures 2.7 et 2.8, respectivement selon Dilts et Trentesaux.

La suite de cette section, fournit plus de détails sur les différentes classes d'architectures.

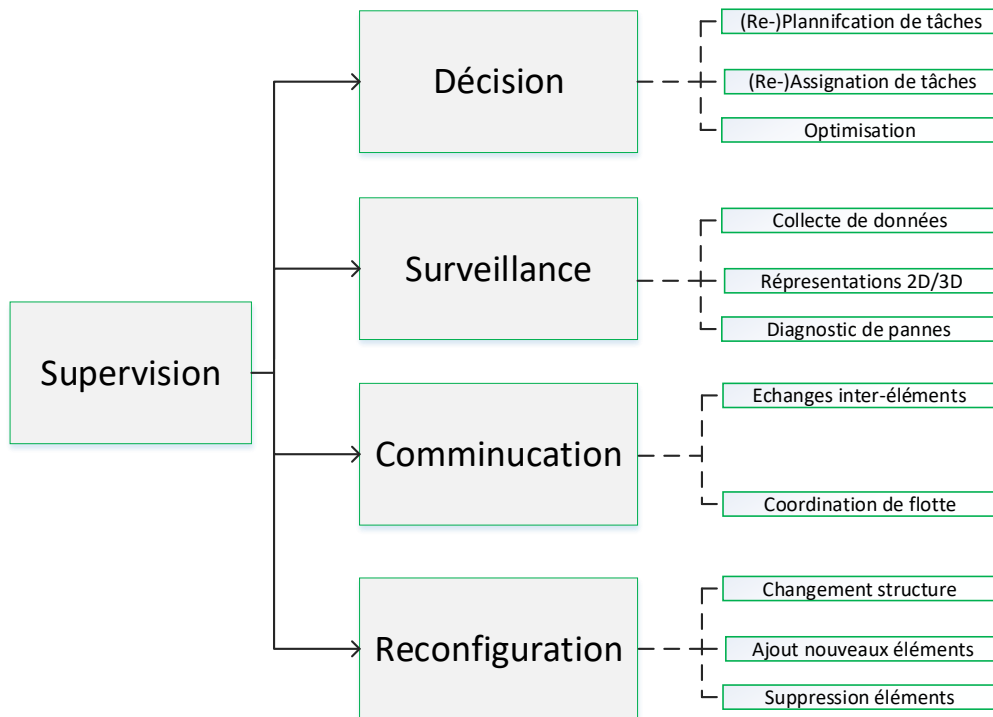


FIGURE 2.6 – Modules de la supervision

2.4.2.1 Centralisée

Ce type de structure se base sur la gestion de tout l'atelier par un seul centre de décision. Les architectures appartenant à cette catégorie sont constituées de deux niveaux : Le premier niveau se charge de la prise de décision et sa transmission au second niveau qui se charge de l'exécution des ordres. Ce dernier niveau remonte également des informations sur l'état de l'atelier pour que le premier niveau puisse intervenir au besoin [8].

Grâce la centralisation des données, les informations sont facilement accessibles. La structure est facilement implémentable avec des coûts non onéreux, raison pour laquelle c'est largement utilisé en milieu industriel. Par ailleurs, un système centralisé présente des résultats de fonctionnement optimal lorsque l'atelier marche sans perturbations [45].

Cependant, le fait que toutes les décisions soient prises par une seule entité conduit à une mauvaise réactivité du système en cas de perturbation, chose récurrente en industrie. En outre, le système souffre d'un temps de réponse relativement faible et d'une rigidité rendant difficile l'intégration de nouveaux éléments [128],[39].

En somme, ce type de contrôle est surtout adapté pour des entités non-autonomes, inactives dans les prises de décision. Pour ce qui est du contrôle des entités actives, il reste désuet [160].

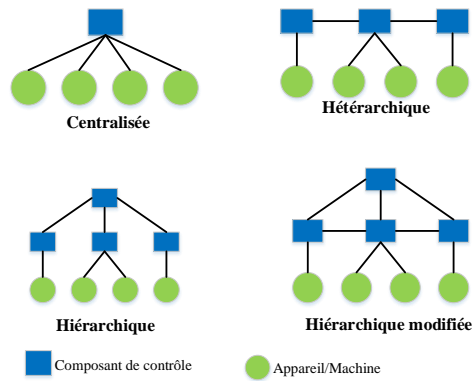


FIGURE 2.7 – Types d'architecture selon Dilts et al. [39]

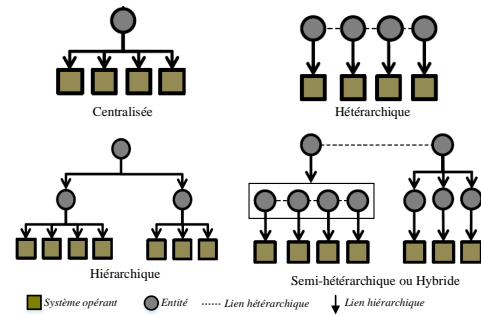


FIGURE 2.8 – Classes d'architecture selon Trentesaux [170]

2.4.2.2 Hiérarchique

Les architectures hiérarchiques sont une forme d'évolution directe des architectures centralisées. De ce fait, elles constituent un ensemble de sous-systèmes centralisés. Dans cette catégorie le nombre de niveaux est alors strictement supérieur à deux vu qu'elle inclut au moins les deux niveaux de la structure centralisée. Le flux d'informations est vertical et de l'autonomie est accordée à certains niveaux même si une entité centrale se trouve au sommet (Le plus haut niveau de la hiérarchie). Le système bénéficie d'un temps de réponse rapide et d'une prédictibilité du comportement global grâce à la présence d'entité de contrôle central [47], [177].

Néanmoins la similitude avec la structure centralisée fait qu'elle souffre également d'une forte rigidité et faible flexibilité, qui engendrent une mauvaise adaptation aux changements. Leitao [96], la situe (structure hiérarchique) d'ailleurs dans la catégorie des systèmes de contrôle traditionnel à l'instar des centralisées.

2.4.2.3 Hétérarchique

Les approches hétérarchiques, bien que faisant partie des alternatives aux systèmes intelligents récents, ont été introduites depuis les années quatre-Vingt par Duffie et Piper [46] par l'appellation d'architecture non-hiérarchique. Hatvany [64] fait également partie des premiers à avoir démontré la nécessité des systèmes de contrôle non-hiérarchique, car selon lui les architectures centralisées et hiérarchiques sont assez rigides, contraintes de suivre les actions prédéterminées.

Ce type de contrôle se caractérise par l'absence total d'unité centrale et de hiérarchie, ce qui se traduit par un flux d'information horizontal. Les entités communiquent essentiellement par négociation [159].

Grâce à une forte flexibilité, le système peut facilement s'adapter aux changements et être tolérant aux pannes et perturbations.

Dans [18], une modélisation industrielle est faite à base de Lego pour pouvoir

mener une étude comparative entre les architectures centralisées et hétérarchiques. Le temps de parcours de palettes (comprenant les produits) est utilisé comme critère d'évaluation. Les résultats de l'expérience montrent que pour un système dont le comportement est connu et prédictible au cours du temps, les architectures centralisées présentent une meilleure gestion. Cependant en cas de perturbations, un système de contrôle beaucoup plus flexible (hétérarchique) réagit mieux.

L'absence de hiérarchie garantit également une atteinte des objectifs locaux, mais les objectifs globaux sont difficilement atteignables à cause de la myopie des entités [121]. Ainsi, le comportement global du système reste fortement imprévisible. De plus, un des éléments majeurs dans les structures hétérarchiques reste la communication, pourtant aucun standard n'est fourni pour les échanges d'information entre entités selon Boccella et al. [18]. Néanmoins, des protocoles comme MTconnect ou OPC UA garantissent la communication entre des entités peu importe la structure architecturale.

2.4.2.4 Hybride

Afin de bénéficier de l'optimisation globale des approches hiérarchiques et de la réactivité des approches hétérarchiques sans souffrir de la myopie des agents, les approches hybrides ont été proposées. Les architectures hybrides utilisent une approche centralisée lorsque l'objectif est l'optimisation globale, et une approche plus hétérarchique en présence d'événements inattendus et de modifications [68], [98]. Le terme de *semi-hétérarchique* est aussi utilisé pour qualifier les architectures hybrides [140], [170].

A l'instar de plusieurs autres domaines (automobile[3], informatique [80], biologie [62], etc.) les hybrides sont connus pour apporter un compromis et de meilleures solutions à des problématiques existantes. Néanmoins, leur contemporanéité fait que leur maîtrise soit incomplète par rapport aux autres. De ce fait, leur application ou implémentation peut s'avérer complexe. En outre, la structure de ces architectures n'est pas des plus simples.

Des alternatives visant à simplifier la structure et l'implémentation des architectures hybrides sur les systèmes intelligents ont été développées. C'est le cas des systèmes Fractals de production (FrMS) inspirés des mathématiques [166], systèmes bioniques de production inspirés (BMS) inspirés de la nature [173], et les systèmes holoniques de production (HMS) inspirés du comportement social [175], [96], [67].

2.4.3 Architectures de supervision notables

Quelques architectures notables sont spécifiées dans cette partie avec leurs noms, le type (défini par développeur de l'approche), et le rôle ou l'application pour laquelle elles ont été initialement développées. Le type d'architecture n'est pas toujours facile à spécifier (Dans le cas où le développeur ne l'a pas fait) si elle dispose de plusieurs fonctionnalités. Par exemple elle peut avoir une relation hiérarchique entre les entités et un système d'optimisation hybride.

- **AARIA** : Cette architecture découle des exigences d'une installation industrielle fabriquant des équipements militaires. Elle se base sur un

système d'agents autonomes pour le contrôle et l'ordonnancement dans les ateliers de production [126].

- **ABAS** : Le domaine d'application de ABAS est la reconfiguration des systèmes manufacturiers (RMS) [87]. Un mécanisme d'enchères et de négociation fondé sur les MAS (Systèmes Multi-Agents) [116], est utilisé pour apporter des solutions de reconfiguration dynamique [101].
- **ADACOR** : Cette architecture de référence, est de type hybride. Elle est implémentée sur les fondements des HMS, et se compose de quatre types de holon : holon produit, holon tâche, holon opérationnel (représentant les ressources) et le holon superviseur en charge de l'optimisation du système. ADACOR permet un fonctionnement hiérarchique dans les conditions normales et un fonctionnement hétérarchique en cas de perturbations [11], [98].
- **ADS2** : Système de supervision hétérarchique basé sur un MAS, dans lequel chaque agent est doté d'une vision locale et de capacités de décision, de diagnostic et de réparation. Cette architecture intègre trois composants : Un modèle décisionnel d'agent ; Un mécanisme distribué de coordination et de re-synchronisation de l'état du système par les agents ; Un algorithme local propre à chaque agent qui entrelace les phases de diagnostic et de réparation afin de réduire la taille de l'espace des diagnostics candidats [70].
- **DIMS** : Architecture complète, offrant des fonctionnalités : d'assignation de tâches d'une façon hiérarchique, d'optimisation indirecte basée sur un GA, de structure générique d'agent pour une implémentation multi-agent et de replanification de tâches [4],[67], [115], [190].
- **MITHRA** : Cette architecture a été développée pour la supervision des robots mobiles autonomes. Elle se base sur coopération entre perception et action. Elle est composée de cinq niveaux : (1) Une interface homme machine [25]; (2) Un superviseur (Responsable de la planification et de l'exécution des tâches); (3) Une couche de perception et de navigation; (4) Une procédure de modélisation de l'environnement; (5) Un contrôleur bas niveau pour les moteurs et autres capteurs [32],[33].
- **ORCA** : Conçue pour être utilisable directement par les industriels et pour plusieurs domaines d'application. l'architecture hybride ORCA est à la fois réactive, car les décisions sont prises là où elles doivent être appliquées (Décision locale) et Optimisée du fait que la myopie des entités prenant les décisions est contrôlée. Deux modes de fonctionnement sont à indiquer : Exécutant (Lorsque les ordres proviennent de la hiérarchie) et Autonome (le cas contraire). Un indicateur est utilisé pour quantifier la myopie des entités en donnant le rapport entre les entités en mode autonome donc myope et le nombre total d'entités [121].
- **PROSA** : Le concept des holons introduit par Arthur Koestler [85], est utilisé pour définir une interaction entre trois types de holons (holon produit, holon ordre et holon ressource). Le holon ressource représente

la partie physique, donc la ressource et les informations de production. Le holon produit contient le processus et la connaissance nécessaire pour assurer sa bonne réalisation. Le holon ordre peut représenter une commande du client ou une tâche dans le FMS et est responsable de l'exécution du travail assigné dans les délais. Un autre holon (Staff) est souvent prévu pour assister les holons de base en cas de problème, en fournissant les directives (conseils) nécessaires aux holons de base, sans jamais être responsable d'une prise de décision. Cette action est réservée aux trois holons de base [175].

- **PROSIS** : Cette architecture également basée sur les HMS, permet de définir un holon produit constitué de deux parties : Une partie matérielle qui représente le produit physique et une partie immatérielle qui contient toutes les informations de fabrication du produit ainsi que sa traçabilité. Ainsi, des holons ressource et ordre interagissent également avec le holon produit. PROSIS permet aussi une assignation de tâche basée sur le Contract Net [74], [130].
- **SCADA** : Considéré comme un facilitateur dans la supervision en temps réel et distante pour les plateformes éloignées, SCADA est une architecture centralisée d'aide à la décision des procédés de fabrication. Elle dispose également d'un outil de surveillance qui garantit une collecte de données sur les ressources du système dont elle est en charge, pour leur représentation graphique. [8].
- **PAC** : Architecture hiérarchique d'ordonnancement de tâches et aussi de replanification en cas de perturbation grâce à un module de surveillance intégré dans l'architecture. En outre, cette dernière permet de garantir également le suivi et l'exécution des tâches planifiées. [12]
- **RSA** : Cette architecture a été mise au point pour la supervision d'une flotte de robot dans le cadre d'un projet d'exploration de la NASA . Elle incorpore plusieurs fonctionnalités telles que : Le contrôle et la surveillance à distance, la planification de tâches, outil de détection de danger, etc.[63].
- **SHFMS** : Architecture de contrôle hybride (semi-hétérarchique) composée d'un niveau de décision globale et un autre pour les décisions locales. L'approche vise à utiliser une technique d'optimisation basée sur la simulation dans le niveau de décision globale pour réduire l'impact de la myopie dans les décisions prises localement [140]. Zambrano part du principe que l'élimination absolue de la myopie n'est pas avantageuse. Du fait que c'est un comportement hérité des systèmes hétérarchiques, elle est nécessaire pour obtenir certaines fonctionnalités désirées dans les FMS à savoir, la réactivité, l'adaptabilité et la tolérance aux pannes [188].

2.4.4 Synthèse sur les architectures de supervision

Le tableau 2.1, fournit un récapitulatif des quatre types d'architectures de supervision. Pour chacune d'elles, les avantages et inconvénients, ainsi que

les types d'entités sur lesquelles elles peuvent être idéalement appliquées sont indiquées.


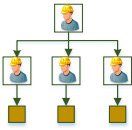
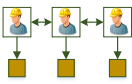
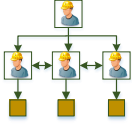
Type	Structure	Avantages	Applicabilité	Limites
Centralisé		Accéssibilité des données, faible coût, Optimisation globale	Entités réactives, non-intelligentes	Rigidité, Faible adaptation
Hiérarchique		Rapidité, Prédicibilité, Robustesse, Optimisation globale	Entités semi-autonomes	Faible adaptation, Manque de flexibilité, rigidité
Hétérarchique		Réactivité, Flexibilité, Adaptation, forte tolérance	Entités autonomes	Myopie, imprédictibilité
Hybride		Rapidité, Réactivité, Flexibilité, Adaptabilité	Entités semi-autonomes et autonomes	Structures complexes, Faible maniabilité

TABLE 2.1 – Récapitulatifs sur les caractéristiques des architectures de contrôle

En définitive, lorsque la prise de décision est centralisée, le système et l'ensemble de ses entités sont mieux contrôlés, ce qui rend le comportement du système prévisible. Le système perd néanmoins de sa flexibilité. Pour que cette dernière croisse, la décision doit être distribuée, ce qui fait perdre le contrôle sur les entités du système. L'absence de contrôle sur les entités rend le système myope, faisant perdre ainsi les objectifs globaux. Ces deux situations sont illustrées par la figure 2.9.

Ainsi pour avoir une architecture de supervision effective, le curseur de la prise de décision doit être bien située entre les types centralisé et distribué pour avoir un compromis entre la flexibilité et le contrôle.

2.5 Ordonnement de tâches

L'ordonnement est utilisé dans divers domaines, tels que : Les systèmes de production dans les usines; Milieu professionnel avec l'aménagement des horaires et tâches des salariés, de même que la gestion de projet; Milieu académique avec l'organisation des emplois du temps des élèves et

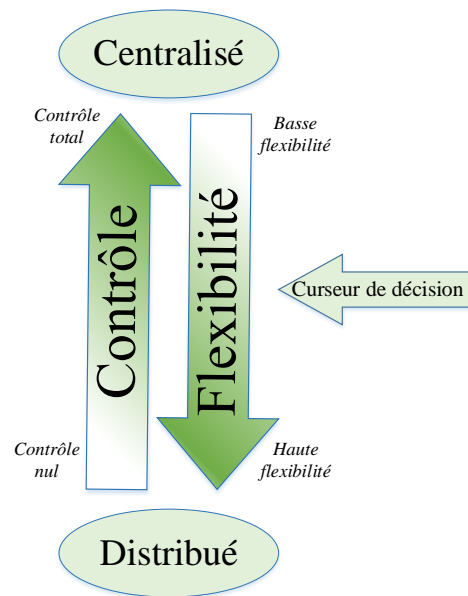


FIGURE 2.9 – Dilemme entre niveaux de flexibilité et contrôle

enseignants; etc [15]. L'ordonnancement de tâches permet d'une part de définir la séquence ou l'ordre entre les différentes tâches, quelle tâche commencer et lesquelles suivront : cette étape s'appelle le **séquencement** ou **planification** [1]. D'autre part, il permet l'assignation des tâches planifiées à des ressources capables de les exécuter : C'est l'**Assignation** [115]. On parle d'assignation de tâches ou d'allocation de ressources.

Selon Wu et al. [184], le rôle de l'ordonnanceur de tâche est d'assigner des tâches à des ressources disponibles tout en satisfaisant les exigences de précedence de ces tâches et en minimisant le temps d'exécution total (Makespan) [119].

Dans son livre paru en 2012, Pinedo relate l'ordonnancement comme un procédé de prise de décision souvent utilisé et incontournable en industrie. Il (l'ordonnancement) gère l'allocation des ressources aux tâches sur une durée donnée en ayant pour but d'optimiser un ou plusieurs objectifs [129]. Cette définition est partagée par Reddy et al. [139], qui considèrent l'ordonnancement comme un processus de décision qui lie les opérations, le temps, le coût et les objectifs globaux d'une compagnie, en allouant un nombre de ressources limitées à des tâches.

Bharadwaj et al. [15] définissent l'ordonnancement comme un processus d'élaboration de plan qui spécifie les temps de début et de fin d'un ensemble de tâches, et les assigne d'une façon efficace à des ressources. Plusieurs paramètres peuvent influencer la réalisation d'un ordonnancement comme : La priorité des tâches, les restrictions de coût, les exigences de date de livraison, la précedence des opérations, les capacités, exigences et disponibilité des ressources.

L'ordonnancement des tâches (*task scheduling* en anglais) permet aussi de prendre en compte la gestion des files d'attente des ressources comme les machines, pour déterminer quelle tâche exécuter en premier. Quelques règles comme : Le premier arrivé premier servi (*First-come-First-Served*, en anglais);

Le temps de processus le plus bas (*Shortest processing time*) ou le délai de livraison le plus proche (*Shortest due date*) permettent de pallier à ce problème de file d'attente [1].

Le choix des critères et méthodes pour la réalisation de l'ordonnancement des tâches est une phase clé, car il impacte fortement l'utilisation des ressources (par la répartition équitables de la charge de travail, prévenant l'usure des ressources et garant de produits de bonne qualité), les délais de livraisons et fabrication des produits. Le fonctionnement global de l'atelier et les coûts de production dépendent en partie d'un bon ordonnancement des tâches [129].

2.5.1 Types d'ordonnancement

Le type d'un ordonnancement de tâche est relatif au moment où se fait l'ordonnancement par rapport à la mise en application. Deux cas sont possibles, selon que l'ordonnancement est fait instantanément (Pour être appliqué à l'instant de la réalisation) ou avec un décalage entre la réalisation et la mise en application par les ressources. Ces deux types sont évoqués dans la littérature comme suit :

- 1) **Statique** : Cette catégorie consiste à réaliser la planification des tâches en avance. Elle suppose une connaissance à priori de l'ensemble des paramètres et contraintes de production. Elle est vaguement utilisée même si l'ensemble des informations relatives à la production est rarement connue à l'avance. Ce Planning prévisionnel est souvent adapté (réajusté) avant ou pendant l'exécution [119],[92],[94]. Néanmoins l'ordonnancement statique est un moyen stratégique et nécessaire sur dans les Plannings qui peuvent susciter plusieurs heures voire jours de calcul. Le terme anglais "*Offline*" pour hors-ligne, est aussi utilisé pour qualifier ce type d'ordonnancement.
- 2) **Dynamique** : L'environnement de production est généralement régi par des paramètres incontrôlés (modéliser par des variables aléatoires) tels que des Pannes des ressources, perturbations, etc. Ce qui rend la fidélité des Plannings prévisionnels difficiles [94]. De ce fait les Plannings en temps réel permettent une adaptation dynamique aux contraintes du système de production [123], [15]. On qualifie ce type d'ordonnancement de "*temps réel*" ou de "*online*" (Mot anglais signifiant littéralement en ligne). L'ordonnancement en temps réel est également utilisé pour une replanification ou une adaptation de l'ordonnancement statique. Cette replanification peut être refaite intégralement ou juste sur la partie du système affecté. L'intérêt de ce second cas, est montré dans [42].

2.5.2 Méthodes d'ordonnement de tâches

Dans un système de production, il est nécessaire d'indiquer les paramètres conduisant à une bonne réalisation des tâches et fonctionnement quasi-optimal du système. L'hexamètre de Quintilien³, permet une connaissance de ces paramètres [131] :

- (i) **Quoi** : Détermine l'ensemble des tâches à réaliser sur un produit.
- (ii) **Qui** : Toutes les ressources (Machines, robots, opérateurs et autres équipements) qui sont nécessaires à la réalisation de la tâche.
- (iii) **Où** : La ou les localisation(s) concernée(s) par la tâche. L'endroit où elle aura lieu.
- (iv) **Quand** : Définit toutes les données relatives au temps. Dates de début et fin potentielles, durée de la tâche.
- (iv) **Comment** : Indique les procédures ou la manière dont la tâche doit se réaliser.

En nous basant sur les travaux de thèse de Zambrano [140], on peut catégoriser les approches d'ordonnement en six groupes synthétisées dans le tableau 2.2 (Le terme *transporteurs* dans le même tableau, fait référence aux ressources de transports). Toutes ces approches peuvent être implémentées en utilisant un système multi-agent (MAS) [122], système holonique (HMS) [37] et/ou un système axés sur le produit (PD, pour Product Driven) [44].

D'autre part des méthodes d'optimisation sont généralement associées à ces approches d'ordonnement. Ces méthodes sont : Les heuristiques (Algorithme glouton, recherche en faisceau, etc.); Méta-heuristiques (Algorithme génétique, recherche tabou, etc.); Méthodes optimales ou quasi-optimales (Programmation linéaire et non-linéaire, Programmation dynamique, etc.); Théorie du contrôle [100].

Comme nous le verrons dans la partie dédiée à l'ordonnement des tâches de transport aux manipulateurs mobiles (Section 2.5.4), les approches basées sur le CNET sont les plus utilisées parmi les méthodes présentées dans le tableau 2.2. Par ailleurs, un intérêt particulier sera accordé également aux approches basées sur les champs potentiels. Même si elles sont moins utilisées que celles du CNET, elles présentent la faculté de résoudre à la fois, deux problèmes majeurs (ordonnement et navigation) qui concernent les ressources de transports. Ainsi ces deux approches sont plus détaillées dans les sous-sections qui suivent.

2.5.2.1 Champs potentiels

L'utilisation des champs potentiels sur des ressources de transport avait pour but initial de mieux gérer la navigation et également de coordonner les déplacements de plusieurs robots par l'évitement d'obstacles [13]. Des

3. <https://fr.wikipedia.org/wiki/qqoqccp>, consulté le 03 juillet 2020

TABLE 2.2 – Approches d’ordonnancement

Approches	Fonctions	Type	Éléments concernés
Blackboard [15],[24],[36]	Collecte de données, Partage d’information, Evaluation et selection d’actions (tâches)	Centralisé	Machines, Transporteurs, Produits
Contract Net (CNET)	Enchères, négociation	Distribué	Machines, Transporteurs
Règles de Dispatching et de priorité [51], [40]	Séquecement des tâches, Gestion des files d’attente	Centralisé	Machines, Produits
Champs potentiels	Principes d’attraction et de répulsion	Distribué	Machines, Transporteurs, Produits
Algorithmes de tirage(Kanban) [154]	Gestion de production et stock, Juste à temps	Centralisé	Machines, Produits
Mécanismes de stigmergie [41]	Communication indirecte, auto-organisation	Distribué	Machines, Transporteurs

champs attractifs sont émis par les points objectifs, pour faire converger les ressources, tandis que l’environnement et d’autres ressources émettent des champs répulsifs pour faire éviter les obstacles aux ressources [171].

L’assignation des tâches se base sur les mêmes fondements que pour la navigation. Des produits ou machines ayant besoin d’un service de transport, émettent des champs attractifs que les transporteurs en mesure de répondre, captent et s’y rendent. Tout en utilisant les mêmes types de champs répulsifs que la navigation.

Cette faculté double, permettant aux approches basées sur les champs potentiels de pouvoir résoudre à la fois des problèmes de navigation et d’ordonnancement, en fait leur efficacité [170].

L’intensité de réception du champs est inversement proportionnelle à la distance par rapport à la source émettrice. Ce rapport entre ces deux grandeurs est représenté sur la figure 2.10. Pour les distances proches de la source émettrice du champs, la réception du champs est au maximum. L’augmentation de cette distance, se traduisant par un éloignement de la source fait décroître l’intensité de réception du champs. Ce critère basé sur la distance permet d’assigner les tâches aux ressources [61].

Une priorité est quelques fois attribuée aux tâches, pour aider les ressources dans l’ordonnancement des tâches. Ainsi lorsque plusieurs champs correspondant chacun à des tâches distinctes sont émises, les ressources seront plus

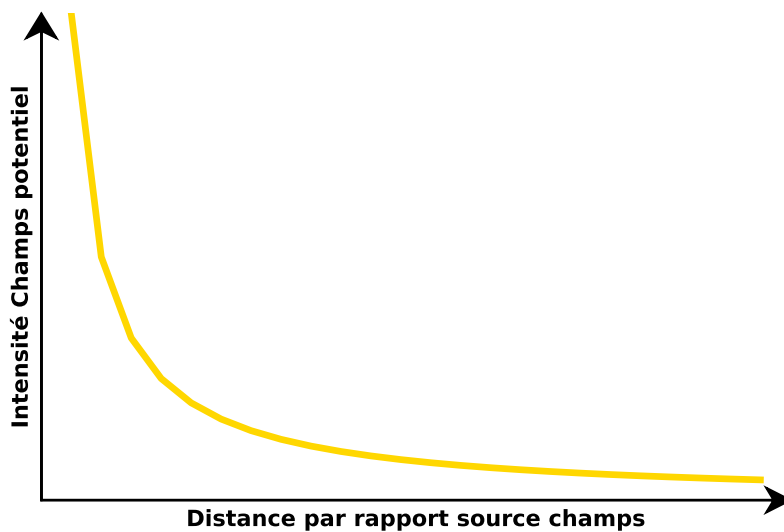


FIGURE 2.10 – Variation du champs potentiel par rapport à la distance

attirées par les tâches ayant des priorités élevées.

Les champs sont émis dans toutes les directions, ce qui peut offrir un système de déplacement flexible pour les entités ayant une autonomie de navigation.

Cependant, l'application des approches basées sur les champs potentiels nécessite une configuration matérielle (Pour l'émission et la réception des champs) et environnementale (Pour la propagation des flux de champs) particulière et bien adéquate. Pour que les champs puissent se propager efficacement, l'entrepôt ou l'atelier de production doit être dégagé. Les obstacles comme les murs, et autres machines imposantes peuvent constituer un frein dans l'application de cette méthode [61]. Ce qui est souvent le cas dans les usines et les entrepôts.

Weyns et al. [182] évoquent aussi les problèmes de minima locaux qui peuvent subvenir avec ce type d'approche. Une ressource peut rester bloquer à un minimum local par l'assignation de tâches sur base des champs potentiels (Exemple : le cas du cul-de-sac), vu que les critères de décisions ne concernent qu'elle.

2.5.2.2 Contract Net

Le concept du CNET (Contract Net) a été introduit dans les années 1980 par Reid Smith [158]. Selon ce dernier, le CNET facilite le contrôle distribué pour l'exécution des tâches coopératives grâce à un protocole de négociation entre agents. La négociation de contrat s'apparente à une forme de discussion établit par deux ou plusieurs entités en vue de parvenir à un accord commun sur une action. Dans le CNET, cette négociation se fait au travers d'un mécanisme d'enchères [158].

Le mécanisme d'enchères, découle des mêmes principes que ceux utilisés

dans le monde des affaires ou socio-économique, désigné par l'expression : *Vente aux enchères* [106].



FIGURE 2.11 – Image⁴ illustrative d'une vente aux enchères

Selon Wikipédia⁵ une vente aux enchères est : "un mécanisme structuré de négociation par lequel un agent économique (vendeur ou acheteur) met en concurrence plusieurs autres agents, durant un laps de temps limité, en vue de leur vendre ou de leur acheter des biens ou des services". L'origine des enchères remonte à l'antiquité avec la traite des personnes⁶ et autres objets précieux.

Ainsi, l'objectif des ventes aux enchères dans le milieu socio-économique est de créer une compétition entre les amateurs d'un bien, afin de faire croître progressivement son prix. Le bien sera attribué au plus offrant.

Dans l'environnement "académico-industriel", les méthodes utilisant le CNET sont réputées efficaces pour résoudre les problèmes d'allocation de ressources [170]. Dans ces méthodes, deux types d'agents sont utilisés : (1) L'agent Manager en charge de l'offre de tâche et de la surveillance son exécution ; (2) Les agents contractuel(s) responsables de l'exécution de la tâche ou de proposer des enchères à l'offre émise.

La figure 2.12 montre le mécanisme d'enchères avec trois étapes distinctes. Lorsqu'un agent Manager annonce un appel d'offre pour une tâche (étape 1 sur la figure 2.12), chaque agent contractuel capable d'y répondre, évalue l'offre puis fait une proposition (étape 2) [67]. Ensuite, l'agent Manager attribue la tâche et l'agent contractuel gagnant et le Manager coordonnent pour effectuer la tâche (étape 3). Trois situations peuvent se présenter dans les FMS [175], décrivant le type de relation entre Managers et contractuels :

- (a) Les Managers et les contractuels sont tous des ressources essayant de répartir la charge de travail.

4. Image créée par macrovector, licence gratuite, <https://www.freepik.com/search?dates=any&format=search&page=1&query=auction&selection=1&sort=popular> consulté le 15 juillet 2020

5. <https://fr.wikipedia.org/wiki/Ench%C3%A8re>, consulté le 09 juillet 2020

6. Originellement utilisé pour vendre des butins et prisonniers de guerre par les hébreux puis les grecs et les romains. Le terme enchère, est dérivé du latin carus qui signifie cher, <https://www.vlimmo.fr/actualites?SvvhJaIigd>, consulté le 09 juillet 2020

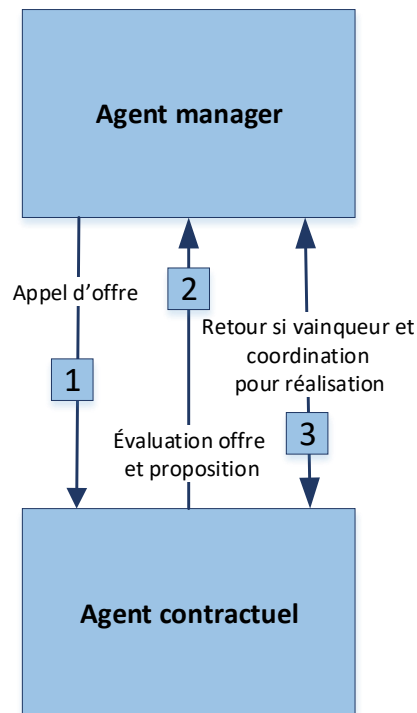


FIGURE 2.12 – Mécanisme d'enchères

- (b) Les Managers sont des composants (produits) proposant des tâches et les contractuels des ressources enchérissant pour obtenir ces tâches.
- (c) Les Managers sont des ressources offrant des services et les contractuels sont des composants utilisant ces services.

Malgré leurs nombreux avantages comme l'efficacité dans l'assignation de tâches, la distributivité et la coopération entre agents, les approches basées sur le CNET disposent aussi de quelques inconvénients. Une des limites des enchères, reste leur réalisation d'une manière séquentielle. Dans les approches classiques, plusieurs tâches ne peuvent être envoyées pour la proposition des enchères. Une ressource qui enchérit pour une tâche, ne peut faire d'offre pour une autre sans que la première n'ait été attribuée [182]. En d'autres termes, l'évaluation de ces performances pour répondre à la seconde tâche pourrait être erronée car la première non assignée ne peut être prise en compte. A titre illustratif, on peut supposer un acheteur disposant d'un montant déterminé à des enchères. Il enrichit pour l'achat d'une voiture à hauteur de tout l'argent à sa disposition. Au même moment, une autre voiture est présentée dans une autre vente aux enchères en ligne. Cet acheteur ne peut logiquement pas prétendre à cette seconde vente aux enchères, car la totalité de ces ressources financières sont bloquées par la première. Une alternative à ce problème est de permettre aux ressources d'effectuer des offres pour plusieurs tâches non assignées, et leur laisser le choix d'accepter ou de refuser la tâche si elle leur est assignée par le Manager des enchères [84].

Le concept des enchères combinatoires (*Combinatorial Auction* en anglais) est une solution à ce problème. Avec cette approche, une ressource peut soumettre plusieurs enchères à un même Manager pour plusieurs tâches à la fois [52].

2.5.3 Problèmes de Job-Shop avec et sans flexibilité de routage

Le problème d'ordonnancement d'un Job-Shop sans flexibilité de routage (*Job Shop Scheduling Problem* en anglais, **JSP**) consiste à rechercher une assignation optimale d'un ensemble de Jobs à un ensemble de machines tout en minimisant ou maximisant la/les fonction(s) objectif(s) du système [43], [152]. La modélisation d'un tel système prend en considération un certain nombre de contraintes et de suppositions.

Dans les problèmes d'ordonnancement de Job-Shop flexible (*Flexible Job-Shop Scheduling Problem* en anglais, **FJSP**) une flexibilité de routage est en plus intégrée [51]. Ainsi, chaque tâche d'un Job dispose de plusieurs machines capables de la réaliser. Chaque tâche a dans ses paramètres d'entrée l'ensemble des machines pouvant l'exécuter avec les différentes performances associées aux machines. La flexibilité de production est alors accrue avec ce type d'atelier même si la complexité du système augmente aussi [43].

2.5.4 Ordonnancement des tâches de transport

Plusieurs types de tâches sont disponibles dans les FMS comme : la production (Assemblage, Usinage, Inspection, ...), la maintenance, le stockage des produits ou encore le transport [169].

Dans la littérature, les travaux dédiés à l'ordonnancement des tâches de transport traitent surtout de l'ordonnancement couplant les tâches de transport et production et les tâches de transport (seules) sous des contraintes multiples [79].

2.5.4.1 Tâche de transport

Le transport d'un produit en industrie consiste à son chargement à un point source, à son transport vers un point destination puis à son déchargement [182], [95]. Ainsi une tâche de transport est constituée de trois opérations principales comme l'indique la figure 2.13 [61] :

- Le **chargement** : Il consiste à saisir un objet puis à le déposer sur une plateforme responsable de son transfert. Il peut être effectué par toute ressource du type prise et dépose (*Pick and Place* en anglais) comme les bras manipulateurs et également les humains grâce à leurs membres supérieurs.
- Le **transport** : Suit la phase de chargement du produit, comme le nom l'indique, il permet de transporter dans des conditions de sécurité le produit vers sa destination qui peut être une machine ou un point de stockage. Seules les ressources mobiles sont capables de réaliser cette

opération. Les chariots élévateurs et les AGVs sont largement utilisés pour le transport, bien que les humains peuvent le faire aussi.

- Le **déchargement** : Clôt la tâche de transport, par la dépose du produit au point de destination. Cette opération dispose des mêmes caractéristiques que celle du chargement.

La tâche dans sa globalité peut être réalisée par un manipulateur mobile grâce à son AGV et son bras manipulateur embarqué. On peut remarquer aussi, sur la figure 2.13 qu'un opérateur peut avoir la même fonction qu'un AGV, un bras et/ou un manipulateur mobile dans le cadre la tâche de transport. Seulement, une différence existe au niveau des performances robotiques et humaines.

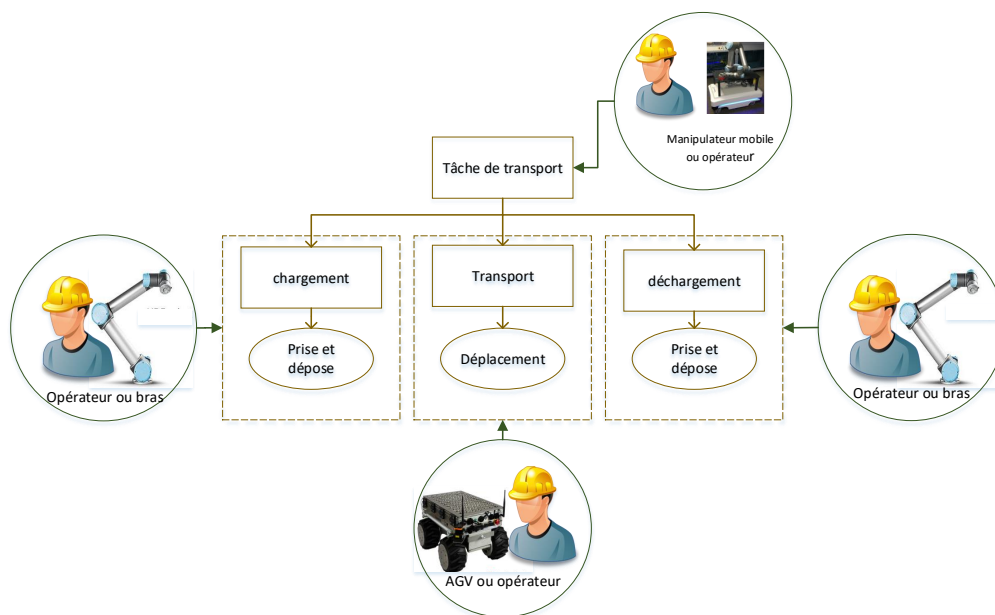


FIGURE 2.13 – Opérations d'une tâche de transport

2.5.4.2 Ordonnancement des tâches de transport indépendamment des tâches de production

Elle consiste à séquencer et assigner des tâches de transport à des ressources de transport.

Dans [182], une approche pour l'attribution de tâches aux AGVs dans un environnement dynamique basé sur les champs potentiels est proposé. Les champs sont émis par les agents de transport, puis les AGVs vides sont attirés par ces champs. Les AGVs émettent à leur tour des champs répulsifs pour éviter les collisions lorsque plusieurs sont attirés par le même champ.

Dans [189], une autre approche basée sur les champs potentiels, dans laquelle chaque ressource peut offrir un ou plusieurs services, est proposée. Chaque service ayant son propre champ potentiel, une même ressource peut émettre plusieurs champs potentiels correspondant aux services qu'elle offre. Les produits sont alors attirés par le champ correspondant au service dont ils

ont besoin.

Dans [103], un algorithme d'attribution des tâches à un système multi-robot basé sur un mécanisme hétérarchique d'enchères en tenant compte de plusieurs contraintes (délais de tâches, capacité des robots,...) a été développé. Chaque robot a un gain pour l'exécution d'une tâche. Ce gain constitue son enchère et permet la soumission à ses voisins lorsqu'une tâche est proposée.

Wawerla et al. [181] ont expérimenté deux méthodes d'assignation de tâches sur un système multi-robots dans un contexte de transport. La première méthode consiste à utiliser un ordonnanceur central disposant d'information sur tout l'atelier pour effectuer l'allocation aux robots. Dans la seconde, chaque robot de transport prend ses décisions localement en fonction des données qu'il a pu collecter. Le chargement et le déchargement des palettes à transporter sont évoqués dans le travail réalisé, mais les temps et la consommation d'énergie associés à ces opérations ne sont pas explicitement pris en compte. Dans [178] une approche permettant le dimensionnement du nombre d'AGVs, l'assignation des tâches aux AGVs, et la gestion des trajectoires pour l'évitement des collisions est proposée et testée. L'assignation des tâches se fait par deux méthodes : Un algorithme basé sur un heuristique utilisant un indice de vieillissement et une méta-heuristique basée sur la recherche tabou. Une étude comparative a permis de montrer que les résultats obtenus pour la recherche tabou sur l'assignation de tâches de seulement 5 véhicules, sont meilleures que celles obtenues pour 6 véhicules dans le premier algorithme. Ce qui démontre l'importance d'avoir un bon algorithme qui distribue d'une façon optimisée les tâches aux véhicules, en dehors des bonnes gestions de trajectoires, et de l'évitement des collisions.

Fauadi et al. [52] ont expérimenté une architecture entièrement distribuée pour traiter l'attribution des tâches aux AGVs. Les enchères sont utilisées pour l'attribution des tâches, avec les machines comme Managers. Cela rend l'approche décentralisée car chaque machine gère l'attribution des tâches en fonction des produits dont elle en a besoin ou des besoins de la machine voisine. L'AGV capable de proposer le temps de transport minimal remporte l'enchère.

Ces approches souffrent des mêmes défauts que les systèmes de contrôle totalement hétérarchiques : Absence de décision globale.

Hernandez-Martinez et al. [69] proposent la modélisation d'un atelier constitué de diverses machines, de points de stockage et d'assemblage, ainsi que des AGVs via les réseaux de Petri. Le transport des produits étant au cœur, l'assignation des tâches aux AGVs est simplement basée sur la plus courte distance par rapport au point de prise du produit. D'autre part, un système de contrôle permet, aux AGVs de coopérer pour réaliser une tâche, tout en gardant une distance de sécurité entre eux, évitant ainsi les collisions. Tous ces travaux, limitent la tâche de transport au seul rôle de l'AGV donc à l'opération de transport.

2.5.4.3 Ordonnancement tâches de transport et de production

Ce type d'ordonnancement consiste à séquencer et allouer simultanément aux ressources de production (machines) et de transport (manipulateurs mobiles, AGVs, bras) des tâches [79].

Dans [35] une approche d'assignation double (transport et production) des manipulateurs mobiles (AGV + Bras) et machines est présentée. Le travail se distingue par la prise en compte de l'activité du bras embarqué sur l'AGV dans l'assignation de tâche. En effet, le bras est utilisé pour des tâches de production telles que l'assemblage et l'inspection.

Néanmoins dans ce travail, les bras embraqués sur les AGVs sont considérés comme des ressources de production. De ce fait la tâche de transport consiste au seul rôle de l'AGV, même si les temps de chargement et déchargement sont inclus dans le temps de déplacement de l'AGV. Une formulation à base de MILP (*Mixed-Integer Linear Programming*) est utilisée pour le problème, et un GA (Algorithme Génétique) combiné à de la recherche tabou permet sa résolution.

De plus l'approche, considère que l'atelier est constitué d'un ensemble de ressources homogènes, tous les AGVs disposent d'un bras. Ainsi, le problème ne prend pas en compte le cas où l'AGV se présente seul, sans bras (Cas des systèmes de transport hétérogènes).

Reddy et al. [139], ont proposé un GA hybride pour la résolution des problèmes d'ordonnancement combinant à la fois les tâches de transport et de production. Un GA est utilisé pour l'allocation des tâches aux machines, tandis qu'une méthode heuristique permet l'assignation des tâches de transport aux véhicules. Cette dernière méthode est intégrée dans le GA, et est sollicitée pendant la phase d'évaluation de Fitness. L'algorithme hybride, vise à minimiser à la fois la durée moyenne des tâches, le retard moyen et le Makespan.

Lee [95], a étudié le problème d'allocation des machines tout en considérant les tâches de transport. Des contraintes telles que la capacité et les temps liés aux ressources de transport sont prises en compte. Les systèmes manufacturiers sont généralement constitués de deux types de transfert : Un premier qui concerne le transfert des produits en cours de fabrication (assuré par des AGVs ou convoyeurs); Un second qui consiste à la livraison des produits finis vers les clients ou les entrepôts de stockage (assuré par des camions dédiés). L'approche proposé par Lee, permet de prendre en considération ces deux cas de transport de produits.

