Thèse de Doctorat en Cotutelle Internationale

Pour obtenir le grade de :

Docteur de l'Université Polytechnique Hauts-de-France et l'INSA Hauts-de-France

Discipline : Automatique et Informatique Industrielle

et le grade de :

Docteur de l'Université Abdelmalek Essaâdi

Discipline : Mathématiques, Spécialité : Logistique et Recherche Opérationnelle

Présentée et soutenue le 25 septembre 2020 à Tétouan, Maroc par :

Tarik CHARGUI

Ecoles doctorales :

Sciences Pour l'Ingénieur (ED SPI 072), Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH UMR CNRS 8201), Valenciennes, France

Centre d'Etudes Doctorales Sciences et Technologies (CED-ST), Equipe de Recherche : Modélisation, Optimisation des Système Industriels et Logistiques (MOSIL), ENSA de Tétouan, Maroc

Optimisation et simulation des opérations de cross-docking dans le contexte de l'Internet Physique

JURY

Rapporteurs

M. El AFIA Abdellatif	Professeur, ENSIAS, Université Mohammed V, Rabat, Maroc (Président du jury)
M. YALAOUI Farouk	Professeur, Université de Technologie de Troyes, France

Examinateurs

Mme. LADIER Anne-Laure	Maître de conférences, INSA de Lyon, France
M. BENMANSOUR Rachid	Maître de conférences, INSEA de Rabat, Maroc
M. EL KADIRI Kamal Eddine	Professeur, Université Abdelmalek Essaâdi, Tétouan, Maroc

Co-directeurs de thèse

M. REGHIOUI Mohamed	Maître de conférences, HDR, ENCG, Université Abdelmalek Essaâdi, Tétouan, Maroc
M. TRENTESAUX Damien	Professeur, Université Polytechnique Hauts-de-France, France

Co-encadrant

M. BEKRAR Abdelghani	Maître de conférences, Université Polytechnique Hauts-de-France, France

Invité

M. EL FALLAHI Abdellah	Maître de conférences, HDR, ENSATE, Université Abdelmalek Essaâdi, Tétouan, Maroc

CC BY NC

Avant-propos

Le travail présenté dans ce document a été réalisé dans le cadre d'une cotutelle de thèse entre l'Université Abdelmalek Essaâdi au Maroc et l'Université Polytechnique Hauts-de-France.

La thèse a été menée au sein de l'équipe de recherche MOSIL (Modélisation, Optimisation des Système Industriels et Logistiques) du Centre d'Etudes Doctorales Sciences et Technologies (CED-ST) de la faculté des sciences de Tétouan au Maroc et au sein du laboratoire LAMIH (Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines, UMR CNRS 8201).

Le travail mené a bénéficié du soutien du Centre National pour la Recherche Scientifique et Technique du Maroc (CNRST) et du Centre National de Recherche Scientifique (CNRS), France.

Il a enfin été partiellement mené dans le cadre du projet ELSAT2020 financé par la région Hauts-de-France.

Remerciements

Tout d'abord, je tiens à remercier toutes celles et ceux qui m'ont aidé à réaliser ce travail de thèse. Je remercie énormément en premier lieu mes parents, mes sœurs, mon frère et toute ma famille pour leur motivation, leur soutien moral et leur patience.

Je tiens à remercie particulièrement mes co-directeurs de thèse M. Mohamed Reghioui de l'Université Abdelmalek Essaâdi ainsi que M. Damien Trentesaux et mon co-encadrant M. Abdelghani Bekrar de l'Université Polytechnique Hauts-de-France pour leur encadrement quasi journalier, leur encouragement, leur disponibilité et surtout pour leur précieux conseils tout au long de la préparation de cette thèse.

Je tiens également à remercier M. Abdellatif El Afia, professeur à l'Université Mohamed V de Rabat, M. Abdellah El Fallahi, professeur à l'Université Abdelmalek Essaâdi et M. Farouk Yalaoui, professeur à l'Université de technologie de Troyes pour avoir accepté d'être rapporteurs de cette thèse. Je tiens également à remercier Mme. Anne-Laure Ladier, maître de conférences à l'INSA de Lyon, M. Rachid Benmansour, maître de conférences à l'INSEA de Rabat et M. Kamal Eddine El kadiri, professeur à l'Université Abdelmalek Essaâdi de Tétouan d'avoir examiné mon travail de thèse.

Enfin, je remercie mes amis et collègues de travail pour leur aide scientifique, professionnelle et personnelle durant cette thèse.