Raman et al. [135] ont fourni un modèle déterministe d'ordonnancement sur la même thématique, en formulant le problème sous forme de programmation linéaire en nombres entiers (ILP, *Integer Linear Programming*). Leur modèle, suppose que les AGVs retournent au point de stockage des matières premières et produits finis après chaque tâche. Cette supposition, peut néanmoins engendrer un grand nombre de déplacement à vide des AGVs. Ce qui impacte l'optimalité des solutions.

Babu et al. [58], en proposant une méta-heuristique basée sur l'évolution différentielle ont résolu le problème d'ordonnancement de tâches à des machines et AGVs dans un FMS. L'algorithme visant à commencer les tâches dès que possible, permet une minimisation du Makespan. L'ensemble des AGVs pris en compte dans le problème dispose de performances identiques (Vitesse, charge maximale) et seul un produit est transporté à la fois. De plus, les machines disposent de stocks, qui permet aux AGVs de pouvoir libérer les produits même si la machine est prise au moment de leur arrivée.

Tous ces travaux [35], [139], [95], [135], [58] traitent des JSP. L'ordonnancement simultané des tâches de transport et de production est abordée dans les travaux de recherche en utilisant les JSP, dans la grande majorité des travaux. L'allocation de machines à des tâches de production dans un Job-Shop avec flexibilité de routage (FJSP) est déjà connue comme un problème du type NP-Hard [43],[54]. L'ajout des tâches de transport à ce problème, ne peut ainsi qu'accroître la complexité.

Peu de travaux comme [72], [90] ont adressé le problème d'ordonnancement simultané (production et transport) en introduisant la flexibilité de routage. J'ai contribué dans le développement d'une approche (IT2P) décrit dans le chapitre 4 permettant également de faire face à ce type de problème.

2.5.5 Mécanisme d'ordonnancement distribué

Les systèmes de décisions centralisées des ressources de transport s'avèrent efficaces pour leur supervision car permet un accès facile à toutes les informations souhaitées pour décider de "**Qui**" fait "**Quoi**" et "**Comment**", en ce qui concerne les chemins à prendre [182]. Seulement, la question du **comment** peut être bien gérée localement par chaque entité de transport grâce son autonomie. En plus la décision centralisée présente les mêmes faiblesses que les architectures de supervision centralisée (Manques de flexibilité et réactivité, etc.).

A l'instar des architectures de contrôle, les mécanismes de décisions et d'ordonnancement distribués ont montré leur efficacité. L'intérêt de l'utilisation d'une méthode d'optimisation et planification distribuée est montrée dans [144], à travers plusieurs bancs de test comparant les méthodes centralisées et décentralisées.

Une architecture composée de trois niveaux pour l'ordonnancement et le contrôle des FMS est proposée par Bongaerts et al. dans [19]. Le plus haut niveau est chargé de l'ordonnancement statique (en avance), qu'il effectue en fonction des retours du deuxième niveau, périodiquement ou en cas de perturbations. Les données des niveaux les plus élevés sont prises comme des conseils et non des ordres par le niveau inférieur. Ce dernier considère l'état dynamique de l'atelier et la planification si nécessaire. Cette méthode permet un ordonnancement à plusieurs niveaux, dans lequel la décision finale est prise par les entités du niveau le plus bas.

Une application de l'architecture développée par Bongaerts est réalisée dans [81] dans un entrepôt de stockage de produit cosmétique, qui déploie des robots. Dans cette application, une autorité absolue est donnée soit au niveau du milieu ou au plus haut niveau contrairement à l'approche de Bongaerts dans laquelle, la décision finale revient aux entités du plus bas niveau. Le

plus haut niveau gère un premier ordonnancement de tâches (optimisé grâce à un GA) que le second niveau récupère. Un second ordonnancement basé sur un système d'enchères est réalisé par les ressources du bas niveau. Si l'ordonnancement réalisé par les entités du plus bas niveau est différent de celui du plus haut niveau, alors le niveau du milieu intervient pour donner une décision finale car il dispose d'une meilleure vue sur le système. Ce mécanisme hybride d'ordonnancement permet une répartition de la charge de travail et une meilleure réactivité du système face aux perturbations.

Dans [118] l'interaction entre un système centralisé d'ordonnancement et un MES (*Manufacturing Execution System*) décentralisé (basé sur l'architecture PROSA) chargé d'exécuter cet ordonnancement est présentée. Cette collaboration est rendue possible grâce à un échange d'information entre les deux systèmes. Le MES en cas de perturbations (Panne d'une ressource, annulation d'une opération...) peut renvoyer une information au système centralisé d'ordonnancement pour que celui-ci délivre un nouveau Planning. Néanmoins un délai existe entre le signalement du MES et le calcul du nouvel ordonnancement. Pour pallier à ce problème, le MES est doté d'une fenêtre de prédiction avec un temps de détection qui doit être inférieur ou égal au temps nécessaire au système d'ordonnancement pour délivrer un nouveau Planning.

Panurak et al. [125] ont étudié un système d'ordonnancement à plusieurs niveaux. Un ordonnanceur central disposant d'une vue globale sur tout le système crée d'abord un ordonnancement non détaillé qui est transmis aux entités de bas niveaux. Ces dernières négocient alors un ordonnancement plus détaillé à leur niveau et peuvent aussi dévier légèrement de l'ordonnancement initial.

Ces méthodes d'ordonnancement multi-niveaux permettent d'obtenir une charge de travail relativement équitable parmi les ressources disponibles et résistantes aux perturbations. Cependant, elles nécessitent une bonne synchronisation entre les différents niveaux de décision pour détecter quand un nouvel ordonnancement doit être effectué par qui et à quel niveau. De plus, comme la décision finale se résume toujours à un niveau, il peut arriver que la planification d'autres niveaux ne soit pas d'une grande utilité.

L'approche DISCOFT développée dans le chapitre 5, se base sur ces types d'ordonnancement avec des décisions distribuées, tout en éliminant l'inconvénient d'effectuer plusieurs fois des ordonnancements et à différents niveaux. Ainsi, dans DISCOFT l'ordonnancement est effectué suite à une collaboration entre deux niveaux, pour répartir la charge de travail à l'instar des méthodes de décision multi-niveaux.

Tous ces mécanismes de décision distribuée sont rendus possibles par l'apparition des éléments autonomes et intelligents. La section suivante explore ces entités, notamment celles utilisées pour les tâches de transport.

2.6 Entités autonomes de transport

Les robots mobiles sont des entités autonomes capables de se déplacer d'un point **A** vers un point **B** pour remplir une mission qui leur a été attribuée. Ils sont utilisés dans plusieurs domaines tels que l'industrie, l'exploration spatiale [63], les opérations militaires, le médical, etc. Lorsque le champs d'application est l'industrie, on parle plutôt d'AGV. En ajoutant un bras robotisé sur l'AGV pour les opérations de chargement et déchargement, il devient un manipulateur mobile [147].

2.6.1 Les véhicules à guidage automatique

Les AGVs sont des équipements de manutention (*Handling Materials* en anglais) à roues pouvant se déplacer sans conducteur et disposant de batteries rechargeables [79], [75]. Ils sont largement utilisés dans les systèmes industriels et de logistique moderne pour leur faculté à accroître la flexibilité des systèmes [102].

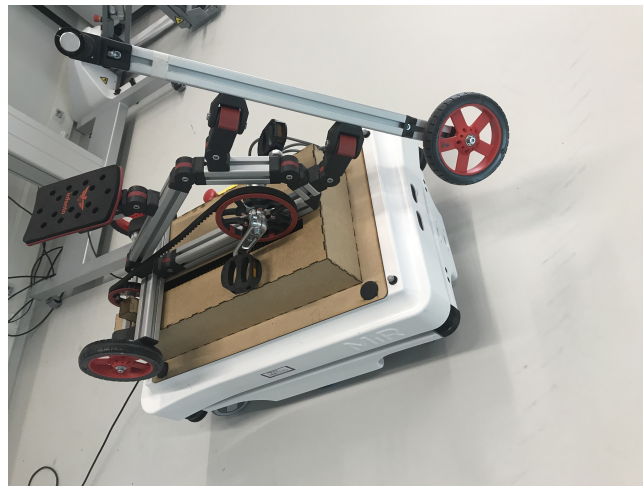


FIGURE 2.14 – Exemple d'AGV (Mir100) adapté et utilisé pour le transport de tricycle

Un système d'AGV est un système de transport industriel entièrement autonome qui déploie plusieurs AGVs pour le déplacement des produits [182]. Une des fonctionnalités clés des AGVs, reste leur mécanisme de navigation. Ce dernier étant l'organe qui régit l'évitement d'obstacles et la création de nouvelles trajectoires dans les systèmes totalement autonomes et le garant d'une livraison à bon port des produits.

2.6.2 Systèmes de navigation

Les systèmes de navigation des AGVs ont beaucoup évolué depuis leur mise en place dans les années 1950 [102]. On peut néanmoins les catégoriser en deux grands groupes : Les systèmes de navigation assistées (semi-autonome) et totalement autonome.

2.6.2.1 Navigation semi-autonome

Dans ce type de navigation les AGVs suivent des trajectoires prédéfinies. Des chemins sont définis à même le sol pour assister et contraindre les AGVs dans leurs déplacements. Les chemins peuvent être dessinés par un système électrique filaire émettant des ondes électromagnétiques, un mécanisme de rails à l'instar des trains ou tout simplement des ondes magnétiques. Les AGVs détectent et se dirigent grâce ces chemins prédéfinis [104]. La création de nouvelles trajectoires est assez compliquée et ne peut pas se faire d'une manière dynamique. En effet, elle implique d'étudier et définir de nouveaux chemins d'une façon optimale pour rentabiliser les coûts et les déplacements des AGVs. Ainsi, ils sont peu adaptés pour les AGVs.

2.6.2.2 Navigation autonome

Grâce à la navigation autonome, les AGVs peuvent évoluer assez librement dans leur environnement de travail grâce à une connaissance du milieu (cartographie, utilisation de capteurs embarqués) ou une assistance souple (lignes de peintures détectées à l'aide de caméra embarquée) [102]. L'utilisation des lignes de peintures pour la détection visuelle par l'AGV offre plus de flexibilité que les systèmes filaires et les autres mécanismes de la navigation semi-autonome. De plus le principe reste moins coûteux.

Des capteurs comme le Laser permettent aux AGVs qui en sont équipés de naviguer dans un environnement doté de réflecteurs facilitant la localisation [104].

Lorsqu'aucun aménagement ou infrastructure externe n'est prévu pour aider dans le système de navigation de l'AGV, celui-ci doit continuellement se localiser d'une façon très précise dans l'environnement de travail. La navigation Slam se faisant à l'aide des Lidars est un exemple courant de système de navigation autonome [76]. La figure 2.15 montre un robot mobile de type Robotnik, utilisant la navigation Slam grâce à des Lidars embarqués à l'avant et arrière du robot. Plusieurs caméras dont dispose le robot peuvent également être utilisées en supplément dans la navigation. La technologie des systèmes de localisation par satellite (GPS) est aussi une possibilité de navigation [192].

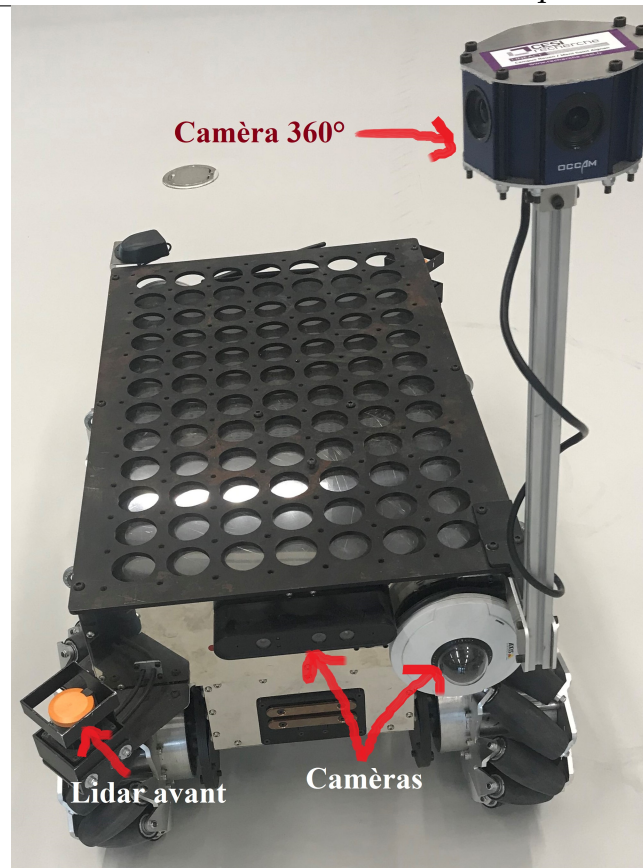


FIGURE 2.15 – AGV Robotnik

2.6.3 Les bras manipulateurs

Aussi appelés bras robotisés ou robotiques, ils sont inspirés de ceux (bras) des humains et utilisés pour plusieurs applications. A l'instar des bras humains, les bras manipulateurs sont caractérisés par un certain nombre d'axes appelé degré de liberté qui varie en fonction des robots. Ils disposent d'une terminaison ou outils (équivalent de la main chez l'humain) qui définit leur fonction [172].

Des outils à ventouse, magnétique, pince ou doigts permettent leur utilisation industrielle pour les applications de manutention (chargement et déchargement).



FIGURE 2.16 – Trois bras manipulateurs Niryo One du laboratoire Lineact-Cesi de Rouen

2.6.4 Les manipulateurs mobiles

Les manipulateurs mobiles sont des bases mobiles (AGVs) sur lesquelles sont montées des bras robotisés [107]. Le terme de bras mobile est aussi utilisé si la fonction principale est de conférer une mobilité aux bras. Ces robots bénéficient ainsi des fonctionnalités de deux entités, leur permettant de pouvoir effectuer les tâches de transport intégralement.

Les manipulateurs mobiles disposent aussi d'une certaine modularité, garantissant l'assemblage ou le désassemblage des deux robots [53].

2.6.4.1 Modularité des manipulateurs mobiles

La modularité peut être définie comme la faculté d'un objet à être transformée ou modifiée en jouant avec ses éléments. Les manipulateurs mobiles modulables, peuvent avoir deux configurations possibles comme le montre la figure 2.17.

2.6.4.1.1 Première configuration : Deux entités jointes physiquement

Dans cette configuration, l'AGV est joint physiquement au bras et les deux robots sont considérés comme une entité unique [165]. Une entité est l'extension de l'autre : Le bras est l'extension de l'AGV si le transport est dominant ; L'AGV est l'extension du bras dans le cas du bras mobile.

La communication se fait par l'intermédiaire d'un seul contrôleur qui comporte plusieurs actionneurs (moteurs du robot mobile et du bras robotique). Cette configuration est assez répandue, surtout lorsqu'on adopte une supervision centralisée. D'autre part l'utilisation des manipulateurs mobiles

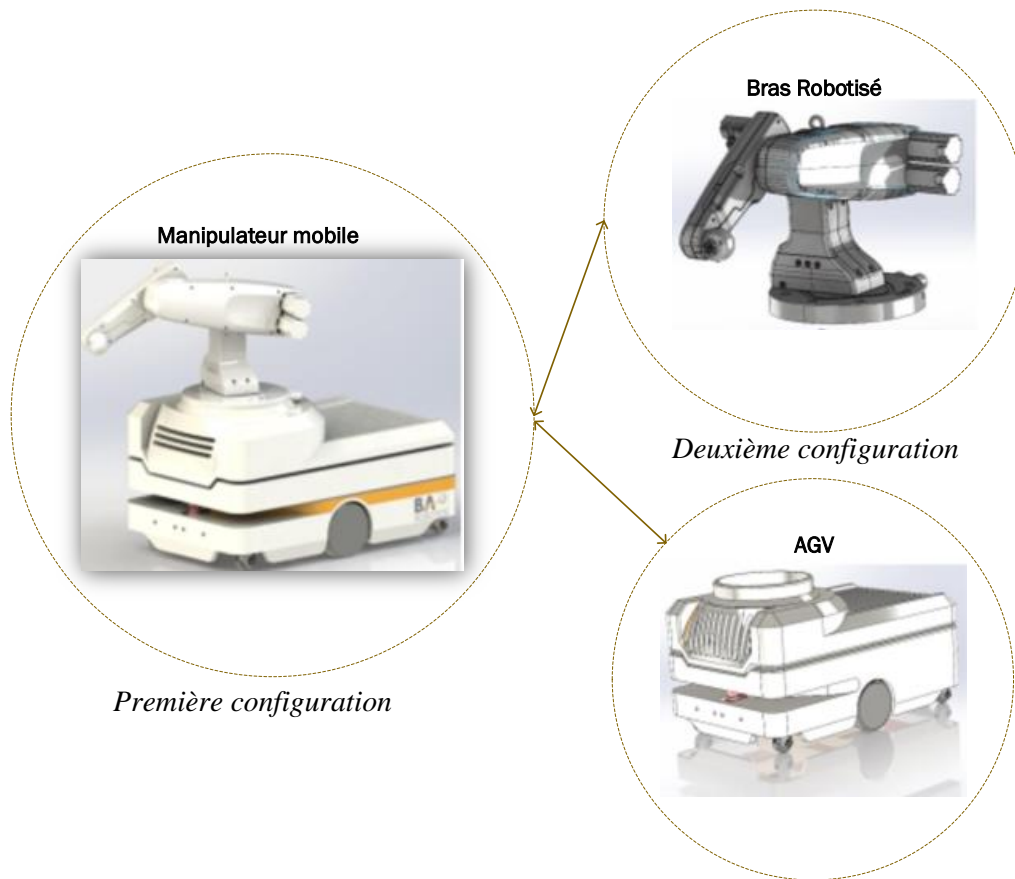


FIGURE 2.17 – Manipulateur mobile modulaire

comme une seule entité réduit la complexité de contrôle, car le système est homogène (un seul type d'entité) [35].

Par contre, l'utilisation de ce type de configuration limite la flexibilité du système car les deux entités ne peuvent travailler qu'ensemble, c'est-à-dire que si l'une d'entre elles est impliquée dans une tâche, l'autre l'est également puisqu'elles sont physiquement dépendantes. Des architectures distribuées sont développées afin de surmonter ce problème, où les robots sont considérés comme deux entités distinctes [38]. Dans ce cas, les deux entités restent jointes physiquement mais séparées d'une façon logicielle. Ainsi, la communication avec le superviseur ou d'autres entités peut se faire individuellement (Chaque entité communique directement sans passer par l'autre) ou d'une façon jumelée au travers d'un organe dédié garant de la synchronisation entre les deux robots. Cette solution d'organe de synchronisation permet aussi de réduire la complexité du mécanisme et la quantité de données échangée avec le milieu extérieur au manipulateur [53].

2.6.4.1.2 Deuxième configuration : Deux entités séparées

Dans ce cas, la base mobile est physiquement séparée du bras et les deux entités (base mobile et bras) sont considérées comme deux entités autonomes différentes, dont chacune possède sa propre unité de décision [71]. Cette configuration est avantageuse quand l'AGV n'a pas besoin des services du

bras, comme dans les cas où le chargement et déchargement des pièces doivent être faits par un humain ou un bras plus efficace et mieux adapté.

Cette seconde configuration est aussi utile dans le cas où le bras doit effectuer des actions de prise et dépose sur place pendant un certain temps [61]. Ainsi, il n'aura pas besoin d'immobiliser l'AGV qui pourrait faire d'autres actions en parallèle. La communication entre les deux entités est régie par les règles de communication entre entités indépendantes.

2.6.5 Les robots collaboratifs

2.6.5.1 Chronologie

Les robots collaboratifs ou cobots (Contraction de l'anglais *CO*llaborative *ro*BOTS) sont une tendance des nouvelles évolutions de la robotique. La cobotique est définie comme une discipline qui traite des relations et collaborations entre l'humain et le robot [27]. Les cobots ont été introduits par Colgate, Wannasuphprasit et Peshking en 1996 [29]. Le groupe *General Motors*, fut l'un des premiers industriels à utiliser un robot collaboratif pour assister les opérateurs à porter des portières de voiture en 1998 [16],[180]. Dix ans plus tard (en 2008), la commercialisation des cobots à commencer avec la compagnie *Universal Robots*⁷ qui a proposé le cobot UR5. Ce dernier est capable de porter une charge de cinq kilogrammes (5kg) et de s'arrêter en cas de contact physique étranger grâce à un retour d'effort. Plusieurs autres fabricants de robots industriels ont suivi la tendance comme les groupes leaders *FANUC*, *KUKA*, *ABB* en proposant respectivement CR35ia, LBR iiwa puis KMR iiwa⁸ et YuMi.

Depuis, des cobots de tout type ont été développés : Des AGVs (MiR100 par exemple, en image sur la figure 2.14), des bras manipulateurs (Comme les Niryo One, image figure 2.16) et des manipulateurs mobiles.

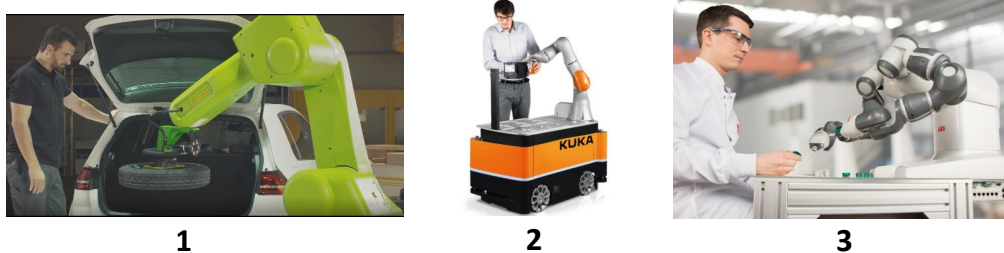


FIGURE 2.18 – Robots collaboratifs : 1 CR35ia de FANUC⁹, 2 KMR iiwa de KUKA¹⁰ et 3 YuMi de ABB¹¹

7. <https://www.universal-robots.com/fr/>, consulté le 21 juillet 2020

8. Le KMR iiwa est le manipulateur mobile qui inclut le bras collaboratif sept axes LBR iiwa

2.6.5.2 Atouts et différences avec la robotique industrielle classique

Le principe de base de la cobotique ou robotique collaborative est de faire travailler des machines dotées d'intelligence entre elles ou avec des humains sans aucun danger [27].

A la différence des robots industriels classiques, les cobots ont été conçus pour travailler avec les humains et non spécifiquement pour eux. Le principe de la cobotique industrielle est de combiner les performances du robot (précision, force, . . .) avec la dextérité et l'expérience de l'humain dans la réalisation des tâches pénibles, répétitives ou délicates (manipuler en sécurité des pièces fragiles, chaudes ou encombrantes) [83].

La cobotique industrielle se distingue également par la suppression des cages dans lesquelles étaient logées les robots classiques par mesure de sécurité pour les humains aux alentours. Ces installations sont plutôt onéreuses et requièrent un suivi régulier. L'intervention ou la maintenance à proximité ou sur ce genre de robots nécessite un arrêt complet de leur fonctionnement. Avec la robotique collaborative, cette contrainte disparaît, le robot étant équipé de capteurs qui scrutent son environnement en permanence. Dès qu'il détecte un humain ou un obstacle, il s'immobilise ou évite l'obstacle par précaution [16].

Une autre distinction des robots collaboratifs est leur facilité de programmation. Alors que les robots industriels classiques demandent des compétences avancées en programmation informatique, les robots collaboratifs se distinguent par leurs facultés d'apprentissage. Ils disposent également d'une capacité à s'adapter rapidement aux changements dans la production, et faire des actions non programmées pour sortir d'une situation de blocage. Ils sont plus flexibles pour pouvoir s'impliquer dans différentes applications [83].

2.6.6 Les opérateurs humains

L'humain joue un rôle non négligeable dans l'industrie 4.0 [150]. En plus des rôles traditionnels de production et de prise de décision à tout niveau, il peut être sollicité pour des tâches de maintenance [65] (Comme pour la maintenance à l'aide de la Réalité Augmentée RA, voir figure 2.20), la surveillance des procédés ou ressources, l'interprétation et l'envoi des ordres de production, l'assemblage, le transport, etc.

L'étude ergonomique du travail industriel des opérateurs, les conséquences liées à la charge supplémentaire de travail, le stress, les nouvelles responsabilités et toutes les méthodes visant à améliorer les performances des humains dans les systèmes de production ont été largement traités dans la littérature [5], [112], [113], [124]. Ainsi, cette partie est dédiée à la nécessité, l'apport et la participation des opérateurs dans la réalisation des tâches de transport.

9. Source image FANUC : <https://www.fanuc.eu/be/fr/robots/page-de-filtre-des-robots/des-robots-collaboratifs>, consulté le 21 juillet 2020

10. Source image KUKA : <https://www.visit.sindex.ch/fr/KUKA-KMR-iiwa,p1501009>, consulté le 21 juillet 2020

11. Source image ABB : <https://new.abb.com/products/robotics/fr/robots-industriels/yumi>, consulté le 21 juillet 2020

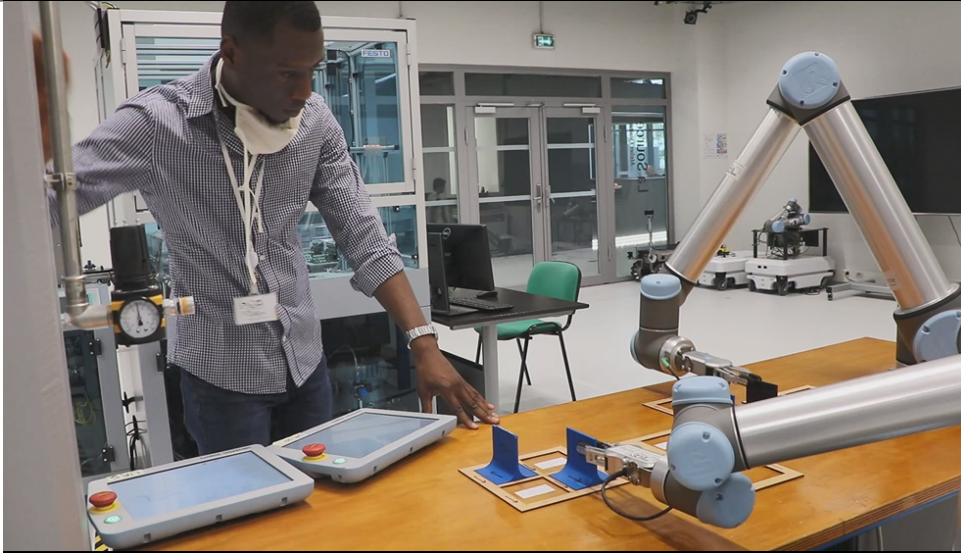


FIGURE 2.19 – Robots collaboratifs travaillant à proximité d'un humain au laboratoire Lineact-CESI de Rouen

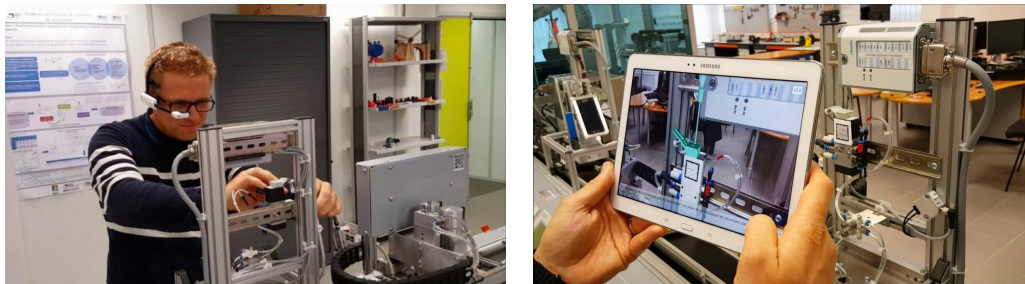


FIGURE 2.20 – Opérateur utilisant la RA pour des tâches de maintenance à l'aide d'une lunette (Image de gauche) et d'une tablette (Image de droite)

2.6.6.1 Exemple d'action nécessitant une intervention humaine : Démonstrateur du futur

L'humain peut effectuer les tâches de transport efficacement grâce à ces différents membres. Il dispose aussi d'une faculté native d'apprentissage et ainsi se déplacer avec des objets tout en évitant avec aisance les obstacles qui pourraient se présenter devant lui et générer de nouveaux chemins. En outre la réalisation de certaines opérations de transport industriel, est plus économique avec un humain qu'un robot à cause de leur rareté ou complexité. Par exemple, sur le démonstrateur du futur du laboratoire LINEACT qui permet d'assembler des téléphones, le module de stockage des produits finis est verrouillé. Quatre serrures devant être fermées à clé quand la ligne de production est en marche, maintiennent les deux trappes du module de stockage des produits finis. La récupération (Chargement sur un transporteur) des produits (téléphones assemblés) est à la charge d'un opérateur, figure 2.21.

Pour se faire, l'opérateur doit s'assurer d'abord que la chaîne est à l'arrêt (**Étape 1**). Ensuite, il part chercher la clé ouvrant le module (Image de



FIGURE 2.21 – Processus de chargement des produits finis du démonstrateur du futur au Laboratoire LINEACT-CESI

la clé sur la deuxième image de la figure 2.21) qui se situe dans un arrangement à quelques mètres de la chaîne d'assemblage (Étape 2). Une fois la clé en main, il revient vers la chaîne et identifie les quatre serrures (Étape 3). L'opérateur doit par la suite, exercer une pression sur la trappe qu'il souhaite ouvrir à l'aide d'une main et introduire la clé dans la serrure (Image du milieu sur la figure 2.21) à l'aide de l'autre main pour l'ouvrir (Étape 4). La quatrième étape doit être répétée pour les trois serrures restantes, en plaçant à chaque fois la main exerçant la pression juste à côté de la serrure pour créer un meilleur jeu.

Dans l'Étape 5, l'opérateur ouvre les deux trappes déverrouillées, de telle sorte qu'il puisse avoir accès aux produits. C'est seulement en ce moment, qu'il saisit les produits et les charge sur le transporteur (Troisième image sur la figure 2.21), sachant que certains produits sont stockés à une hauteur de 1,60m (Étape 6). L'opérateur referme les trappes par le procédé de l'étape 4, une fois qu'il a fini de retirer toutes les pièces. Il doit également remettre la clé dans son arrangement pour faciliter les utilisations futures.

Cette description donnée, démontre la complexité de certains procédés industriels. La robotisation de l'ensemble des étapes conduisant à la récupération des produits depuis le module de stockage du démonstrateur et à leur chargement sur un transporteur nécessite d'avoir :

- Un robot doté d'un système de vision, pour détecter la position des serrures, des produits et de la clé. La vision permettra aussi d'insérer la clé avec précision dans le trou de la serrure ayant une circonférence de l'ordre du millimètre.
- Un robot à deux bras pour les actions simultanées de pression et d'ouverture des serrures. Utiliser deux robots peut être aussi une solution.
- Il faut un outil adapté au robot pour pouvoir maintenir fermement la clé et la faire tourner dans la serrure.

- La hauteur de l'emplacement de certaines pièces (1.60m) est aussi un paramètre non négligeable, qui nécessite d'avoir un robot avec un rayon d'action conséquent.
- La clé doit être à proximité de la chaîne, et placer de telle sorte que le robot puisse la saisir. Il est impératif que son emplacement ne change jamais autrement le robot serait bloqué.
- etc.

La robotisation de cette tâche d'ouverture et de chargement peut être onéreuse en coût de conception et en temps de développement algorithmique. Pourtant, cette tâche est assez basique pour un opérateur. Ce qui prouve, le rôle indispensable de l'opérateur même dans l'industrie 4.0.

2.6.6.2 Moyens de communication des opérateurs

La communication entre les agents humain et robotique relève de la discipline **IHM** (Interfaces ou Interactions Homme-Machine). L'IHM se définit comme l'ensemble des moyens et outils mis en oeuvre pour qu'un humain puisse interagir, communiquer ou contrôler une machine [26].

Divers outils technologiques permettent aux opérateurs d'interagir avec les robots et de communiquer avec le poste de supervision [148]. Ces outils vont de simples gadgets connectés portables (Tablette, téléphone, casque, etc.), au poste de contrôle utilisant des ordinateurs fixes. Dans le contexte du

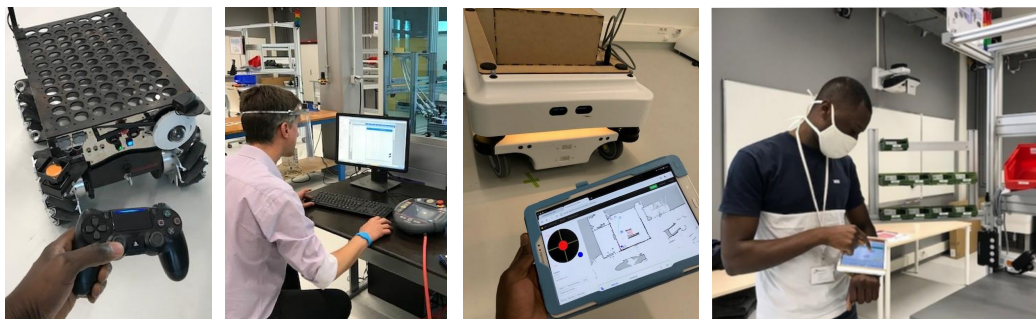


FIGURE 2.22 – Quelques outils d'interaction, de gauche à droite : Un Joystick¹² connecté en Bluetooth, un opérateur communiquant grâce un ordinateur, une tablette connectée en Wifi et un opérateur avec une tablette reliée à sa main, connectée en Wifi

transport, un outil portable, facilement maniable et léger serait plus adapté. L'image la plus à droite de la figure 2.22, illustre un moyen pragmatique de communication pour les opérateurs. Une tablette simple est adaptée au bras de l'opérateur, lui permettant d'échanger avec les autres entités de l'atelier tout en ayant une mobilité complète et une disponibilité de ses deux mains pour pouvoir intervenir dans les tâches de transport.

12. Manette de jeu Playstation, utilisée également pour le contrôle de certains robots

2.7 Coalitions

Les coalitions sont utilisées dans plusieurs domaines : la politique avec les coalitions entre les partis [21], la médecine avec les équipes de garde et pour autres travaux collaboratifs [93], les forces militaires formées par des armées de plusieurs pays [183], l'industrie, etc.

Le transport industriel des produits est fait à travers un ensemble de tâches suscitant la coopération et la collaboration de plusieurs ressources pour sa réalisation. Des coalitions qui sont alors une forme d'entente entre plusieurs ressources peuvent être formées pour mener à bien la mission de transport.

Au sein d'une coalition des règles internes doivent être suivies pour garantir la synchronisation et la coordination entre les différentes ressources : C'est la collaboration [89]. Par ailleurs, les membres d'une équipe doivent également prêter une attention particulière aux règles externes, pour permettre la cohabitation avec d'autres ressources ou équipes : C'est la coopération [73]. Par exemple deux AGVs appartenant à des coalitions différentes, coopèrent pour s'éviter mutuellement en cas de croisement.

La compréhension du principe de coalition passe nécessairement par une meilleure appréhension de la coopération et la collaboration entre ressources. La suite de cette section apporte plus d'éclaircissement sur ces deux termes et survole également plusieurs travaux ayant utilisé et démontré l'intérêt de faire travailler en équipe des ressources de même nature (coalition entre robots) et de natures différentes (coalition entre humains et robots).

2.7.1 Coopération

Les études basées sur la coopération ont pour but, de développer des techniques de raisonnement et des connaissances dont les entités ont besoin pour la résolution de problèmes complexes. La coopération est nécessaire dans le cas où une entité ne dispose pas suffisamment d'expertise ou d'informations pour venir à bout d'un problème [48]. Durfee et al. [48], trouve un objectif générique quadruple à la coopération :

- (i) Accroître le temps d'accomplissement des tâches à travers le parallélisme.
- (ii) Augmenter le nombre de tâches réalisables par le partage d'expertise, information, matériels physiques, etc.
- (iii) Diminuer l'interférence entre les tâches en évitant les interactions nuisibles.
- (iv) Augmenter la probabilité d'achèvement des tâches en entreprenant plusieurs méthodes pour les effectuer.

2.7.1.1 Différence entre coopération et collaboration

La coopération et la collaboration sont souvent utilisés à tort comme étant identiques. Néanmoins, une différence existe entre les deux termes que ce soit sur les plans socio-politique, économique et industriel. Cependant, les

deux termes sont utilisés à juste titre pour évoquer un travail collectif, de deux ou plusieurs entités ou organismes travaillant ensemble.

Shirley Hord [73], considère la coopération et la collaboration comme deux concepts différents mais pouvant être complémentaires des fois. Selon lui, on parle de coopération lorsque deux individus ou organisation trouvent un accord mutuel, pour des intérêts pouvant être différents. Par ailleurs, la collaboration intervient lorsque le travail vise à atteindre un but commun. Il illustre ses propos par plusieurs exemples sociaux. Comme deux équipes de football se mettant en accord pour jouer un match tout en respectant les règles de jeu, ce qui serait une coopération entre les deux équipes. Chaque équipe dispose de son objectif qui est de remporter la partie. Les membres d'une même équipe collaborent pour atteindre ce but final qui est la victoire. Le mariage peut aussi être considéré comme une forme de collaboration car le couple vise un ou plusieurs objectifs communs. La crise pandémique de la COVID-19 a nécessité la coopération et la collaboration de plusieurs gouvernements du monde pour réduire la propagation et gérer au mieux les personnes affectées [86].

Dans son article dédié à la différence entre collaboration et coopération, Olga Kozar [89] situe la différence entre les deux concepts au niveau des objectifs attendus. Dans une activité collaborative, deux ou plusieurs éléments forment un groupe pour effectuer un travail parvenant ainsi à un objectif commun grâce à un ensemble d'apports individuels. Elle implique, une interaction, négociation, coordination et discussion entre les éléments collaborant. La coopération quant à elle consiste à travailler ensemble indépendamment des objectifs de chacun.

Le travail coopératif est une tâche divisible, dont chaque participant est responsable de la résolution d'une portion du problème tandis que la collaboration est une coordination des efforts de chaque participant pour la résolution d'un problème [141].

2.7.2 Coalitions robotiques

Une coalition robotique se définit comme un groupe de robots, combinant leurs capacités afin d'exécuter une tâche que chacun d'entre eux ne peut pas exécuter seul [156]. Il est en général difficile pour un seul robot d'exécuter une tâche d'une nature complexe, d'où la nécessité de former des coalitions. La figure 2.23 illustre la nécessité et le principe de formation de coalition par des robots. Trois tâches consistant à former des couleurs (jaune, blanche, marron) doivent être faites. Des robots pouvant chacun générer une des trois couleurs primaires (rouge, vert et bleu) sont en charge de la réalisation des tâches. Aucun des robots ne dispose des capacités requises pour effectuer une tâche intégralement. Néanmoins, en combinant leurs expertises d'une façon proportionnée (Chaque couleur sur la figure 2.23 correspond à une quantité de couleur bien définie, illustrée par la taille des robots sur la figure), ils pourront y parvenir. C'est ainsi que les robots rouge et vert ont formé une équipe pour accomplir la tâche 1, qui consiste à réaliser du jaune. Il est quand même important de noter que les robots formant les trois couleurs primaires peuvent réaliser l'ensemble des tâches présentées et même au delà, tant qu'elles consisteront à former des couleurs. Ainsi dans un système

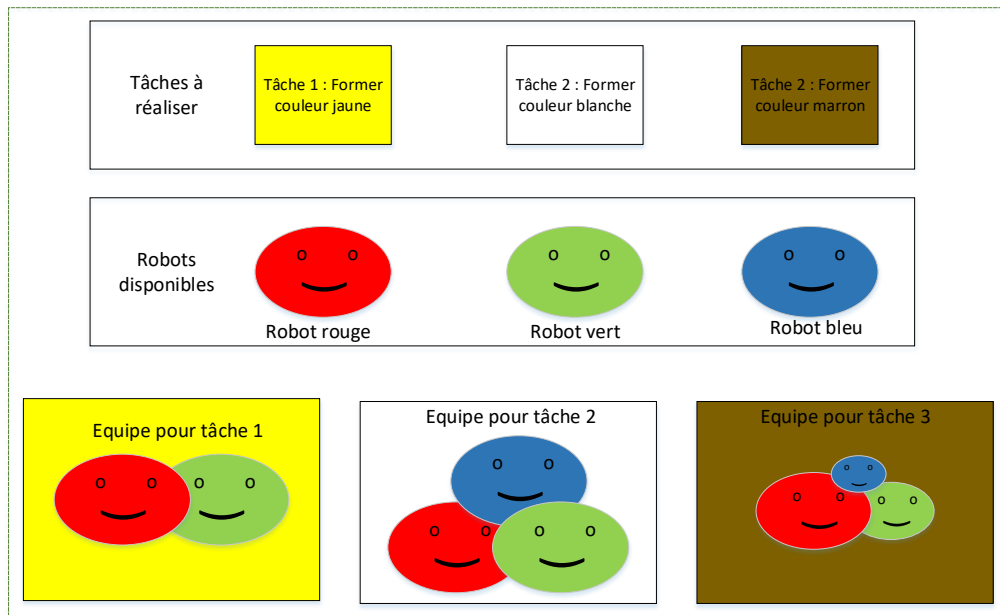


FIGURE 2.23 – Exemple de formation de coalitions pour la réalisation de tâches complexes

de coalition les exigences des tâches doivent être comprises dans la fusion des compétences d'un groupe ou de l'ensemble des robots. Toute tâche qui ne remplit pas cette condition ne pourra pas être réalisée, à moins d'ajouter des robots pouvant satisfaire le besoin. De nombreux travaux de recherche ont porté sur la formation de coalitions de robots, connue comme un problème de type NP-Hard [146], [49].

Gerkey et al. [56] ont analysé le problème d'assignation de tâche multi-robots en proposant une répartition et taxonomie des problèmes traités. Ainsi les problèmes récurrents sont :

- **Assignation de tâches uniques ou multiples** : Spécifie le nombre de tâches que chaque robot est capable de gérer simultanément (Une ou plusieurs tâches à la fois).
- **Allocation mono ou multi-robots par tâche** : Détermine la complexité des tâches, en spécifiant si chaque tâche est capable d'être traitée par un seul robot ou si un groupe de robots est exigé.
- **Assignation simultanée ou étendue dans le temps** : Le premier cas concerne l'ordonnancement dynamique des robots et intervient lorsqu'une visibilité des robots et des tâches sur le long terme est inconnue. Dans la seconde, l'ensemble des paramètres sont connus.

Le problème de formation de coalition robotique intervient lorsqu'on souhaite assigner une tâche unique à un groupe de robots que ce soit en temps réel ou en statique, ce qui correspond à des combinaisons de plusieurs problèmes de la répartition de Gerkey et al. [56].

Dans [137], un problème de coalition multi-robots est abordé et une approche centralisée, basée sur un GA, est utilisée pour le résoudre. Un robot central,

qui a des connaissances sur les autres robots, dirige la coordination entre les robots pour effectuer des tâches. Toutefois, le modèle reste générique car il ne précise pas la nature et la complexité des tâches. Dans [185], une approche hybride pour la coalition de plusieurs robots est utilisée pour les tâches délicates (missions de sauvetage ou environnement dangereux). La méthode vise à maximiser le nombre de tâches accomplies par la coalition tout en minimisant l'énergie consommée.

Un autre algorithme de coalition multi-robots dans le même domaine d'application (missions de sauvetage) est présenté dans [114]. Plusieurs contraintes comme le niveau de batterie, les exigences de capacité des tâches et les contraintes de localisation sont prises en compte dans la formation de coalition. Par exemple, en fonction de la durée possible d'une tâche, un niveau de batterie minimum peut être exigé pour intégrer le groupe (ou la coalition). Le nombre de robots pouvant faire partie d'une coalition est également limité. Cette restriction pourrait être un inconvénient pour d'autres domaines comme l'industrie, qui peut comprendre des tâches nécessitant l'intervention de plusieurs ressources en même temps. Ainsi, cette contrainte restreint l'utilisation de l'approche dans d'autres domaines.

Dans [49] un algorithme permettant d'assigner des tâches à un ensemble de robots mobiles par le principe de coalition est présenté. La fonction objective vise à minimiser la distance parcourue par les robots. Le travail se base sur la supposition que le nombre robots mobiles doit toujours être supérieur au nombre de tâches à assigner. Aussi le nombre de robots requis dans une coalition est prédéfini, et la somme des nombres de robots requis par chaque tâche équivaut au nombre total de robots. En d'autres termes, chaque robot est utilisé dans une seule tâche. Ceci conduit à une utilisation non optimisée des robots car pour des tâches similaires, un robot aurait pu intervenir dans plusieurs coalitions.

2.7.3 Coalitions humains robots

Le concept fondamental d'une coalition est d'unir des compétences pour augmenter les chances de réalisation d'une tâche ou d'exceller dans l'exécution. Les robots en combinant leurs fonctionnalités parviennent à réaliser efficacement un grand nombre de tâches complexes. En effet les robots sont capables de se déplacer rapidement, de porter de lourdes charges et sont immunisés contre la fatigue. Par contre, les humains en plus d'être nativement multi-tâches disposent d'une faculté anticipatoire, à évoluer dans des environnements non-structurés ou développer une conscience face à certaines situations [138]. De plus, comme démontré dans la section 2.6.6, il est souvent indispensable de faire appel aux opérateurs humains dans la réalisation de certaines tâches.

C'est dans ce cadre que l'expertise des humains est combinée à celle des robots pour une meilleure exécution des tâches complexes.

Smirnov et al. [155] se sont intéressés au problème de formation de coalition incluant les humains et les robots. Ainsi, une approche basée sur un mécanisme ontologique d'échange d'information permet la création de coalition entre robots mobiles uniquement ou entre robots mobiles et opérateurs humains. Les coalitions sont formées au besoin, et la ressource initiatrice de la

coalition exprime son besoin dans une sorte de tableau de bord commun. La ressource pouvant satisfaire le besoin forme d'abord une coalition virtuelle avec l'initiatrice (La ressource ayant spécifié le besoin) dans le tableau de bord avant de descendre vers une coalition physique au niveau des ressources. Bien que le domaine d'application de l'approche reste générique, un exemple d'évitement d'obstacle est donné.

Sangseok et al. [186] présentent une étude sur les coalitions formées par les humains et les robots, notamment sur l'impact d'avoir des sous-groupes dans une même équipe dû à la préférence des humains pour certains robots.

Tang et al. [164] partent du principe que les robots et les humains disposent de capacités différentes et peuvent donc former des coalitions et échanger des informations pour accomplir des tâches. Ainsi, ils se sont focalisés sur certaines difficultés que peuvent engendrer ce travail collaboratif, tout en proposant des solutions permettant d'y remédier.

Le travail présenté dans [99] présente une approche distribuée permettant l'intégration de robots industriels pour travailler en équipe avec des opérateurs humains pour des tâches d'assemblage. L'approche permet de prendre en considération divers paramètres tels que : l'attention de l'opérateur, la prise en compte des compétences de l'opérateur par le robot pour une meilleure gestion des tâches et plusieurs données collectées par des capteurs pour une représentation de l'environnement de travail. En effet, la connaissance des compétences des membres d'une même équipe est importante pour mener à bien les tâches.

[142] a évoqué la complexité pour un humain non expert à travailler en équipe avec un groupe de robots. Ainsi, un agent intelligent fournissant des conseils aux opérateurs humains collaborant avec des robots est développé.

2.8 Conclusion

La supervision des ateliers de production flexibles passe nécessairement par le choix d'une architecture de contrôle et la mise en place d'une approche permettant l'ordonnancement de tâches. Le type d'une architecture définit l'interaction et les responsabilités à accorder aux éléments de l'atelier, pendant que l'ordonnancement permet de rendre leur travail efficace et optimisé. Par rapport aux types d'architecture, les approches distribuées se sont avérées plus efficaces que les centralisées grâce à leur réactivité et l'apport en flexibilité qu'elles offrent au système.

Plusieurs approches sont utilisées pour l'ordonnancement des tâches dans les ateliers de production flexibles, mais les approches du CNET principalement basées sur les enchères ont montré leur efficacité, raison pour laquelle elles sont largement utilisées.

Par ailleurs l'ordonnancement se fait également de plusieurs manières. Une seule entité peut être en charge de la sélection et de l'attribution des tâches, tout comme elles (sélection et attribution) peuvent provenir de plusieurs entités appartenant à des niveaux de hiérarchie différents. La seconde méthode, permettant l'ordonnancement distribué est plus adapté aux entités dotées d'intelligence car elles sont capables de prendre des décisions de haut niveau.

Le concept des coalitions permet la résolution collective de tâches complexes, difficilement ou non solvable d'une façon individuelle.

Les coalitions industrielles peuvent être constituées par des entités identiques comme un ensemble de robots ou des éléments différents à l'instar des robots et des humains grâce à la collaboration. Celle-ci permet de faire converger les intérêts de plusieurs éléments grâce au travail participatif, contrairement à la coopération qui permet également un travail collectif mais sans forcément avoir un objectif commun.

La démonstration du rôle crucial de l'humain dans la réalisation de certaines actions complexes comme celles du transport, motive notre choix d'utiliser le second type de coalition qui englobent à la fois robots et opérateurs humains. En définitive cette recherche bibliographique a permis d'étudier les travaux portant sur les entités de transport, l'ordonnancement des tâches de transport et le contrôle des FMS en général. Cette étude a ainsi permis un positionnement du travail effectué dans cette thèse et de montrer son originalité par le dégagement des limites des travaux existants.

Les chapitres suivants, se focalisent sur les développements effectués.

Chapitre 3

Architecture générique de supervision

3.1 Introduction

Une architecture de supervision constituée de plusieurs modules permettant, une gestion quasi-optimale et flexible des ressources autonomes dans le concept de l'Industrie 4.0 est détaillée dans ce chapitre. Cette architecture s'appuie sur les fondements de l'agent générique du Système Manufacturier Dynamiquement Intégré, DIMS (Dynamically Integrated Manufacturing System, en anglais). Ainsi, dans les sections suivantes, la motivation d'adapter une architecture existante plutôt que de développer une nouvelle est justifiée. En outre les spécificités de l'architecture DIMS montrant notre intérêt sont explicitées. Par la suite, les limites de DIMS et la contribution qu'apporte ce travail de thèse sont indiquées. Enfin, un atelier de production flexible est modélisé pour pouvoir tester l'efficacité de DIMS améliorée.

3.2 Architecture proposée

3.2.1 Évolution des architectures

Plusieurs architectures ont été développées pour régir la supervision des entités dans le contexte industriel (Voir des exemples dans la section 2.4.3 du chapitre 2). Notre choix, s'est porté sur l'adaptation et l'amélioration d'une architecture existante plutôt que le développement d'une nouvelle. Ce choix se justifie par la perpétuité des architectures existantes et offre une chance à celles-ci d'aller au delà de la phase théorique et d'atteindre la phase expérimentale réelle en laboratoire, voire même en industrie [128], [77], [121]. Cette philosophie de la pérennisation des méthodes développées est partagée par un grand nombre de chercheurs de la communauté. Les architectures sont ainsi modifiées et améliorées avec un changement de nom quelques fois pour suivre l'adaptation.

L'architecture holonique PROSA constituée de trois holons basiques (Holon produit, Holon ressource et Holon ordre) et développée par Van Brussel et al. en 1998 dans [175] est un exemple standard de ce concept. Cette architecture a été reprise et modifiée par Paulo Leitao et al. en 2005 dans [98]. Un quatrième holon (holon superviseur) a donc été ajouté aux trois basiques de PROSA et un nouveau nom a été proposé : ADACOR. Cette approche à son

tour a subi plusieurs évolutions en 2007 dans [97], puis en 2011 dans [10] pour aboutir à ADACOR² en 2015 dans [11].

Par ailleurs l'architecture PROSIS, émanant aussi de PROSA se distingue par l'association d'un holon produit à chaque produit en cours de fabrication dans l'atelier. Ces holons contiennent ainsi des données sur le produit. Patrick Pujo et al. ont mis en oeuvre PROSIS en 2009 dans [130].

Une des dernières applications de PROSA est faite par Valckenaers en 2020 [174]. Dans ce travail, PROSA évolue vers ARTI (pour une meilleure adaptation aux systèmes intelligents et connectés tels que les jumeaux numériques. L'architecture ATLAS proposée dans [50] en 1997 pour le contrôle en temps réel des robots, a évolué vers RSA [63] en 2006. Cette dernière, comporte plus de fonctionnalités (coordination de flotte, téléprésence, etc.) et se veut plus complète.

Anoskie dans ses travaux de thèse en 2005 [4] [191], a étudié l'architecture DIMS basée sur une structure MAS en mettant l'accent sur la composition des agents et sur l'intégration d'une simulation distribuée pour répondre rapidement et efficacement aux changements dans les FMS. Dans [190], Zhang a amélioré DIMS en intégrant un algorithme d'assignation de tâches basées sur les enchères et utilisant un GA comme méthode d'optimisation. La même méthode a été approfondie et explicitée par Naihui dans ces travaux de thèse [67] en 2011.

Nous conduirons une adaptation de l'approche au contexte des entités robotiques de l'industrie 4.0 et une amélioration dans l'approche d'assignation intégrée dans DIMS. De ce fait, l'originalité de cette approche justifiant son choix sera mise en avant mais aussi les limites pour mettre en lumière notre contribution à l'approche d'assignation des tâches.

3.2.2 Architecture des agents dans DIMS

L'architecture hybride des agents DIMS, offre une configuration fractale aux agents qui la constitue [143]. En effet, chaque agent est créé sur la base d'une structure générique comprenant une unité de contrôle, un environnement commun. Un agent peut disposer aussi d'agents enfants qu'on appellera sous-agent ayant la même configuration que le parent grâce à la structure fractale. Cette configuration particulière constitue un des atouts de l'architecture car elle permet une gestion facile de l'ensemble des agents, vu qu'ils sont tous des instances d'un agent générique. En outre la composition de ce dernier offre une panoplie de fonctionnalités visible sur la figure 3.1, permettant l'adaptation et l'utilisation de la méthode dans plusieurs domaines. Certaines fonctions sont détaillées dans les sections qui suivent.

3.2.2.1 Unité de Contrôle

L'unité de contrôle d'un agent contient quatre parties clés incluant un dispositif de communication inter-agents, une base de connaissances (KB, pour Knowledge Base), un contrôleur et un composant de simulation.

La communication entre les agents est assurée grâce au dispositif de communication d'inter-agents. Le module de communication d'un agent est basé

sur le langage de manipulation de la requête de connaissances KQML.

Trois types de connaissances sont enregistrés dans le KB :

- (1) Connaissance des missions : Spécifie les actions et les objectifs locaux d'un agent.
- (2) Résolution de problèmes : Ensemble des méthodes que peut utiliser l'agent pour s'acquitter de ses missions.
- (3) Connaissances coopératives : Elles constituent un ensemble de règles de décision implanté sur l'agent garantissant la négociation et les protocoles permettant les coordinations entre agents.

Le contrôleur permet de déterminer les comportements d'exécution ou les décisions des agents en fonction des connaissances contenues dans le KB, des données existantes dans la base de données (dans l'environnement commun) et des connaissances extraites des messages entrants.

La composante de simulation permet grâce à un modèle de réseau de Petri (PN, pour Petri Net en anglais) de générer et d'évaluer dynamiquement de nouvelles configurations. Cette composante se base sur une simulation d'événements discrets et peut garantir aussi un inter-échange entre agents pour une simulation distribuée.

3.2.2.2 Environnement commun

L'environnement commun d'un agent se compose de trois parties principales :

- (i) Une base de données : Comporte les différentes informations relatives à l'agent et à son environnement.
- (ii) Manager d'agent : Gère la création de nouveaux agents ou la suppression des existants.
- (iii) Facilitateur : Simplifie et fluidifie la communication entre un agent parent et ses enfants.

3.2.3 Application de DIMS à la robotique

L'adaptation faite de l'architecture DIMS dans ce travail de thèse, permet d'avoir un agent pour le superviseur et des agents pour les ressources de transports, pouvant avoir autant de sous-agents qu'ils en ont besoin. Les agents et sous-agents dédiés aux robots sont génériques (Les mêmes structures et compositions pour tous) et se distinguent de l'agent du superviseur que d'un module : Le superviseur dispose d'un module de Monitoring (Surveillance de l'atelier) contrairement aux robots qui disposent d'un contrôleur (organe d'exécution des tâches).

Des modules apparaissant dans l'approche originale (figure 3.1) tels que le facilitateur et la base de connaissances KB sont absentes de l'adaptation dans la figure 3.2. En effet les ressources de transport utilisées dans l'Industrie 4.0

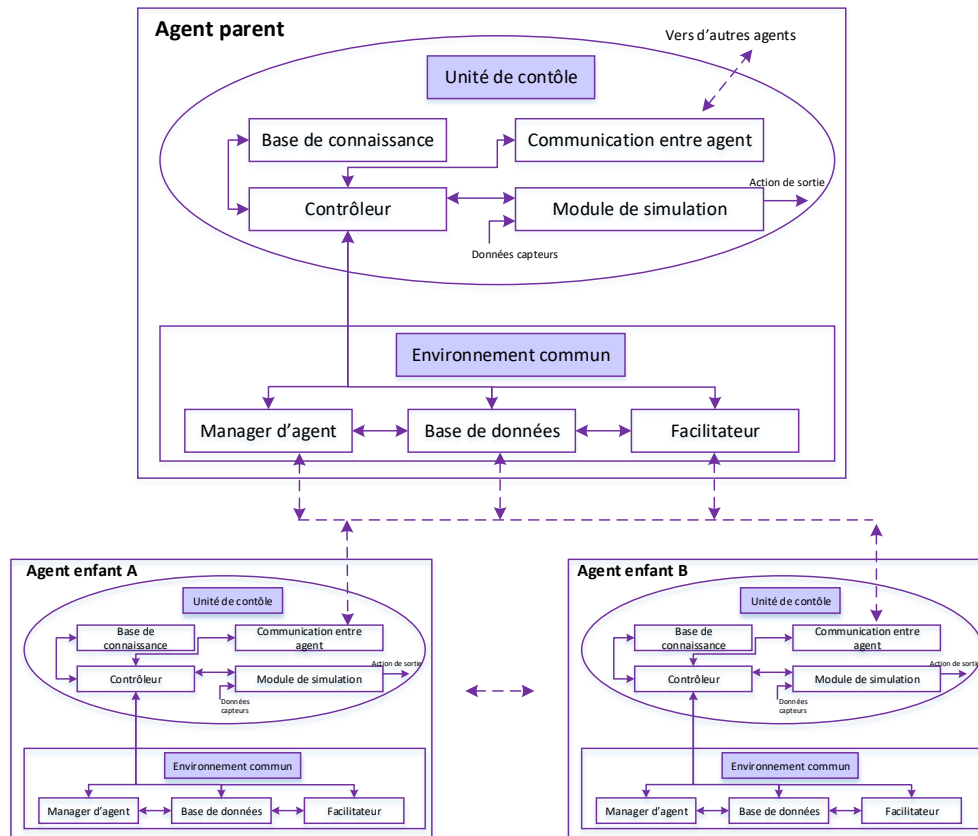


FIGURE 3.1 – Architecture DIMS proposée par Zhang

disposent basiquement ces organes dès leur conception.

On remarque également la fusion de l'unité de contrôle et de l'environnement commun et la création d'une base de donnée centrale permettant la sauvegarde des données de l'ensemble des entités de l'atelier. Sept (7) différents modules sont utilisés au total dans cette nouvelle adaptation ainsi que treize (13) types de liens régissant les relations entre les différents modules.

3.2.3.1 Module d'interaction

L'unité d'interaction est l'organe qui régit toutes les communications extérieures d'un agent. Deux types d'échanges sont possibles : L'échange direct à travers le Middleware ROS combiné à autres protocoles de communication comme MTconnect; Et l'échange indirect à travers la base de donnée. Dans ce second cas, l'information est mise à disposition et les entités peuvent la consulter ou la modifier au besoin. L'échange direct présente deux possibilités également :

- La communication verticale** : Concerne toutes les informations échangées entre des agents et/ou sous-agents de niveaux hiérarchiques différents. Par exemple le superviseur et les ressources de l'atelier utilisent ce genre de communication.

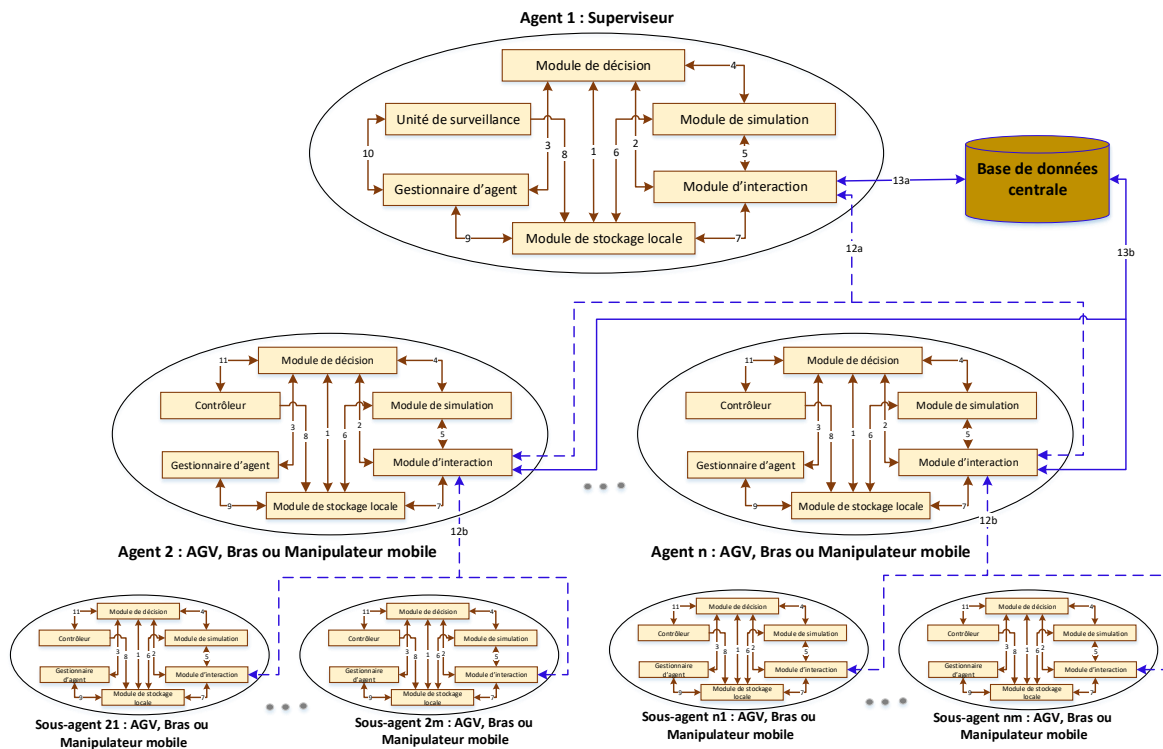


FIGURE 3.2 – Adaptation de DIMS

b) **La communication horizontale** : Elle se fait entre des entités de même niveau hiérarchique.

3.2.3.2 Base de données

Entité indépendante (n'appartient à aucun agent), la base de donnée permet le stockage des données provenant des éléments de l'ensemble de atelier. Elle garantit une gestion dynamique des données hétérogènes provenant des différents systèmes (Robots, machines, applications utilisées par les opérateurs, etc.). La base de données est accessible à tous les éléments tant en lecture qu'en écriture de données [66]. Elle est au centre de la communication indirecte entre les agents et se fonde sur le système de gestion de base de données MongoDB [9].

3.2.3.3 Module de décision

Cet outil contient un ensemble d'algorithmes permettant à l'agent de prendre des décisions. Ces algorithmes peuvent être des méthodes d'assignation de tâches et/ou d'optimisation pour des mécanismes complexes ayant plusieurs possibilités. Par exemple le choix des sous-agents et le choix d'un chemin parmi plusieurs sous certaines contraintes.

3.2.3.4 Unité de surveillance

Grâce à des représentations graphiques en temps réel, la surveillance permet au superviseur d'acquérir continuellement des informations sur l'état

de l'atelier. Ceci lui permet d'agir rapidement en cas de problème. L'unité de surveillance sert aussi dans les actions de maintenance des ressources. L'humain joue un rôle essentiel dans l'interprétation des données de la surveillance ainsi que dans la prise de décision de différents niveaux, rendue facile grâce aux représentations graphiques.

3.2.3.5 Module de simulation

Dans cette application de DIMS à des robots, la simulation peut avoir un rôle de prédiction de la bonne faisabilité des actions ou dans les prises de décisions. Les robots utilisent des outils de simulation comme *Gazebo* et *Rviz* [132], en guise d'outil de planification de trajectoire, tous deux composants du Middleware robotique ROS.

Le superviseur dispose de simulateur de flux tel que *NetLogo* [168] ou *Anylogic*, comme outil de simulation. Ce dernier plus complexe et complet, répond aux besoins du superviseur en charge de tout l'atelier .

3.2.3.6 Gestionnaire d'agent

A l'instar du Manager dans la première version de DIMS (figure 3.1), ce module permet une gestion d'un agent et de ses sous-agents. Il constitue d'abord un registre dans lequel est recensé tous les sous-agents d'une ressource avec leurs caractéristiques (identité, type, fonctions, etc.) et spécifie aussi le besoin d'intégrer ou de supprimer d'autres sous-agents.

3.2.3.7 Contrôleur

C'est l'organe exécutif des robots. Le contrôleur se charge de prendre des décisions de bas niveau (évitement d'obstacles, arrêt d'urgence, génération de trajectoires) ou ne suscitant pas de calculs complexes. Il joue un rôle important dans la surveillance locale du robot et la bonne réalisation des tâches car il dispose des données de tous les capteurs intégrés au robot en temps réel.

3.2.3.8 Module de stockage de données

Ce module est relié à tous les autres car permet le stockage local ou temporaire des données de l'agent. Son organe physique est celle de la mémoire vive dédiée à la ressource (La mémoire de l'ordinateur dédié au superviseur représente son module de stockage).

3.2.4 Informations échangées entre les modules

Dans la figure 3.2, représentant l'adaptation de DIMS, chaque numéro dans la structure des agents représente un type d'interaction entre les différentes composantes d'un agent, d'un agent vers d'autres agents ou vers la base de donnée. Ces informations sont les suivantes, par ordre de numéro croissant :

- **1** : Ce lien bidirectionnel entre les modules de décision et de stockage local de données fait transiter les décisions prises pour des mises à jour et sauvegarde de l'état de l'agent (*Décision* \rightarrow *Stockage*). Il assure par suite une consultation des données courantes sur l'agent pour une meilleure prise de décision (*Stockage* \rightarrow *Décision*).
- **2** : Ce canal permet la transition des données prises par le module de décision vers l'extérieur de l'agent (*Décision* \rightarrow *Interaction*) ou la réception des données provenant d'autres agents ou de la base de donnée pour les intégrer dans la prise de décision (*Interaction* \rightarrow *Décision*).
- **3** : Interaction bidirectionnelle entre le gestionnaire d'agent et l'entité décisionnelle. Cette dernière a besoin d'avoir en entrée les agents à charge (dans le cas du superviseur) et/ou les sous-agents (pour les entités tierces) pour les algorithmes d'allocation de tâches (*Gestionnaire* \rightarrow *Décision*). Par ailleurs, le gestionnaire pour les décisions tactiques (Nombre d'agents ou sous-nécessaire, leurs emplacements ou configuration) peut solliciter les services du module de décision et recevoir ainsi les résultats (*Décision* \rightarrow *Gestionnaire*).
- **4** : La simulation joue un rôle de vérificateur sur la faisabilité des solutions prises par le module de décision notamment celles prises par l'humain, en prenant le sens du lien *Décision* \rightarrow *Simulation*. Dans l'autre (*Simulation* \rightarrow *Décision*) la simulation peut faire appel aux services d'optimisation du module de décision sur le choix à faire sur les paramètres d'entrée de la simulation par exemple. Ce second principe se fonde sur les concepts de l'optimisation basée sur la simulation (*Simulation-based optimization*, en anglais) [145],[88],[140].
- **5** : Le lien 5 permet l'extériorisation des résultats obtenus par la simulation (*Simulation* \rightarrow *Interaction*) ou la prise en compte des données externes comme l'environnement (emplacement des objets statiques et dynamiques), l'état d'autres ressources (dans le cas de la simulation distribuée) en entrée dans la simulation (*Interaction* \rightarrow *Simulation*).
- **6** : Les résultats de la simulation sont stockés à travers ce canal au besoin (*Simulation* \rightarrow *Stockage*) et les informations internes à l'agent et ses sous-agents peuvent aussi servir d'entrée dans le module de simulation (*Stockage* \rightarrow *Simulation*).
- **7** : Permet de transmettre les informations collectées ou stocker des informations depuis l'environnement extérieur et aussi d'envoyer si nécessaire les informations en dehors de l'agent (*Stockage* \rightleftharpoons *Interaction*).
- **8** : Ce canal unidirectionnel sert à l'envoi des données de la surveillance vers le module de stockage local. Ces données qui renseignent essentiellement sur l'état physique des agents (Visualisation des données des capteurs : Consommation énergétique, localisation, vitesse courante,...) peuvent également être utilisées lors des futures prises de décision.
- **9** : Les différentes informations sur les agents et/ou sous-agents passent par l'interaction 9 autant pour les mises à jour (*Gestionnaire* \rightarrow *Stockage*)

que pour la collecte (*Stockage* → *Gestionnaire*) des données à jour. Ces données peuvent être l'historique de l'ensemble des ressources ayant été en contact avec l'agent concerné et/ou les actions (Les contributions) qu'elles ont mené (apporté) avec (à) cet agent. Ceci afin de mieux évaluer l'activité d'un agent et son rôle dans l'atelier.

- **10** : Ce lien existe que dans l'agent superviseur car rattaché au module de surveillance. Il garantit les fonctionnalités suivantes : La création d'un espace de surveillance ou/et de visualisation d'un agent (ou sous-agent) en cas d'une nouvelle intégration, d'une mise à jour sur ses fonctionnalités ou un redémarrage après une maintenance. Ce transfert d'informations se fait dans le sens *Gestionnaire* → *Surveillance*. L'autre sens sert principalement d'action de mise à jour sur les performances des ressources.
- **11** : Présent dans tous les agents en dehors du superviseur, ce lien permet une prise en compte des données perçues par les capteurs d'un agent (caméras et lidars pour l'environnement et la position courante, capteur d'effort pour la présence des humains et obstacles) pour les algorithmes de décision dans le sens *Contrôleur* → *Décision*. L'autre sens consiste à l'envoi des ordres ou missions que doit exécuter l'agent après décision (*Décision* → *Contrôleur*).
- **12a** et **12b** : Représentés par des traits discontinus de couleur bleue, ces liens assurent respectivement les échanges d'informations directs d'agent à agent et de sous-agent à agent (*Sous-agent* ⇌ *Agent*). Les informations peuvent servir dans la collaboration ou coopération entre entités. De ce fait, ces liens ne relient que les modules d'interaction.
- **13a** et **13b** : Représentés par des traits continus bleus, ces connexions régissent d'une part les échanges entre le superviseur et la base de données (13a) puis entre les autres agents et la base (13b) tantôt pour l'accès aux données de tout l'atelier que pour la mise à jour de ces données.

3.3 Ordonnancement de tâches basé sur DIMS

3.3.1 Particularité de l'approche

Le module de décision des agents DIMS propose une méthode d'assignation de tâches basée sur les enchères combinée à une optimisation utilisant un algorithme génétique. La spécificité de l'approche est de permettre une allocation doublement hiérarchique. Grâce à la structure hiérarchique des agents, lorsque qu'un Job est émis l'agent essaie de voir si un de ses sous-agents disposent des capacités requises pour réaliser le Job intégralement. Si cette condition n'est pas satisfaite alors l'agent vérifie si le Job dispose d'une structure hiérarchique permettant une décomposition. Dans le cas où cette décomposition est possible l'agent tente à nouveau de voir la faisabilité avec ses sous-agents et descend au fur à mesure dans la hiérarchie du Job pour trouver une solution avec ces sous-agents tant que la décomposition du Job le permet.

L'optimisation au travers du GA visant à minimiser le coût total d'exécution des tâches par les ressources, se base sur un processus indirect qui fait son originalité car offrant plus de flexibilité. En effet, le codage du chromosome (ch) utilisé est assez particulier car contrairement aux chromosomes classiques, celui-ci ne contient pas les solutions directes qu'on souhaite optimiser. Il se comporte plutôt d'un ensemble de prix virtuels associés aux tâches (P_i) et de profits minimums virtuels reliés aux ressources ($Fmin_r$). Ces valeurs à optimiser constituent des sortes de guides et lignes directrices qui conduiront à une solution (ensemble de ressources) "optimale" (meilleure). L'expression 3.1 illustre le codage utilisé pour le chromosome.

$$ch = \{P_1, \dots, P_n, Fmin_1, \dots, Fmin_m\} \quad (3.1)$$

Avec n le nombre total d'opérations et m le nombre de ressource

3.3.2 Limites de l'approche d'assignation de DIMS

L'approche initialement proposée dans [190] et [67] ne permet pas une bonne gestion de plusieurs Jobs pouvant se faire en parallèle. La méthode ne prévoit aucun mécanisme permettant l'allocation aux ressources d'un ensemble de Jobs s'exécutant simultanément. De ce fait, en ayant plusieurs Jobs, on ne peut que les assigner d'une façon séquentielle (Assignation complète du Job_1 , puis Job_2 jusqu'au Job_n). Cette façon de faire, rallonge considérablement le temps d'exécution total des Jobs (Makespan). Par ailleurs, seul le coût total est optimisé (Optimisation mono-objectif).

3.3.3 Amélioration de la méthode

La révision que nous avons effectuée sur la méthode vise à garantir une bonne gestion de plusieurs Jobs pouvant s'exécuter en parallèle avec des contraintes de précédence entre les éléments d'un Job [115]. En outre, l'optimisation minimise dans cette nouvelle version, le coût total et également le temps total d'exécution des Jobs. D'autres contraintes comme la disponibilité des machines de production et le respect d'un *Due date* (Délai, qui doit être inférieur au temps total d'exécution des Jobs) sont également prises en compte.

Par ailleurs, chaque ressource dispose de performances (Coût et temps) qui lui sont propres en exécutant une tâche. On supposera qu'un plan de production est déjà fourni, et régit celui du transport.

3.3.3.1 Notations

Indices et ensembles

\mathcal{R}	Ensemble des ressources de transport, $\mathcal{R} = \{1, 2, \dots, \mathcal{R} \}$
\mathcal{J}	Ensemble des Jobs, $\mathcal{J} = \{1, 2, \dots, \mathcal{J} \}$
\mathcal{I}^j	Ensemble des tâches d'un Job $j \in \mathcal{J}$

BFP	Ensemble des Buffers ¹ associés aux produits
$Cand_{j,k}$	Ensemble des candidats de la tâche k du Job j
\mathcal{W}	Ensemble des gagnants
r	Indice des ressources de transport, $r \in \mathcal{R}$
j	Indice des Jobs
k	Indice des tâches
p	Indice des produits, chaque p est associé à un Job j
Paramètres	
$P_{j,k}$	Prix virtuel associé à la tâche k du Job j
$Fmin_r$	Profit minimum virtuel associé à la ressource r
N	Plus grande valeur possible pour le coût d'une tâche
ch	Un chromosome
$C_{j,k,r}$	Coût d'exécution associé à la tâche k du Job j par la ressource r
$T_{j,k,r}$	Temps d'exécution associé à la tâche k du Job j par la ressource r
$C_{unit,r}$	Coût par unité de temps associé à la ressource r
$dest_{j,k}$	Point de destination de la tâche k du Job j
bf_r	Buffer de temps associé à la ressource r
bf_p	Buffer de temps associé au produit p
$Disp_{r,p}$	Moment auquel la ressource r et le produit p sont disponibles
$T_{j,k,r}^f$	Date de fin de la tâche k par la ressource r
$totalCost$	Coût total associé à l'ensemble des Jobs
$totalTime$	Temps total associé à l'ensemble des Jobs
$nbrPop$	Nombre d'éléments dans une population
$nbrGen$	Nombre de générations

3.3.3.2 Mécanisme d'enchères et d'optimisation

La méthode se base sur un ordonnancement statique (*Offline Scheduling*), autrement dit elle est faite en avance bien, avant le début de l'exécution des tâches. Elle se compose de cinq phases principales reparties en fonction comme l'indique le schéma 3.3. Chaque fonction joue un rôle clé et bien spécifique dans le processus de l'allocation. La fonction objectif étant de minimiser les coûts et temps totaux :

$$\text{Minimiser}(totalCost, totalTime) \quad (3.2)$$

3.3.3.2.1 Création de chromosome

Dans cette première fonction les chromosomes à optimiser et permettant l'obtention d'un plan d'assignation optimisé sont générés. La création d'un chromosome se fait aléatoirement, et contient les prix virtuels de l'ensemble des tâches quel que soit le Job ainsi que les profits minimums virtuels associés aux ressources. La fonction Création de chromosome est représentée par l'algorithme 1.

1. Le Buffer d'un produit ou d'une ressource constitue un registre qui peut contenir des informations comme : les dates de début et fin des opérations, le type opération, position actuelle... Ici on l'utilise pour enregistrer les dates de fin de process sur des produits, pour avoir leur moment de disponibilité.

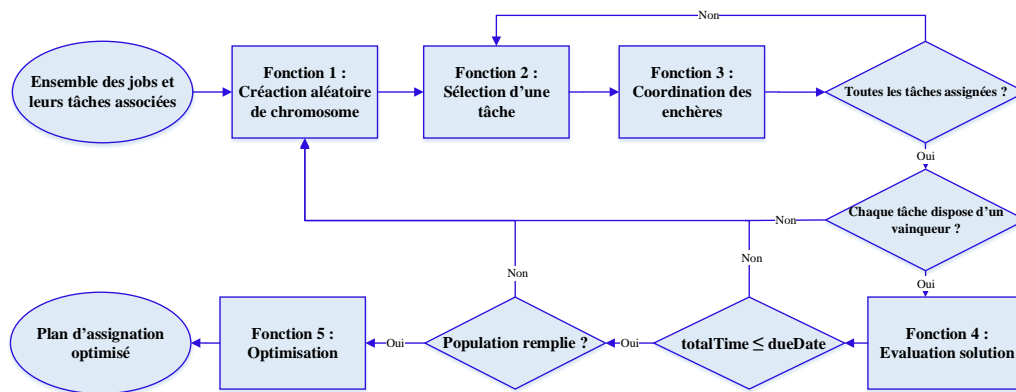


FIGURE 3.3 – Mécanisme global d'ordonnancement

Fonction Création chromosome($\mathcal{J}, \mathcal{R}, N$) :

Pour tout j compris dans \mathcal{J} **faire**

Pour tout k compris dans \mathcal{I}^j **faire**

$P_{j,k} \leftarrow random(0, N)$
 [Nombre aléatoire entre 0 et N]
 $Prix[j, k] \leftarrow P_{j,k}$

Fin Pour

Fin Pour

Pour tout r compris dans \mathcal{R} **faire**

$Fmin_r \leftarrow random(0, N)$
 $Profits[r] \leftarrow Fmin_r$

Fin Pour

$ch \leftarrow [Prix, Profits]$

Retourner ch ;

Fin

ALGORITHME 1 – Création aléatoire d'un chromosome

3.3.3.2.2 Sélection de tâche

Cette phase décrite par l'algorithme 2, permet de choisir efficacement une tâche d'un Job qui respecte la précédence entre les tâches du Job. La sortie de cette fonction est donc une tâche k appartenant à un Job j .

Le choix s'effectue sur la base de deux critères. Premièrement, le produit associé au Job doit avoir le temps de Buffer le plus bas. En d'autres termes, il doit être le produit disponible le plus tôt possible. Cette condition garantit un faible *Makespan*, car tous les Jobs s'exécutant en parallèles se termineront presque en même temps. Deuxièmement, la machine de destination du produit doit être disponible. Dans le cas où la machine n'est pas disponible, la tâche ne peut pas être effectuée pour le moment et une autre doit être choisie.

```

Fonction Sélection tâche(ch) :
  Etape 1 : Choix du Job et tâche
   $j = indice(min(BFP))$ 
   $k = k_{-1} + 1$ 
  Etape 2 : Disponibilité machine
  Si ( $dest_{j,k}$  est libre) Alors
    | Validation  $j$  et  $k$ 
  Sinon
    | Aller à(Étape 1)
  Fin Si
  Retourner  $j, k$ ;
Fin

```

ALGORITHME 2 – Sélection d'un Job et d'une tâche

3.3.3.2.3 Coordination des enchères

La fonction dans l'algorithme 3 est en charge de la transcription du chromosome composé de prix et de profits, en un plan de transport constitué des ressources en charge des tâches.

Ainsi, parmi ses entrées figurent le chromosome(ch) et la tâche sélectionnée (k). La gestion des enchères s'effectue à ce niveau. Après la transmission de l'offre (la tâche) aux ressources, ces dernières estiment et proposent leurs performances (coût et temps de livraison du produit) liées à la tâche. Le temps de livraison ($T_{j,k,r}^f$) du produit est obtenu en ajoutant au temps nécessaire ($T_{j,k,r}$) à la ressource pour effectuer la tâche k , le temps de disponibilité du produit et de la ressource ($Disp_{r,p}$). Après la réception des offres, le superviseur sélectionne les candidats potentiels et le gagnant de la tâche en utilisant les mêmes critères que dans [190]. Toutes les ressources générant un gain sont candidats à la tâche et celle qui minimise le temps de livraison parmi les candidats remporte l'enchère.

```

Fonction Coordination enchères( $ch, j, k, \mathcal{R}$ ) :
  Etape 3 : Enchères
  Pour  $r$  tout compris dans  $\mathcal{R}$  faire
     $C_{j,k,r} = T_{j,k,r} \times C_{unit,r}$ 
     $Disp_{r,p} = \max(bf_p, bf_r)$ 
     $T_{j,k,r}^f = Disp_{r,p} + T_{j,k,r}$ 
    Si ( $P_{j,k} - C_{j,k,r} \geq Fmin_r$ ) Alors
       $Cand_{j,k} \leftarrow r$ 
    Fin Si
    Si ( $Cand_{j,k} = \emptyset$ ) Alors
      [Aucun candidat]
      Aller à(fonction Création chromosome)
    Sinon
       $\mathcal{W} \leftarrow \min(cand_{j,k})$ 
    Fin Si
  Fin Pour
  Tant que (toutes les tâches ne sont pas assignées) faire
    fonction(Sélection tâche)
    Aller à(Etape 3)
  Fait
  Retourner  $\mathcal{W}$ ;
Fin

```

ALGORITHME 3 – Coordination des enchères

Le processus est repris depuis la sélection d'une tâche jusqu'à la désignation du gagnant, pour l'ensemble des tâches. La fonction doit s'assurer aussi que toutes les tâches ont un vainqueur autrement le chromosome est supprimé et un nouveau est généré pour l'utiliser dans le processus d'enchère. La fonction renvoie l'ensemble des vainqueurs (\mathcal{W}) et confirme d'un côté que le chromosome permet d'avoir une solution faisable.

3.3.3.2.4 Évaluation solution

Après l'étape des enchères, le chromosome doit encore passé par une autre phase de validation. L'algorithme 4 permet l'évaluation de la solution trouvée, on calcule les valeurs des temps et coût totaux qu'engendrent les gagnants contenus dans \mathcal{W} en effectuant les tâches de tous les Jobs.


```

Fonction Évaluation solution( $\mathcal{W}$ ) :
   $totalCost = \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{I}^j} \sum_{r \in \mathcal{W}} C_{j,k,r}$ 
   $totalTime = \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{I}^j} \sum_{r \in \mathcal{W}} T_{j,k,r}^f$ 
  Si ( $totalTime \leq dueDate$ ) Alors
     $population \leftarrow ch$ 
     $fitnessTime \leftarrow totalTime$ 
     $fitnessCost \leftarrow totalCost$ 
  Sinon
    Aller à(creation chromosome)
  Fin Si
  Si (Création population initiale vraie et longueur( $population$ ) <  $nbrPop$ ) Alors
    faire fonction(Création chromosome, Sélection tâche, Coordination enchères, Evaluation solution)
  Sinon
    trier( $fitnessCost$ )
    trier( $fitnessTime$ )
    Pour tout  $ch$  compris dans  $population$  faire
       $fitness \leftarrow rang_{ch}(fitnessCost) + rang_{ch}(fitnessTime)$ 
    Fin Pour
    trier( $fitness$ )
  Fin Si
  Retourner  $population, fitness$ ;
Fin

```

ALGORITHME 4 – Évaluation solution

Le cumul des temps de livraison ($totalTime$) doit être plus petit que le $dueDate$ fixé. La validation de cette dernière étape de vérification permet l'enregistrement du chromosome dans la population. En cas de non satisfaction du $dueDate$, le chromosome est supprimé et un autre est généré. S'ensuit alors les fonctions de sélection de tâche et coordination d'enchères. Ces fonctions sont également reprises pour le remplissage de la population initiale. Deux fonctions de Fitness ($fitnessTime$, $fitnessCost$) associées respectivement aux $totalTime$ et $totalCost$ que fournissent chaque élément de la population sont à dénoter. Ces deux Fitness sont triées par ordre de valeurs croissantes, et une troisième Fitness intervient pour sauvegarder la somme des rangs de chaque chromosome dans les $fitnessTime$ et $fitnessCost$. Ainsi $fitness$ qui représente la troisième fonction de fitness est évaluée à son tour. Cette méthode de résolution d'un problème d'optimisation multi-objectif basé sur le GA a été étudiée dans [145].

3.3.3.2.5 Optimisation

La procédure d'optimisation comprenant les différentes phases comme la sélection, le croisement et la mutation sur un certain nombre de génération est décrite dans l'algorithme 5. Elle reste identique à celle utilisée dans l'approche initialement proposée dans [67] et [191]. Le meilleur chromosome utilisé après un certain nombre de génération est transcrit en plan d'assignation comprenant les ressources grâce à la fonction Coordination enchères.

```

Fonction Optimisation(population, fitness) :
  Tant que (generation < nbrGen) faire
    crossover(population)
    mutation(population)
    fonction(Coordination enchères, Évaluation solution, Opti-
      misation)
  Fait
    meilleur_ch ← fitness[0] [Premier chromosome dans fitness]
    plan d'assignation= faire fonction(Coordination
      enchères(meilleur_ch))
  Retourner plan d'assignation;
Fin

```

ALGORITHME 5 – Optimisation

3.4 Cas d'étude

Ce cas d'étude vise à mettre en oeuvre l'amélioration de l'approche d'ordonnancement de tâches basé sur DIMS. L'expérience est conduite à travers une simulation. En outre, les résultats expérimentaux sont implantés sur un cas d'étude réel en vue de pouvoir mettre en place les modules d'interaction des différents agents, qui assurent l'échange d'information. Les détails de l'implémentation sont disponibles en annexe A.

3.4.1 Description du cas d'étude

L'expérience, relate un atelier de production flexible miniaturisé composé de quatre machines de production (M1, M2, M3, M4) qui assurent la fabrication de quatre produits. Deux stocks principaux sont également utilisés (Stocks d'entrée et de sortie) pour le stockage des matières premières et des produits finis. Trois robots sont en charge du transfert des produits entre les différents postes. L'ensemble des composants (machines, stocks, robots) de l'atelier miniature est représenté sur la figure 3.4. Le délai de réalisation (*dueDate*) est fixé à 300secondes.

Quatre Jobs sont considérés sachant que chaque Job est associé à un produit. Ces derniers ont chacun le même nombre de tâches de transport : quatre (4),

soit seize au total. La séquence de fabrication de chaque produit est prédéfinie comme l'indique la figure 3.5.

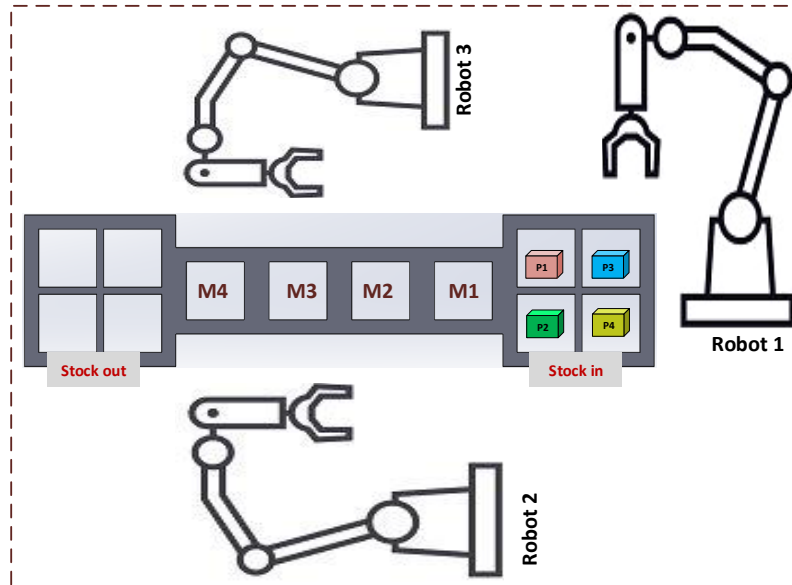


FIGURE 3.4 – Mini-atelier de production

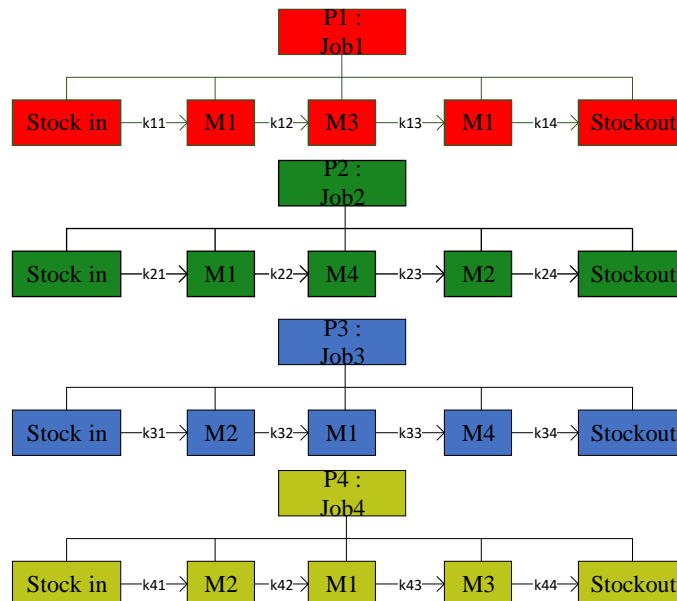


FIGURE 3.5 – Sequences des produits

3.4.2 Résultats

Le diagramme de Gantt dans la figure 3.6, permet de voir le plan d'assignation des tâches de transport associées aux produits sur les robots. On

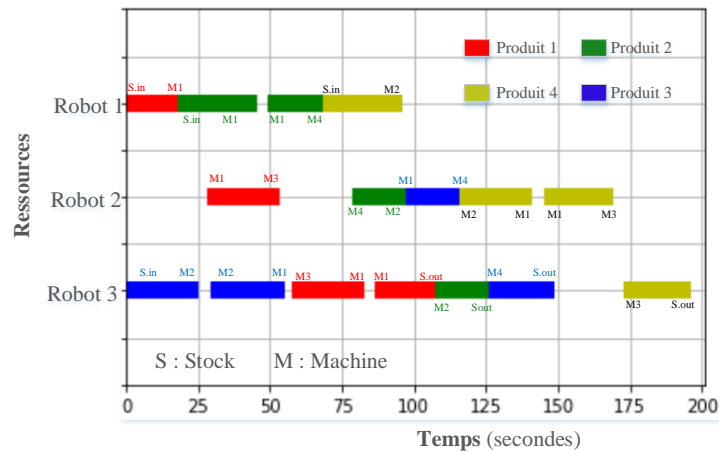


FIGURE 3.6 – Diagramme de gantt de l'activité des trois robots

peut voir le point de prise et la destination de chaque produit ainsi que la ressource de transport en charge (en ordonnées sur la figure). La traçabilité des produits sur le Gantt est facilitée grâce au code couleur associé à chaque produit. Les temps de production sont relativement plus petits que ceux du transport (10s pour chaque machine quel que en soit le produit), mais ne sont pas explicitement mis en évidence sur le Gantt (Ce dernier mettant en évidence que les tâches de transport). Néanmoins, à chaque fois qu'un produit est transportée au niveau d'une machine, il subit un process de 10s.