Résumé

Dans cette thèse, nous étudions les problèmes d'optimisation des opérations de cross-docking dans le contexte de l'Internet Physique (PI). Ce nouveau concept a été introduit pour améliorer la flexibilité, la synchronisation et la durabilité des systèmes logistiques. Nous nous intéressons particulièrement à la planification des opérations dans plusieurs types de cross-docks (les PI-hubs), à savoir le Road-Road, le Road-Rail et le Rail-Road PI-hub. Nous commençons par modéliser mathématiquement le problème de planification des véhicules dans un cross-dock Road-Road classique et puis nous développons deux méta-heuristiques stables et performantes par rapport aux solutions obtenues par le solveur CPLEX. Cette première étude a été réalisée afin d'entamer par la suite le passage vers le Road-Road PI-hub à travers une simulation à événements discrets pour évaluer la robustesse des deux types de cross-docks et en proposant également un couplage simulation-optimisation qui permet d'obtenir des solutions robustes efficaces même en cas des perturbations. Ensuite, nous proposons un modèle multi-agent MAS pour la planification des opérations dans un Road-Rail PI-hub. La performance et la stabilité du MAS ont été validées par un modèle mathématique MILP implémenté sur CPLEX et ensuite sur un benchmark d'instances de la littérature. La réactivité du MAS a été également validée dans un environnement dynamique sous perturbations. Enfin, nous étudions la planification durable dans un Rail-Road PI-hub. Nous proposons un modèle mathématique de programmation lexicographique multi-objectifs MO-MIP résolu sur CPLEX. Puis, deux méta-heuristiques hybrides sont proposée pour minimiser à la fois la consommation énergétique des PI-convoyeurs ainsi que le coût d'utilisation des véhicules. Les deux méta-heuristiques génèrent des solutions optimales et quasi-optimales dans des temps de calculs très réduits.

Mots-clés : Logistique, Internet Physique, Cross-dock, Recherche Opérationnelle, Programmation linéaire, Planification des véhicules, Méta-heuristiques, Simulation, Systèmes multi-agents, Perturbations, Robustesse, Durabilité.

Abstract

This doctoral thesis addresses the cross-docking optimization problems in the context of the Physical Internet (PI). This new paradigm was introduced to improve the flexibility, synchronization and sustainability of the supply chain and logistics systems. We focus on the operations scheduling in several types of cross-docks (called PI-hubs in the context of the PI), namely the Road-Road, Road-Rail and the Rail-Road PI-hub. After analyzing the state of the art of the Physical Internet optimization problems in the global supply chain, especially for classical cross-docks and PI-hubs, we address the problem of truck scheduling in a classical Road-Road cross-dock through different approaches (MILP, meta-heuristics, etc.) in order to evaluate its performances compared to the Road-Road PI-hub through a discrete event simulation study to evaluate the robustness of the two cross-docks and also by proposing a simulation-optimization approach. Next, we propose a multi-agent model MAS for scheduling operations in a Road-Rail PI-hub. The performance and the stability of the MAS are validated by a MILP model and then on a benchmark instances from the literature. The MAS is also evaluated in a dynamic environment under perturbations. The results obtained showed the ability of the MAS to react to perturbations. Finally, we study the Rail-Road PI-hub sustainable scheduling. We propose a multi-objective mathematical model MO-MIP using lexicographic programming to minimize the energy consumption of PI-conveyors as well as the cost of using trucks. Then, two hybrid and multi-objective meta-heuristics are proposed, namely MO-VNSSA and MO-VNSTS.

Keywords : Logistics, Physical Internet, Cross-Dock, Operational Research, Linear Programming, Truck Scheduling, Meta-heuristics, Simulation, Multi-agent systems, Perturbations, Robustness, Sustainability.

Table des matières

Avant-propos **2**

Remerciements **3**

Introduction générale **19**

1 Du cross-docking vers l'Internet Physique : Etat de l'art **23**

1.1 Introduction 24

1.2 Les cross-docks classiques 24

1.2.1 Classification des problèmes du cross-dock classique 26

1.2.2 Les différentes formes des cross-docks 27

1.2.3 Les cross-docks à un quai d'entrée et un quai de sortie 27

1.2.4 Les cross-docks à quais multiples 29

1.2.5 La durabilité dans les cross-docks classiques 33

1.3 L'Internet Physique : concept et définitions 34

1.3.1 Présentation 34

1.3.2 Définitions et notions de base 35

1.3.2.1 Les PI-containers 35

1.3.2.2 Les PI-movers 37

1.3.2.3 Les PI-nodes 37

1.4 L'Internet Physique : vers une chaîne logistique interconnectée, durable et résiliente 38

1.4.1 L'interconnectivité dans la chaîne logistique PI 38

1.4.2 La durabilité dans la chaîne logistique 39

1.4.3 La résilience de la chaîne logistique 40
1.5 Les PI-cross-docks 42
1.5.1 Les Rail-Road PI-hubs 42
1.5.2 Les Road-Road PI-hubs 44
1.6 Synthèse 46
1.7 Conclusion 47