Ce diagramme permet de montrer la faisabilité des solutions trouvées et la prise en compte des contraintes : précédence entre les tâches d'un même Job, plusieurs Jobs s'effectuant en parallèle, toutes les machines travaillant sur un seul produit à la fois et également le respect du *dueDate* de 300s. Les résultats de l'optimisation représentée par la figure 3.7 démontrent l'efficacité de la fonction bi-objectif. Pour les premières générations, le coût total et le temps sont inversement proportionnels (l'une des valeurs est élevée et l'autre faible). Après 8 générations, les deux valeurs coût total et temps sont minimisées.

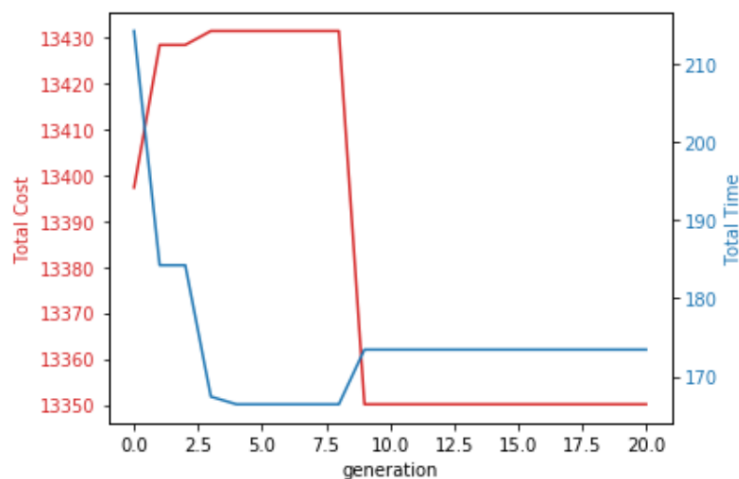


FIGURE 3.7 – Évolution des coûts et temps totaux par génération

3.5 Conclusion

En définitive, la configuration des agents dans l'architecture DIMS permet une meilleure supervision des FMS. En effet, la configuration des agents est d'abord du type fractal (mêmes éléments constitutifs pour tous) avec des fonctionnalités multiples permettant des prises de décisions optimisées, et une communication inter-agent fluide et simplifiée. L'amélioration apportée à la structure générique de l'agent par l'ajout de certains modules tels que celui de la surveillance de l'atelier, et la redéfinition des autres modules avec les outils permettant leur implémentation garantit : une bonne gestion et fiabilité des éléments autonomes de transport de l'industrie 4.0 et une réduction de la complexité de l'architecture pour faciliter l'intégration.

L'ajout d'une base de donnée permet de faire face et de mieux gérer l'afflux de données non homogènes générées par les différents systèmes. De plus, la base de données permet également une centralisation de l'information pour une accessibilité rapide aux données de l'atelier pour les éléments et une mise à jour continue de celle-ci.

Un accent particulier a été mis sur le module de décision au niveau du superviseur par le développement d'une approche d'assignation de tâche basé sur un mécanisme d'enchères. Cette approche bénéficie de la structure hiérarchique des agents mais aussi d'une possibilité de décomposition à plusieurs niveaux des Jobs pour une meilleure flexibilité pour les agents lors du processus d'allocation. J'ai également amélioré cette méthode par la prise en compte de plusieurs Jobs pouvant s'exécuter en parallèle avec une contrainte de précédence entre les tâches d'un Job. La méthode d'optimisation basée sur un algorithme génétique qui se veut indirect par le codage d'un chromosome composé de paramètres (prix et profits) menant à une solution optimisée de nature différente (Plan d'assignation composé de ressources de transport), a également bénéficié d'une modification par le passage de la fonction mono-objectif (minimisation du coût) à une bi-objectif (minimisation coût et temps). L'efficacité de la méthode avec son processus d'optimisation indirecte a été prouvée dans un cas d'étude.

Le prochain chapitre portera sur le développement du module de décision des agents de bas niveaux (ressources de l'atelier).

Chapitre 4

Ordonnancement de tâches par formation de coalitions

4.1 Introduction

Dans ce chapitre une méthode de décision distribuée permettant l'ordonnancement des tâches de transport incluant les opérations de chargement et déchargement a été développée. Cette approche, combine un modèle de décision globale basé sur les enchères (CNET) et un modèle de décision locale basé sur la programmation linéaire en nombres entiers.

Ce dernier est formulé d'une façon mathématique et vise à minimiser le temps d'accomplissement global des tâches avec une prise en compte de plusieurs contraintes. Cette approche est appelée DISCOFT pour Formation de COalition DIStribuéée pour les tâches de Transport (*DIStributed COalition Formation for Transportation tasks* en anglais).

Un second algorithme développé dans le même contexte industriel est couplé à DISCOFT pour garantir ainsi un plan de production optimisé, qui engendrera les tâches de transport. Cette seconde méthode qu'on nommera IT2P (*Integrated Transportation and Production Plans*), est également décrite à travers un algorithme.

La section 4.2 de ce chapitre décrit la relation qui existe entre les différentes activités de production à savoir les ordres, les Jobs, les tâches et les opérations. Ensuite, la section 4.3 donne un mécanisme général de l'ordonnancement de tâche qui va de l'émission d'un ordre de production à l'obtention d'un plan d'ordonnancement de production et de transport. Le rôle et les liens entre les deux approches (IT2P et DISCOFT) sont aussi spécifiés. La section 4.4, décrit le modèle de l'approche IT2P.

Enfin, la section 4.5 traite des deux algorithmes de décision (globale et locale) qui composent l'approche DISCOFT. La nécessité d'utiliser une méthode d'optimisation pour le modèle de décision locale est également spécifiée.

4.2 Les tâches industrielles

Dans le processus de fabrication ou d'assemblage d'un produit en milieu industriel, deux types de tâches sont associés : Production et Transport. Ces

deux tâches sont reliées, car assurant à la fois les actions sur les produits auprès des machines (tâches de production) et le transfert des produits entre les machines (tâche de transport).

Les activités des ateliers de production sont organisées d'une façon hiérarchique. L'appréhension de cette organisation hiérarchique est indispensable pour mieux cerner la répartition des rôles dans l'atelier.

Le schéma 4.1 montre une structure hiérarchique à quatre niveaux détaillant les liens entre les ordres, les Jobs, les différentes tâches et les opérations. Ces quatre éléments constituent les différentes activités de la production.

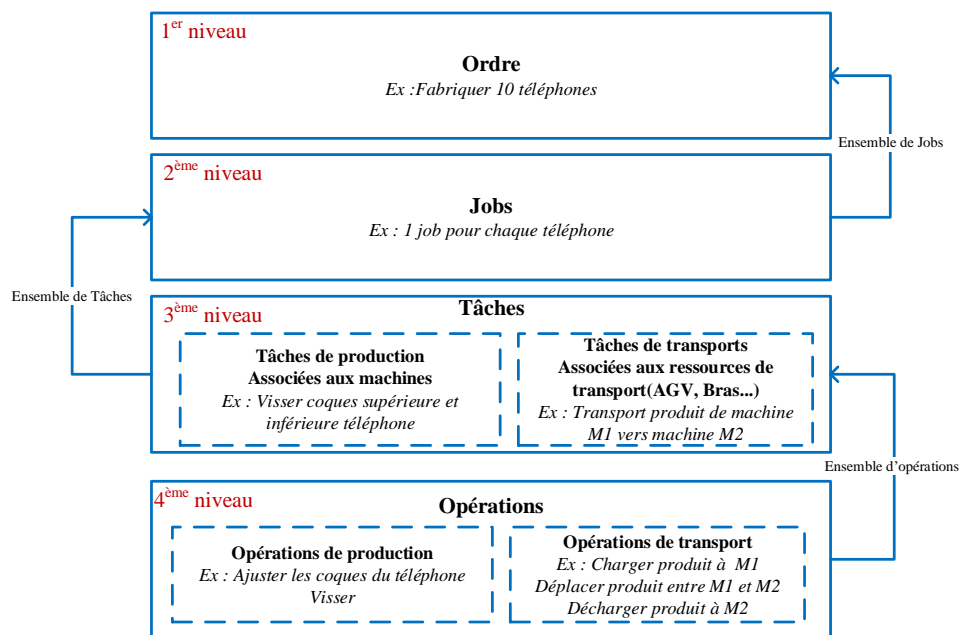


FIGURE 4.1 – Organisation hiérarchique des activités de production

4.2.1 Ordre

Les ordres de production représentent le plus haut niveau de la hiérarchie. Elles contiennent les détails des produits à fabriquer et émanent généralement de l'ERP. Dans l'approche du juste-à-temps qui consiste à produire à la demande, les ordres correspondent directement aux commandes clients [154].

4.2.2 Job

Situé au deuxième niveau dans la hiérarchie, les Jobs sont une transcription des ordres que les algorithmes d'assignation utilisent comme entrée. Un Job est ainsi associé à chaque produit, donc un ordre de dix produits correspond à dix Jobs. A noter que chaque exemplaire d'un même produit générera un Job distinct.

4.2.3 Tâches

Un Job est une succession de plusieurs tâches comme celles de production, transport, etc. Les tâches de production sont associées aux ressources de production telles que les machines ou les postes d'assemblage. Les tâches de transport permettent le transfert des produits entre les différentes ressources de production ou entre les stocks, avec des ressources comme les manipulateurs mobiles, les AGVs ou les humains.

4.2.4 Opérations

Les opérations ou encore sous-tâches, sont au dernier niveau dans la hiérarchie et comportent les détails élémentaires des tâches. La définition des opérations (ce en quoi elles consistent) peut être propre à chaque application.

4.3 Formalisme général de l'assignation des tâches

Dans les processus industriels, le transport et la production sont liées et fortement dépendants. La fabrication d'un produit requiert son transfert entre plusieurs points. Une suite séquentielle appelée aussi Job, régit la mise au point d'un produit de la manière générique suivante :

$(transport \rightarrow production) \times f \rightarrow transport$, f étant le nombre de machines par lequel doit transiter le produit. Ce qui se traduit par l'acheminement de produit vers une machine, puis la récupération après processus et le transfert vers un point de stockage ou une autre machine si la séquence de fabrication inclut plusieurs postes de fabrication. Le cycle commence et finit par une tâche de transport car on suppose que chaque produit fait une entrée dans l'atelier depuis le stock des matières premières puis une sortie en fin de production vers la zone de stockage des produits finis). Le temps de production dans cette dernière zone est généralement considérée comme nulle, raison pour laquelle seul le transport est indiqué.

A la réception d'un ordre de production comme l'indique la figure 4.2, comprenant des détails sur l'ensemble des produits à fabriquer (les Jobs) ainsi que les alternatives de production de chaque tâche, une première approche d'assignation prend en charge les Jobs. Cette approche, IT2P, fournit un planning de production ainsi qu'un plan de transport partiel. Le plan production de l'approche IT2P sert ainsi d'entrée à l'approche DISCOFT. Cette dernière, renvoie un planning de transport complet, et un plan de production avec des dates de début et de fin de tâches ajustées par rapport à ceux obtenus initialement dans IT2P.

Les sections suivantes, fournissent plus de détails sur les deux approches et sur leur liaison.

4.4 Approche IT2P

L'approche IT2P est une méthode d'ordonnancement permettant la prise en compte simultanée des tâches de transport et de production. Elle a été développée dans le cadre du projet CoRoT dans une collaboration entre le

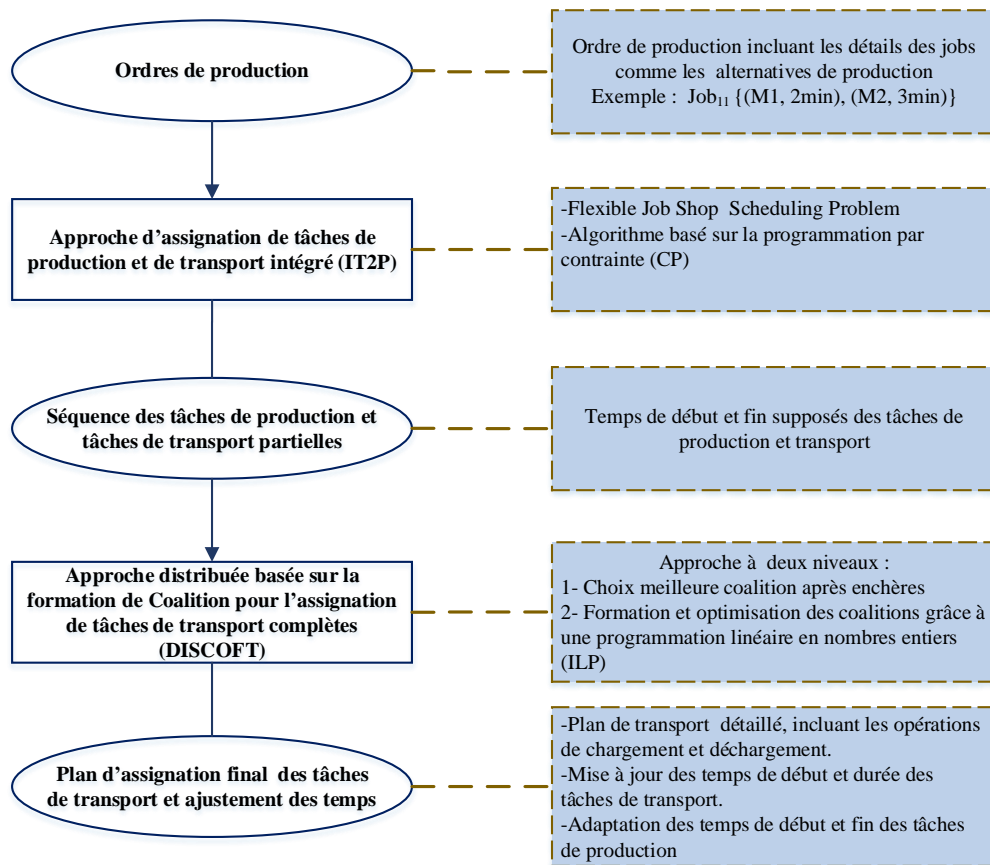


FIGURE 4.2 – Diagramme global de l'assignation

laboratoire de recherche et d'innovation LINEACT de Rouen en France et l'université d'EXETER au Royaume-Uni. Ma participation dans le développement de cette approche s'est fait donc au travers du laboratoire LINEACT.

4.4.1 Intérêt pour notre étude

L'objectif de cette méthode est de choisir le couple (machine, AGV) minimisant le temps d'accomplissement pour l'ensemble des tâches provenant des Jobs. Hormis la prise en compte simultanée des plans de transport et de production, l'approche traite aussi des FJSP. Ces deux caractéristiques en font son originalité.

L'apport principal de l'approche IT2P pour notre étude est le plan de production qui est utilisé comme support pour les modèles de décision intégrés dans l'approche DISCOFT. En outre, étant donné que IT2P fournit un plan de transport sans prise en compte des opérations de chargement et déchargement, elle fera office d'outil de comparaison avec DISCOFT dans la partie expérimentale au chapitre 5.

4.4.2 Différence entre les deux approches

Les différences majeures entre les plannings de transport obtenus avec les deux approches résident d'abord dans la décomposition et l'ajout de sous-tâches à DISCOFT. De plus, les méthodes de résolution utilisées, une est basée sur la programmation par contrainte (IT2P) et l'autre sur les enchères et la programmation linéaire en nombres entiers (DISCOFT). Enfin, dans cette dernière approche, le principe de coalition est utilisé avec une intégration des humains comme ressource de transport.

Par ailleurs, les deux méthodes disposent de la même allocation aux machines mais avec un décalage des dates de début et de fin des tâches de production. Ce décalage est causé par le transport qui prend en compte plus d'opérations dans l'approche DISCOFT. Néanmoins les durées opératoires sur les machines restent identiques dans les deux approches.

4.4.3 Mécanisme d'ordonnement de l'approche IT2P

4.4.3.1 Notations

Indices et ensembles

\mathcal{J}	Ensemble des Jobs
J	Un job constitué de plusieurs tâches, $J \in \mathcal{J}$
\mathcal{V}	Ensemble des AGVs, $\mathcal{V} = \{1, 2, \dots, \mathcal{V} \}$
\mathcal{M}	Ensemble des machines, $\mathcal{M} = \{1, 2, \dots, \mathcal{M} \}$
$Cand_{j,k}^m$	Sous-ensemble comprenant les machines candidates à la tâche k du Job j , $Cand_{j,k}^m \subset \mathcal{M}$
j	Indice d'un Job J dans l'ensemble \mathcal{J}
k	Indice des tâches d'un Job, $k \in J$
m	Indice des machines, $m \in \mathcal{M}$
v	Indice des AGVs, $v \in \mathcal{V}$
mg	Indice machine gagnante
vg	Indice AGV gagnant

Paramètres et variables

$\delta_{j,k,m}$	Temps d'exécution de la tâche k du Job j par la machine m
$Disp_r$	Date ou moment auquel la ressource r est disponible pour effectuer une tâche, $r = m$ ou v
p_v	Position actuelle de l'Agv v
$tt_{p_v,m}$	Temps de déplacement entre la position p_v et la machine m pour l'AGV v
$T_{j,k,m}^d$	Date de début de la tâche de production k par la machine m
$TR_{j,k,v}^d$	Date de début de la Tâche de tRansport k par l'Agv v
$T_{j,k,m}^f$	Date de fin de la tâche de production k par la machine m
$TR_{j,k,v}^f$	Date de fin de la Tâche de tRansport k par l'Agv v
$Plan_{prod}$	Tableau pour contenir les détails sur les assignations faites aux machines
$Plan_{trans}$	Tableau pour contenir les détails sur les assignations faites aux AGVs

4.4.3.2 Algorithme d'allocation

Le mécanisme d'assignation déterminé par l'algorithme 6, commence par la sélection d'un Job et d'une tâche. Dans IT2P, les tâches ayant les indices (*Job, tâche*), les plus petits sont d'abord sélectionnées. La phase de sélection d'une tâche à assigner est importante car détermine les premiers temps d'occupation des ressources et conditionne ainsi l'objectif final qui est la minimisation du Makespan. Dans une version ultérieure de l'approche IT2P, un algorithme génétique (GA) est utilisé pour l'ordre de sélection des tâches, en codant un chromosome qui donnait une priorité aux tâches (Voir plus de détails en annexe B sur le codage du chromosome et le modèle mathématique de l'approche IT2P). Celles qui ont une priorité élevée sont choisies en premières. Cependant, grâce à la programmation par contrainte (CP) qui permet une assignation par intervalle, la problématique de sélection est bien gérée. En effet, la tâche 1 du Job 2 notée (2, 1) peut être choisie en première pour l'allocation mais programmée dans un intervalle [*début_tâche, fin_tâche*] bien tardif par rapport à une tâche d'un autre Job sélectionnée après la (2, 1). Il faut par ailleurs spécifier la contrainte de *non-chevauchement* entre les intervalles de temps pour une même ressource et aussi d'autres contraintes comme celle de précedence entre les tâches d'un même Job dans le programme du CP. La figure 4.3 illustre les possibilités d'intégration d'une nouvelle tâche dans le Buffer d'une ressource par l'approche IT2P et les autres méthodes d'assignation de tâche.

```

Fonction Approche_IT2P( $\mathcal{J}, \mathcal{V}, \mathcal{M}$ ) :
  Pour tout  $J$  compris dans  $\mathcal{J}$  faire
    Pour tout  $k$  compris dans  $J$  faire
      [Initialisation de la date de fin de la tâche à un grand nombre]

       $Tmin_{j,k}^f \leftarrow \infty$ 
      Pour tout  $m$  compris dans  $cand_{j,k}^m$  faire
        Si ( $m$  équivaut à  $m^{source}$ ) Alors
          [Pas de transport, car machine source = machine destination]
           $T_{j,k,m}^f \leftarrow Disp_m + \delta_{j,k,m}$ 
          Si ( $T_{j,k,m}^f < Tmin_{j,k}$ ) Alors
            ( $mg, vg$ )  $\leftarrow (m, None)$ 
            ( $T_{j,k,mg}^d, T_{j,k,mg}^f$ )  $\leftarrow (Disp_m, T_{j,k,m}^f)$ 
             $Tmin_{j,k} \leftarrow T_{j,k,m}^f$ 
          Fin Si
        Sinon
          [Machine source différente machine destination, donc nécessité de transport]
          Pour tout  $v$  compris dans AGV faire
             $TR_{j,k,v}^f \leftarrow \max(Disp_v +$ 
               $tt_{p_v, m^{source}}, TR_{j,k-1}^f) + tt_{m^{source}, m}$ 
             $T_{j,k,m}^f \leftarrow \max(Disp_m, TR_{j,k,v}^f) + \delta_{j,k,m}$ 
            Si ( $T_{j,k,m}^f < Tmin_{j,k}$ ) Alors
              ( $mg, vg$ )  $\leftarrow (m, v)$ 
              ( $TR_{j,k,vg}^d, TR_{j,k,vg}^f$ )  $\leftarrow$ 
                ( $Disp_v, TR_{j,k,v}^f$ )
               $T_{j,k,mg}^d \leftarrow \max(Disp_m, TR_{j,k,v}^f)$ 
               $T_{j,k,mg}^f \leftarrow T_{j,k,m}^f$ 
               $Tmin_{j,k} \leftarrow T_{j,k,m}^f$ 
            Fin Si
          Fin Pour
        Fin Si
      Fin Pour
      [Ajout machine et AGV gagnants aux plans de production et de transport, ainsi que les dates de début et fin de tâche]
       $Plan_{prod}[j, k] \leftarrow (mg, T_{j,k,mg}^d, T_{j,k,mg}^f)$ 
       $Plan_{trans}[j, k] \leftarrow (vg, T_{j,k,vg}^d, T_{j,k,vg}^f)$ 
    Fin Pour
  Retourner  $Plan_{prod}, Plan_{trans}$  ;
Fin

```

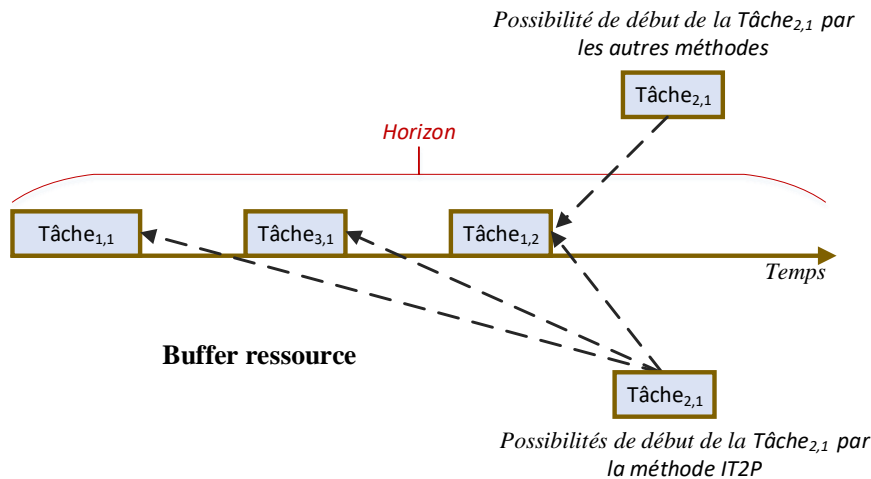


FIGURE 4.3 – Positionnement de IT2P par rapport aux autres approches

Après la sélection, s'en suit la phase de vérification d'une nécessité de transport. Ainsi, si la machine sur laquelle se trouve le produit est la candidate désignée pour la prochaine tâche alors il n'y a aucun besoin d'AGV pour le transport. De plus, si cette machine dispose d'une date d'accomplissement ou de fin d'exécution ($T_{j,k,m}^f$) plus bas que le temps $Tmin_{j,k}^f$ initialisé à une grande valeur, elle devient la gagnante temporaire en attendant de vérifier les autres machines candidates. La valeur de $Tmin_{j,k}^f$ est ajustée vers celle de $T_{j,k,m}^f$.

Lorsqu'il y a un besoin de transporter (La machine source est différente de la machine destination), alors il faudrait en plus trouver un AGV v capable d'exécuter cette tâche de transport. Celui-ci devrait permettre de minimiser la date de fin de la tâche de production qui est impactée par le transport. Le calcul du temps de livraison du produit à la machine m par l'AGV v passe d'abord par la détermination du moment de disponibilité des ressources et du produit concernés. La disponibilité s'obtient en prenant le maximum entre le temps de disponibilité ($Disp_v$) de l'AGV (Déterminé par le CP avec respect de toutes les contraintes) et le moment de fin de la précédente tâche de production associée au produit ($T_{j,k-1,m}^{source}$).

Ensuite, les temps de transport de l'AGV sont ajoutés au moment de disponibilité à la fois des ressources et du produit concernés. Le transport concerne le déplacement à vide de l'AGV vers le point de prise ($tt_{p_v, m}^{source}$) s'il y a lieu, et le transfert du produit vers sa destination.

Une fois que l'AGV aura livré le produit, ce qui correspond à la fin du transport, la tâche de production peut commencer si la machine est disponible. Dans ce cas de figure, le début de la production $T_{j,k,m}^d$ correspond au maximum entre la fin du transport et du temps de disponibilité de la machine ($\max(TR_{j,k,v}^f, Disp_m)$). Si la date de fin de la production est inférieure à la

date minimale actuelle, associée à la tâche ($T_{j,k,m}^f < T_{j,k}^{min,f}$) alors le couple (machine, Agv) est désigné comme gagnant temporaire. Ce mécanisme est vérifié pour l'ensemble des AGVs avec la machine candidate choisie, puis on passe à une autre machine jusqu'à trouver le couple gagnant définitif pour la tâche.

A chaque fois qu'un duo gagnant est trouvé pour une tâche, les tableaux des plans de production et de transport sont ajustés. Cette mise à jour consiste à mettre dans les cellules dédiées les détails sur les gagnants tels que : *Resource gagnante, date de début, date de fin*. Les indices du tableau renseignent sur l'identité de la tâche (ligne pour Job, colonne pour tâche).

4.5 Approche DISCOFT

DISCOFT est une approche d'ordonnancement de tâches de transport mise en lumière au cours de cette thèse. La motivation est de permettre une utilisation optimisée des ressources intelligentes dans le contexte de l'I4.0, un accroissement de la flexibilité dans le transport et une allocation complète et plus précise des ressources aux tâches de transport et de production dans ce même contexte.

4.5.1 Caractéristiques générales et originalité

Les éléments caractéristiques de DISCOFT sont les suivants :

- (a) **Une distributivité** : L'approche propose une prise de décision partagée entre l'ensemble des éléments de l'atelier. Toutes les entités participent à la prise de décision. La charge de travail du superviseur est réduite grâce à cette participation collective.
- (b) **Une assignation hybride** : DISCOFT combine deux algorithmes dans la prise de décision. Le premier est basé sur la programmation linéaire en nombres entiers et le second sur un mécanisme d'enchères.
- (c) **Une optimisation double** : Grâce à l'approche IT2P qui fournit un plan de production issu de la programmation par contrainte, la sortie de DISCOFT garantit à la fois un plan de transport et de production optimisé.
- (d) **Une collaboration multiple** : Le plan de transport est obtenu grâce à une collaboration entre les ressources du bas niveau (Robots et humains) par le principe de coalition et aussi avec l'entité de haut niveau (superviseur) qui détermine le gagnant. Ce qui se traduit par une collaboration horizontale (entre entités de même niveau) et verticale (entre entités de hiérarchie différente).
- (e) **Ordonnancement élémentaire** : Toutes les tâches de transport sont considérées comme un ensemble d'opérations. Ainsi l'ordonnancement par l'approche DISCOFT consiste à trouver des gagnants pour chacune de

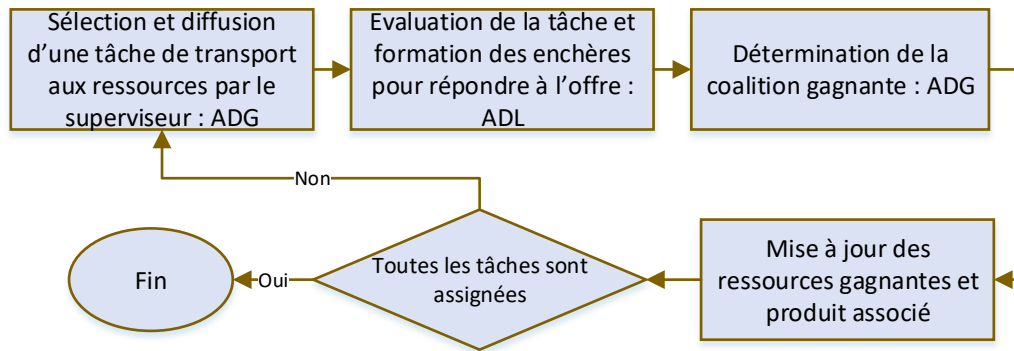


FIGURE 4.4 – Mécanisme d'ordonnancement des tâches

ces opérations contrairement aux travaux existants qui assignent juste la tâche de transport en la considérant comme une seule opération.

4.5.2 Principe d'assignation avec DISCOFT

Une fois que l'approche IT2P basée sur la programmation par contrainte aura fourni un plan de production pour l'ensemble des tâches de tous les Jobs, DISCOFT se charge de récupérer les tâches de transport qui en découlent. Ces dernières sont alors subdivisées en sous-tâches (opérations) puis assignées une tâche après l'autre.

Le mécanisme d'ordonnancement de l'ensemble des tâches de transport est illustré par la figure 4.4. Le superviseur disposant d'un Algorithme de Décision Globale (ADG) basé sur les enchères, sélectionne la tâche de transport associée au produit ayant la date de début de tâche de production la plus petite (minimum parmi les dates de début de toutes les tâches production fournies par IT2P).

La tâche de transport choisie est alors diffusée aux ressources de bas niveau avec l'ensemble des détails sur les opérations la constituant. Les ressources de bas niveau évaluent puis répondent à l'offre en formant des coalitions optimisées grâce à un Algorithme de Décision Locale (ADL) implémenté sur les ressources.

Le superviseur procède à la désignation des ressources gagnantes, puis à la mise à jour des Buffers associés aux concernés (ressources et produit). Le mécanisme est repris de la sélection de tâche à la mise à jour des ressources gagnantes et produit jusqu'à l'assignation de toutes les tâches.

La subdivision des tâches, les différents algorithmes de décision et la procédure de mise à jour des ressources sont détaillés dans les parties qui suivent.

4.5.3 Les opérations possibles d'une tâche de transport

Outre les trois opérations primaires d'une tâche de transport, d'autres opérations, secondaires (elles sont faites dans certains cas), sont à dénoter.

Ces opérations ne concernent pas directement les produits mais sont essentielles dans le cas où les ressources de transport sont absentes des lieux de chargement et/ou déchargement. L'ensemble des opérations (primaires et secondaires) sont mises en évidence sur le figure 4.5. Les notations OS_1 , OS_2 et OS_3 sont utilisées pour les types d'Opération Secondaire.

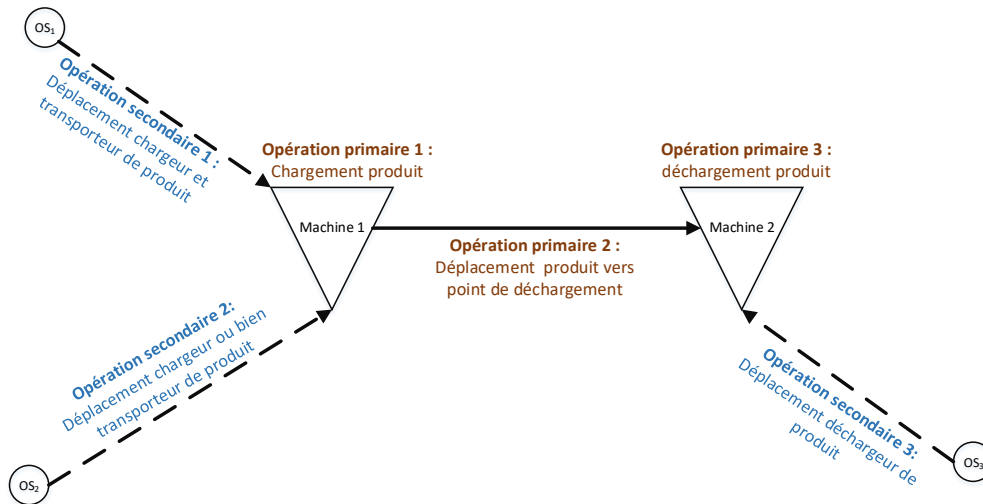


FIGURE 4.5 – Modèle éclaté d'une tâche de transport

4.5.3.1 Opérations secondaires au point de chargement

Deux types d'opération secondaire (OS_1 et OS_2) sont possibles avant la phase de chargement du produit comme l'indique la figure 4.5). Ces opérations secondaires interviennent en fonction de la présence ou non des ressources sur le lieu de chargement.

La variable plc (présence lieu chargement) utilisée dans le tableau 4.2, permet déterminer le type d'opération secondaire qui aura lieu avant le chargement. Ainsi, lorsque plc prend la valeur 0 pour une ressource, cela implique que cette ressource est absente du lieu de chargement. Elle doit de ce fait se déplacer pour s'y rendre.

TABLE 4.2 – Opérations secondaires possibles avant le chargement

Variable	Ressources		Type secondaire
	Chargeur	Transporteur	
plc	0	0	OS_1
	0	1	OS_2
	1	0	OS_2
	1	1	-

Comme le montre le tableau 4.2, l'absence du transporteur et du chargeur du produit sur le lieu de chargement conduit à une ou des opérations secondaires du type OS_1 . Le nombre d'opérations de type OS_1 à faire dépendra de la configuration des ressources.

Si elles sont jointes alors une seule opération OS_1 sera faite avant l'opération primaire de chargement car les deux ressources (Chargeur et transporteur du produit) forment un manipulateur mobile. Par contre, si les deux ressources sont séparées alors deux OS_1 sont à faire, chacune correspondant au déplacement des deux ressources concernées vers le lieu de chargement. Ces deux OS_1 , peuvent se faire en parallèle et n'ont donc aucune contrainte de précedence entre elles. Néanmoins, tant qu'elles ne sont pas effectuées l'opération primaire de chargement ne peut commencer (figure 4.6).

Le type OS_2 intervient si et seulement si une des ressources (Chargeur ou transporteur) n'est pas présente au point de prise du produit. Sur le tableau 4.2, L' OS_2 est obtenue si le plc d'une des ressources est à zéro (0).

Si le plc des deux ressources est à 1, ce qui implique qu'elles sont toutes présentes sur le lieu de chargement alors il n'y a aucune opération secondaire à faire.

4.5.3.2 Opération secondaire au point de déchargement

L'opération secondaire de type OS_3 intervenant au point de déchargement (voir figure 4.5), concerne la ressource responsable du déchargement dans le cas où elle est absente de l'endroit désiré. L' OS_3 est assez similaire à l' OS_2 car elle concerne aussi l'absence de seulement une des ressources sur le point de la mission.

La majeure différence entre ces opérations secondaires réside dans leurs possibilités de début. Contrairement à l' OS_2 qui doit commencer en première, l' OS_3 au point de déchargement n'a aucune contrainte et peut se faire en parallèle ou après toutes les opérations de la tâche de transport.

La ressource qui décharge peut : attendre le produit, faire attendre le produit ou arriver au même moment que le produit. L'idéal étant d'avoir un décalage de temps très léger voire nul entre l'arrivée de la ressource et du produit pour que le déchargement puisse commencer au plus tôt.

4.5.3.3 Choix effectué pour notre modèle

Pour le développement de l'approche DISCOFT, certaines opérations secondaires ont été fusionnées avec les primaires. Ainsi, dans le modèle compact des opérations possibles d'une tâche de transport, figure 4.6, on retrouve quatre opérations primaires et aucune secondaire.

La **première opération** concerne le déplacement du transporteur (AGV) vers le point de charge. Cette opération peut avoir une durée nulle, dans le cas où l'AGV est déjà sur place. L'intérêt de cette opération est de faciliter l'expression temporelle des contraintes de précedence entre les différentes opérations.

Dans la **deuxième opération**, on peut retrouver un éventuel mouvement vers le point de chargement de la ressource en charge, puis la prise et dépose du produit sur l'AGV. Ce dernier est présent sur le lieu grâce à la précedence

avec la première opération. Ainsi la deuxième opération débute avec le déplacement de la ressource qui charge (Si elle est absente) et prend fin après la dépose du produit.

La **troisième opération** reste inchangée de celle du modèle éclaté dans la figure 4.5 et permet donc le transfert du produit.

Enfin, la **quatrième opération** affecte la ressource de déchargement tant pour son action possible de déplacement vers la destination du produit que par sa prise et dépose du produit sur la machine ou stock adéquat. Elle doit ainsi être précédée par toutes les autres opérations.

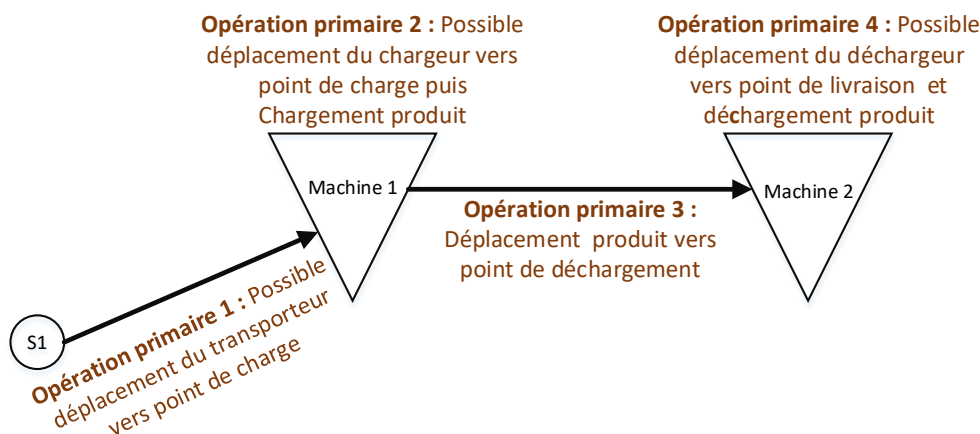


FIGURE 4.6 – Modèle compact d'une tâche de transport

4.5.4 Conditions d'appartenance aux coalitions

Le but d'une coalition étant de réunir plusieurs compétences au sein d'un même groupe, certains critères ou conditions permettent l'accès à une coalition dans le contexte industriel du transport [53]. Ainsi pour chaque coalition donnée, un ensemble d'exigences est émis en fonction de la tâche à effectuer et les ressources souhaitant intégrer la coalition doivent remplir au moins une de ces conditions.

Du point de vue des ressources, les critères ouvrant l'entrée à une coalition représentent un ensemble de performances ou de fonctionnalités. Pour ce qui est des tâches qui émettent ces critères, ce sont plutôt des pré-requis ou obligations que les ressources doivent satisfaire.

Par exemple, soit k une tâche consistant à récupérer un cadre de verre trempé dans un liquide savonneux après ponçage, puis à le poser sur un AGV adapté. Le cadre en verre qui servira à la fabrication d'un pare-brise de voiture, pèse 11kg. L'AGV déplace le cadre vers un autre point où il doit être placé sur une plateforme pour recevoir un calque de verre. Toutes ces actions doivent se faire dans un temps de 45 secondes. De la tâche k , les exigences suivantes peuvent être déduites :

- Type : Le contexte étant le transport, des ressources capables de charger/décharger et transporter sont requises
- Outil : Une ventouse pour saisir le verre
- Performances physiques : Une charge utile de minimum 11kg est exigée pour toutes les ressources. Des vitesses élevées pour le délai (45 secondes) de réalisation de la tâche et aussi une bonne précision pour déposer le verre sur le transporteur. Une étanchéité est aussi requise pour la ressource effectuant l'opération de récupération dans la solution savonneuse.
- Délicatesse : Pour pouvoir manipuler le verre avec précaution

Les critères possibles varient en fonction du domaine d'application (Industrie automobile, pharmaceutique, agro-alimentaire, etc). Néanmoins, le tableau 4.3 recense et décrit un grand nombre de paramètres utilisés pour la formation de coalitions concernant les tâches de transport. Les robots ayant été conçus pour imiter les humains, les paramètres décrits restent valables pour les deux (robots et humains) [22].

A noter que, dans le contexte de l'industrie 4.0, les humains travaillant dans les usines comme opérateurs, communiquent avec les autres éléments grâce à des objets (Tablette, casque...) qu'ils portent [60].

TABLE 4.3 – Critères des coalitions

N°	Paramètres	Description
1	Type	Le domaine d'application pour laquelle la ressource est conçue (robot) ou déployée (humain). Autrement dit, le type de tâche ou d'opération que la ressource est capable de réaliser, comme transporter des objets, souder des pièces, etc. Dans le cadre du transport, les types varient des ressources de chargement/déchargement (prise et dépose) à celle du transport de produit. Les humains, grâce à leurs bras et pieds peuvent satisfaire les deux types d'opération.
2	Outil	Est déterminé par la terminaison ou métier de la ressource. Chaque outil est une spécification du type. Par exemple : une ressource de type prise et dépose (<i>Pick and Place</i>) peut le réaliser avec un outil ventouse ou pince.
3	Polyvalence	Ensemble des fonctions ou métiers que la ressource peut effectuer. Les humains ou les bras manipulateurs avec des changeurs d'outils sont polyvalents.
4	Modularité	Capacité et facilité d'une ressource à se transformer en une ou plusieurs autres pour s'adapter à de nouvelles situations.

N°	Paramètres	Description
5	Performances physiques : a) Charge utile b) Vitesse c) Précision et répétabilité d) Autonomie énergétique	Englobent les capacités diverses intégrées à la ressource, telles que : a) La charge maximale que peut porter ou déplacer une ressource de transport. b) Vitesse de déplacement ou d'exécution des opérations c) Faculté à minimiser l'erreur dans la réalisation d'une action une ou plusieurs fois. d) Durée que peut tenir la batterie d'une ressource qui en dispose, sans besoin d'être rechargée.
6	Performance logicielle	Faculté à réaliser assez rapidement des calculs (Générer de nouvelles trajectoires par exemple), résoudre des problèmes complexes et stocker des informations.
7	Coût de travail	Représente le coût que peut susciter une ressource pour exécuter une opération sur une certaine durée. Ce coût peut être la consommation énergétique dans le cas des robots ou un prix par unité de temps (ou par opération) pour les humains.
8	Délicatesse	Aptitude d'une ressource à pouvoir manipuler ou transporter des objets fragiles.
9	Collaborativité	Faculté d'une ressource à travailler avec d'autres ressources sans danger et pour un objectif commun.
10	Coopérativité	Faculté d'une ressource à travailler efficacement avec d'autres sans forcément avoir les mêmes objectifs.
11	Autonomie décisionnelle	Permet aux ressources de pouvoir prendre des décisions de haut niveau.
12	Sécurité	Concerne tous les aléas et risques (Bruit, éblouissement lumineuse) que peuvent générer le travail d'une ressource dans un environnement donné. Un grand nombre d'aspect sécuritaire est inclus dans la collaborativité des ressources.
13	Localisation	L'emplacement ou position géographique d'une ressource peut être un atout stratégique surtout pour les ressources immobiles
14	Mobilité	Fonction permettant à une ressource de pouvoir se déplacer dans les endroits accessibles. Ce paramètre accru la participation aux coalitions dans le cas des tâches de transport.
15	Accessibilité	Capacité d'une ressource à pouvoir récupérer ou déposer des objets dans des endroits difficilement accessibles ou à risque.

N°	Paramètres	Description
16	Interopérabilité	Capacité d'une ressource à pouvoir communiquer ou échanger des informations directement avec les autres éléments de l'atelier

4.5.5 Notations

Indices et ensembles

\mathcal{V}	Ensemble des AGVs capables de former une coalition pour répondre à une tâche de transport proposée par le superviseur
\mathcal{A}	Ensemble des bras manipulateurs
\mathcal{H}	Ensemble des opérateurs humains
\mathcal{R}^v	Ensemble des ressources capables d'appartenir à la coalition avec v , $\mathcal{R}^v = \{v\} \cup \mathcal{A} \cup \mathcal{H}$ et $\mathcal{R} = \{1, \dots, \mathcal{R}^v \}$
\mathcal{K}	Ensemble des tâches de transport
K	Ensemble des indices relatives aux éléments de l'ensemble \mathcal{K}
\mathcal{I}^k	Ensemble des opérations d'une tâche, $\mathcal{I}^k = \{1, \dots, \mathcal{I}^k \}$
v	Indice des AGVs, chaque AGV est en charge d'une coalition
k	Indice des tâches, chaque tâche est composée d'un ensemble d'opérations, $k \in K$
l_k	Nombre d'opérations dans une tâche d'indice k , i.e. $l_k = \mathcal{I}^k $
i	Indice des opérations, $i \in \mathcal{I}^k$
r	Indice des ressources de transport, $r \in \mathcal{R}^v$
j	Indice des produits
σ	Indice des stocks, i.e. l'emplacement des zones tampon d'entrée\sortie des machines et d'autres postes, où sont stockés les matières premières et produits finis

Paramètres

p_r	Position actuelle de la ressource r
$d_{\sigma, \sigma'}$	Distance entre la position σ et du stock σ'
s_r	Vitesse de déplacement de la ressource mobile r
<i>horizon</i>	Temps maximum dans lequel s'effectuera l'ensemble des tâches prévues.
$t_{i,r,j}$	Moment auquel la ressource r et le produit p seront disponibles pour l'opération i
tl_r	Temps requis par r pour charger ou décharger n'importe quel produit, avec $r \in \mathcal{A} \cup \mathcal{H}$
$tt_{r,\sigma,\sigma'}$	Temps requis par la ressource r pour aller du stock σ vers le stock σ' , avec $tt_{r,\sigma,\sigma'} = \frac{d_{\sigma,\sigma'}}{s_r}$
$\sigma_{i,k}^s$	Indice du stock source ou point de départ de l'opération i de la tâche d'indice k
$\sigma_{i,k}^d$	Indice du stock destination ou point d'arrivée de l'opération i de la tâche d'indice k
$\tau_{r,i,k}$	= tl_r si l'opération i de la tâche d'indice k est une opération de chargement ou déchargement; $tt_{r,\sigma_{i,k}^s,\sigma_{i,k}^d}$ sinon (Opération de transport)

$Cmax_k^v$	Date de fin de la tâche d'indice k ou de livraison du produit associé à k par la coalition v
γ_r	= 1 si la ressource r est mobile; 0 sinon
$\psi_{r,i,k}$	= 1 si la ressource r est capable d'exécuter l'opération i de la tâche d'indice k ; 0 sinon
$\lambda_{r,\sigma}$	= 1 si la ressource r est située au niveau de σ ; 0 sinon
$\delta_{r,i,k}$	Durée de l'opération i de la tâche d'indice k par la ressource r = $\tau_{r,i,k}$ si $\lambda_{r,\sigma_{i,k}^s} = 1$; $\gamma_r \cdot (\tau_{r,i,k} + tt_{r,p_r,\sigma_{i,k}^s})$ sinon
Variabes de décision	
$Y_{r,i,k}$	= 1 si la ressource r est en charge de l'opération i de la tâche d'indice k ; 0 sinon
$S_{r,i,k}$	Date de début de l'opération i de la tâche d'indice k par la ressource r
W_k	Coalition gagnante de la tâche d'indice k choisie par le superviseur dans l'algorithme de décision globale

4.5.6 Algorithme de Décision Locale : ADL

Le modèle de décision local basé sur l'ILP est implémenté sur chaque AGV. Il vise à trouver la meilleure coalition de ressources, tout en minimisant le temps d'exécution de la tâche de transport. Une coalition est nécessairement formée par : (i) un AGV chargé de déplacer le produit et *leader* de la coalition; et (ii) une ou deux ressources chargées des opérations de chargement et de déchargement du produit aux endroits prédéfinis.

4.5.6.1 Complexité du problème

Le problème de formation de coalition robotique est connu comme un problème NP-complet (NP-Hard problem) [146]. Dans notre cas, il faut en plus prendre en compte les opérateurs humains qui entrent dans le processus de formation de coalition, mais aussi les possibilités de début d'opération que peuvent avoir l'ensemble des ressources éventuellement candidates à la coalition. Ainsi deux facteurs permettent d'évaluer la complexité du problème : le nombre de ressources éligibles à une coalition pour un AGV ($|\mathcal{C}^v|$), et les dates de début de chaque ressource (\mathcal{S}^v).

Le nombre de coalitions possibles pour chaque AGV peut s'exprimer à travers l'expression (4.1). Ce nombre se traduit par une somme d'arrangements (A_n^k).

L'AGV peut faire une coalition avec un bras ou un humain parmi tous les bras ou les humains (respectivement $A_{|\mathcal{A}|}^1$ ou $H_{|\mathcal{H}|}^1$). Il peut aussi choisir deux bras ou deux humains. Tout comme il a la possibilité de choisir un bras parmi les bras puis un humain parmi l'ensemble des humains ($A_{|\mathcal{A}|}^1 \times A_{|\mathcal{H}|}^1$). Ce dernier produit est doublé dans l'expression (4.1), car l'ordre du choix est important. En d'autres termes, une coalition $\{r_1, v_2, r_4\}$ (avec r = un bras ou un humain et v l'AGV en charge de la coalition) est différente d'une coalition formée dans l'ordre $\{r_4, v_2, r_1\}$. Cette différence s'explique par le fait que le temps (ou coût) de réalisation d'une opération par une ressource dépend de plusieurs paramètres comme la position de la ressource par rapport au lieu

$$\begin{aligned}
 |\mathcal{C}^v| &= A_{|\mathcal{A}|}^1 + A_{|\mathcal{H}|}^1 + A_{|\mathcal{A}|}^2 + 2 \times A_{|\mathcal{A}|}^1 \times A_{|\mathcal{H}|}^1 + A_{|\mathcal{H}|}^2 \\
 &= |\mathcal{A}| + |\mathcal{H}| + \frac{|\mathcal{A}| \times (|\mathcal{A}| - 1) \times \cancel{(|\mathcal{A}| - 2)!}}{\cancel{(|\mathcal{A}| - 2)!}} + 2 \times |\mathcal{A}| \times |\mathcal{H}| + \\
 &\quad \frac{|\mathcal{H}| \times (|\mathcal{H}| - 1) \times \cancel{(|\mathcal{H}| - 2)!}}{\cancel{(|\mathcal{H}| - 2)!}} \tag{4.1} \\
 &= |\mathcal{A}| + |\mathcal{H}| + |\mathcal{A}|^2 - |\mathcal{A}| + 2 \times |\mathcal{A}| \times |\mathcal{H}| + |\mathcal{H}|^2 - |\mathcal{H}| \\
 &= (|\mathcal{A}| + |\mathcal{H}|)^2
 \end{aligned}$$

Avec $A_n^k = \frac{n!}{(n-k)!}$ le nombre d'arrangements de k éléments parmi n , $|X|$ le cardinal de ensemble X , et $\mathcal{C}^v = \mathcal{A}^2 \cup \mathcal{A} \times \mathcal{H} \cup \mathcal{H}^2$ ensemble des coalitions possibles pour l'AGV v .

La date de début des opérations par les différentes ressources quant à lui est plus complexe à exprimer, car il prend en compte plusieurs paramètres tels que la disponibilité des ressources, celle du produit, la durée de l'opération et aussi les contraintes liées aux opérations (Contraintes de précédence par exemple). Néanmoins on peut l'exprimer de la façon suivante :

$$\mathcal{T}^v = \sum_{i \in \mathcal{I}^k} \sum_{r \in \mathcal{R}^v} (\text{horizon} - t_{i,r,k}) \forall k \in K \tag{4.2}$$

L'expression de \mathcal{T}^v dans la relation (4.2), permet d'écrire par la suite l'ensemble des possibilités de début d'opération $\mathcal{T}_{unitaire}^v$ quelle que soit la ressource r pour une opération i :

$$\mathcal{T}_{unitaire}^v \geq \text{horizon} - t_{i_k,r,j}, \quad \forall r \in \mathcal{R}^v \tag{4.3}$$

Le $\mathcal{T}_{unitaire}^v$ est déduit en fonction des possibilités de dates de début de la dernière opération dans une tâche, qui correspond à la seconde partie de l'inéquation. Celle-ci étant logiquement la moins grande de toutes les autres opérations de la tâche, vue la précédence entre les opérations d'une même tâche.

la Taille du Problème (TP) étudié peut être approximée grâce aux expressions de $|\mathcal{C}^v|$ dans (4.1) et de $\mathcal{T}_{unitaire}^v$ dans (4.3) comme suit :

$$TP > |\mathcal{C}^v| \times \mathcal{T}_{unitaire}^v$$

En remplaçant $|\mathcal{C}^v|$ et $\mathcal{T}_{unitaire}^v$ par leurs expressions on obtient :

$$TP > (|\mathcal{A}| + |\mathcal{H}|)^2 \times (\text{horizon} - t_{i_k,r,j}) \tag{4.4}$$

L'inéquation (4.4), montre une estimation minorée de la taille du problème pour un seul AGV.

TP peut être très large pour un nombre de ressources de chargement/déchargement ($|\mathcal{A}|$ et $|\mathcal{H}|$) important **et/ou** un *horizon* grand. Ceci montre la nécessité d'utiliser une méthode d'optimisation pour la résolution de ce problème de formation de coalition incluant des humains et des robots.

L'estimation de la taille du problème permet aussi de voir l'intérêt de la distributivité dans la prise de décision. En effet, si l'approche était centralisée, ce qui implique une optimisation faite par une seule entité, alors la taille du problème estimée dans (4.4) serait encore multipliée par le nombre total d'AGV dans l'atelier ($TP \times |\mathcal{V}|$). Le temps de calcul ou de recherche d'une solution optimale pour une tâche aurait aussi été multiplié par le nombre d'AGV ($|\mathcal{V}|$). Ce qui n'est pas négligeable surtout dans les entrepôts déployant un grand nombre d'AGVs.

4.5.6.2 Hypothèses

A l'instar d'un grand nombre de problème de recherche opérationnelle, nous avons émis un ensemble d'hypothèses (suppositions) en vue de formaliser un modèle mathématique.

- (S1) La configuration de l'atelier est définie et ne peut être modifiée. L'assemblage ou le désassemblage des manipulateurs mobiles (AGV + bras robotique) a déjà été fait au niveau décisionnel tactique, c'est-à-dire que la décision de ces actions n'est pas prise à ce niveau qui reste opérationnel (Voir section 2.2.2 du chapitre 2 pour les différents types de décision). Cependant, même si un bras robotique donné est déjà joint à un AGV, il peut faire partie d'une coalition dirigée par un autre AGV.
- (S2) Seuls les AGVs sont en charge de la formation de coalition.
- (S3) Chaque AGV ne peut transporter qu'un seul produit à la fois et dispose de suffisamment d'espace pour transporter n'importe quel type de produit, même s'il est équipé d'un bras robotisé.
- (S4) Chaque AGV conserve l'endroit où il a exécuté la dernière opération (opération de déchargement du produit) de la dernière tâche de transport qu'il a exécuté, et ceci jusqu'à ce qu'une autre tâche lui soit assignée. Néanmoins, l'AGV se positionne de façon à ne pas entraver le déplacement des autres AGVs.
- (S5) Les bras robotiques reliés aux AGVs sont considérés comme des bras mobiles et ont des emplacements variables. Les autres bras sont considérés comme des bras fixes, qui sont affectés à des emplacements spécifiques ne pouvant être modifiés.
- (S6) Les opérateurs ne peuvent effectuer que des opérations de chargement/déchargement et sont considérés comme des bras fixes, c'est-à-dire affectés à des emplacements spécifiques.
- (S7) Chaque bras manipulateur et opérateur dispose d'une charge utile, c'est-à-dire le poids maximum qu'il peut manipuler en toute sécurité.

(S8) Les emplacements de prise et dépose des produits sont connus et correspondent aux emplacements des différents stocks : de matières premières, d'entrée et de sortie des machines et de produits finis. Les stocks d'entrée et de sortie d'une machine sont localisés au même endroit (Coordonnées géographiques confondues).

4.5.6.3 Fonction objectif

La fonction objectif du modèle de décision locale permet de minimiser la date de fin de tâche par la coalition, en d'autres termes la date de livraison du produit ($Cmax$). Elle représente la somme de l'ensemble des dates de fin d'opérations ($début+durée$) constituant la tâche de transport comme le monte la fonction suivante :

$$\text{Minimiser } \sum_{i \in \mathcal{I}^k} \sum_{r \in \mathcal{R}^v} Y_{r,i,k} \times (\delta_{r,i,k} + S_{r,i,k}), \quad \forall k \in K \quad (4.5)$$

Néanmoins l'enchère qui sera transmis au superviseur après optimisation sera la date de fin de la dernière opération. Le but de la fonction (4.5) est de minimiser les dates de fin de toutes les opérations afin d'aboutir à une date de livraison optimisée (minimisée) du produit, étant donné que les opérations sont reliées et s'enchaînent.

Linéarisation de la fonction objectif

L'expression de la fonction objectif dans (4.5) n'est pas linéaire, dans la mesure où nous avons le produit entre des variables de décision booléenne ($Y_{r,i,k}$) et entière ($S_{r,i,k}$). La résolution de ce type problème au travers d'un ILP nécessite alors une linéarisation.

Ce type de problème a déjà été traité en recherche opérationnelle, notamment par la linéarisation sous contrainte utilisant l'enveloppe de McCormick [108], [30]. Ainsi le produit d'une variable continue y_1 bornée par $[y_1^{low}, y_1^{up}]$ et d'une variable booléenne $y_2 \in [0, 1]$, peut être linéarisé par une variable $z = y_1 \times y_2$ contrainte par :

$$\begin{aligned} z &\leq y_1^{up} \times y_2 \\ z &\leq y_1 - y_1^{low} \times (1 - y_2) \\ z &\geq y_1^{low} \times y_2 \\ z &\geq y_1 - y_1^{up} \times (1 - y_2) \end{aligned} \quad (4.6)$$

Par analogie à notre cas, on peut poser : $Z_{r,i,k} = Y_{r,i,k} \times S_{r,i,k}$, $\forall r \in \mathcal{R}^v, \forall i \in \mathcal{I}^k$ et $k \in K$. En remplaçant le produit $Y_{r,i,k} \times S_{r,i,k}$ par $Z_{r,i,k}$ dans (4.5), la fonction objectif devient linéaire et s'exprime par :

$$\text{Minimiser } \sum_{i \in \mathcal{I}^k} \sum_{r \in \mathcal{R}^v} Y_{r,i,k} \times \delta_{r,i,k} + Z_{r,i,k} \quad (4.7)$$

Sachant que la variable de début d'opération $S_{r,i,k}$ est bornée par $[0, horizon]$ et $Y_{r,i,k}$ est booléenne, les inéquations (contraintes) de McCormick pour $Z_{r,i,k}$

sont :

$$\begin{aligned}
Z_{r,i,k} &\leq \text{horizon} \times Y_{r,i,k} \\
Z_{r,i,k} &\leq S_{r,i,k} \\
Z_{r,i,k} &\geq 0 \\
Z_{r,i,k} &\geq S_{r,i,k} - \text{horizon} \times (1 - Y_{r,i,k})
\end{aligned}
, \forall r \in \mathcal{R}^v, \forall i \in \mathcal{I}^k \text{ et } \forall k \in K \quad (4.8)$$

4.5.6.4 Contraintes

Hormis les contraintes de McCormick, la fonction objectif telle que définie par l'expression (4.7), est assujettie à un ensemble de contraintes.

(C1.1) : Exprimée par (4.9), cette contrainte permet de garantir que chaque opération ne peut avoir qu'une seule ressource gagnante.

$$\sum_{r \in \mathcal{R}^v} Y_{r,i,k} = 1, \quad \forall i \in \mathcal{I}^k \text{ et } \forall k \in K \quad (4.9)$$

(C1.2) : Celle-ci est complémentaire à la contrainte (C1.1), et garantit que les ressources ayant gagnées les tâches disposent des performances ($\psi_{r,i,k}$) physiques requises par l'ensemble des opérations. Par exemple, seul un bras robotisé ou un opérateur humain ayant une charge utile de plus 5kg peut réaliser une opération nécessitant le chargement d'un produit de 5kg. (C1.2) est traduite par l'équation (4.10).

$$\sum_{i \in \mathcal{I}^k} \sum_{r \in \mathcal{R}^v} Y_{r,i,k} \times \psi_{r,i,k} = l_k, \quad \forall k \in K \quad (4.10)$$

(C2.1) : La contrainte géographique dans la relation (4.11) indique que, seules les ressources étant localisées à l'emplacement de l'opération ($\lambda_{r,i,k,\sigma}$) **et/ou** ayant la faculté de s'y rendre à cet emplacement grâce à leurs mobilités (γ_r) sont aptes à remporter les opérations. Par exemple une ressource fixe située à l'emplacement σ_1 ne peut en aucun cas remporter une opération se déroulant au niveau de σ_3 (Exception faite pour le cas où les deux emplacements sont très proches et peuvent être joints par une même ressource fixe, sans que la position de sa base ne change).

$$\sum_{r \in \mathcal{R}^v} Y_{r,i,k} (\gamma_r + \lambda_{r,i,k,\sigma}) \geq 1, \quad \forall i \in \mathcal{I}^k \text{ et } \forall k \in K \quad (4.11)$$

(C2.2) : Pour renforcer la contrainte géographique, les ressources immobiles (fixes) et qui ne sont pas bien situées pour réaliser une opération ($\gamma_r = 0$ et $\lambda_{r,\sigma_{i,k}} = 0$) ont une durée d'opération nulle ($\delta_{r,i,k} = 0$) car γ et λ conditionnent δ . Étant donné que notre fonction objectif vise à minimiser, une ressource fournissant une durée d'opération nulle, représente une bonne solution bien qu'elle ne satisfait pas la condition géographique. Néanmoins, grâce à la contrainte (C2.2) reliée à l'équation (4.12), ce type de ressource ne peut sortir gagnant de l'opération encourue, car elle contraint toutes les ressources gagnantes à avoir une durée d'opération au moins égal à $\tau_{r,i,k}$. Ce dernier étant un temps minimum propre à

chaque ressource pour l'exécution d'une opération i . Une autre solution consisterait à octroyer une durée d'opération $\delta_{r,i,k}$ infinie (ou très grande) à ces ressources, pour qu'elles ne soient jamais choisies pour ces opérations dont elles ne satisfont pas les conditions géographiques. Cependant, cette dernière solution ne s'applique que pour les fonctions objectifs qui vise à minimiser un paramètre.

$$\sum_{r \in \mathcal{R}^v} Y_{r,i,k} \times \delta_{r,i,k} \geq \sum_{r \in \mathcal{R}^v} Y_{r,i,k} \times \tau_{r,i,k}, \quad \forall i \in \mathcal{I}^k \text{ et } \forall k \in K \quad (4.12)$$

(C3) : Aucune ressource ne peut commencer une opération i d'une tâche d'indice k , sans que la ressource et le produit associé à la tâche ne soient disponibles ($t_{r,i,k}$). Ainsi, la date de début d'une opération doit être toujours plus grande que le moment auquel la ressource en charge et le produit sont libres. La relation (4.13) traduit cette contrainte.

$$Z_{r,i,k} \geq Y_{r,i,k} \times t_{r,i,k}, \quad \forall r \in \mathcal{R}^v, \forall i \in \mathcal{I}^k \text{ et } \forall k \in K \quad (4.13)$$

(C4) : Au sein d'une tâche, il existe une contrainte de précédence entre les opérations. Ainsi, le produit ne peut être chargé (opération i_1) sans que l'AGV ne soit sur place (opération i_0), et l'AGV ne peut partir (opération i_2) sans que le chargement ne soit terminé (opération i_1). De même, le déchargement (opération i_3) ne peut commencer sans que l'AGV ne soit arrivé au lieu de déchargement. Cette contrainte permet donc le respect des séquences entre les opérations ($i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow i_3$). La contrainte (C4) est écrite sous forme temporelle dans l'inéquation (4.14). Dans cette dernière, la date de fin d'une opération doit être inférieure ou égale à la date de début de la prochaine opération. Cette condition reste valable pour toutes les opérations d'une tâche.

$$\sum_{r \in \mathcal{R}^v} (Y_{r,i,k} \times \delta_{r,i,k} + Z_{r,i,k}) \leq \sum_{r' \in \mathcal{R}^v} Z_{r',i',k}, \quad \forall k \in K, \text{ et } i, i' \in (\mathcal{I}^k)^2$$

avec $i' = i + 1$

(4.14)

4.5.6.5 Soumission d'enchère

Après le processus d'optimisation, chaque AGV ayant trouvé une solution représentant un ensemble de ressources qui minimise la livraison du produit, envoie son offre au superviseur. La proposition envoyée correspond à son $Cmax^v$, qui se traduit par l'équation (4.15).

$$Cmax_k^v = Y_{r,l_k,k} \times \delta_{r,l_k,k} + Z_{r,l_k,k} \quad (4.15)$$

4.5.7 Algorithme de Décision Globale : ADG

L'ADG est une instance de l'approche d'ordonnancement développée dans le chapitre 3 (voir Section 3.3.3). La partie optimisation basée sur un GA est

néanmoins supprimée dans cette instance, car les tâches sont assignées une après l'autre. Il n'y a donc pas de besoin d'optimiser.

L'objectif de la décision globale est de s'assurer que la coalition optimale de robots est sélectionnée pour chaque tâche de transport soumise aux ressources de l'atelier.

L'ADG est basé sur un processus d'enchères dans lequel le superviseur, représentant l'entité de haut niveau, soumet une tâche de transport et reçoit les offres de la part des AGVs. Comme le montre la figure 4.7, l'entité de décision globale sélectionne et envoie une tâche aux AGVs en charge des coalitions. Ces derniers, répondent en proposant la coalition qui minimise le temps de livraison du produit de la tâche (C_{max}^v) grâce à un modèle de décision local basé sur un ILP. La sélection d'un gagnant par l'ADG parmi les propositions reçues se fait au travers d'une approche déterministe. Ainsi, la coalition ayant le plus petit C_{max}^v remporte l'enchère, comme l'indique l'équation (4.16).

$$W_k = \arg \min_{v \in \mathcal{V}} (C_{max}^v), \quad \forall k \in K \quad (4.16)$$

Chaque tâche de transport soumise aux ressources de bas de niveau doit avoir au moins une proposition de coalition même si le C_{max}^v est élevé, afin de garantir un gagnant pour l'ensemble des tâches de transport.

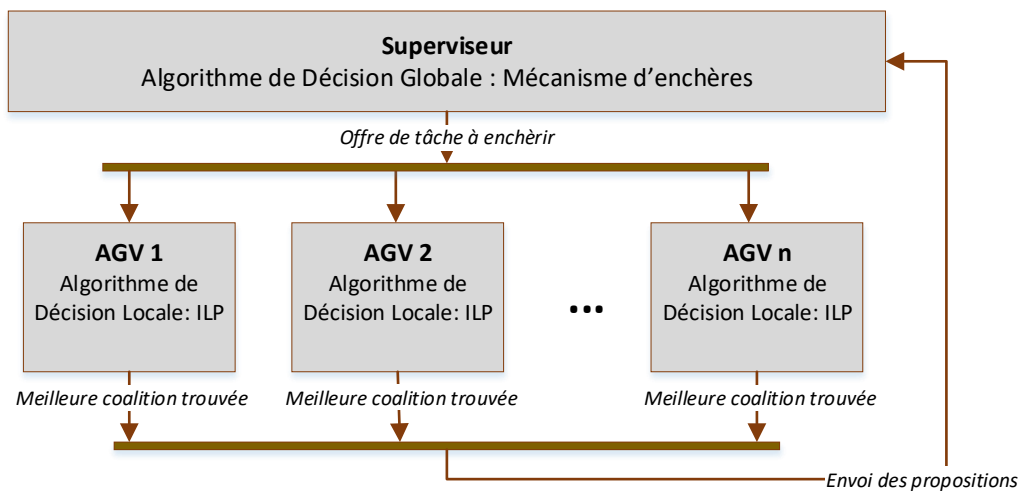


FIGURE 4.7 – Structure de l'approche DISCOFT

4.5.7.1 Ajustement des Buffers des gagnants

Après avoir déterminé un gagnant pour la tâche dans l'ADG, le superviseur se charge de prévenir les ressources gagnantes pour qu'elles puissent actualiser leurs Buffers. Cette étape est importante car permet d'évaluer la disponibilité des ressources et des produits pour les prochaines tâches. Les valeurs de $t_{r,i,k,j}$ changent alors chaque fois que l'algorithme 7 est exécutée.

La mise à jour pour les ressources de transport est effectuée seulement après

une tâche de transport, mais pour les produits elle est faite aussi pour les tâches de production. En effet, un produit qui doit subir sa prochaine tâche de production sur la même machine que précédemment n'a pas besoin de transport, on parle dans ce cas de *transport muet*. Ce type d'action n'a nullement besoin des ressources de transport, donc aucun changement pour sur leurs Buffer et position par rapport au *transport muet*. Par contre, le Buffer de temps du produit change, et prend en compte la durée de la nouvelle tâche de production sans que sa position ne change.

Dans l'algorithme 7 toutes les ressources appartenant à la coalition ($r \in W_k$) voient leurs Buffer et position mis à jour. Cet ajustement commence d'abord par l'AGV responsable du groupe. L'AGV v encaisse toute la durée de la tâche car elle reste mobilisée pour l'ensemble des opérations et prend la position de la dernière opération ($\sigma_{l_k,k}^d$). Ainsi pour les prochaines tâches, l'AGV v ayant remporté la tâche $k-1$ sera indisponible dans l'intervalle $[S_{r_0,0,k-1}, Cmax^v]$, avec $S_{r_0,0,k-1}$ la date de début la première opération ($i=0$) par la ressource $r_0 \in W_{k-1}$.

Le produit j transporté pendant la tâche k , reçoit la même mise à jour que l'AGV, car il reste aussi actif durant à peu près toute la tâche.

Pour les gagnants des autres opérations ($r \in W_k$, avec $r \neq v$), ils reçoivent les temps d'occupation des opérations qui les affectent ainsi que les dernières positions de ces opérations.

Par ailleurs, lorsque deux ressources sont jointes (r et r_joint forment un manipulateur mobile), alors leurs Buffer et position restent identiques. Ainsi, lorsqu'une opération ou tâche est assignée à un des éléments d'un manipulateur mobile, l'ensemble du manipulateur (AGV+Bras) demeure occupé sur la toute durée de l'action.

```

Pour tout  $r$  compris dans  $W_k$  faire
  [Mise à jour AGV en charge coalition et produit j]
  Si ( $r$  correspond à  $v$ ) Alors
     $Buffer_v \leftarrow Cmax^v$ 
     $p_v \leftarrow \sigma_{l_k,k}^d$ 
     $(Buffer_j, p_j) \leftarrow (Buffer_v, p_v)$ 
  Fin Si
  Si ( $r$  est jointe à une ressource  $r\_joint$ ) Alors
    Si ( $r\_joint$  et  $r$  différents de  $v$ ) Alors
       $Buffer_r \leftarrow S_{r,i,k} + \delta_{r,i,k}$ 
       $p_r \leftarrow \sigma_{i,k}^d$ 
       $(Buffer_{r\_joint}, p_{r\_joint}) \leftarrow (Buffer_r, p_r)$ 
    Sinon
      Si ( $r\_joint$  équivaut à  $v$ ) Alors
         $(Buffer_r, p_r) \leftarrow (Buffer_v, p_v)$ 
      Fin Si
    Fin Si
  Sinon
    Si ( $r$  n'est jointe à aucune ressource et  $r$  différent de  $v$ ) Alors
       $Buffer_r \leftarrow S_{r,i,k} + \delta_{r,i,k}$ 
       $p_r \leftarrow \sigma_{i,k}^d$ 
    Fin Si
  Fin Si
Fin Pour

```

ALGORITHME 7 – Fonction mise à jour ressources et produits

4.6 Conclusion

L'approche qui a été mise en oeuvre dans ce chapitre permet de répondre efficacement à plusieurs problématiques industrielles en particulier celle du transport robotisé dans le contexte de l'I4.0.

L'approche DISCOFT intègre plusieurs fonctionnalités telles que : Une prise de décision distribuée permettant une réduction de la charge du superviseur et une exploitation de l'intelligence des ressources ; Un plan de transport plus précis, complet et optimisé avec un plan de production adapté, ce dernier est optimisé aussi grâce à l'approche IT2P ; Une prise en compte de la modularité des ressources grâce au principe de coalition.

En définitive, une description détaillée allant de l'émission d'un ordre de production à l'allocation aux ressources (Machines, robots et humains) en passant par plusieurs mécanismes de décisions et d'optimisation est fournie dans ce chapitre. L'efficacité de l'approche DISCOFT proposée sera expérimentée dans le chapitre suivant.

Chapitre 5

Cas d'étude : Démonstrateur du futur

5.1 Introduction

Dans ce chapitre nous présentons l'étude expérimentale qui a été réalisée en vue de mettre en oeuvre et valider l'approche DISCOFT. Nous avons modélisé un cas d'étude sur la plateforme du démonstrateur du futur au LI-NEACT du Cesi de Rouen.

L'ensemble des ressources tant logicielles que matérielles utilisé dans les expériences est décrit, ainsi que les différents essais ayant permis de montrer la faisabilité, l'efficacité et la comparaison de l'approche par rapport à d'autres approches existantes.

5.2 Choix du solveur

La résolution des problèmes complexes par une méthode d'optimisation requiert l'utilisation d'un programme ou logiciel informatique dédié. Ces logiciels, portent l'appellation de solveur (*en français solveur*). Il en existe une large gamme [55] : ILOG CPLEX¹, LiPS², MINOS³, Excel⁴, OR-Tools⁵, etc. Le choix d'un solveur dépend du type de problème, du domaine d'application et de la complexité du problème. Dans notre cas, un solveur capable de prendre en charge les langages *Python*, *C++* et *Java* et disponible pour les plateformes *Linux* et *Windows* est visé. Ces paramètres réunis, facilitent l'implémentation sur les robots et les différentes ressources dont nous disposons.

OR-Tools

Pour la résolution des problèmes formulés (Sections 4.4 et 4.5 du chapitre 4) nous avons utilisé des composants d'OR-Tools développée par Google.

1. <https://www.ibm.com/fr-fr/products/ilog-cplex-optimization-studio>, consulté le 19 août 2020

2. <http://lipside.sourceforge.net/>, consulté le 20 septembre 2020

3. http://www.sbsi-sol-optimize.com/asp/sol_product_minos.htm, consulté le 20 septembre 2020

4. http://www.bernardcordier.com/excel_solveur.htm, consulté le 20 septembre 2020

5. <https://developers.google.com/optimization/install>, consulté le 20 septembre 2020

OR-Tools est un meta-solveur compatible avec la plus part des systèmes d'exploitation (Linux, MAC OS, Windows) avec une prise en charge de quatre langages de programmation : Python, C++, Java et C# (Dot Net) [176]. C'est une plateforme open source avec une communauté de développeurs très active et largement utilisée pour la résolution des problèmes d'optimisation comme l'illustre la figure 5.1 [110], [117].

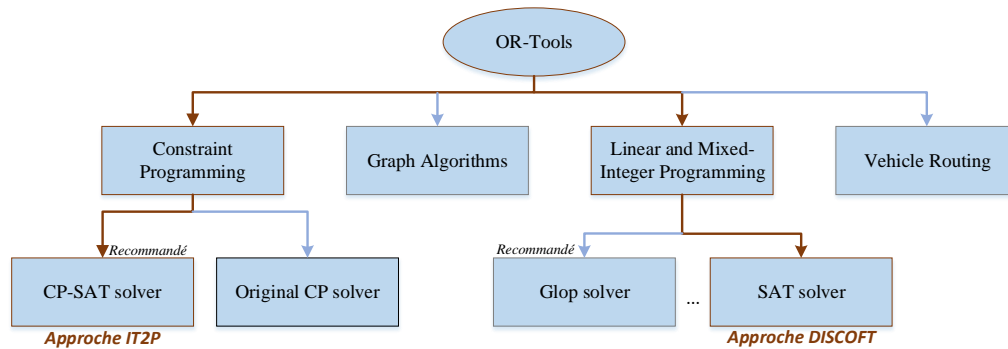


FIGURE 5.1 – Positionnement des solveurs utilisés

Problèmes traités et solveurs

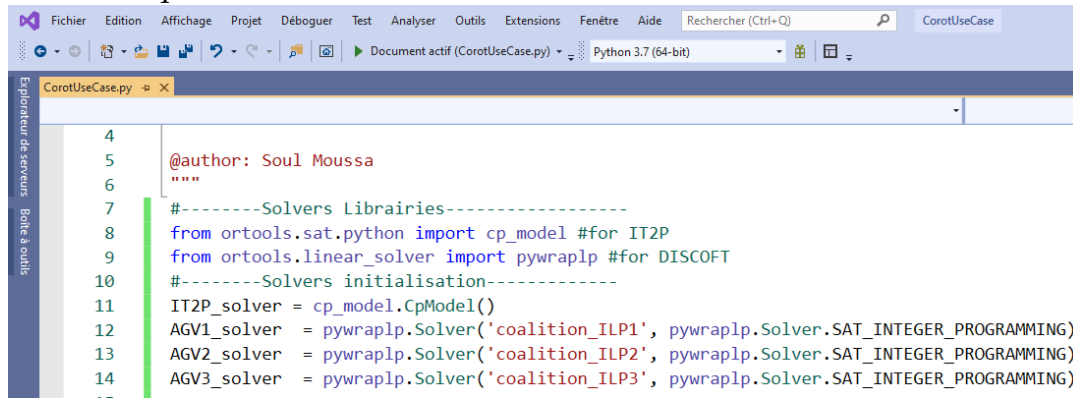
On s'intéresse à deux méthodes d'optimisation dans notre cas d'étude : CP et ILP. L'approche IT2P qui est formalisée comme un problème de type CP [111], est résolue avec le solveur recommandé par OR-Tools pour cette catégorie, CP-SAT solver.

L'approche DISCOFT, est basée sur un ILP au niveau de l'ADL (Algorithme de Décision Locale) avec un ensemble de variables binaire et entière. Le solveur recommandé sur OR-Tools est le Glop solveur qui se base sur l'*algorithme du simplexe* [82]. Néanmoins, Glop solveur considère toutes les variables comme étant réelles, donc il est plus adapté plus pour le MILP que l'ILP. Le SAT solveur⁶ appartenant à la même catégorie (Voir figure 5.1), est quant à lui plus orienté vers l'ILP car permet spécifiquement la résolution des modèles comportant des variables booléennes et entières.

Cependant, SAT solver de son nom complet SAT_INTEGER_PROGRAMMING, n'est pas fourni lors de l'installation *binaire* de OR-Tools, il faut donc procéder à une installation depuis le *code source*. L'installation depuis le *code source*, nécessite une configuration *software* adéquate.

Sur Windows, pour le langage Python les versions : *Microsoft Visual Studio 2019*, *CMake 3.5* ou plus récent et au moins *Python 3.6* sont requises pour installer OR-Tools depuis le code source. Ce type d'installation est plus complète et permet aussi une personnalisation des bibliothèques. Nous l'avons opté pour pouvoir résoudre efficacement notre problème.

6. http://google.github.io/or-tools/python/ortools/linear_solver/pywraplp.html, visité le 20 septembre 2020



```

4
5 @author: Soul Moussa
6
7 #-----Solvers Librairies-----
8 from ortools.sat.python import cp_model #for IT2P
9 from ortools.linear_solver import pywraplp #for DISCOFT
10 #-----Solvers initialisation-----
11 IT2P_solver = cp_model.CpModel()
12 AGV1_solver = pywraplp.Solver('coalition_ILP1', pywraplp.Solver.SAT_INTEGER_PROGRAMMING)
13 AGV2_solver = pywraplp.Solver('coalition_ILP2', pywraplp.Solver.SAT_INTEGER_PROGRAMMING)
14 AGV3_solver = pywraplp.Solver('coalition_ILP3', pywraplp.Solver.SAT_INTEGER_PROGRAMMING)
--

```

FIGURE 5.2 – Instanciation des sovs sous Visual studio 2019

5.3 Description du cas d'étude

Le cas d'étude est réalisé sur la plateforme LINEACT du Cesi de Rouen, constituée de plusieurs postes d'assemblages et de robots. La figure 5.3 est un schéma global du cas d'étude avec l'ensemble des ressources et postes, des Types de tâche qui sont réalisées au niveau de chaque poste et de leurs emplacements (p_0, s_0, \dots, s_5). Les ressources utilisées et produits assemblés, sont décrits dans les sections suivantes.

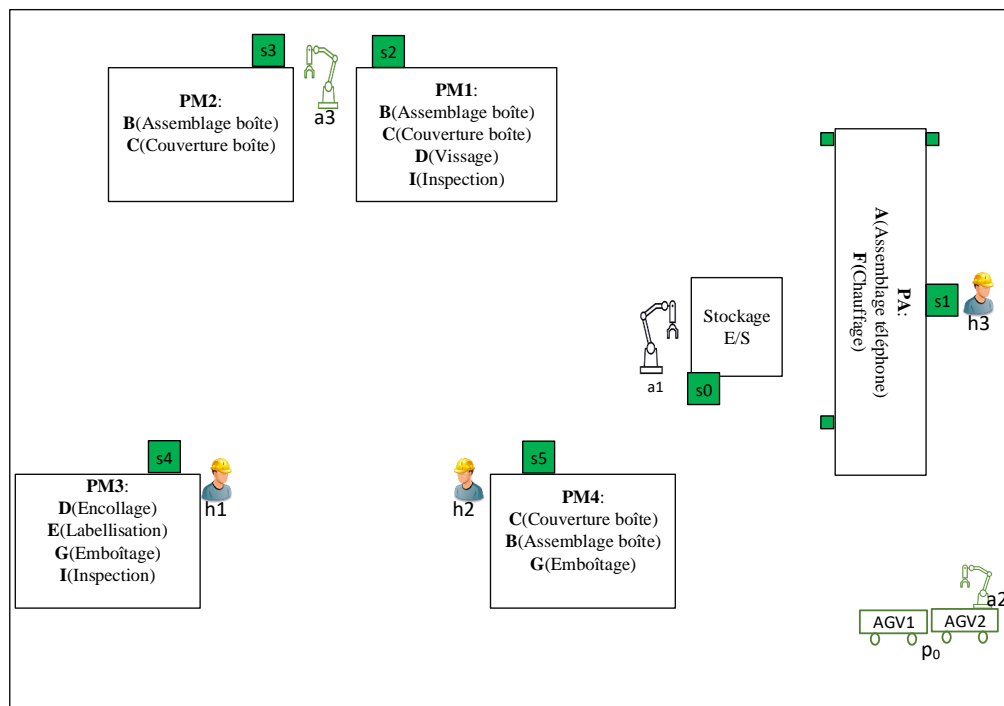


FIGURE 5.3 – Vue d'ensemble du cas d'étude

5.3.1 Ressources de production

Ces ressources effectuent toutes les tâches dans l'atelier en dehors de celles de transport. Ces tâches qui sont réparties sur les postes de l'atelier, figure 5.3. Deux types de poste sont utilisés dans notre cas étude, les manuels et une automatique.

5.3.1.1 Poste automatique

Le Poste Automatique (PA) ou démonstrateur du futur est une ligne d'assemblage flexible de téléphone, figure 5.4. Il est constitué de quatre modules principaux :

- **Module 1** : En charge du stockage des matières premières (coques supérieures des téléphones) et produits finis (téléphone), avec un robot SCARA (quatre axes) qui récupère et/ou dépose des produits du convoyeur vers les stocks et vice-versa.
- **Module 2** : Constitué de deux unités séparées, chacune en charge d'une opération. La première permet le perçage des produits, et la seconde clipse, après ajustement, les deux parties du téléphone (Supérieure et inférieure).
- **Module 3** : Dans ce bloc un four réalise les tâches de chauffage d'un côté et un jumeleur automatique ferme le téléphone en déposant la partie inférieure. Le jumeleur a une réserve de coques inférieures.
- **Module 4** : Un module disposant d'un robot six axes avec trois outils (préhenseurs) possibles. Cette partie de la chaîne grâce à son robot polyvalent gère l'insertion et la manipulation des composants internes du téléphone. Ces derniers sont disponibles dans un stock associé à ce module.

Outre les points de stockage du module 1, le poste PA dispose ainsi de deux autres stocks dont l'un est associé au jumuleur (Coques inférieures) et l'autre au module 4 (Composants internes téléphones). L'ensemble des points de stockage de ce poste sont visibles sur la figure 5.3 avec la même appellation s1⁷.

La chaîne est autonome⁸, et le déplacement des produits entre les différents modules est assuré par des convoyeurs.

5.3.1.2 Postes manuels

Au nombre de quatre, les Postes Manuels (PM) sont occupés par des opérateurs en charge de réaliser toutes les tâches que le poste automatique ne peut réaliser (Labellisation, Inspection, ...), figure 5.5. Chaque poste dispose d'un stock infini de l'ensemble des accessoires et outils nécessaires à son fonctionnement. Les stocks sont préalablement remplis.

7. Le même nom est utilisé pour tous les stocks du poste PA, pour éviter de lui définir plusieurs points géographiques

8. Les ordres d'assemblage sont données à distance, aucune intervention n'est nécessaire pour l'assemblage



FIGURE 5.4 – La chaîne automatisée et ses modules

5.3.1.3 Poste de stockage

Annoté par $E \setminus S$ sur la figure 5.3, ce poste stocke les matières premières de PA et aussi les produits P1 et P2 lorsque qu'ils sont terminés. Notons, que le stockage en soi n'est pas considéré comme une tâche (Aucun temps de production n'est associée) dans notre cas d'étude.

5.3.1.4 Tâches de production

Huit Types de tâche de production sont effectuées sur les postes de l'atelier. Ces tâches vont de l'assemblage des produits, à l'inspection du produit fini. La table 5.1 synthétise les types de tâche.

Le problème traité dans ce cas d'étude est un FJSP [187], ce qui implique qu'une tâche pourrait être réalisée par plusieurs machines avec les mêmes ou différentes performances, comme l'indique la table 5.2.

TABLE 5.1 – Tâches effectuées dans l'atelier

Tâche	Description
A	Assemblage du téléphone
B	Assemblage de la boîte
C	Recouvrement de la boîte
D	Encollage des différentes parties de la boîte
E	Labellisation de la boîte par rapport à la référence du téléphone
F	Chauffage
G	Mise en boîte du téléphone
H	Peinture
I	Inspection du produit fini



FIGURE 5.5 – Poste manuel

TABLE 5.2 – Temps de processus en seconde des postes par type de tâche

Machines	Types de tâche								
	A	B	C	D	E	F	G	H	I
PM1		180		120				240	120
PM2		180	120						
PM3				180	60		60		120
PM4		180		120			60		
PA	240					60			

5.3.2 Produits

Deux types de produits sont assemblés après un ensemble de tâche dans l'atelier. Le premier produit P1, est obtenu en assemblant un téléphone (principalement assemblé par la chaîne automatique) et une boîte dans laquelle sera emballée le téléphone. Le produit P1 est illustré par l'image de la figure 5.11. Le second produit est une boîte vide, figure 5.8. Les différentes parties de la boîte sont façonnées à la découpe laser.

TABLE 5.3 – Séquence des produits assemblés

Type de produit	Séquence
P1	A→F→B→C→D→E→I
P2	B→C→D→E→H→I



FIGURE 5.6 – Partie inférieure boîte



FIGURE 5.7 – Partie supérieure boîte



FIGURE 5.8 – Boîte : Produit 2



FIGURE 5.9 – Partie inférieure téléphone



FIGURE 5.10 – Composants internes téléphone



FIGURE 5.11 – Téléphone emboîté

5.3.3 Ressources de transport

Les ressources de transport utilisés dans ce cas d'étude sont : des opérateurs humains (symbolisés par la lettre h_x , x correspondant au numéro de

l'opérateur), des bras manipulateurs (a_x) et des AGVs. Les manipulateurs mobiles sont considérés comme un AGV avec un bras embarqué. En d'autres termes, deux entités jointes physiquement mais chacune avec son unité de décision.