2 Planification des véhicules dans un cross-dock Road-Road classique à quais multiples 48

2.1 Introduction 49
2.2 Présentation de la problématique 49
2.3 Modèle mathématique 52
2.3.1 Paramètres 53
2.3.2 Variables de décision 54
2.3.2.1 Variables continues 54
2.3.2.2 Variables entières 55
2.3.2.3 Variables binaires 55
2.3.3 Fonction objective 56
2.3.4 Contraintes 56
2.3.5 Contraintes de flux 59
2.3.5.1 Contraintes d'affectation et de séquencement 59
2.3.6 Contraintes de planification 60
2.4 Proposition de deux méta-heuristiques 61
2.4.1 Heuristique 61
2.4.1.1 Recherche locale par insertion aléatoire (LS) 63

2.4.2 Tabu Search (TS) 64
2.4.3 Simulated Annealing (SA) 65
2.5 Résultats expérimentaux 68
2.5.1 Instances 68
2.5.2 Analyse des résultats expérimentaux 70
2.5.2.1 Heuristique et CPLEX 70
2.5.2.2 Résultats de la recherche locale (LS) et des deux méta-heuristiques (TS et SA) 71
2.5.2.3 Comparaison entre les méta-heuristiques proposées . . . 72
2.6 Comparaison avec des instances benchmark de la littérature 73
2.6.1 Analyse statistique 74
2.7 Conclusion 75

3 Du cross-dock classique au PI-hub : Etude de robustesse 77

3.1 Introduction 78
3.2 Etude de la robustesse du cross-dock classique et du PI-hub 79
3.2.1 Vue d'ensemble sur le cross-dock classique 79
3.2.2 Vue d'ensemble sur le PI-hub 81
3.2.3 Présentation des scénarios de simulation 81
3.2.4 Analyse des résultats de la simulation 82
3.2.4.1 Le temps d'attente des véhicules 83
3.2.4.2 Le nombre de véhicules en attente 84
3.2.4.3 Le temps de cycle des produits 85
3.2.5 Le taux d'utilisation des ressources 86
3.3 Couplage simulation-optimisation : Etude de robustesse 87

3.3.1 Indicateur de performance 88
3.3.2 Approches de résolution : Simulateur intégré dans la méta-heuristique (TS) 88
3.3.3 Mécanisme d'évaluation de robustesse 88
3.3.4 Résultats expérimentaux 90
3.4 Conclusion 93

4 Planification durable et réactive des opérations dans un cross-dock Road-Rail PI-hub 94

4.1 Introduction 95
4.2 Présentation du problème 95
4.3 Modèles proposés 97
4.3.1 Modèle mathématique : MILP 98
4.3.1.1 Données et paramètres 98
4.3.1.2 Variables de décision 99
4.3.2 Modèle multi-agents : MAS 103
4.4 Résultats expérimentaux 111
4.4.1 Implémentation et instances 111
4.4.2 Résultats du MILP et du MAS 113
4.5 Evaluation de la robustesse du MAS 117
4.6 Conclusion 122

5 Planification multi-objective et durable des opérations dans un cross-dock Rail-Road PI-hub 123

5.1 Introduction 124
5.2 Présentation du problème 124

5.3 Modèle mathématique 126
5.3.1 Données et paramètres du problème 126
5.3.2 Variables de décision 127
5.3.2.1 Variables binaires 127
5.3.2.2 Variables entières 128
5.3.2.3 Variables continues 128
5.3.3 Fonction objective 128
5.3.4 Contraintes 129
5.4 Méthodes de résolution proposées 131
5.4.1 Heuristique de construction 132
5.4.2 Opérateurs de recherche locale 133
5.4.3 MO-VNSSA 134
5.4.4 MO-VNSTS 135
5.5 Résultats expérimentaux 136
5.5.1 Implémentation et instances 137
5.5.2 Résultats 137
5.5.3 Analyse statistique 140
5.5.3.1 Analyse de variance : ANOVA 140
5.5.3.2 Convergence des méta-heuristiques 140
5.5.4 Synthèse 141
5.6 Conclusion 141