5.3.3.1 Robots et opérateurs

Les tâches de transport incluant le chargement/déchargement et transfert des produits entre les postes sont assurées par des AGVs, bras robotisés et opérateurs. Deux AGVs de type MIR 100, trois Universal Robots (Deux UR5 et un UR10) et trois opérateurs sont utilisés dans cette expérience. Le nombre d'opérateurs dans l'atelier est limité pour des raisons de fluidité, et pour faciliter ainsi le déplacement des AGVs.

5.3.3.2 Ressource virtuelle

Un troisième AGV, virtuel cette fois-ci, est déclaré pour prendre en charge la création de coalition des tâches de transport ne suscitant pas de déplacement entre deux points. Ainsi ce type de tâche est disponible entre les transferts pouvant être effectués intégralement par des ressources de chargement ou de déchargement. Les postes PM1 et PM2 illustrent bien le rôle de l'AGV virtuel (voir position des deux postes sur figure 5.3). Les deux postes étant presque joints, une ressource de chargement/déchargement peut réaliser la tâche de transport.

Ainsi les ressources de déplacement (AGVs) disposent aussi de la caractéristique "virtuel" ou "réel" qui sera considérée comme une performance et une exigence pour les tâches. L'ensemble des caractéristiques des ressources est résumé dans la table 5.4.

5.3.3.3 Aspect sécuritaire

Hormis la réduction du nombre d'opérateurs présents et l'ergonomie des postes (voir figure 5.14) assurant une fluidité, on remarque dans la table 5.4, de longs temps de processus pour les bras et de petites vitesses (Inférieures à $0.3 \text{ m} \setminus \text{s}$) de déplacements des ressources mobiles. Il faut noter que ces vitesses sont des valeurs moyennes, en effet les vitesses de translation (Déplacements en ligne droite) sont plus élevées mais les AGVs effectuent les rotations lentement pour changer de direction. Pour ce qui est de la vitesse associée aux bras (*Exemple : a_2*), elle correspond à la vitesse de l'AGV porteur et non à la vitesse du mouvement du bras.

Les restrictions sur les vitesses (Pour les bras et les ressources mobiles) qu'on s'est imposé, permettent un respect de la *directive "Machines" 2006\42\CE* et la *norme ISO 10218-1 2011* prévalant sur les exigences et recommandations liées à l'utilisation des robots industriels collaboratifs [167].

Une vision 3D de l'atelier est montrée par la figure 5.14. Ce modèle de l'atelier a été construit sur *Sweet Home 3D*⁹, et a permis la définition des coordonnées géographiques et l'aménagement des postes et ressources.

9. <http://www.sweethome3d.com/fr/>, consulté le 20 septembre 2020

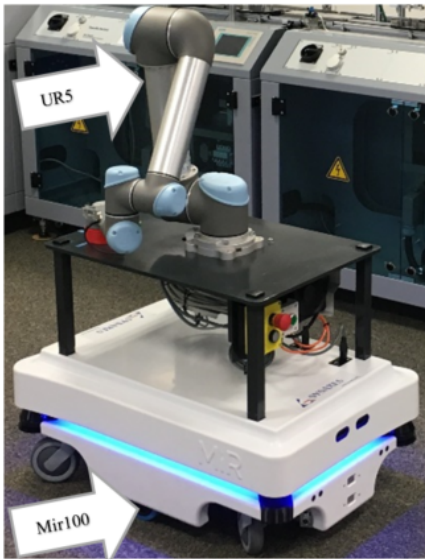


FIGURE 5.12 – Manipulateur mobile : Mir100+UR5

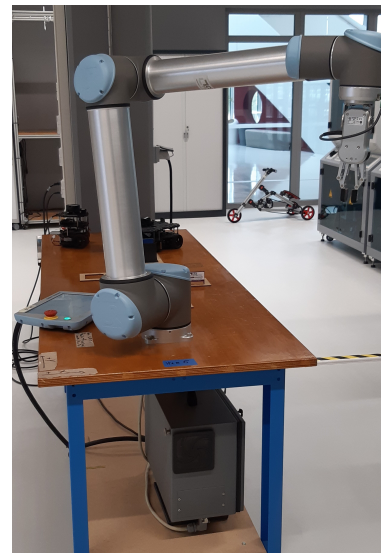


FIGURE 5.13 – Bras manipulateur : UR10

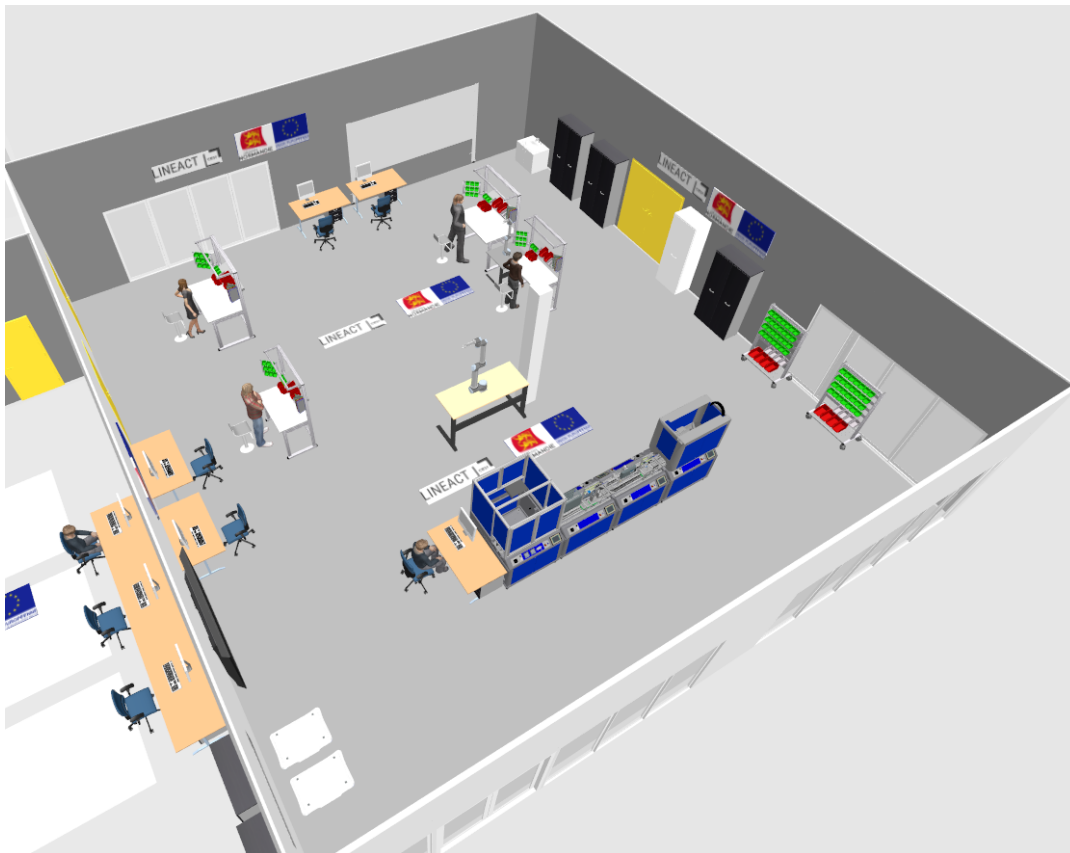


FIGURE 5.14 – Vision 3D de l'atelier

TABLE 5.4 – Caractéristiques des ressources de transport.

Ressource	Charge utile(Kg)	Emplacement initial	Type	Vitesse (m/s)	Temps de process(s)
AGV1	200	p0	Mobile, réel	0.3	-
AGV2	200	p0	Mobile, réel	0.2	-
Agv Virtuel	200	s2	Mobile, virtuel	0.4	-
a1	10	s0	Fixe	0	50
a2	5	p0	Mobile	0.2	55
a3	5	s2	Fixe	0	65
h1	7	s4	Fixe	0	35
h2	7	s5	Fixe	0	30
h3	7	s1	Fixe	0	60

5.4 Expériences et Discussion

5.4.1 Expérience 1

Cette première expérience constitue une mise en oeuvre de DISCOFT en vue de tester sa faisabilité et son efficacité. Les résultats présentés ci-dessous, sont obtenus après la simulation de l'assemblage d'un exemplaire de chaque produit (P1 et P2). En outre, une comparaison entre le planning des ressources proposé par l'approche DISCOFT et celui issu de l'approche l'IT2P, est effectuée. Nous rappelons que cette dernière approche (IT2P) ne prend pas en compte les opérations de chargement et déchargement.

5.4.1.1 Les enchères et la sélection des gagnants

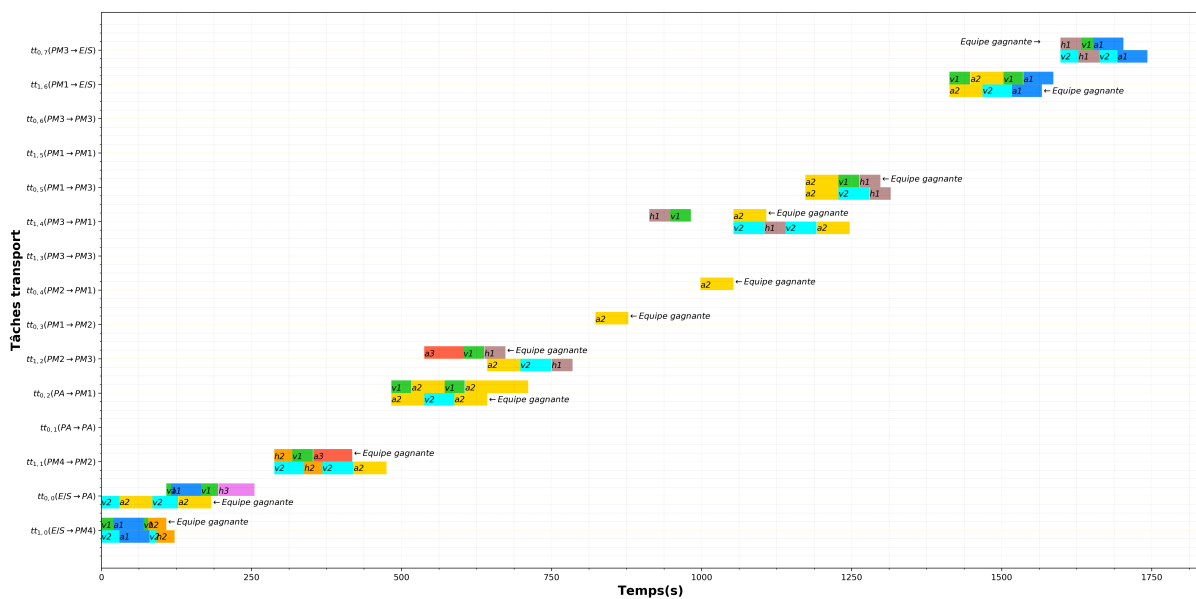


FIGURE 5.15 – Enchères reçues pour l'assemblage de deux produits

Dans cette partie, nous expérimentons les rôles et fonctionnements des deux couches de l'approche. La composition des coalitions pour répondre aux enchères est faite par les ADL.

L'ADG se charge de sélectionner les gagnants des tâches proposées.

Canevas

Les graphes présentés pour cette partie ne sont constituées que des tâches de transport. Bien que les tâches de production aient été prises en compte dans l'assignation. Les espaces ou vides présents entre les tâches de transport correspondent généralement aux temps de production. Les graphes illustrant à la fois les tâches de transport et de production dans cette première expérience sont visibles dans la section 5.4.1.4.

Grâce au diagramme de Gantt dans la figure 5.15, on peut voir les propositions d'enchères reçues de la part des AGVs satisfaisant les exigences des tâches. L'ensemble des tâches de transport sont décrites en ordonnées du graphe (*source* → *destination*). La notation tt_{jk} fait référence à tâche de transport k du Job j . À chaque ressource est associée une couleur, pour une meilleure lisibilité des coalitions auxquelles elles participent. Aussi, chaque ressource est en charge d'une opération. Le nombre de ressources présentes dans la représentation d'une tâche, correspond ainsi au nombre d'opérations que contient cette tâche. À l'exception de l'Agv virtuel qui n'apparaît pas, car ces temps de déplacements sont nuls.

Le Gantt obtenu permet d'observer les propositions optimisées de tous les AGVs éligibles à la tâche en cours. Pour une tâche donnée, l'algorithme de décision locale des AGVs peut converger à l'obtention des mêmes coalitions mais avec des temps de livraison différents.

La figure 5.16 donne un aperçu visuel dans l'atelier sur la formation de

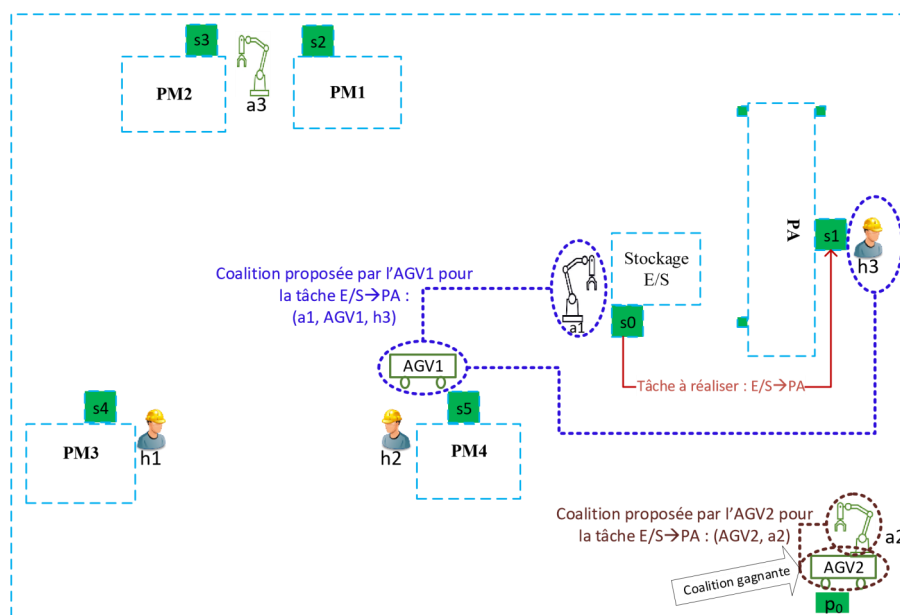


FIGURE 5.16 – Exemple graphique de formation de coalition

coalition pour répondre à une tâche de transport. La tâche consiste à déplacer un produit entre le point de stockage (E/S) et le poste automatique (PA). L'AGV1 après avoir lancé son ADL, sort comme coalition optimale une équipe constituée du bras a1 (fixé au poste E/S) pour le chargement et l'opérateur h3 (fixé au poste PA) pour le déchargement. Du côté de l'AGV2, la meilleure coalition est obtenue avec le bras a2 (monté sur l'AGV2), pour qu'il puisse effectuer les opérations de chargement et déchargement.

Éligibilité et performance des ressources

La ressource *a2*, illustrée par la couleur jaune est présente dans beaucoup d'enchères. Ceci s'explique par son éligibilité à l'ensemble des opérations de chargement et déchargement de toutes les tâches dû à sa mobilité et sa charge utile (5kg) supérieure à la masse de tous les produits en processus. Ainsi, dans un système de coalition la faculté d'une ressource à être sollicitée dépend ses performances. Plus celles-ci sont variées et élevées (Mobilité et bonne charge utile, rapidité etc.), plus la ressource aura une participation accrue.

Atteinte des exigences des tâches

Le tableau 5.5 permet de voir à la fois les exigences des tâches et les propositions faites par les AGVs avec des valeurs numériques précises. Contrairement au diagramme de Gantt dans la figure 5.15 où le nombre de ressources correspond au nombre d'opérations dans une tâche, dans le tableau 5.5 les ressources sont écrites par appartenance à la coalition dans la colonne *Coalitions*. En d'autres termes, même si une ressource gagne deux opérations dans la tâche, son nom n'apparaît qu'une fois dans la colonne *Coalitions*. Les numéros des tâches, dans la première colonne, correspondent au couple (Job tâche). On remarque l'absence de certains numéros (01, 13...), due au fait que le transport est muet. Autrement dit, le produit ne change pas de machine. Dans ces cas, la source équivaut à la destination, comme on peut le voir pour la tâche $tt_{01}(PA \rightarrow PA)$ sur la figure 5.15.

Parmi les exigences se trouve le type de déplacement (Réel ou Virtuel). Les tâches suscitant un type virtuel (03 et 04) ont eu des propositions que de la part de l'*Agv virtuel*, qui est la seule capable de satisfaire cette exigence. Pour les autres tâches, elle n'est pas candidate, contrairement aux AGVs du type réel qui le sont.

Pour les exigences de localisation, seules les ressources mobiles ou bien se trouvant aux emplacements où s'effectuent les opérations (Dans le cas des ressources fixes), apparaissent dans les coalitions. Par exemple, la ressource fixe *a1*, est retrouvée dans les propositions des tâches 10 et 00 parce qu'elle se trouve à l'emplacement *E/S*.

Le respect de l'ensemble de ces exigences, permet d'affirmer la bonne marche de l'ADL implantée sur chaque AGV. Par ailleurs la colonne *Gagnants* du tableau 5.5, permet de retrouver l'intervention de l'ADG. La date de fin de la coalition gagnante, qui est colorée dans la colonne *Fin tâche(s)*, et représente le minimum parmi les propositions.

Coalitions gagnantes

Toutes les coalitions gagnantes des différentes tâches sont représentées dans le diagramme Gantt de la figure 5.17. Dans ce dernier, on peut voir le nom de chaque opération par la nomination tt_{jki} (avec j pour Job, k pour tâche et i pour opération) ainsi que le nom de la ressource gagnante de l'opération, auquel s'ajoute le code couleur associé à la ressource. Ce code couleur, permet d'assurer une traçabilité des ressources sur le Gantt pour s'assurer que tout marche correctement. On doit s'assurer que la diagramme ne présente pas : de présence de ressources à plusieurs endroits à la fois, de chevauchement de temps d'opérations pour la même ressource, de coalition incomplètes, ...

On peut aussi remarquer sur la figure 5.17 le respect des contraintes de précedence entre les opérations d'une tâche puis entre les tâches d'un Job.

TABLE 5.5 – Récapitulatifs de l'assignation des tâches de transport de deux produits.

Tâches	Exigences			Meilleures Coalitions			Gagnants
	Localisation	Déplacement	Masse (Kg)	Coalitions	Début tâche (s)	Fin tâche (s)	
10	$E/S \rightarrow PM4$	Réel	< 1	a1, Agv1, h2 a1, Agv2, h2	0 0	108 122	a1, Agv1, h2
00	$E/S \rightarrow PA$	Réel	< 1	a1, Agv1, h3 a2, Agv2, a2	108 0	255 183	a2, Agv2, a2
11	$PM4 \rightarrow PM2$	Réel	< 1	h2, Agv1, a3 h2, Agv2, a2	288 288	418 475	h2, Agv1, a3
02	$PA \rightarrow PM1$	Réel	< 1	a2, Agv1, a2 a2, Agv2, a2	483 483	711 643	a2, Agv2, a2
12	$PM2 \rightarrow PM3$	Réel	< 1	a3, Agv1, h1 a2, Agv2, h1	538 643	673 785	a3, Agv1, h1
03	$PM1 \rightarrow PM2$	Virtuel	< 1	a2, AgvVirtuel	823	878	a2, AgvVirtuel
04	$PM2 \rightarrow PM1$	Virtuel	< 1	a2, AgvVirtuel	998	1053	a2, AgvVirtuel
14	$PM3 \rightarrow PM1$	Réel	< 1	h1, Agv1, a2 h1, Agv2, a2	913 1053	1108 1247	h1, Agv1, a2
05	$PM1 \rightarrow PM3$	Réel	< 1	a2, Agv1, h1 a2, Agv2, h1	1173 1173	1298 1315	a2, Agv2, h1
16	$PM1 \rightarrow E/S$	Réel	< 1	a2, Agv1, a1 a2, Agv2, h1	1413 1413	1586 1567	a2, Agv2, a1
07	$PM3 \rightarrow E/S$	Réel	< 1	h1, Agv1, a1 h1, Agv2, a1	1598 1598	1703 1743	h1, Agv1, a1

5.4.1.2 Longueur des coalitions

Il est intéressant d'observer le nombre de ressources déployées dans une coalition pour effectuer une tâche, comme le montre la figure 5.18. Ce nombre doit toujours être supérieur ou égal à 2 pour toute tâche car il faut au moins deux entités : une ressource de chargement et déchargement et une ressource pour le transport (en supposant qu'un opérateur humain joue le même rôle qu'un bras manipulateur fixe dans le cadre du transport).

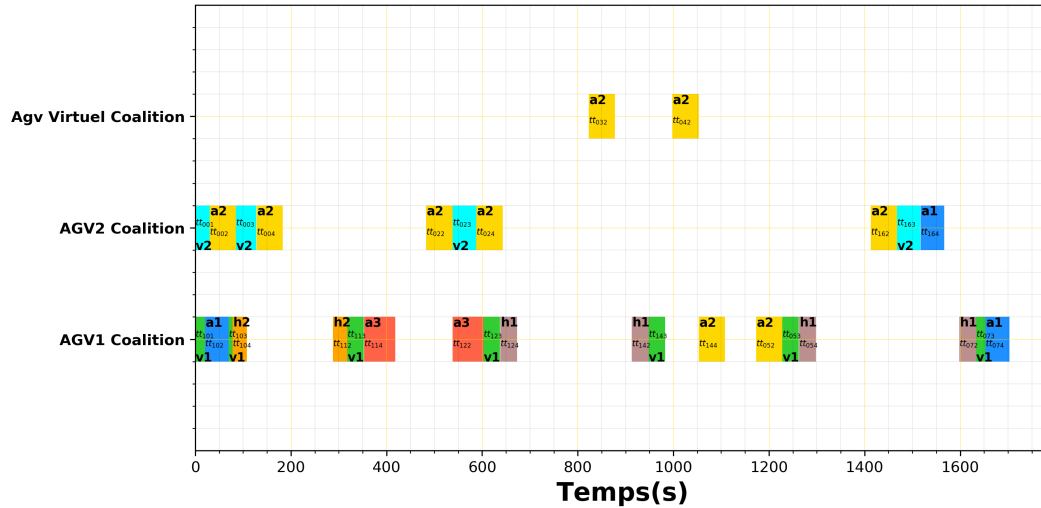


FIGURE 5.17 – Diagramme gantt des coalitions gagnantes

Le nombre de ressources mobilisées indique également la manière dont la tâche est accomplie et la nature des ressources sélectionnées. Par exemple, lorsqu'il équivaut à 2 ça implique soit que l'AGV en charge dispose d'un bras embarqué pour les opérations de chargement et de déchargement, ou bien que l'AGV est virtuel (Un AGV virtuel a besoin d'un seul bras qui prend dépose la pièce d'un poste vers un autre car les deux postes sont liés).

D'autre part cet indicateur (Nombre de ressource déployé), permet de prendre en considération les actions des ressources de transport qui se passent en "coulisses". C'est le cas d'un AGV joint à un bras. Lorsque ce bras est sollicité dans une coalition externe (Différente celle de l'AGV qui le porte), l'AGV (porteur du bras) n'apparaît pas explicitement parmi les gagnants de la tâche malgré sa contribution. Son action n'est visible que dans le temps d'opération du bras et aussi dans le nombre de ressources déployées pour la tâche.

5.4.1.3 Activité des ressources

Dans cette partie, une série de tests sera effectuée en vue d'étudier l'activité des ressources, autrement dit leur participation dans les différentes tâches de transport.

Deux paramètres, exprimés par les équations (5.1) et (5.2) sont étudiés dans ces tests. Le premier (équation (5.1)), correspond au Pourcentage d'Opérations Gagnées (POG) par une ressource. Le second (équation (5.2)), est le Pourcentage du Temps d'Occupation (PTO) d'une ressource, qui correspond à la durée totale au cours de laquelle la ressource est en mouvement.

$$POG = \frac{\text{nombre opérations gagnées}}{\text{nombre total opérations}} \times 100 \quad (5.1)$$

$$PTO = \frac{\sum \text{durée opérations gagnées}}{\text{makespan}} \times 100 \quad (5.2)$$

Au cours des tests, les ressources sont triées selon les valeurs de leurs activités puis supprimées au fur et à mesure par valeur d'activité croissante.

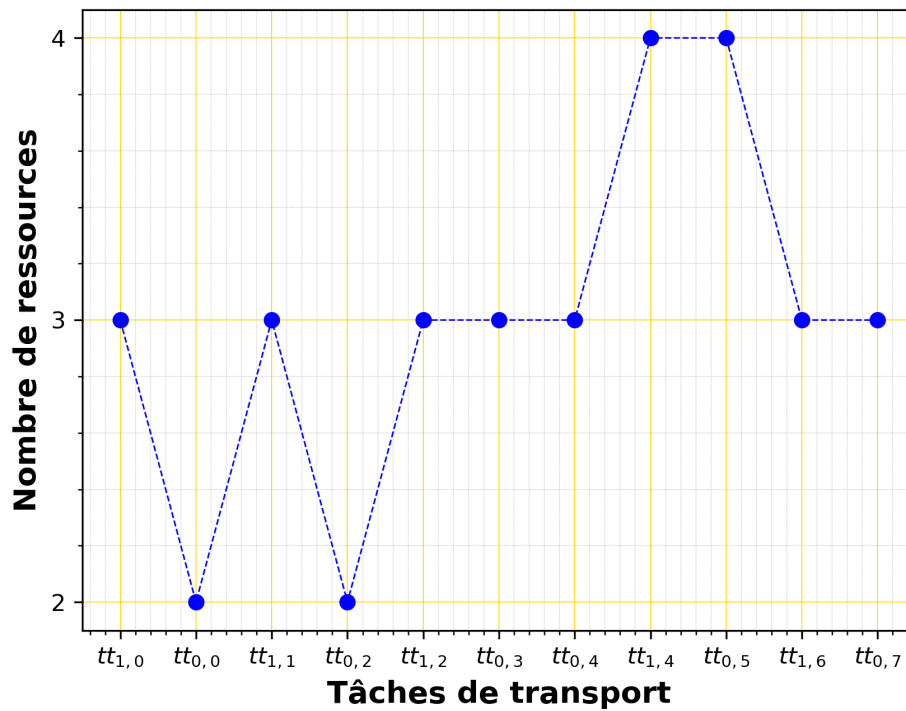


FIGURE 5.18 – Nombre de ressources par tâche

Ceci, afin d'observer les réactivité et flexibilité des algorithmes (ADG et ADL) face à l'absence de ces ressources et aussi l'impact sur le makespan. En fonction de leur activité, nous proposons la répartition suivante des ressources :

Soit $\alpha = \frac{2}{\text{nombre total opérations}} \times 100$ un indicateur d'évaluation du taux d'activité d'une ressource¹⁰ tel que :

- (i) Pour $POG = 0$, la ressource a une "Activité nulle"
- (ii) Pour $POG \leq \alpha$, la ressource a une "Activité faible"
- (iii) Pour $\alpha < POG \leq 2 \times \alpha$, la ressource a une "Activité moyenne"
- (iv) Pour $POG > 2 \times \alpha$, la ressource a une "Activité importante"

Test 0 : Présence de toutes les ressources

Ce test initial permet de s'imprégner des paramètres étudiés et voir l'état des ressources. Ainsi, aucune ressource n'a été supprimée dans ce test. La figure 5.19 montre la représentation des POG et PTO de l'ensemble des ressources ainsi que leur activité (Nulle, faible, moyenne ou importante).

10. Le choix de deux opérations au numérateur de α se justifie par le fait que deux correspond au nombre maximum d'opérations que peut gagner une ressource qui ne participe qu'à une seule tâche

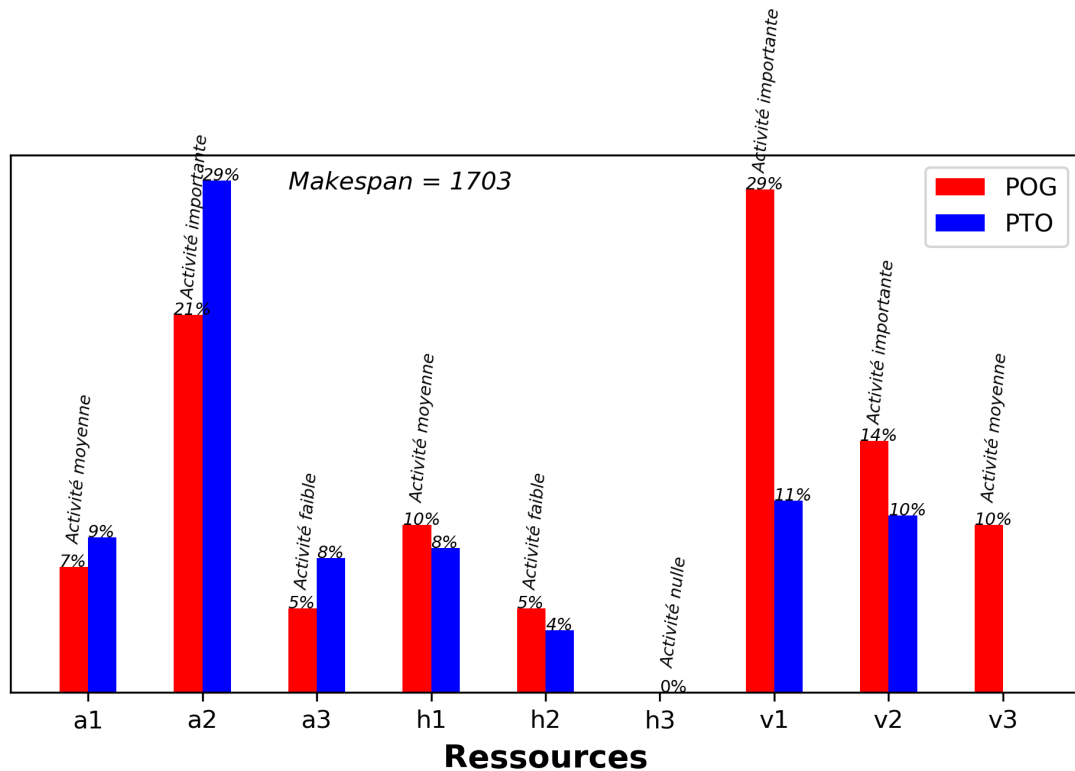


FIGURE 5.19 – Activités avec toutes les ressources

La somme des POG¹¹ sur toutes les ressources, équivaut à 1 car elle est déduite en fonction des opérations et chaque opération n'a qu'une ressource gagnante même si elle peut faire intervenir d'autres ressources.

La somme des PTO est toujours supérieure ou égale à 1. Cette supériorité est due au fait que les tâches s'exécutent en parallèle. Par exemple, deux ressources peuvent être simultanément en mouvement de t_0 jusqu'à t_{final} (makespan) et en faisant la somme de leur PTO on obtient 2, car chacune des ressources dispose d'une PTO de 100% autrement dit 1.

La figure 5.19 montre, aussi, que les AGVs ont des activités importantes, à part l'AGV virtuel (v3) qui participe moyennement dû au nombre de tâches qu'il peut effectuer. Cette forte participation des AGVs réels est confirmée par leurs PTO (tous supérieurs à 10% du makespan). A noter que le makespan, inclut aussi les temps de production.

Par ailleurs, on peut remarquer que le PTO de l'AGV virtuel est nul. En effet ce dernier n'effectue aucun déplacement, même s'il est en charge de certaines coalitions.

Test 1 : Suppression des ressources de catégorie "Activité nulle"

En nous basant sur les résultats présentés sur la figure 5.19 du *test0*, on remarque que la ressource (h3) dispose d'une activité nulle. Cette ressource

11. Les pourcentages affichés sur tous les graphes des activités, sont arrondis à un nombre entier près avec la fonction *round* du langage Python

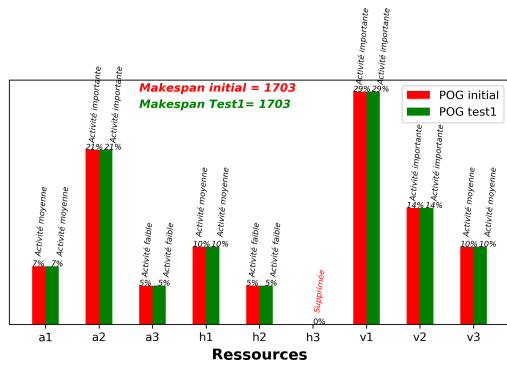


FIGURE 5.20 – Variation des POG au test 1

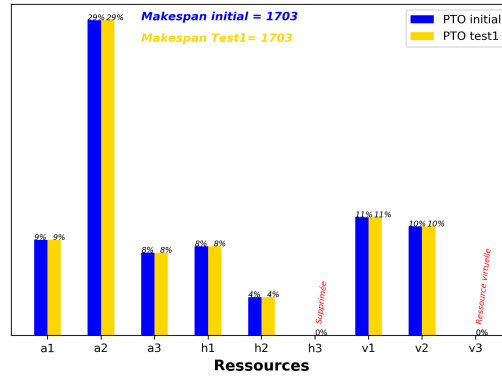


FIGURE 5.21 – Variation des PTO au test 1

est supprimée alors dans ce *test1* pour voir si elle impacte le système.

Les figures 5.20 et la figure 5.21 présente les résultats du *test1*. Ils comparent les POG et PTO des ressources du *test initial* (*test0* : Aucune suppression de ressource) et du *test1* (Suppression d’une ressource à activité nulle). Sur l’ensemble des deux figures 5.20 et 5.21, aucune valeur n’a changé. Ce qui nous permet de conclure que la suppression d’une ressource qui participe aux enchères mais qui n’a remporté aucune opération, n’a aucun impact (négatif) sur le fonctionnement de l’atelier. Par contre, elle peut faire gagner du temps dans la recherche des solutions optimales pour les ADL, car une réduction du nombre de ressource réduit aussi la complexité du système.

Test 2 : Suppression des ressources de catégorie "Activité faible"

Sur la figure 5.21, on peut voir que deux ressources appartiennent à cette catégorie d’activité faible (h2 et a3). Nous écartons pour ce *test2* toutes les ressources ayant une activité inférieure à *Activité faible*. Les résultats du *test2*,

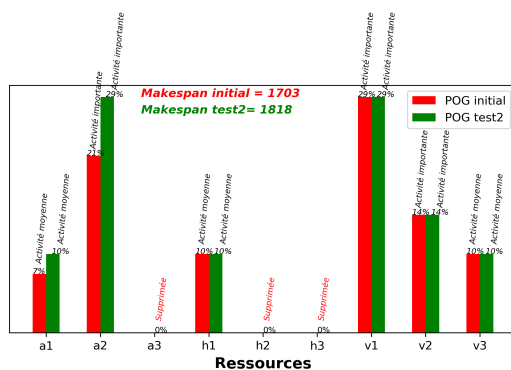


FIGURE 5.22 – Variation des POG au test 2

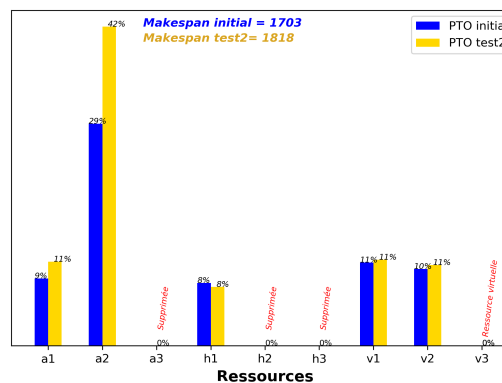


FIGURE 5.23 – Variation des PTO au test 2

sont présentés sur la figure 5.22 et la figure 5.23, montrent une hausse des activités de certaines ressources et une augmentation du makespan. En effet, la suppression de deux ressources de chargement et déchargement à faible

activité a fait varier le nombre d'opérations gagnées et le temps d'occupation des autres ressources. Cette variation démontre une bonne réactivité de l'approche, car les opérations dont les ressources supprimées étaient en charge ont systématiquement été allouées à d'autres ressources tout en minimisant l'impact sur le temps. Ce qui explique la légère augmentation du makespan qui passe de 1703 à 1818 *secondes* soit une hausse de 6.7% pour une réduction de 33.3% de l'effectif total des ressources de transport.

Test 3 : Suppression des ressources de catégorie "Activité moyenne"

En suivant la même logique que les tests précédents, toutes les ressources n'ayant pas une activité importante ont été supprimées (a1, a3, h1, h2, h3). Une exception est faite pour l'AGV virtuel, qui est la seule ressource capable de satisfaire une exigence de type *virtuel*. Cette modification réduit donc le nombre de ressource de chargement et déchargement à 1, auquel s'ajoute les AGVs dont le nombre reste inchangé.

L'approche DISCOFT, s'adapte à cette modification du nombre ressource comme l'illustre les figures 5.24 et 5.25. On peut ainsi voir que l'activité de l'AGV1 est passée d'importante à faible (figure 5.24), donc le nombre d'opération qui lui avait été attribué, a fortement été réduit de même que son temps d'occupation (figure 5.25). Par ailleurs on remarque que ces opérations que l'AGV1 a perdu ont été réaffecté à l'AGV2, qui s'avère être plus performante car disposant d'un bras. C'est grâce à cette réallocation systématique, que l'approche assure un makespan optimisé.

D'autre part, ce quatrième test, permet d'anticiper et d'apporter des solutions aux perturbations que peuvent subir les ressources dans l'atelier (résilience) [134]. Ainsi, en cas d'entrave de certaines ressources, l'approche DISCOFT grâce au principe des coalitions peut toujours assurer le fonctionnement de l'atelier tout en essayant de minimiser les retards comme le montre la variation du makespan obtenu dans ce test.

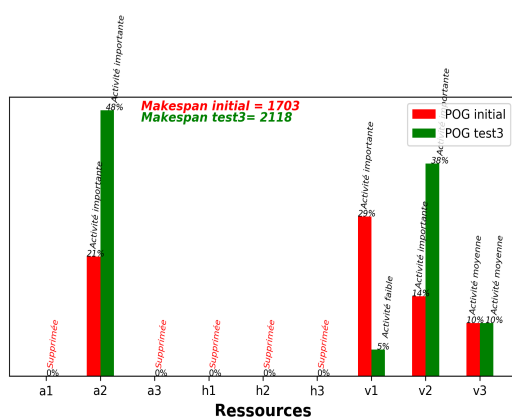


FIGURE 5.24 – Variation des POG au test 3

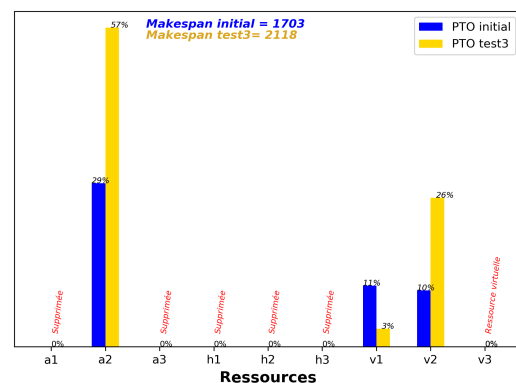


FIGURE 5.25 – Variation des PTO au test 3

Synthèse des tests

En nous basant sur le tableau 5.6 qui résume les résultats des quatre tests, on peut conclure que la suppression de toute ressource ayant une activité non nulle entraîne une augmentation du temps total d'accomplissement des tâches (Makespan). Cette variation du makespan croît en fonction de l'activité et du nombre des ressources écartées.

Un dilemme s'impose entre la réduction du nombre de ressources pour plus

TABLE 5.6 – Synthèse des tests effectués

Test	Catégorie ressources supprimées	Nbr ressources supprimées	Makespan
0	-	0	1703
1	Activité Nulle	1	1703
2	Activités nulle et Faible	3	1818
3	activités Nulle, Faible et Moyenne	5	2118

de gain mais des éventuels retards dans les délais de livraisons ou conserver un nombre raisonnable pour répondre assez rapidement aux demandes du marché. Le meilleur compromis découlant de la série de tests que nous avons menée est de réduire uniquement les ressources ayant des activités nulles et faibles. Ce qui conduit à un léger retard dans les livraisons mais à un gain considérable sur les ressources.

5.4.1.4 Tâches de production et de transport

Le graphe présenté sur la figure 5.26 représente l'ensemble des tâches de toutes les ressources de production et de transport. Pour les ressources de transport, seul le nom de l'AGV en charge de la coalition est marqué. La notation $t_{Job,t\grave{a}che}$ est utilisée pour les tâches de production. Celle des tâches de transport restent $tt_{Job,t\grave{a}che}$.

Les résultats sur ce diagramme Gantt (figure 5.26), montre pour les deux Jobs un grand décalage sur les dates de fin des tâches entre l'approche DISCOFT (qui prend en compte les opérations de chargement et déchargement) et l'approche IT2P (qui ne les considèrent pas). Le cumule de ce décalage de temps fait qu'on se retrouve avec une différence de 500 *secondes* entre les deux approches ($Cmax_{DISCOFT} - Cmax_{IT2P} > 500$). Le $Cmax$ obtenu avec DISCOFT reflète au mieux la réalité car elle permet d'assigner la tâche de transport dans son intégralité. Ainsi, ne pas prendre en compte les opérations de chargement et déchargement dans l'assignation de tâche de transport peut conduire à un décalage considérable sur les prévisions de livraisons.

5.4.2 Expérience 2

Dans cette seconde expérience, les valeurs maximales possibles des temps de chargement et déchargement seront ajoutées à chaque tâche de transport pour l'approche IT2P. On supposera que cette valeur maximale est identique

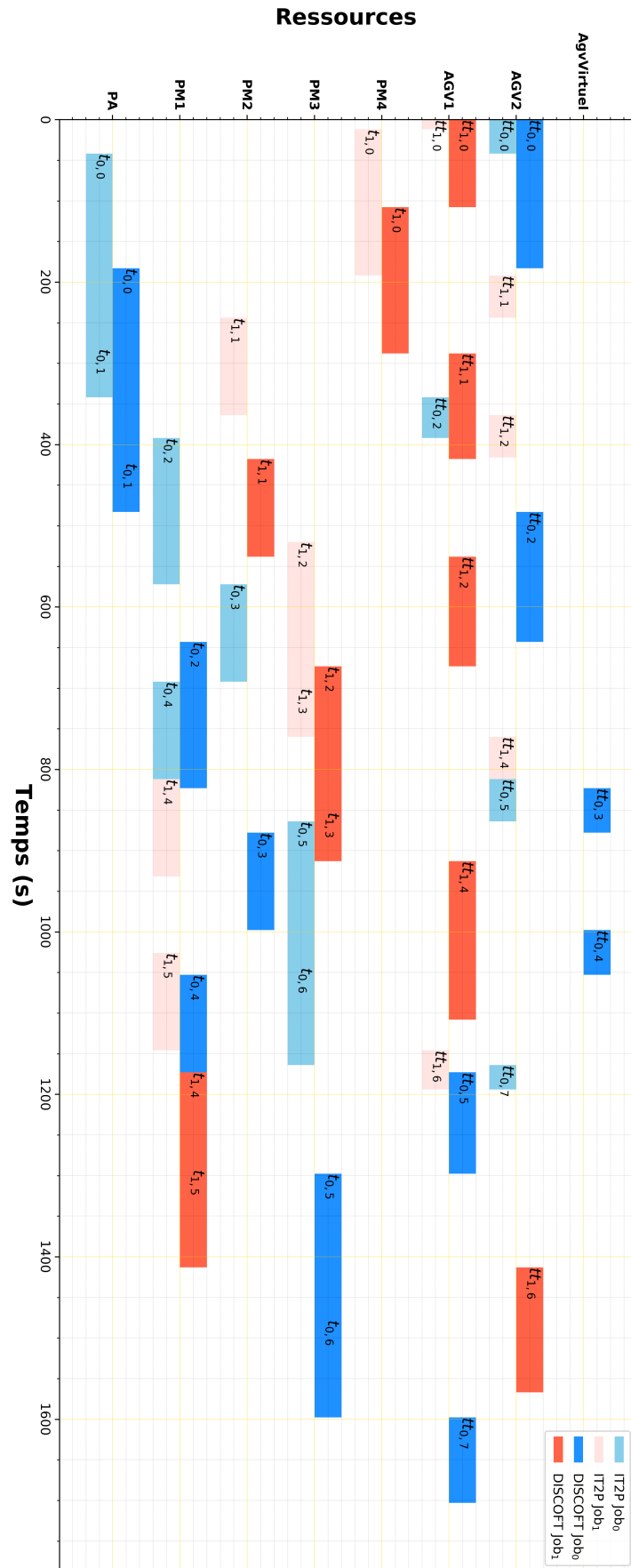


FIGURE 5.26 – Comparaison entre DISCOFT et IT2P

pour le chargement et le déchargement.

Ainsi pour l'approche IT2P la durée des tâches de transport tt exprimée par l'expression (5.3) est : la somme du temps nécessaire à l'AGV pour effectuer le transport du produit et le double du temps maximum de chargement ou déchargement que peut avoir un produit. Ce dernier temps est doublé car une ressource met le même temps pour charger ou décharger un produit, si elle est déjà sur place.

$$\delta_{tt}^{IT2P} = \delta_{AGV,i} + 2 \times \max(\delta_{lu}), \quad \forall tt^{IT2P} \in \mathcal{T}^{IT2P} \quad (5.3)$$

Avec : δ la durée d'une action, i opération de déplacement effectué par l'AGV, lu une opération de chargement ou déchargement et \mathcal{T}^{IT2P} ensemble des tâches de transport trouvé avec l'approche IT2P.