Conclusion générale et perspectives **143**

References **156**

Annexe A **157**

Annexe B **160**

Annexe C **164**

Table des figures

1.1 Processus d'un cross-dock classique de type Road-Road 25

1.2 Les éléments clés de l'Internet Physique 36

1.3 Le Rail-Road PI-hub 43

1.4 Le Road-Road PI-hub Montreuil et al. (2013c) 45

1.5 Approches de résolution des problèmes du PI-cross-dock dans la littérature 45

2.1 Processus et flux des produits dans le cross-dock étudié 50

2.2 Paramètres et variables de décision principales du modèle proposé (MILP) 53

2.3 Exemple de retard des véhicules entrants i et $i+1$ sur le même quai . . . 54

2.4 Processus de résolution 61

2.5 Exemple d'un mouvement de recherche locale 63

2.6 Déviations entre SA et TS1(VB) pour les 810 instances (groupées par 10 instances) 75

3.1 Vue d'ensemble sur le cross-dock classique 79

3.2 Vue d'ensemble sur le PI-hub (PI-cross-dock) 81

3.3 Déviation Moyenne des temps d'attente des véhicules aux quais 84

3.4 Déviation moyenne du nombre de véhicules en attente dans les quais . . . 85

3.5 Déviation du temps de cycle des produits 86

3.6 Déviation des taux d'utilisation des ressources 87

3.7 Les différentes étapes du processus d'évaluation de la robustesse des solutions TS 88

3.8 Mécanisme du couplage simulation-optimisation pour la robustesse de la TS 89

3.9 Déviations Moyennes pour les différentes valeurs des probabilités de pannes 92

4.1 Le Road-Rail PI-hub (section 2) 96

4.2 Les principales données et variables des deux modèles proposés (MAS et MILP) 97

4.3 Les wagons utilisés doivent être consécutifs 102

4.4 Architecture du modèle MAS proposé 104

4.5 Diagramme de séquence du modèle MAS proposé 105

4.6 Fonction objective (wagons, distances et retards) dans le MILP et le MAS 114

4.7 Impact du nombre de destinations sur les wagons utilisés et sur les distances pour le MILP et le MAS 115

4.8 Déviation entre le MAS et le TS1 de Van Belle et al. (2013) 116

4.9 Processus de replanification dans le MAS sous perturbations 117

4.10 Un exemple illustratif pour la replanification des véhicules après une panne de 2 heures au niveau du quai 17 118

4.11 Fonctions objectives (wagons, distance et retard) dans le MAS avec et sans perturbations 120

4.12 Fonctions objectives dans les 5 réplications de chaque instance 121

4.13 Les p-values de l'ANOVA pour chacune des instances 121

5.1 Le Rail-Road PI-hub 125

5.2 Données et variable du modèle MILP proposé 126

5.3 Opérateurs de recherche locale dans MO-VNSSA et MO-VNSTS : Insertion, Permutation et Insertion+Permutation 133

5.4 Fonctions objectives pour CPLEX MO-VNSSA et MO-VNSTS pour les deux objectifs F_1 et F_2 138

5.5 F1 et F2 pour MO-VNSSA et MO-VNSTS dans les grandes instances . . 139

5.6 Convergence de MO-VNSSA et MO-VNSTS pour le deuxième objectif F_2 140

B.1 Vue d'ensemble du cross-dock étudié 161

B.2 Retard des véhicules entrant/sortants pour les 10 réplications 162

B.3 Planification des véhicules entrants/sortants par les deux approches . . . 163

Liste des tableaux

1.1 Les travaux réalisés sur la planification des véhicules dans un cross-dock à quais multiples 32

1.2 Travaux sur les PI-cross-docks et les approches de résolution utilisées . . 46

2.1 Paramètres des instances générées pour tester les algorithmes proposés . 69

2.2 Fonctions objectives et temps de calcul de LS, TS, SA et CPLEX 71

2.3 Comparaison de LS, TS at SA avec les résultats de CPLEX 72

2.4 Comparaison des intervalles de confiance 75

3.1 Paramètres de la simulation 80

3.2 Distribution des temps entre chaque deux arrivées 82

3.3 Distribution des temps entre chaque deux arrivées 83

3.4 Nombre de véhicules en attente dans les quais de réception/expédition . . 84

3.5 Le temps de cycle des produits 85

3.6 Taux d'utilisation des ressources dans les deux cross-docks 86

3.7 Le retard des véhicules entrants/sortants sans perturbations 91

3.8 Déviations moyennes pour chaque probabilité de perturbation (10 réplications) 91

4.1 Variation des paramètres dans les instances générées 112

4.2 Comparaison entre le MILP et le MAS 113

4.3 Résultats du MAS pour les grandes instances 115

4.4 Comparaison entre la planification au début de la journée (S_{BEST}) et la replanification à la fin de la journée ($S_{RESCHEDULED}$) dans le modèle MAS 119