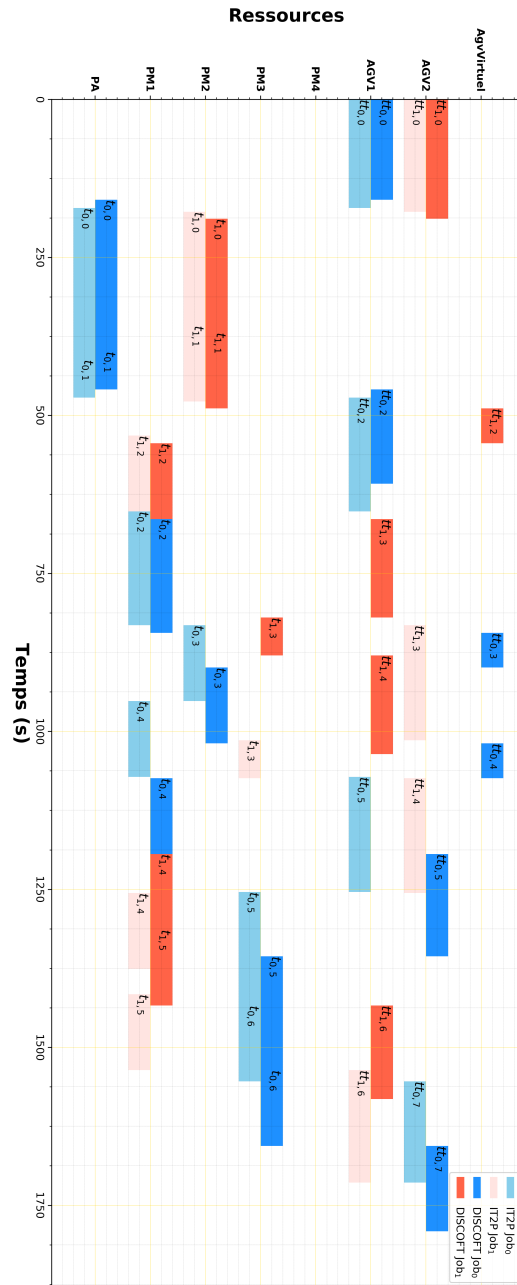


FIGURE 5.27 – Comparaison incluant le max des temps de chargement et déchargement

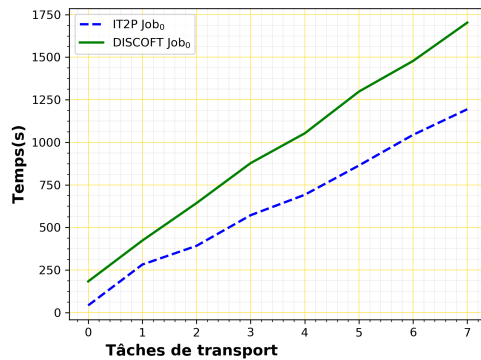


FIGURE 5.28 – Variation des temps de livraison du Job_0 sans temps de chargement et déchargement

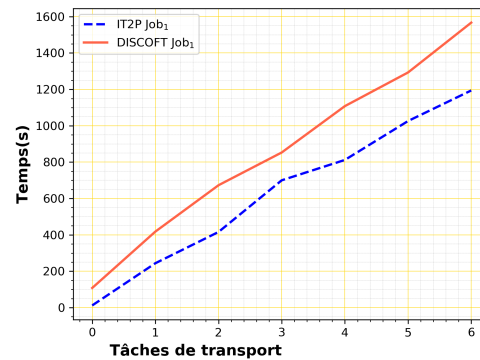


FIGURE 5.29 – Variation des temps de livraison du Job_1 sans temps de chargement et déchargement

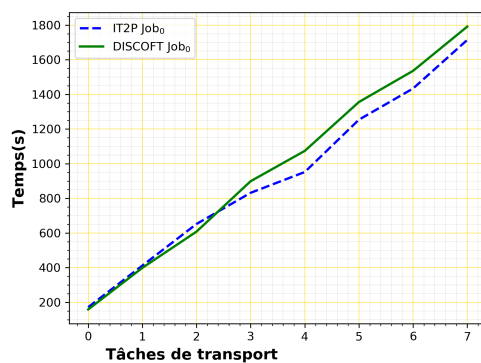


FIGURE 5.30 – Variation des temps de livraison du Job_0

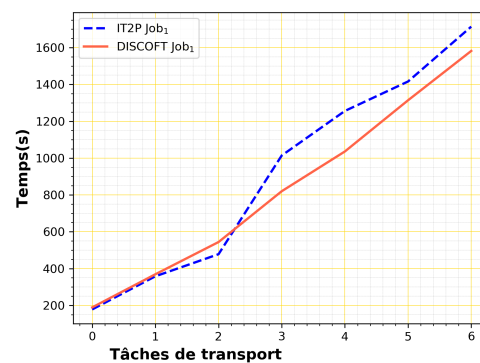


FIGURE 5.31 – Variation des temps de livraison du Job_1

Après simulation, on observe toujours un léger décalage entre les C_{max} des deux approches comme l'indique la figure 5.27. Cependant, on peut remarquer une diminution considérable de l'écart entre les deux approches en observant les évolutions des figures 5.28 et 5.29 respectivement vers les figures 5.30 et 5.31. Sur la figure 5.31, on peut remarquer que le temps de livraison du produit 1 par l'approche IT2P (Job_1) est supérieur à celui obtenu par l'approche DISCOFT pour le même produit. Néanmoins, la détermination du C_{max} se fait par rapport au temps de livraison de l'ensemble des produits : $C_{max} = \max(\text{livraison } Job_0, \text{livraison } Job_1)$. Le produit 0 (Job_0) ayant le temps de livraison le plus élevé, il détermine alors le C_{max} . Ainsi, sur la figure 5.30 qui montre l'évolution des temps de livraison du produit 0, l'infériorité de des temps de livraison de IT2P par rapport à DISCOFT est visible. Malgré l'ajout des maximums de temps de chargement et déchargement à l'approche IT2P, cette dernière reste moins précise. Cette imprécision vient du fait que la disponibilité des ressources de chargement et déchargement n'est pas prise en compte.

L'ajout des temps de chargement et déchargement estimé permet juste une considération de la durée des opérations mais elle n'assure pas la prise en compte des dates de début des opérations. Par exemple, un bras peut mettre 45secondes pour charger un produit. Il peut commencer cette opération à $H = 10h00min$ ou à $H = 12h45min$, tout est fonction de sa disponibilité.

La meilleure façon, de prendre en compte cette information sur les dates de début des tâches qui dépend aussi de la disponibilité des ressources, c'est de considérer les tâches de transport dans leur globalité (opérations de chargement, déplacement AGV et déchargement).

5.5 Conclusion

Après avoir spécifié nos besoins et exigences logiciels nous avons porté notre choix sur deux solveurs d'OR-Tools (SAT-CP et SAT solver) pour l'implémentation des approches utilisées dans le cas d'étude. Les caractéristiques des ressources utilisées, contraintes quelques fois par des aspects sécuritaires ont été décrites.

Deux expérimentations ont été conduites dans ce chapitre afin de valider l'approche DISCOFT. La première a permis de dégager les critères d'éligibilité aux tâches proposées et les performances qui font croître la participation des ressources aux coalitions. En outre, une série de tests a été réalisée en classant les ressources en quatre catégories en fonction de leur activité qui est soit nulle, faible, moyenne ou importante en vue de valider la réactivité de DISCOFT. La suppression des ressources par catégorie a permis d'observer et de trouver un compromis entre le nombre de ressources travaillant en parallèle dans l'atelier et la variation du makespan.

Les diagrammes de Gantt réalisés dans cette expérience, ont attesté du respect de toutes les contraintes liées aux produits, tâches, ressources et aux emplacements géographiques des postes.

La seconde expérience a été focalisée sur la nécessité de prendre en compte les opérations de chargement et déchargement dans l'ordonnancement des tâches de transport. J'ai testé une pratique utilisée tant auprès des industriels que des chercheurs académiques. Cette pratique consiste à estimer les temps des opérations de chargement et déchargement puis les ajouter à l'assignation trouvée pour l'opération de déplacement de l'AGV. Les résultats comparatifs ont montré, que ce genre d'estimation conduit toujours à une limitation de la précision des dates de début et de fin des tâches et opérations dans l'atelier car elle ne prend pas en compte la disponibilité des ressources. Ainsi, la meilleure façon est d'introduire les opérations de chargement et déchargement de façon adéquate sans accroître fortement la complexité du problème. L'approche DISCOFT, comme l'a montrée ce chapitre, permet une bonne gestion de cette problématique.

Conclusion générale

Ce travail de thèse a permis la mise en oeuvre d'une architecture de supervision distribuée pour les entités robotiques de transport dans le contexte de l'Industrie 4.0. Cette architecture découle de l'amélioration et de l'adaptation robotique d'une architecture existante. Parmi les modules qui constituent l'architecture, nous avons porté un accent particulier sur celui de la décision responsable de l'ordonnement des tâches. Ainsi, une approche permettant l'allocation des ressources robotiques autonomes à des tâches de transport a été mise en oeuvre. Cette approche, pour l'ordonnement distribué des tâches de transport (DISCOFT), a été étudiée et développée pour répondre à un certain nombre de problématiques encore non traitées dans les travaux de recherche pour la plupart. Parmi ces problématiques figurent la considération des tâches de transport comme des tâches constituées de plusieurs opérations incluant le chargement et le déchargement des produits. En outre, l'hétérogénéité des systèmes de transport due en partie à la modularité de certaines ressources utilisées comme les manipulateurs mobiles, a également été prise en compte dans l'approche DISCOFT. Cette hétérogénéité, complexifie l'ordonnement dans la mesure où les ressources ont des fonctions et performances différentes.

Après un premier chapitre (1) visant à situer le contexte général et le positionnement de la thèse, un second chapitre (2) a apporté une revue littéraire sur les thématiques qui cadrent avec cette thèse dont : les systèmes de production flexible, l'Industrie 4.0, l'ordonnement de tâches, la supervision des entités autonomes comme les manipulateurs mobiles et la formation de coalition.

La supervision des manipulateurs mobiles intelligents, diffère de celle des éléments réactifs, incapables de prendre des décisions de haut niveau. Le premier grand point à définir pour leur supervision, était une architecture régissant les interactions entre les entités de l'atelier. Les travaux de recherche ayant porté leur attention sur le développement des architectures de supervision ont démontré, qu'une bonne architecture doit avoir à la fois un comportement hiérarchique et hétérarchique.

Le côté hiérarchique permet au système, d'avoir un temps de réponse rapide et une prédictibilité du comportement global. Ce qui est rendu possible par la présence d'une entité de contrôle centrale. Quant au côté hétérarchique, il garantit une réactivité et une tolérance aux perturbations qui peuvent affecter le système. Le comportement hétérarchique permet également d'augmenter la flexibilité du système en donnant une autonomie dans les prises de décision.

Le chapitre 3 de cette thèse, a été focalisée sur la proposition d'une architecture de supervision robotique. Cette dernière a été inspirée par une approche de production dynamiquement intégrée (DIMS).

L'architecture dispose ainsi, d'un modèle d'agent générique qui lui donne une structure fractale. Le rôle et l'importance de chaque module constituant les agents ont été explicitement spécifiés. Le module de décision est responsable de l'ordonnancement et l'optimisation des tâches. L'approche initiale, est constituée d'une méthode d'ordonnancement basée sur les enchères permettant une assignation hiérarchique (Possibilité de décomposer une tâche en sous-tâche pour l'assigner). J'ai amélioré cette approche, en ajoutant des fonctionnalités manquantes comme la prise en compte des exécutions en parallèles.

La méthode d'optimisation basée sur un algorithme mono-objectif a également été modifiée pour prendre en compte une optimisation bi-objectif.

L'ensemble des apports sur l'architecture a été validé sur un cas d'étude simulé, qui par la suite a bénéficié d'une implémentation sur des robots réels .

Cette méthode d'ordonnancement, a permis de mettre en avant la manière par laquelle l'entité centrale (le superviseur) responsable des enchères gère les décisions globales. L'auto-organisation et la façon par laquelle les ressources de transport de bas niveau répondent aux enchères proposés, ont été abordées dans le chapitre 4.

Pour répondre efficacement aux enchères des tâches de transport qui sont de structure macroscopique (constitué de sous-tâches), une autonomie a été accordée aux véhicules à guidage automatique pour former des coalitions avec d'autres robots et/ou des humains. Ce mécanisme est rendu possible grâce à un modèle mathématique basé sur une programmation linéaire en nombres entiers. Ce modèle est implémenté sur chacun des véhicules à guidage automatique présents, pour des prises de décision locale. L'approche DISCOFT dans son ensemble, est un couplage de l'organe de décision globale (le superviseur en charge de l'émission des enchères et de la désignation des gagnants) et des différents organes de décision locale.

Grâce à la modélisation d'un cas d'étude dans le chapitre 5, l'approche DISCOFT a été testée et validée. Un ensemble de représentations graphiques a permis de voir la manière dont sont constituées et optimisées localement les coalitions, avec respect des différentes contraintes. Les expérimentations conduites ont aussi permis de voir les paramètres qui influencent la participation des ressources aux coalitions. Ainsi, la rapidité, la charge utile et la mobilité font partie des éléments influents.

Par ailleurs l'intérêt de prendre en considération le rôle du bras des manipulateurs mobiles dans l'assignation des tâches fut démontrée par une étude comparative. Les résultats des diverses simulations ont prouvé que le fait de négliger les opérations de chargement et déchargement dont le bras (ou l'opérateur humain) est responsable, conduit à une forte imprécision dans l'ordonnancement des tâches. Ce manquement concerne surtout les dates de début et de fin des tâches ainsi que leur durée. Par la prise en compte de ces opérations (chargement et déchargement), l'approche DISCOFT s'est avérée

plus précise. En plus, cette approche permet aussi une utilisation plus intelligente des ressources robotiques en exploitant l'intelligence des robots non seulement pour des tâches basiques telles que l'évitement des obstacles mais aussi pour définir des équipes de travail d'une manière locale. Ce qui reste pratique, dans la mesure où une partie de l'optimisation de la décision est distribuée, déchargeant ainsi le superviseur.

Perspectives

L'implémentation sur un système de production réel fait partie des perspectives à court terme de mes travaux. La plateforme du démonstrateur du futur au laboratoire LINEACT a déjà été aménagée pour recevoir la validation réelle de l'approche DISCOFT.

Dans le cadre du projet CoRoT, l'approche sera également implémentée sur des environnements industriels. Par ailleurs, l'assemblage et le désassemblage dynamique est un paramètre que nous comptons intégrer dans les algorithmes de décision locale de DISCOFT. Des critères comme le niveau de batterie des AGVs ou les contraintes sur les chemins qu'ils peuvent emprunter, sont également des paramètres qui viendront enrichir le modèle de l'approche proposée.

Productions scientifiques

Publications

- 1) **Part B : Journal of Engineering Manufacture** (Impact factor : 1.98). *Accepté et publié* : "Modular mobile manipulators coalition formation through distributed transportation tasks allocation", **MOUSSA G. Souleymane**, BETTAYEB Belgacem, SAHNOUN M'hammed, DUVAL Fabrice, BENSRAHAIR Abdelaziz, 2020.
- 2) **International Journal of Production Economics, Elsevier** (Impact factor : 5.13). *Soumis, en attente d'acceptation* : "Integrated Scheduling of Production and Transportation Tasks for Flexible Job-Shop Manufacturing Systems", M'hammed Sahnoun, Naihu HE, Belgacem BETTAYED, Mario J. GONZALEZ-ROMO, **Souleymane MOUSSA G.**, David Z. ZHANG, 2020.
- 3) **IEEE Electron Devices Technology & Manufacturing Conference (EDTM)**. *Accepté et publié* : "Bi-Objective Indirect Optimization of Robotic Transportation Task Assignment Based on Auction Mechanism", **Goumeye S. M.**, Sahnoun M., Duval F., & Bensrhair A., 2020, Penang.
- 4) **Proceedings of the 17th International Conference on Manufacturing Research (ICMR)**. *Accepté et publié* : Modular Mobile Manipulators Task Allocation Auction-based Approach, **MOUSSA GOUMEYE Souleymane**, SAHNOUN M'hammed, DUVAL Fabrice, BENSRAHAIR Abdelaziz, 2019.
- 5) **25ème colloque des Sciences de la conception et de l'innovation (CONFERE'18)**. *Accepté* : "Interopérabilité des systèmes de production robotisés flexibles", Kalb Eddine CHELLI, M'hammed SAHNOUN, **Souleymane MOUSSA**, Anne LOUIS, David BAUDRY, Fabrice DUVAL, 2018.

Communications et Séminaires

- **11^{ème} édition de la Journée des écoles doctorales PSIME et MIS 2019**. *Présentation orale* : "Modular mobile manipulator tasks assignment in Flexible Manufacturing System", MOUSSA G. Souleymane, UFR des Sciences et Techniques, Saint-Etienne du Rouvray.
- **Journée des doctorants du laboratoire Litis 2019**. *Présentation orale* : "Supervision d'un atelier de production flexible constitué de robots collaboratifs mobiles", MOUSSA G. Souleymane, Université Le havre Normandie.

- **Séminaire des doctorants du laboratoire Lineact 2018.** *Présentation orale* : "Architecture distribuée pour le contrôle des FMS : Implémentation réelle via ROS", MOUSSA G. Souleymane, Cesi de Rouen.
- **25^{èmes} Journées STP du GdR MACS 2018.** *Présentation orale* : "Supervision d'un atelier de production flexible constitué de robot collaboratifs", Campus des Arts et Métiers de Lille.
- **Semaine de l'Innovation du Transport & de la Logistique, SITL 2018.** *Responsable Stand CoRoT* : "Présentation de la solution robotique du Projet CoRoT", MOUSSA G. Souleymane, Parc d'exposition porte de Versailles, Paris.
- **Doctoriales Vallée de Seine 2018 : Séminaire résidentiel de formation pour l'insertion professionnelle des doctorants.** *Présentation orale et participation* : Vulgarisation scientifique de la supervision des robots autonomes, 16 au 20 avril 2018 à Saint-Valery-en-Caux.

Annexe A

Implémentation réelle de DIMS sur un atelier robotique

Cette annexe décrit les procédures d'implémentation l'approche d'ordonnancement de tâches de DIMS, sur un cas d'étude réel. Les différents pré-requis et configurations matérielles et logicielles sont détaillés.

A.1 Modélisation du cas d'étude

Le cas d'étude modélisé est un atelier de production flexible. Trois bras manipulateurs de type Universal Robots (Deux UR5 et un UR10) ont été initialement prévus dans la réalisation du cas d'étude. La simulation dans la dernière section du chapitre 3, a été faite en utilisant les trois robots. Néanmoins, la taille miniaturisée de la maquette servant d'atelier et le fait que les trois robots soient fixes, nous a confronté à plusieurs difficultés ayant conduit à la suppression d'un robot pour mener à bien l'implémentation réelle.

La figure A.1, nous permet de retrouver certaines de ces difficultés :

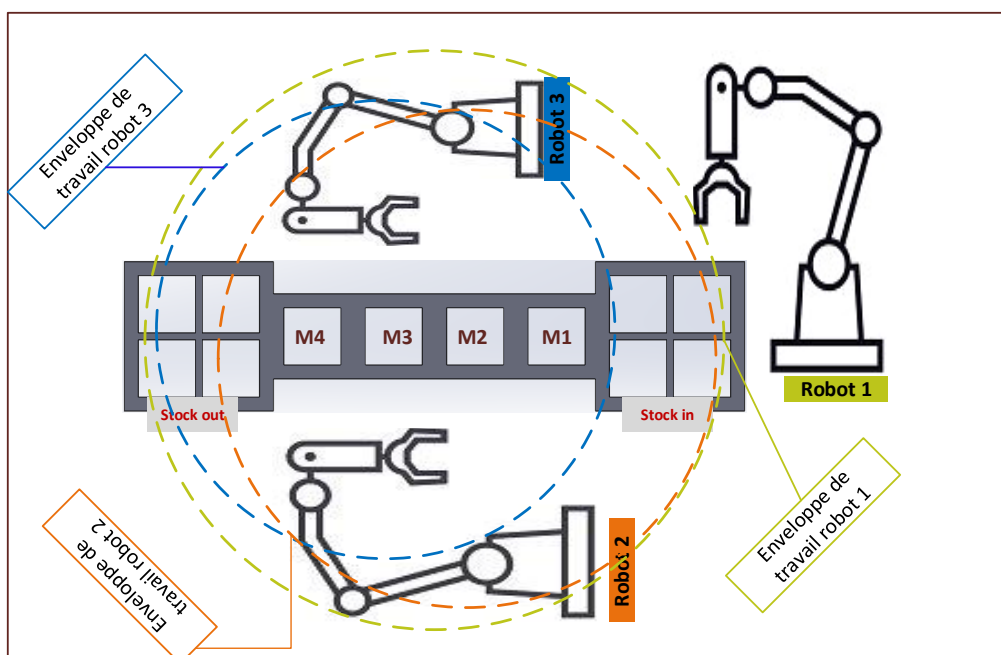


FIGURE A.1 – Enveloppes de travail des robots

- **Le risque de collisions** : Les trois enveloppes de travail se chevauchent. Les robots sont certes collaboratifs (Ils s'arrêtent en cas de choc, et ne présentent pas de danger imminent pour leur entourage), mais il y a un risque continu d'abîmer les produits et nuire ou retarder les process en cas d'arrêts répétés.
- **L'atteignabilité des points** : Avec la configuration sur la figure A.1, les robots 2 et 3 ne peuvent pas atteindre l'ensemble des points de l'atelier miniature. De ce fait, l'ensemble des tâches ne peuvent être réalisées que par le robot 1.

En utilisant les outils de modélisation et de planification de ROS (URDF, Rviz), le cas d'étude a été reformulé. Cette nouvelle modélisation permet de réduire au maximum les risques de collisions (utilisation de 2 robots) et de garantir à tous les robots d'atteindre l'ensemble des points de l'atelier grâce à une étude ergonomique. Cette étude a été menée sous l'environnement Rviz grâce à une modélisation 3D, visible sur la figure A.2.

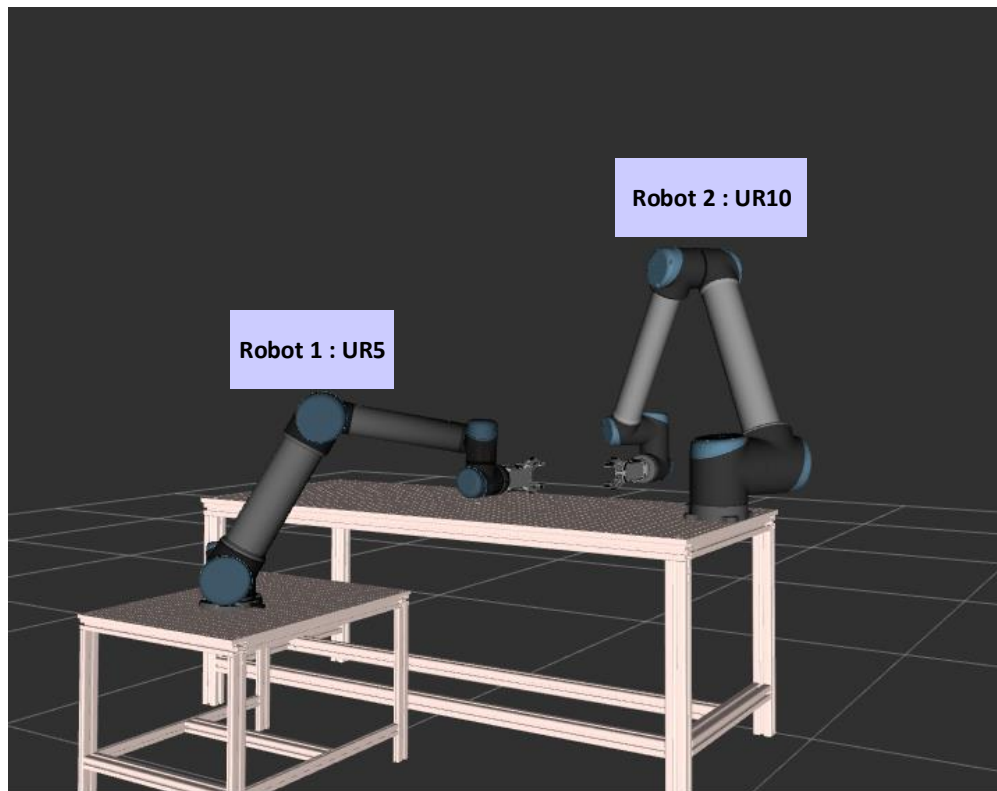


FIGURE A.2 – Modélisation atelier sous Rviz



FIGURE A.3 – Application de l'étude ergonomique sur l'atelier réel

A.2 Enregistrement des coordonnées cartésiennes via ROS

En robotique il existe deux façons de faire déplacer un bras manipulateur. La première consiste à donner des coordonnées angulaires (q) aux articulations pour que le robot puisse se rendre au point cartésien (M) voulu ($M = f(q)$). La seconde consiste à lui donner directement les coordonnées cartésiennes pour que l'outil du robot puisse s'y rendre ($q = f^{-1}(M)$). ROS offre la possibilité d'utiliser ces deux méthodes. Pour le cas d'étude, nous avons utilisé la seconde approche qui consiste à donner la position cartésienne du point désiré au robot, pour avoir une meilleure traçabilité des produits.

Ainsi grâce au meta-package Moveit! de ROS, on peut définir des coordonnées cartésiennes dans l'espace de travail du robot pour qu'il puisse s'y rendre.

A.2.1 Installation de Moveit!

Tous les développements ont été réalisés sur ROS indigo, installé sous Ubuntu 14.04. Le cas d'étude a été fait en Avril 2018.

Moveit!¹ est inclus dans le ROS full desktop, mais il est toujours bien d'essayer de l'installer pour pouvoir avoir la dernière version.

```
sudo apt-get install ros-indigo-moveit
```

Une bonne configuration de Moveit et toutes ses dépendances (dépendances) est requise pour pouvoir piloter les robots via moveit!. C'est pour cela

1. <https://moveit.ros.org/install/>

qu'il faut bien voir les erreurs affichées après compilation et d'y remédier en les téléchargeant. Par exemple si l'erreur dit qu'il manque `moveit-resources`, il faut taper la commande qui permet de la télécharger :

```
sudo apt-get install ros-indigo-moveit-resources
```

Et après il faut mettre à jour les nouvelles installations :

```
sudo apt-get update
```

Avec le meta-package `universal robots`² déjà installé, on peut lancer le fichier `launch` :

```
roslaunch ur5_moveit_config demo.launch
```

qui charge le robot virtuel UR5 avec son `moveit group` « `manipulator` ». On peut ainsi contrôler ce robot virtuel, lui envoyer des points cartésiens en nous basant sur le tutoriel de la version `hydro`³ du robot PR2, téléchargeable en utilisant le lien suivant :

https://github.com/ros-planning/moveit_pr2/tree/hydro-devel

La compilation de ce méta-package depuis l'espace de travail fait souvent ressortir des erreurs qui sont généralement dues à une configuration incomplète de `moveit`!. Après la compilation, il faut adapter le fichier :

`move_group_python_interface_tutorial.py`. Cette adaptation consiste à remplacer, le nom du `group` « `left_arm` » par « `manipulator` ». Ainsi :

```
group = moveit_commander.MoveGroupCommander("left_arm")
```

devient :

```
group = moveit_commander.MoveGroupCommander("manipulator")
```

On peut par la suite modifier les positions cartésiennes qui sont en mètre et les orientations quaternion (plus compliquées à manipuler).

Pour une bonne exécution des déplacements, un réglage de la vitesse et/ou de l'accélération s'impose. Autrement, le robot se mettra dans un état de blocage ou ne finira pas le déplacement lorsqu'on essaye de régler la vitesse avec le Teach-pendant (tablette de contrôle) du robot. Quelques recherches, nous ont permis de trouver tout un arsenal (contrôle vitesse-accélération, obtention de l'état actuel du robot, des angles, tolérance des positions, temps de planification, etc) de classes pouvant être utilisé avec `moveit`! :

http://docs.ros.org/kinetic/api/moveit_ros_planning_interface/html/functions_s.html#index_s

En plus, elles sont faciles à utiliser. Dans tous les cas, beaucoup ont été utilisés dans le script du tutorial `move_group_python_interface_tutorial.py`, donc on peut s'en inspirer. Par exemple pour fixer la vitesse et l'accélération maximale de l'UR5 :

```
group.set_max_acceleration_scaling_factor(0.1)
```

2. https://github.com/ros-industrial/universal_robot/tree/indigo-devel

3. Le tutoriel de la version de Hydro de ROS, était plus stable que celui de Indigo en février 2018.

```
group.set_max_velocity_scaling_factor(0.1)
```

Dans cet exemple, on fixe les deux grandeurs à dix pour cent (10%) de leurs valeurs maximales.

```

1  #!/usr/bin/env python
2  import time
3  import roslib; roslib.load_manifest('ur_driver')
4  import rospy
5  import actionlib
6  from control_msgs.msg import *
7  from trajectory_msgs.msg import *
8  import sys
9  import copy
10 import moveit_commander
11 import moveit_msgs.msg
12 from geometry_msgs.msg import *
13
14
15 JOINT_NAMES_UR5 = ['shoulder_pan_joint', 'shoulder_lift_joint', '
    elbow_joint',
16                    'wrist_1_joint', 'wrist_2_joint', 'wrist_3_joint']
17
18
19 #Definition de la position home
20 state_up_ur5 = [0, -1.56, 0, -1.56, 0, 0]
21
22 client = None
23
24 def main():
25     global client
26     try:
27         ## INITIALISE
28         rospy.init_node('points_save',
29                         anonymous=True)
30         moveit_commander.roscpp_initialize(sys.argv)
31         robot = moveit_commander.RobotCommander()
32         #scene = moveit_commander.PlanningSceneInterface()
33         ur5 = moveit_commander.MoveGroupCommander('ur5_manipulator'
34 )
35         print "===== START POSITION"
36         ur5.get_current_joint_values()
37         ur5.set_joint_value_target(state_up_ur5)
38         plan1=ur5.plan()
39         a = input("Le robot exécute-il la trajectoire sur Rviz en
    toute sécurité?appuyer
40                 y pour si oui")
41         if a == "y":
42             ur5.execute(plan1)
43         else:
44             rospy.signal_shutdown("KeyboardInterrupt") #Ne pas exé
45             cuter la trajectoire
46         rospy.sleep(2)
47         #initialisation de la position courante
48         ur5.clear_pose_targets()
49         stock21 = geometry_msgs.msg.Pose()
50         stock21.position.x = -0.190
51         stock21.position.y = -0.999

```

```

50 stock21.position.z = 0.063
51 stock21.orientation.x =0.703
52 stock21.orientation.y =-0.007
53 stock21.orientation.z =-0.009
54 stock21.orientation.w =0.711
55
56 print "====execution des points===="
57
58 ur5.set_planning_time(5)
59 ur5.set_num_planning_attempts(10)
60 ur5.set_max_acceleration_scaling_factor(0.1)
61 ur5.set_max_velocity_scaling_factor(0.1)
62 (plan2, fraction) = ur5.compute_cartesian_path(stock_in, 0.01,
63 0, False)
64 ur5.set_pose_target(stock_in)
65 ur5.go(wait=True)
66 print "====fin===="
67
68 except KeyboardInterrupt:
69     rospy.signal_shutdown("KeyboardInterrupt")
70     moveit_commander.roscpp_shutdown()
71     #raise
72
73 if __name__ == '__main__': main()

```

LISTING A.1 – Script de déplacement de l'UR5 au travers de moveit!

A.3 Enregistrement des points sur la maquette

L'ensemble des points qui seront utilisés dans le cas d'étude sont définis par des petits points noirs sur la figure A.4. Pour connaître les coordonnées ces points, nous avons d'abord défini le repère de l'UR10 comme repère de référence (tous les points seront définis dans ce repère). Par la suite, le gripper du robot est placé au point désiré, et grâce à la classe *GetCurrentPose* de moveit! qui se traduit par la commande *get_current_pose()* on peut lire la position à laquelle se trouve le robot. Cette dernière aurait pue être lue directement grâce à la tablette de contrôle du robot, seulement il y'a une petite différence entre les coordonnées lues sur la tablette et les coordonnées lues par le script sur l'ordinateur.

```

1 # -----Machine M1-----
2 m1 = geometry_msgs.msg.Pose()
3 m1.position.x = -0.225
4 m1.position.y = -0.530
5 m1.position.z = 0.058
6 m1.orientation.x =0.703
7 m1.orientation.y =-0.007
8 m1.orientation.z =-0.009
9 m1.orientation.w =0.711
10 # -----Machine M2-----
11 m2 = geometry_msgs.msg.Pose()
12 m2.position.x = -0.225
13 m2.position.y = -0.648
14 m2.position.z = 0.063
15 m2.orientation.x =0.703

```

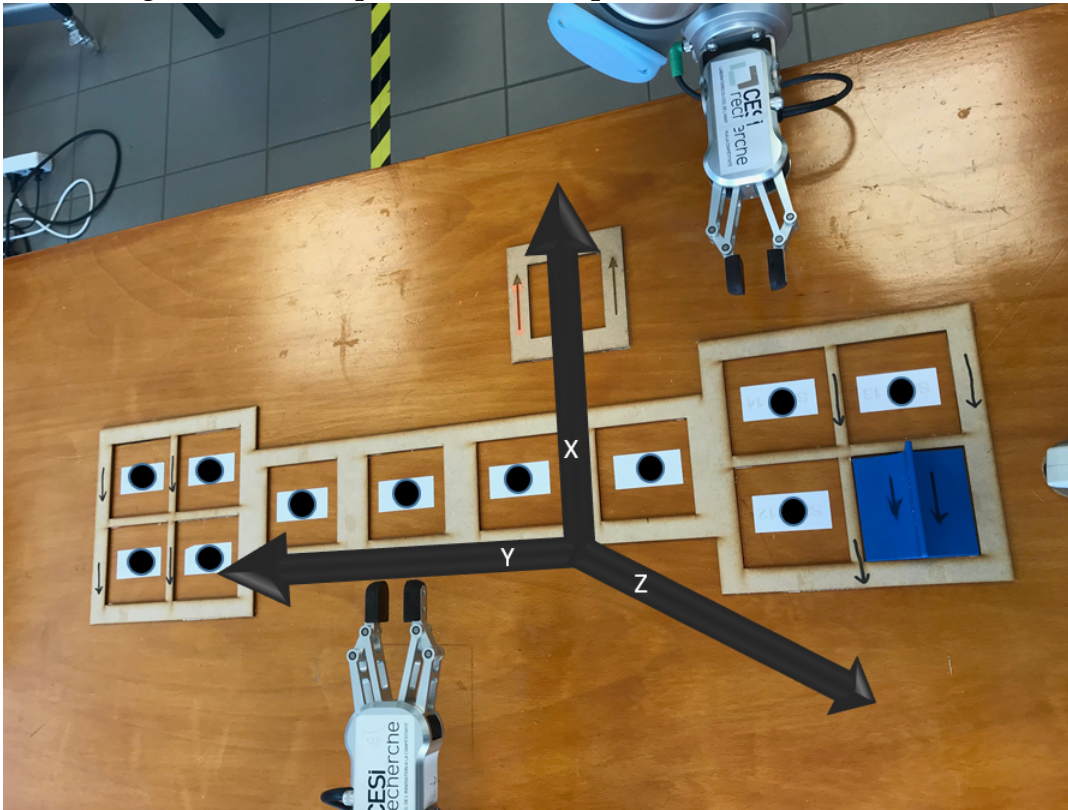


FIGURE A.4 – Points à enregistrer sur la maquette

```

16 m2.orientation.y = -0.007
17 m2.orientation.z = -0.009
18 m2.orientation.w = 0.711
19 # -----Machine M3-----
20 m3 = geometry_msgs.msg.Pose()
21 m3.position.x = -0.225
22 m3.position.y = -0.766
23 m3.position.z = 0.063
24 m3.orientation.x = 0.703
25 m3.orientation.y = -0.007
26 m3.orientation.z = -0.009
27 m3.orientation.w = 0.711
28 # -----Machine M4-----
29 m4 = geometry_msgs.msg.Pose()
30 m4.position.x = -0.225
31 m4.position.y = -0.881
32 m4.position.z = 0.063
33 m4.orientation.x = 0.703
34 m4.orientation.y = -0.007
35 m4.orientation.z = -0.009
36 m4.orientation.w = 0.711

```

LISTING A.2 – Coordonnées cartésiennes des quatre machines dans le repère de l'UR10

A.4 Changement de repère pour l'UR5

Pour que l'UR5 puisse lire et exécuter les points définis dans le repère de l'UR10, il a fallu lui faire un changement de repère virtuel. En d'autres termes, il fallait trouver les valeurs à ajouter à un point M_{UR5} défini dans le repère de l'UR5, pour qu'il corresponde à la position M_{UR10} . Ceci permettant aux robots de lire les mêmes valeurs des points et de l'adapter (dans le cas de l'UR5) ou l'utiliser directement (dans le cas de l'UR10). A noter que, les deux repères (UR5 et UR10) sont parallèles. Donc on n'aura pas de rotation, juste une translation à faire.

Pour faire le changement, on a défini les coordonnées d'un point dans les deux repères (Durant tout le cas d'étude, on a seulement agi sur les positions x, y et z. les orientations en quaternion ont été fixées). La matrice de

TABLE A.1 – Coordonnées de M en mètre dans les deux repères

M	UR10	UR5
x	-0.225	-0.460
y	-0.530	0.300
z	0.057	-0.183

transformation $trans_{mat}$ peut s'exprimer comme suit :

$$\begin{pmatrix} x_{trans} \\ y_{trans} \\ z_{trans} \end{pmatrix} = \begin{pmatrix} x_{ur10} \\ y_{ur10} \\ z_{ur10} \end{pmatrix} - \begin{pmatrix} x_{ur5} \\ y_{ur5} \\ z_{ur5} \end{pmatrix} \quad (A.1)$$

De la table A.1, on peut ainsi obtenir ces valeurs :

$$trans_{mat} = \begin{pmatrix} 0.235 \\ -0.830 \\ -0.400 \end{pmatrix} \quad (A.2)$$

A.5 Communication entre entités

Pour l'envoi et la réception des données, nous avons utilisé les protocoles de communications de ROS (Topics et services principalement). Les données à envoyer sont entre autres le numéro de l'opération, ce en quoi elle consiste (par exemple, déplacer le produit d'un point m1 vers un point m2) et le numéro de la ressource (par exemple 0 pour l'ur10 et 1 pour l'ur5) qui a gagné l'opération. Ces données sont transmises par le superviseur. Une fois que les robots auront accomplis ces tâches, ils doivent remonter l'information au superviseur pour que ce dernier puisse envoyer la prochaine.

Sur ROS, le protocole de type Topic, qui permet l'envoi de données par un Publisher et la souscription à ce Topic par autant de Subscriber le désirant, réponds bien à nos critères d'envoi d'information par le superviseur aux robots. En outre, le Topic sous ROS peut avoir une structure, c'est-à-dire qu'on la possibilité d'envoyer plusieurs types de messages en utilisant le même

type, en spécifiant le type de message (booléen, entier, réel, points cartésiens. . .) dans un fichier de définition, au préalable. Quant à l'envoi des informations par les robots, nous avons besoin d'un type qui permettra d'envoyer les informations justes une fois et à la demande. Donc le Topic ne correspond pas à ce critère, car il publie continuellement et sans qu'on le lui demande. Par ailleurs le service, basé sur le principe de Request/Response permet de répondre à ce besoin. Donc le superviseur lance une requête pour connaître l'état de l'opération actuelle, et dès qu'il reçoit une réponse disant qu'elle est finie, il passe à l'opération suivante.

A.5.1 ROS multi-Machine

ROS multi-machine est une fonctionnalité de ROS qui permet de faire communiquer plusieurs machines, robots ou éléments équipés de ROS. Etant donné que nous voulons tester une architecture de supervision distribuée, nous avons alors utilisé trois ordinateurs, connectés en ROS multi-machine, pour le contrôle de nos entités.

La communication multi-machine sur ROS consiste à définir un des ordinateurs comme étant le Ros-master, et à le faire reconnaître aux autres ordinateurs en utilisant l'adresse IP du Ros-master. Les ordinateurs doivent évidemment être connectés dans le même réseau.

Parmi les trois ordinateurs utilisés, le premier symbolise le superviseur et est aussi le master. Il comporte le programme d'algorithme d'allocation des tâches, les positions des différents stocks et machines, le topic *operation_info*, le service *operation_processed*, le tout dans un noeud Ros nommé *supervisor_pub.py*. Les deux autres ordinateurs symbolisent les robots qui en plus des topics et service incluent dans le superviseur, un service pour le contrôle du gripper et le noeud *ur_driver* et l'ensemble du meta-package universal robot lancés dans des namespace *"/ur10"* et *"/ur5"* respectivement pour le robot UR10 et UR5. Il est important de lancer les noeuds et topics dans des namespace différents pour éviter les des conflits de noms. Les robots et les ordinateurs sont connectés au même réseau via Ethernet. Ainsi les ordinateurs peuvent envoyer des commandes aux robots (déplacement vers un point, ouverture/-fermeture gripper) et les robots peuvent aussi envoyer des informations sur leur état actuel, d'où l'interaction entre eux(robot et ordinateur). L'architecture dans la figure A.5, ci-dessous donne un aperçu de la description faite.

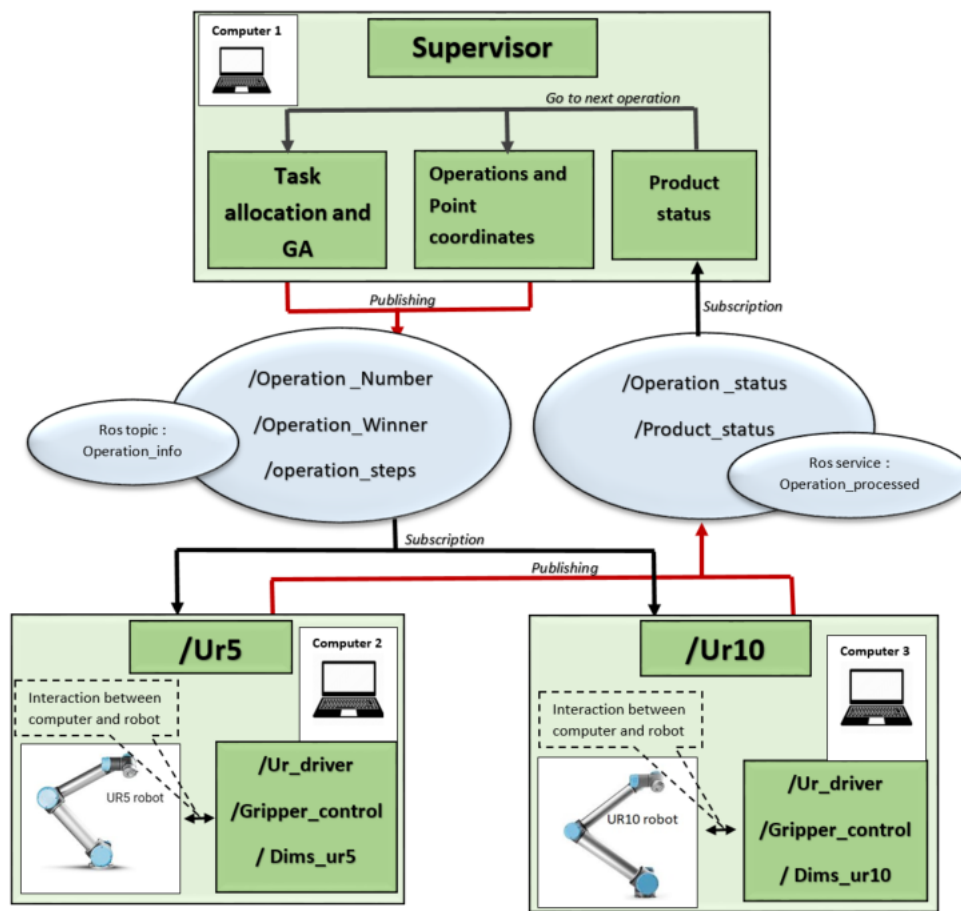


FIGURE A.5 – Architecture implémentation via ROS

Annexe B

Modèle de l'approche de plans de transport et de production intégrés : IT2P

Cette annexe décrit l'approche IT2P développée par l'université d'EXETER et le laboratoire Lineact du CESI de Rouen. La section B.1 de cette annexe décrit le modèle mathématique associé, et la section B.2 explique le codage du chromosome utilisé dans l'algorithme génétique.

B.1 Modèle mathématique de l'approche IT2P

L'approche IT2P a été développée pour la résolution simultanée des problèmes d'ordonnancement des tâches de transport et production. Une flexibilité de routage est en plus intégrée.

B.1.1 Notations

Ensembles et indices

\mathcal{M}	Ensemble des indices des machines
\mathcal{R}	
\mathcal{J}	Ensemble des indices des AGVs
j	Indice des Jobs, $j \in \{1, \dots, J\}$ avec $J = \mathcal{J} $.
m	Indice des machines de production $m \in \{1, \dots, M\}$ avec $M = \mathcal{M} $
r	Indice des robots, $r \in \{1, \dots, R\}$ avec $R = \mathcal{R} $.
i	Indice des tâches, $i \in \{1, \dots, I^j\} \forall j \in \mathcal{J}$ avec I^j le nombre de tâches d'un Job j
$\mathcal{M}^{j,i}$	Sous-ensemble de machines capable de réaliser la tâche i du Job j avec $\mathcal{M}^{j,i} \subset \mathcal{M}$

Paramètres :

$p_{j,i,m}$	Temps d'exécution de la tâche i du Job j sur la machine m
$d_{m,m'}$	Temps de transport entre m et m' , $m, m' \in \mathcal{M}$

Variables de décision :

$X_{j,i,m}$	Variable binaire; égale à : 1 si la machine m est allouée à la tâche i du Job j ; 0 sinon
$Y_{j,i,r}$	Variable binaire; égale à : 1 si l'AGV r est en charge de transporter le produit associé au Job j à la machine allouée à la tâche i ; 0 sinon

$H_{j,i,m}^-$	Variable binaire; égale à : 1 si l'AGV r est en charge de transporter le produit associé au Job j à la machine allouée à la tâche i est en train d'attendre à la machine m ; 0 sinon;
$H_{j,i,r}^{m',m}$	Variable binaire; égale à : 1 si l'AGV r est en charge de transporter le produit associé au Job j à la machine allouée à la tâche i est au niveau de la machine m' ; 0 sinon;
$S_{j,i}$	Variable entière; Temps de début de la tâche i du Job j
$E_{j,i}$	Variable entière; Temps de fin de la tâche i du Job j
$S_{j,i}^t$	Variable entière; Temps de début de la tâche de transport du Job j vers la machine allouée à la tâche i
$E_{j,i}^t$	Variable entière; Temps de fin de la tâche de transport du Job j vers la machine allouée à la tâche i

B.1.2 Fonction objectif et contraintes

La fonction objectif vise à minimiser le Makespan, peut s'écrire selon l'équation **B.1** :

$$\text{Minimiser } \max_{j \in \mathcal{J}} C_j, \quad (\text{B.1})$$

Contrainte par :

$$C_j = E_{j,I^j+1}^t = S_{j,I^j+1}^t + \sum_{m \in \mathcal{M}^{i,I^j}} \sum_{r \in \mathcal{R}} d_{m,0} \times Y_{j,I^j+1,r} \times X_{j,I^j,m} \forall j \in \mathcal{J} \quad (\text{B.2})$$

$$S_{j,i+1}^t \geq S_{j,i} + \sum_{m \in \mathcal{M}^{i,i}} p_{j,i,m} \times X_{j,i,m} \forall j \in \mathcal{J} \forall i \in 1, \dots, I^j \quad (\text{B.3})$$

$$S_{j,i+1}^t \geq \sum_r \sum_{j'} \sum_{i'=1}^{I_{j,i+1,j'}^+} \sum_{m \in \mathcal{M}^{i,i}} \sum_{m' \in \mathcal{M}^{i',i'}} \left(E_{j',i'}^t + d_{m,m'} \right) \times W_{j,i,j',i',m,m',r} \forall j \in \mathcal{J} \forall i \in 1, \dots, I^j \quad (\text{B.4})$$

Avec : $I_{j,i,j'}^+ = i$ si $j = j'$; $I^j + 1$ sinon; et $W_{j,i,j',i',m,m',r} = X_{j',i',m} \times Y_{j',i',r} \times H_{j,i+1,m'}^- \times Y_{j,i,r}$

La contrainte **B.2** est relative au temps de livraison d'un produit associé à un Job j , ce qui correspond à la fin de sa dernière tâche de transport. Cette tâche consiste à la ramener au point de stockage des produits.

Les contraintes **B.3** et **B.4** assurent le fait qu'une tâche de transport ne peut commencer sans l'AGV et le produit ne soient disponibles.

B.2 Codage du chromosome

L'algorithme génétique fait partie des solutions de sélection de tâches à assigner. L'autre étant la programmation par contrainte, vue dans la section **4.4** du chapitre **4**.

Le chromosome codé pour le GA, contient une série de numéro. Chaque numéro étant associé à une tâche d'un Job, la longueur du chromosome correspond au nombre total de tâches de l'ensemble des Jobs.

La figure **B.1**, représente un exemple de codage de chromosome. On y re-

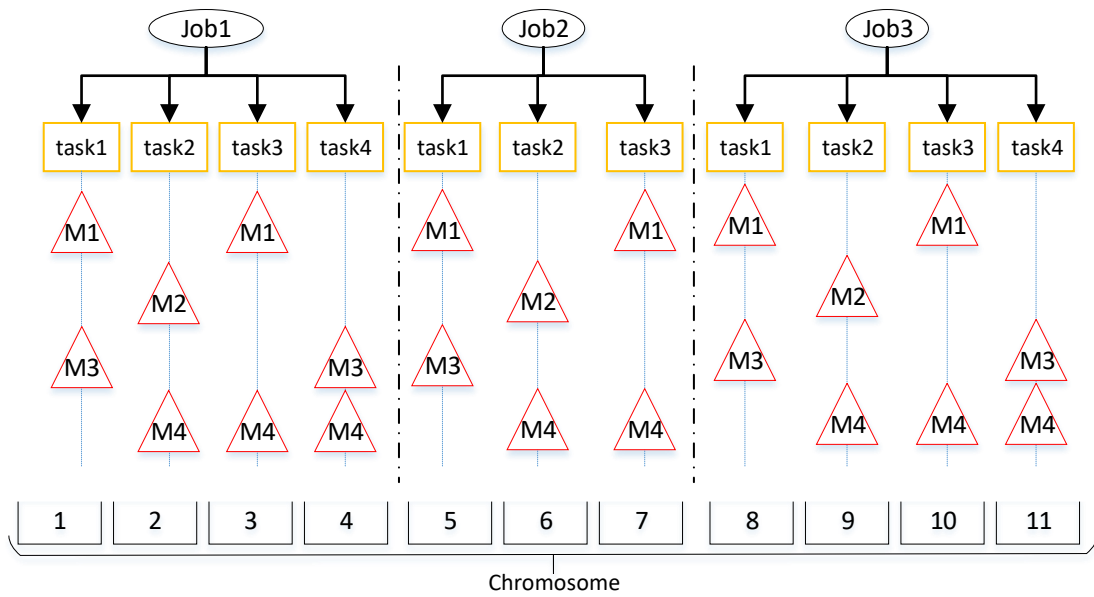


FIGURE B.1 – Exemple de chromosome

trouve, trois Jobs et onze tâches au total. A chaque tâche est associée les machines capables de la réaliser, et aussi un numéro correspondant à sa priorité ou ordre d'exécution. La tâche ayant le plus petit numéro dispose de la haute priorité, donc elle sera assignée en première et ainsi de suite.

Bibliographie

- [1] Imtiaz Ahmed and Warren W Fisher. Due date assignment, job order release, and sequencing interaction in job shop scheduling. *Decision sciences*, 23(3) :633–647, 1992.
- [2] AA Aly and M Subramaniam. Design of an fms decision support system. *International Journal of Production Research*, 31(10) :2257–2273, 1993.
- [3] Yacine Amara. *Contribution à la conception et à la commande des machines synchrones à double excitation. Application au véhicule hybride*. PhD thesis, UNIVERSITE PARIS XI, 2001.
- [4] Anthony Ikechukwu Anosike. *Agent-based modelling simulation and control of dynamically integrated manufacturing systems*. PhD thesis, University of Exeter, 2005.
- [5] Dennis A Attwood, Joseph M Deeb, Mary E Danz-Reece, et al. *Ergonomic solutions for the process industries*. Elsevier, 2004.
- [6] Jakob Axelsson, Joakim Fröberg, and Peter Eriksson. Architecting systems-of-systems and their constituents : A case study applying industry 4.0 in the construction domain. *Systems Engineering*, 22(6) :455–470, 2019.
- [7] NK Baid and NN Nagarur. An integrated decision support system for fms : using intelligent simulation. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 32(4) :951–965, 1994.
- [8] George H Baker and Allan Berg. Supervisory control and data acquisition (scada) systems. *The Critical Infrastructure Protection Report*, 1(6), 2002.
- [9] Kyle Banker. *MongoDB in action*. Manning Publications Co., 2011.
- [10] José Barbosa, Paulo Leitão, Damien Trentesaux, and Emmanuel Adam. Enhancing adacor with biology insights towards reconfigurable manufacturing systems. In *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pages 2746–2751. IEEE, 2011.
- [11] José Barbosa. *Self-organized and evolvable holonic architecture for manufacturing control*. PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2015.
- [12] Alfred Bauer, Jimmie Browne, R Bowden, and James Duggan. *Shop floor control systems : from design to implementation*. Springer Science & Business Media, 2012.

- [13] Joseph L Baxter, EK Burke, Jonathan M Garibaldi, and Mark Norman. Multi-robot search and rescue : A potential field based approach. In *Autonomous robots and agents*, pages 9–16. Springer, 2007.
- [14] Farouk Belkadi, Mourad Messaadia, Alain Bernard, and David Baudry. Collaboration management framework for oem–suppliers relationships : a trust-based conceptual approach. *Enterprise Information Systems*, 11(7) :1018–1042, 2017.
- [15] Anandhi Bharadwaj, Ajay S Vinze, and Arun Sen. Blackboard architecture for reactive scheduling. *Expert Systems with Applications*, 7(1) :55–65, 1994.
- [16] David Bitonneau. *Conception de systèmes cobotiques industriels : approche robotique avec prise en compte des facteurs humains : application à l'industrie manufacturière au sein de Safran et ArianeGroup*. PhD thesis, Bordeaux, 2018.
- [17] Jacek Blazewicz, Horst A Eiselt, Gerd Finke, Gilbert Laporte, and Jan Weglarz. Scheduling tasks and vehicles in a flexible manufacturing system. *International Journal of Flexible Manufacturing Systems*, 4(1) :5–16, 1991.
- [18] Anna Rosaria Boccella, Centobelli Piera, Roberto Cerchione, Teresa Murino, et al. Evaluating centralized and heterarchical control of smart manufacturing systems in the era of industry 4.0. *Applied Sciences*, 10(3) :755, 2020.
- [19] L Bongaerts. Integration of scheduling and control in holonic manufacturing systems (phd thesis). *Heverlee (Leuven), Belgium, Katholieke Universiteit Leuven*, 1998.
- [20] Ivo Boškoski, Camilla Gallo, Michael B Wallace, and Guido Costamagna. Covid-19 pandemic and personal protective equipment shortage : protective efficacy comparing masks and scientific methods for respirator reuse. *Gastrointestinal endoscopy*, 2020.
- [21] Shaun Bowler, Indridi H Indridason, and Thomas Gschwend. Coalition policy perceptions. *The Journal of Politics*, 82(4) :000–000, 2020.
- [22] Cynthia Breazeal and Brian Scassellati. Robots that imitate humans. *Trends in cognitive sciences*, 6(11) :481–487, 2002.
- [23] Jim Browne, Didier Dubois, Keith Rathmill, Suresh P Sethi, Kathryn E Stecke, et al. Classification of flexible manufacturing systems. *The FMS magazine*, 2(2) :114–117, 1984.
- [24] Vadim V Bulitko and David C Wilkins. Using petri nets to represent context in blackboard scheduling. In *Proceedings of the American Association for Artificial Intelligence (AAAI) Workshop on Reasoning in Context for AI Applications*. Orlando, Florida, USA, 1999.

- [25] Olivier Causse. Man-machine interface and supervisor for a mobile robot. In *Mobile Robots VII*, volume 1831, pages 166–176. International Society for Optics and Photonics, 1993.
- [26] David John Chapman, Andrew J Dowling, and Rohan J Mc Adam. Human machine interface, June 30 2009. US Patent 7,555,706.
- [27] Bernard Claverie, Benoît Le Blanc, and Pascal Fouillat. La cobotique. la robotique soumise. *Communication et organisation*, (44) :203–214, 2013.
- [28] Aurelie Clodic. *Supervision pour un robot interactif : Action et Interaction pour un robot autonome en environnement humain*. PhD thesis, Université Paul Sabatier-Toulouse III, 2007.
- [29] J Edward Colgate, J Edward, Michael A Peshkin, and Witaya Wannasuphprasit. *Cobots : Robots for collaboration with human operators*. 1996.
- [30] Alberto Costa and Leo Liberti. Relaxations of multilinear convex envelopes : dual is better than primal. In *International Symposium on Experimental Algorithms*, pages 87–98. Springer, 2012.
- [31] M Courvoisier, M Combacau, and A de Bonneval. Control and monitoring of large discrete event systems : a generic approach (to ems). In *Industrial Electronics, 1993. Conference Proceedings, ISIE'93-Budapest., IEEE International Symposium on*, pages 571–576. IEEE, 1993.
- [32] James L Crowley. Coordination of action and perception in a surveillance robot. *IEEE Expert*, 2(4) :32–43, 1987.
- [33] James L Crowley, Frank Wallner, and Bernt Schiele. Position estimation using principal components of range data. *Robotics and Autonomous Systems*, 23(4) :267–276, 1998.
- [34] Marcos R Da Silveira and Michel Combacau. Supervision and control of heterarchical discrete event systems. *Sba : Controle & Automação Sociedade Brasileira de Automatica*, 17(1) :1–9, 2006.
- [35] Quang-Vinh Dang, Cong Thanh Nguyen, and Hana Rudová. Scheduling of mobile robots for transportation and manufacturing tasks. *Journal of Heuristics*, 25(2) :175–213, 2019.
- [36] Keith Decker, Alan Garvey, Marty Humphrey, and Victor Lesser. Control heuristics for scheduling in a parallel blackboard system. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(02) :243–264, 1993.
- [37] William Derigent, Olivier Cardin, and Damien Trentesaux. Industry 4.0 : contributions of holonic manufacturing control architectures and future challenges. *Journal of Intelligent Manufacturing*, pages 1–22, 2020.
- [38] Bernhard Dieber and Benjamin Breiling. Security considerations in modular mobile manipulation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 70–77. IEEE, 2019.

- [39] David M Dilts, Neil P Boyd, and HH Whorms. The evolution of control architectures for automated manufacturing systems. *Journal of manufacturing systems*, 10(1) :79–93, 1991.
- [40] Panneer DD Dominic, Sathya Kaliyamoorthy, and M Saravana Kumar. Efficient dispatching rules for dynamic job shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 24(1-2) :70–75, 2004.
- [41] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4) :28–39, 2006.
- [42] Jon Doyle and MP Wellman. Rational distributed reason maintenance for planning and replanning of large-scale activities. In *Proceedings of the DARPA Workshop on Planning and Scheduling*, pages 28–36. Morgan Kaufmann, 1990.
- [43] IMEN Driss, KINZA NADIA Mouss, and ASSIA Laggoun. An effective genetic algorithm for the flexible job shop scheduling problems. In *11th Congres Int. de Genine Industriel–CIGI2015, Québec, Canada*, pages 26–28, 2015.
- [44] Dmitrii Drozdov, Udayanto Dwi Atmojo, Cheng Pang, Sandeep Patil, Muhammad Irfan Ali, Antti Tenhunen, Taavi Oksanen, Kiril Chernetiev, Valeriy Vyatkin, et al. Utilizing software design patterns in product-driven manufacturing system. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, 2020.
- [45] Neil A Duffie. Challenges in design of heterarchical controls for dynamic logistic systems. In *Dynamics in Logistics*, pages 3–24. Springer, 2008.
- [46] Neil A Duffie and Rex S Piper. Nonhierarchical control of manufacturing systems. *Journal of Manufacturing systems*, 5(2) :141, 1986.
- [47] Neil A Duffie and Vittaldas V Prabhu. Real-time distributed scheduling of heterarchical manufacturing systems. *Journal of Manufacturing Systems*, 13(2) :94–107, 1994.
- [48] Edmund H Durfee, Victor R Lesser, and Daniel D Corkill. Trends in cooperative distributed problem solving. *IEEE Transactions on knowledge and data Engineering*, 1989.
- [49] Ayan Dutta, Vladimir Ufimtsev, and Asai Asaithambi. Correlation clustering based coalition formation for multi-robot task allocation. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 906–913, 2019.
- [50] Alberto Elfes. Incorporating spatial representations at multiple levels of abstraction in a replicated multilayered architecture for robot control. In *Intelligent Robots—Sensing, Modeling And Planning*, pages 249–266. World Scientific, 1997.

- [51] José Eloundou. *Modélisation multi-contraintes d'un système de production flexible*. PhD thesis, INSA de Rouen, 2016.
- [52] Muhammad Hafidz Fazli Bin Md Fauadi, Wan-Ling Li, and Tomohiro Murata. Combinatorial auction method for decentralized task assignment of multiple-loading capacity agv based on intelligent agent architecture. In *2011 Second International Conference on Innovations in Bio-inspired Computing and Applications*, pages 207–211. IEEE, 2011.
- [53] Souleymane Moussa G., Belgacem Bettayeb, M'hammed Sahnoun, Fabrice Duval, and Abdelaziz Bensrhair. Modular mobile manipulators coalition formation through distributed transportation tasks allocation. *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture*, 0(0) :0954405420961225, 0.
- [54] Michael R Garey, David S Johnson, and Ravi Sethi. The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*, 1(2) :117–129, 1976.
- [55] Jared L Gearhart, Kristin L Adair, Richard J Detry, Justin D Durfee, Katherine A Jones, and Nathaniel Martin. Comparison of open-source linear programming solvers. *Sandia National Laboratories, SAND2013-8847*, 2013.
- [56] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9) :939–954, 2004.
- [57] Donald Gerwin. An agenda for research on the flexibility of manufacturing processes. *International Journal of Operations & Production Management*, 2005.
- [58] A Gnanavel Babu, J Jerald, A Noorul Haq, V Muthu Luxmi, and TP Vigneswaralu. Scheduling of machines and automated guided vehicles in fms using differential evolution. *International Journal of Production Research*, 48(16) :4683–4699, 2010.
- [59] Sergio R González, Ivan Mondragón, Gabriel Zambrano, Wilson Hernandez, and Henry Montaña. Manufacturing control architecture for fms with agv : A state-of-the-art. In *Advances in Automation and Robotics Research in Latin America*, pages 157–172. Springer, 2017.
- [60] Dominic Gorecky, Mathias Schmitt, Matthias Loskyll, and Detlef Zühlke. Human-machine-interaction in the industry 4.0 era. In *2014 12th IEEE international conference on industrial informatics (INDIN)*, pages 289–294. IEEE, 2014.
- [61] Souleymane MOUSSA GOUMEYE, Fabrice DUVAL M'Hammed SAH-NOUNa, and Abdelaziz BENSRAHAIR. Modular mobile manipulators task allocation auction-based approach b. In *Advances in Manufacturing*

- Technology XXXIII : Proceedings of the 17th International Conference on Manufacturing Research, incorporating the 34th National Conference on Manufacturing Research, 10-12 September 2019, Queen's University, Belfast, volume 9, page 99. IOS Press, 2019.*
- [62] Manman Guo, Hui Feng, Jun Zhang, Wenqin Wang, Yi Wang, Yuqing Li, Chunhui Gao, Huanchun Chen, Ying Feng, and Zheng-Guo He. Dissecting transcription regulatory pathways through a new bacterial one-hybrid reporter system. *Genome Research*, 19(7) :1301–1308, 2009.
- [63] Ehud Halberstam, Luis Navarro-Serment, Ronald Conescu, Sandra Mau, Gregg Podnar, Alan D Guisewite, H Benjamin Brown, Alberto Elfes, John M Dolan, and Marcel Bergerman. A robot supervision architecture for safe and efficient space exploration and operation. In *Earth & Space 2006 : Engineering, Construction, and Operations in Challenging Environment*, pages 2–8. 2006.
- [64] Józef Hatvany. Intelligence and cooperation in heterarchic manufacturing systems. *Robotics and computer-integrated manufacturing*, 2(2) :101–104, 1985.
- [65] Vincent Havard. *Développement de méthodes et outils basés sur la réalité augmentée et virtuelle pour l'assistance ou l'apprentissage d'opérations dans un contexte industriel*. PhD Thesis, 2018.
- [66] V Havarda, M Sahnouna, B Bettayeba, and D Baudrya. An architecture for data management, visualisation and supervision of cyber-physical production systems. In *Advances in Manufacturing Technology XXXIII : Proceedings of the 17th International Conference on Manufacturing Research, incorporating the 34th National Conference on Manufacturing Research, 10-12 September 2019, Queen's University, Belfast, volume 9, page 81. IOS Press, 2019.*
- [67] Naihui He. Agent-based hierarchical planning and scheduling control in dynamically integrated manufacturing system. 2011.
- [68] Sunderesh S Heragu, Robert J Graves, Byung-In Kim, and Art St Onge. Intelligent agent based framework for manufacturing systems control. *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans*, 32(5) :560–573, 2002.
- [69] Eduardo G Hernandez-Martinez, Sergio A Foyo-Valdes, Erika S Puga-Velazquez, and Jesús A Meda-Campaña. Hybrid architecture for coordination of agvs in fms. *International Journal of Advanced Robotic Systems*, 11(3) :41, 2014.
- [70] Cédric Herpson. *Approche multi-agents pour la supervision adaptative des systèmes distribués*. PhD thesis, Université Pierre et Marie Curie-Paris 6, 2012.

- [71] B Hichri, JC Fauroux, L Adouane, I Doroftei, and Y Mezouar. Design of cooperative mobile robots for co-manipulation and transportation tasks. *Robotics and Computer-Integrated Manufacturing*, 57 :412–421, 2019.
- [72] Seyed Mahdi Homayouni and Dalila BMM Fontes. Joint scheduling of production and transport with alternative job routing in flexible manufacturing systems. In *AIP Conference Proceedings*, volume 2070, page 020045. AIP Publishing LLC, 2019.
- [73] Shirley M Hord. Working together : Cooperation or collaboration?. 1981.
- [74] Fu-Shiung Hsieh. Holarchy formation and optimization in holonic manufacturing systems with contract net. *Automatica*, 44(4) :959–970, 2008.
- [75] Dengya Huo, Yaohua Wu, and Shasha Wu. The scheduling technology development of multi-agv system in ai era. In *Journal of Physics : Conference Series*, volume 1575, page 012154. IOP Publishing, 2020.
- [76] Constantin Ilas. Electronic sensing technologies for autonomous ground vehicles : A review. In *2013 8th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pages 1–6. IEEE, 2013.
- [77] Jacquie Jarvis, Ralph Rönquist, D McFarlane, and L Jain. A team-based holonic approach to robotic assembly cell control. *Journal of Network and Computer Applications*, 29(2-3) :160–176, 2005.
- [78] Diebold John. Automation. the advent of the automatic factory. *New York*, 1952.
- [79] Essam Kaoud, Mahmoud A El-Sharief, and MG El-Sebaie. Scheduling problems of automated guided vehicles in job shop, flow shop, and container terminals. In *2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 60–65. IEEE, 2017.
- [80] R Rangadurai Karthick, Vipul P Hattiwale, and Balaraman Ravindran. Adaptive network intrusion detection system using a hybrid approach. In *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, pages 1–7. IEEE, 2012.
- [81] Byung-In Kim, Sunderesh S Heragu, Robert J Graves, and A St Onge. A hybrid scheduling and control system architecture for warehouse management. *IEEE Transactions on Robotics and Automation*, 19(6) :991–1001, 2003.
- [82] Victor Klee and George J Minty. How good is the simplex algorithm. *Inequalities*, 3(3) :159–175, 1972.
- [83] Édouard Kleinpeter. Le cobot, la coopération entre l'utilisateur et la machine. *Multitudes*, (1) :70–75, 2015.

- [84] Tore Knabe, Michael Schillo, and Klaus Fischer. Improvements to the fipa contract net protocol for performance increase and cascading applications. In *In International Workshop for Multi-Agent Interoperability at the German Conference on AI (KI-2002*. Citeseer, 2002.
- [85] Arthur Koestler. *The ghost in the machine*. 1968.
- [86] Norihiro Kokudo and Haruhito Sugiyama. Call for international cooperation and collaboration to effectively tackle the covid-19 pandemic. *Global Health & Medicine*, 2(2) :60–62, 2020.
- [87] Yoram Koren and Moshe Shpitalni. Design of reconfigurable manufacturing systems. *Journal of manufacturing systems*, 29(4) :130–141, 2010.
- [88] Przemysław Korytkowski, Tomasz Wiśniewski, and Szymon Rymaszewski. An evolutionary simulation-based optimization approach for dispatching scheduling. *Simulation Modelling Practice and Theory*, 35 :69–85, 2013.
- [89] Olga Kozar. Towards better group work : Seeing the difference between cooperation and collaboration. In *English teaching forum*, volume 48, pages 16–23. ERIC, 2010.
- [90] MV Satish Kumar, Ranga Janardhana, and CSP Rao. Simultaneous scheduling of machines and vehicles in an fms environment with alternative routing. *The International Journal of Advanced Manufacturing Technology*, 53(1-4) :339–351, 2011.
- [91] Shailendra Kumar, Ajay Goyal, and Ankit Singhal. Manufacturing flexibility and its effect on system performance. *Jordan Journal of Mechanical & Industrial Engineering*, 11(2), 2017.
- [92] Yu-Kwong Kwok and Ishfaq Ahmad. Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys (CSUR)*, 31(4) :406–471, 1999.
- [93] Tanguy Lapegue. *Planification de personnel avec affectation de tâches fixées : méthodes et application dans un contexte médical*. PhD thesis, 2014.
- [94] Tuan Le-Anh and MBM De Koster. A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, 171(1) :1–23, 2006.
- [95] Chung-Yee Lee and Zhi-Long Chen. Machine scheduling with transportation considerations. *Journal of scheduling*, 4(1) :3–24, 2001.
- [96] Paulo Leitão. Agent-based distributed manufacturing control : A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7) :979–991, 2009.
- [97] Paulo Leitão, João Mendes, Armando W Colombo, and Francisco Restivo. Reconfigurable production control systems : beyond adacor. *IFAC Proceedings Volumes*, 40(3) :129–134, 2007.

- [98] Paulo Leitão and Francisco Restivo. Adacor : A holonic architecture for agile and adaptive manufacturing control. *Computers in industry*, 57(2) :121–130, 2006.
- [99] Claus Lenz, Suraj Nair, Markus Rickert, Alois Knoll, Wolfgang Rosel, Jurgen Gast, Alexander Bannat, and Frank Wallhoff. Joint-action for humans and industrial robots for assembly tasks. In *RO-MAN 2008-The 17th IEEE International Symposium on Robot and Human Interactive Communication*, pages 130–135. IEEE, 2008.
- [100] Daniel Leonard, Ngo Van Long, and Van Long Ngo. *Optimal control theory and static optimization in economics*. Cambridge University Press, 1992.
- [101] Niels Lohse. *Towards an ontology framework for the integrated design of modular assembly systems*. PhD thesis, University of Nottingham Nottingham, 2006.
- [102] Jing Long and Chun Liang Zhang. The summary of agv guidance technology. In *Advanced Materials Research*, volume 591, pages 1625–1628. Trans Tech Publ, 2012.
- [103] Lingzhi Luo, Nilanjan Chakraborty, and Katia Sycara. Distributed algorithms for multirobot task assignment with task deadline constraints. *IEEE Transactions on Automation Science and Engineering*, 12 :1–13, 07 2015.
- [104] Liam Lynch, Thomas Newe, John Clifford, Joseph Coleman, Joseph Walsh, and Daniel Toal. Automated ground vehicle (agv) and sensor technologies-a review. In *2018 12th International Conference on Sensing Technology (ICST)*, pages 347–352. IEEE, 2018.
- [105] Ali Ahmad Malik, Tariq Masood, and Rehana Kousar. Repurposing factories with robotics in the face of covid-19. 2020.
- [106] Gerard Marty. *Sociologie des institutions marchandes : le cas de la mise en vente des bois publics en Lorraine*. PhD thesis, Université Paris Sorbonne (Paris 4), 2013.
- [107] Jeremy Marvel and Roger Bostelman. Towards mobile manipulator safety standards. In *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 31–36. IEEE, 2013.
- [108] Garth P McCormick. Computability of global solutions to factorable nonconvex programs : Part i—convex underestimating problems. *Mathematical programming*, 10(1) :147–175, 1976.
- [109] Ahmad Mehrabian, Reza Tavakkoli-Moghaddam, and Kaveh Khalili-Damagham. Multi-objective routing and scheduling in flexible manufacturing systems under uncertainty. *Iranian Journal of Fuzzy Systems*, 14(2) :45–77, 2017.

- [110] Tarek Menouer and Bertrand Le Cun. A parallelization mixing or-tools/gecode solvers on top of the bobpp framework. In *2013 Eighth international conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 242–246. IEEE, 2013.
- [111] Tarek Menouer, Bertrand Le Cun, and Pascal Vander-Swalmen. Partitioning methods to parallelize constraint programming solver using the parallel framework bobpp. In *Advanced Computational Methods for Knowledge Engineering*, pages 117–127. Springer, 2013.
- [112] Mourad Messaadia, Hadrien Szigeti, Magali Bosch-Mauchand, Matthieu Bricogne, Benoît Eynard, and Anirban Majumdar. Ict for design and manufacturing : A strategic vision for technology maturity assessment. In *ICoRD'13*, pages 913–924. Springer, 2013.
- [113] G Mossa, F Boenzi, S Digiesi, G Mummolo, and VA Romano. Productivity and ergonomic risk in human based production systems : A job-rotation scheduling model. *International Journal of Production Economics*, 171 :471–477, 2016.
- [114] Carla Mouradian, Jagruti Sahoo, Roch H Glitho, Monique J Morrow, and Paul A Polakos. A coalition formation algorithm for multi-robot task allocation in large-scale natural disasters. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1909–1914. IEEE, 2017.
- [115] Souleymane Moussa Goumeys, M'hammed Sahnoun, Fabrice Duval, and Abdelaziz Bensrhair. Bi-Objective indirect optimization of robotic transportation task assignment based on auction mechanism. In *2020 4th IEEE Electron Devices Technology & Manufacturing Conference (EDTM) (EDTM2020)*, Penang, Malaysia, March 2020.
- [116] Jörg P Müller and Klaus Fischer. Application impact of multi-agent systems and technologies : a survey. In *Agent-oriented software engineering*, pages 27–53. Springer, 2014.
- [117] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*, pages 9839–9849, 2018.
- [118] Juan M Novas, Jan Van Belle, Bart Saint Germain, and Paul Valckenaers. A collaborative framework between a scheduling system and a holonic manufacturing execution system. In *Service orientation in holonic and multi agent manufacturing and robotics*, pages 3–17. Springer, 2013.
- [119] Fatma A Omara and Mona M Arafa. Genetic algorithms for task scheduling problem. In *Foundations of Computational Intelligence Volume 3*, pages 479–507. Springer, 2009.

- [120] TA Ottaway and JR Burns. An adaptive production control system utilizing agent technology. *International Journal of Production Research*, 38(4) :721–737, 2000.
- [121] Cyrille Pach. *ORCA : Architecture hybride pour le contrôle de la myopie dans le cadre du pilotage des Systèmes Flexibles de Production*. PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2013.
- [122] Adrià Salvador Palau, Maharshi Harshadbhai Dhada, and Ajith Kumar Parlikad. Multi-agent system architectures for collaborative prognostics. *Journal of Intelligent Manufacturing*, 30(8) :2999–3013, 2019.
- [123] Michael A Palis, Jing-Chiou Liou, Sanguthevar Rajasekaran, SUNIL SHENDE, and David SL Wei. Online scheduling of dynamic trees. *Parallel Processing Letters*, 5(04) :635–646, 1995.
- [124] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. Situation awareness, mental workload, and trust in automation : Viable, empirically supported cognitive engineering constructs. *Journal of cognitive engineering and decision making*, 2(2) :140–160, 2008.
- [125] H Van Dyke Parunak. Fractal actors for distributed manufacturing control. In *Proc. 2nd Conference on AI Applications*, pages 653–660, 1985.
- [126] H Van Dyke Parunak, Albert D Baker, and Steven J Clark. The aaria agent architecture : An example of requirements-driven agent-based system design. In *Agents*, pages 482–483, 1997.
- [127] AC Pereira and Fernando Romero. A review of the meanings and the implications of the industry 4.0 concept. *Procedia Manufacturing*, 13 :1206–1214, 2017.
- [128] M Peschl. *An architecture for flexible manufacturing systems based on task-driven agents*. PhD thesis, Thesis to Doctoral program–Faculty of Information Technology and Electrical Engineering, Oulu University, Finland, 2014.
- [129] Michael Pinedo. *Scheduling*, volume 29. Springer, 2012.
- [130] Patrick Pujo, Nicolas Broissin, and Fouzia Ounnar. Prosis : An isoarchic structure for hms control. *Engineering Applications of Artificial Intelligence*, 22(7) :1034–1045, 2009.
- [131] Patrick Pujo and Fouzia Ounnar. Vers une approche holonique des systèmes mécatroniques complexes : proposition d’un système de pilotage auto organisé et isoarchique. *Journal Européen des Systèmes Automatisés (JESA)*, 41(6) :673–706, 2007.
- [132] Morgan Quigley, Brian Gerkey, and William D Smart. *Programming Robots with ROS : a practical introduction to the Robot Operating System*. "O’Reilly Media, Inc.", 2015.

- [133] Azrul Azwan Abdul Rahman. Revolution of production system for the industry 4.0. In *Mass Production Processes*. IntechOpen, 2020.
- [134] Ragesh K Ramachandran, James A Preiss, and Gaurav S Sukhatme. Resilience by reconfiguration : Exploiting heterogeneity in robot teams. *arXiv preprint arXiv :1903.04856*, 2019.
- [135] N Raman. Simultaneous scheduling of machines and material handling devices in automated manufacturing. In *Proc. of the First ORSA/TIMS Conf. on Flexible Manufacturing Systems, Ann Arbor, MI*, pages 455–465, 1986.
- [136] Paul G Ranky. *Flexible manufacturing cells and systems in CIM*. CIMware, 1990.
- [137] Amit Rauniyar and Pranab K Muhuri. Multi-robot coalition formation problem : Task allocation with adaptive immigrants based genetic algorithms. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 000137–000142. IEEE, 2016.
- [138] Harish Ravichandar, Kenneth Shaw, and Sonia Chernova. Strata : unified framework for task assignments in large teams of heterogeneous agents. *Auton. Agents Multi Agent Syst.*, 34(2) :38, 2020.
- [139] BSP Reddy and CSP Rao. A hybrid multi-objective ga for simultaneous scheduling of machines and agvs in fms. *The International Journal of Advanced Manufacturing Technology*, 31(5-6) :602–613, 2006.
- [140] Gabriel Zambrano Rey, Thérèse Bonte, Vittaldas Prabhu, and Damien Trentesaux. Reducing myopic behavior in fms control : A semi-heterarchical simulation–optimization approach. *Simulation Modelling Practice and Theory*, 46 :53–75, 2014.
- [141] Jeremy Roschelle and Stephanie D Teasley. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, pages 69–97. Springer, 1995.
- [142] Ariel Rosenfeld, Noa Agmon, Oleg Maksimov, and Sarit Kraus. Intelligent agent supporting human–multi-robot team collaboration. *Artificial Intelligence*, 252 :211–231, 2017.
- [143] Kwangyeol Ryu and Mooyoung Jung. Agent-based fractal architecture and modelling for developing distributed manufacturing systems. *International Journal of Production Research*, 41(17) :4233–4255, 2003.
- [144] Georgios KD Saharidis. Supply chain optimization : Centralized vs decentralized planning and scheduling. *Supply Chain Management*, pages 3–26, 2011.
- [145] M’hammed Sahnoun, Belgacem Bettayeb, Samuel-Jean Bassetto, and Michel Tollenaere. Simulation-based optimization of sampling plans to reduce inspections while mastering the risk exposure in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 27(6) :1335–1349, 2016.

- [146] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1) :209–238, 1999.
- [147] Michael Schneier, Michael Schneier, and Roger Bostelman. *Literature review of mobile robots for manufacturing*. US Department of Commerce, National Institute of Standards and Technology, 2015.
- [148] Ali Sekmen, Ahmet Bugra Koku, and Saleh Zein-Sabatto. Human robot interaction via cellular phones. In *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, volume 4, pages 3937–3942. IEEE, 2003.
- [149] Andrea Krasa Sethi and Suresh Pal Sethi. Flexibility in manufacturing : a survey. *International journal of flexible manufacturing systems*, 2(4) :289–328, 1990.
- [150] Saqib Shamim, Shuang Cang, Hongnian Yu, and Yun Li. Management approaches for industry 4.0 : A human resource management perspective. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 5309–5316. IEEE, 2016.
- [151] Ramesh Sharda, Dursun Delen, Efraim Turban, J Aronson, and T Liang. Business intelligence and analytics. *System for Decesion Support*, 2014.
- [152] Liji Shen, Stéphane Dauzère-Pérès, and Janis S Neufeld. Solving the flexible job shop scheduling problem with sequence-dependent setup times. *European Journal of Operational Research*, 265(2) :503–516, 2018.
- [153] John P Shewchuk and Colin L Moodie. Definition and classification of manufacturing flexibility types and measures. *International Journal of Flexible Manufacturing Systems*, 10(4) :325–349, 1998.
- [154] Shiego Shingo. Maîtrise de la production et méthode kanban. *Le cas Toyota*, 1983.
- [155] Alexander Smirnov, Alexey Kashevnik, Mikhail Petrov, and Vladimir Parfenov. Context-based coalition creation in human-robot systems : approach and case study. In *International Conference on Interactive Collaborative Robotics*, pages 229–238. Springer, 2017.
- [156] Alexander Smirnov, Leonid Sheremetov, and Nikolay Teslya. Fuzzy cooperative games usage in smart contracts for dynamic robot coalition formation : Approach and use case description. In *Proceedings of the 21st International Conference on Enterprise Information Systems, ICEIS*, pages 361–370, 2019.
- [157] Jeffrey S Smith, Walter C Hoberecht, and Sanjay B Joshi. A shop-floor control architecture for computer-integrated manufacturing. *IIE transactions*, 28(10) :783–794, 1996.

- [158] Reid G Smith. The contract net protocol : High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, (12) :1104–1113, 1980.
- [159] Reid G Smith. The contract net protocol : High-level communication and control in a distributed problem solver. In *Readings in distributed artificial intelligence*, pages 357–366. Elsevier, 1988.
- [160] Moussa Goumeye Souleymane, M’Hammed Sahnoun, David Baudry, and Anne Louis. SUPERVISION D’UN SYSTEME DE PRODUCTION CONSTITUE DE ROBOTS COLLABORATIFS. In *confere 2017*, volume 17, seville, Spain, July 2017.
- [161] Kathryn E Stecke and Narayan Raman. Fms planning decisions, operating flexibilities, and system performance. *IEEE Transactions on Engineering Management*, 42(1) :82–90, 1995.
- [162] SH Sundarani, MN Qureshi, et al. Study of adoption barriers for flexible manufacturing system in industry. *Voice of Research*, 5(4) :64–67, 2017.
- [163] Hadrien Szigeti, Mourad Messaadia, Anirban Majumdar, and Benoit Eynard. Steep analysis as a tool for building technology roadmaps. In *Internationale challenges e-2011 conference, Florence*, pages 26–28, 2011.
- [164] Fang Tang and Lynne E Parker. Peer-to-peer human-robot teaming through reconfigurable schemas. In *AAAI Spring Symposium : To Boldly Go Where No Human-Robot Team Has Gone Before*, pages 26–29, 2006.
- [165] Herbert G Tanner and Kostas J Kyriakopoulos. Nonholonomic motion planning for mobile manipulators. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 1233–1238. IEEE, 2000.
- [166] A Tharumarajah. Comparison of the bionic, fractal and holonic manufacturing system concepts. *International Journal of Computer Integrated Manufacturing*, 9(3) :217–226, 1996.
- [167] D Tihay and N Perrin. Human-robot coactivity : need’s analysis. *Scientific committee*, page 40, 2018.
- [168] Seth Tisue and Uri Wilensky. Netlogo : A simple environment for modeling complexity. In *International conference on complex systems*, volume 21, pages 16–21. Boston, MA, 2004.
- [169] Abdoul Karim Armand Toguyéni, Pascal Berruet, and Etienne Craye. Models and algorithms for failure diagnosis and recovery in fmss. *International Journal of Flexible Manufacturing Systems*, 15(1) :57–85, 2003.
- [170] Damien Trentesaux. Distributed control of production systems. *Engineering Applications of Artificial Intelligence*, 22(7) :971–978, 2009.

- [171] David Trimoulet, Madeleine El Zahera, Celine Viazzia, M'hammed SAHNOUN, and David Baudry. Multi robot path planning approach for dynamic environment. In *Advances in Manufacturing Technology XXXI : Proceedings of the 15th International Conference on Manufacturing Research, Incorporating the 32nd National Conference on Manufacturing Research, September 5–7, 2017, University of Greenwich, UK*, volume 6, page 275. IOS Press, 2017.
- [172] Shivadev K Ubhayakar and Robert D Baker. Robotic arm systems, September 4 1990. US Patent 4,954,952.
- [173] Kanji Ueda. A concept for bionic manufacturing systems based on dna-type information. In *Human Aspects in Computer Integrated Manufacturing*, pages 853–863. Elsevier, 1992.
- [174] Paul Valckenaers. Perspective on holonic manufacturing systems : Prosa becomes arti. *Computers in Industry*, 120 :103226, 2020.
- [175] Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, and Patrick Peeters. Reference architecture for holonic manufacturing systems : Prosa. *Computers in industry*, 37(3) :255–274, 1998.
- [176] Nikolaj van Omme, Laurent Perron, and Vincent Furnon. or-tools user's manual. Technical report, Google, 2014.
- [177] Guilherme Ernani Vieira and Claudimar Pereira Veiga. Hierarchical and centralized architectures for distributed production planning, scheduling and control activities. In *2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*, pages 341–346. IEEE, 2009.
- [178] Kelen Vivaldini, Luís F Rocha, Nádia Junqueira Martarelli, Marcelo Becker, and A Paulo Moreira. Integrated tasks assignment and routing for the estimation of the optimal number of agvs. *The International Journal of Advanced Manufacturing Technology*, 82(1-4) :719–736, 2016.
- [179] Steven Edward Walterscheid. Hand spinner novelty with integrated lights, February 20 2018. US Patent 9,895,620.
- [180] Witaya Wannasuphoprasit, Prasad Akella, Michael Peshkin, and J Edward Colgate. Cobots : A novel material handling technology. In *in proceedings of International Mechanical Engineering Congress and Exposition, Anaheim, ASME 98-WA/MH-2*. Citeseer, 1998.
- [181] Jens Wawerla and Richard T Vaughan. A fast and frugal method for team-task allocation in a multi-robot transportation system. In *2010 IEEE International Conference on Robotics and Automation*, pages 1432–1437. IEEE, 2010.
- [182] Danny Weyns, Nelis Boucké, and Tom Holvoet. Gradient field-based task assignment in an agv transportation system. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 842–849. ACM, 2006.

- [183] Scott Wolford. *The politics of military coalitions*. Cambridge University Press, 2015.
- [184] Annie S Wu, Han Yu, Shiyuan Jin, K-C Lin, and Guy Schiavone. An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE Transactions on parallel and distributed systems*, 15(9) :824–834, 2004.
- [185] Danfeng Wu, Guangping Zeng, Lingguo Meng, Weijian Zhou, and Linmin Li. Gini coefficient-based task allocation for multi-robot systems with limited energy resources. *IEEE/CAA Journal of Automatica Sinica*, 5(1) :155–168, 2017.
- [186] Sangseok You and Lionel Peter Robert. Subgroup formation in human-robot teams. 2019.
- [187] Seyedeh Sarah Zabihzadeh and Javad Rezaeian. Two meta-heuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time. *Applied Soft Computing*, 40 :319–330, 2016.
- [188] Gabriel Zambrano, Cyrille Pach, Emmanuel Adam, Thierry Berger, and Damien Trentesaux. Myopic behaviour in heterarchical control of fms. In *International Conference on Industrial Engineering and Systems Management IESM*, 2011.
- [189] Nadine Zbib, Cyrille Pach, Yves Sallez, and Damien Trentesaux. Heterarchical production control in manufacturing systems using the potential fields concept. *Journal of Intelligent Manufacturing*, 23(5) :1649–1670, 2012.
- [190] David Zhengwen Zhang, Anthony Anosike, and Ming Kim Lim. Dynamically integrated manufacturing systems (dims)—a multiagent approach. *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans*, 37(5) :824–850, 2007.
- [191] David Zhengwen Zhang and Anthony Ikechukwu Anosike. Modeling and simulation of dynamically integrated manufacturing systems. *Journal of Intelligent Manufacturing*, 23(6) :2367–2382, 2012.
- [192] Xiaoxia Zhang, Chuanlong Wang, Liuzhu Han, and Xiaolin Li. On the gps/dr integrated navigation system for agv. *Journal of Shenyang Jianzhu University(Natural Science)*, 27(1) :173–177, 2011.
- [193] Ray Y Zhong, Xun Xu, Eberhard Klotz, and Stephen T Newman. Intelligent manufacturing in the context of industry 4.0 : a review. *Engineering*, 3(5) :616–630, 2017.
- [194] Keliang Zhou, Taigang Liu, and Lifeng Zhou. Industry 4.0 : Towards future industrial opportunities and challenges. In *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*, pages 2147–2152. IEEE, 2015.