5.1 Valeurs des paramètres et des données d'entrée pour les petites instances 137

5.2 Valeurs des paramètres dans les méta-heuristiques : MO-VNSSA et MO-VNSTS pour les petites instances 137

5.3 Résultats expérimentaux pour CPLEX, MO-VNSSA et MO-VNSTS sur les instances générées 139

5.4 Résultats expérimentaux pour MO-VNSSA et MO-VNSTS pour les grandes instances 139

5.5 Résultats de l'ANOVA pour MO-VNSSA et MO-VNSTS 140

A.1 Comparaison entre le SA et le TS1 (VB) de Van Belle et al. (2013) ($w_1 = 1$, $w_2 = 2$) 157

A.2 Comparaison entre le SA et le TS1 (VB) de Van Belle et al. (2013) ($w_1 = 1$, $w_2 = 5$) 158

A.3 Comparaison entre le SA et le TS1 (VB) de Van Belle et al. (2013) ($w_1 = 1$, $w_2 = 10$) 159

B.1 Exemple d'une journée (24h) dans le cross-dock étudié 160

B.2 Retard des véhicules entrants/sortants pour 10 réplications 161

B.3 Arrivée et départ préférés des véhicules entrant/sortants 162

B.4 Début et fin de déchargement/chargement des véhicules entrants/sortants 163

C.1 Résultats expérimentaux des trois méta-heuristiques du MAS sur les instances benchmark 164

Liste des Algorithmes

2.1 Algorithme de l'heuristique proposée 62

2.2 Algorithme de la recherche locale par insertion aléatoire (LS) proposé . . . 64

2.3 Algorithme du tabu search (TS) proposé 66

2.4 Algorithme du Simulated Annealing (SA) proposé 67

4.1 Heuristique de construction de la solution initiale de l'agent synchroniseur 107

4.2 Simulated Annealing pour la planification des véhicules 109

4.3 Algorithme du VNS-SA proposé 109

4.4 Algorithme du GRASP-SA proposé 110

4.5 Algorithme du TS-SA proposé 111

5.1 Heuristique de construction (H_0) de la solution initiale 132

5.2 Algorithme de la méta-heuristique MO-VNSSA 134

5.3 Algorithme de la méta-heuristique MO-VNSTS 136

Liste des acronymes

ANOVA	Analysis Of Variance
DSPEA	Delayed Start Parallel Evolutionary Algorithm
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
HGATS	Hybrid Genetic Algorithm with Tabu Search
LS	Local Search
MAS	Multi-Agent System
MILP	Mixed Integer Linear Programming model
MO-MIP	Multi-Objective Mixed Integer Programming model
MO-PSO	Multi-Objective Particle Swarm Optimization
MO-VNSSA	Multi-Objective Variable Neighborhood Search-Simulated Annealing
MO-VNSTS	Multi-Objective Variable Neighborhood Search-Tabu Search
ORCA	Optimized and Reactive Control Architecture
PI	Physical Internet
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SA-PSO	Simulated Annealing-Particle Swarm Optimization
TL	Tabu List
TS	Tabu Search
VNS	Variable Neighborhood Search

Introduction générale

Dans un contexte industriel et économique concurrentiel, l'optimisation et la gestion de la chaîne logistique ou le Supply Chain Management (SCM) sont devenues des facteurs incontournables pour les entreprises de production et de distribution afin de maîtriser et contrôler efficacement l'ensemble des opérations logistiques. La gestion de la chaîne logistique consiste à organiser les flux physiques, financiers et d'informations entre les fournisseurs et les clients en passant par trois phases principales : approvisionnement, production et distribution. Cette dernière phase nécessite une synchronisation efficace entre les différents points de distribution : plateformes de cross-docking, centres et magasins de distribution jusqu'aux clients finaux.

Cette thèse s'inscrit dans le cadre de l'optimisation de la planification des opérations de cross-docking afin d'évaluer les apports et les limites du concept de l'Internet Physique (PI ou π) sur la performance des cross-docks et leur robustesse face aux différentes perturbations. Le PI a été introduit comme un réseau logistique universel ouvert et utilisable par l'ensemble des acteurs dans la chaîne logistique globale. L'objective de cette thèse est d'évaluer l'apport du PI sur la performance de la chaîne logistique et plus spécifiquement sur les plateformes du cross-docking. De manière générale, le processus d'un cross-dock classique commence par le déchargement des véhicules entrants (camions, trains, etc.). Les produits déchargés peuvent être ensuite stockés temporairement, ou bien transférés directement s'ils sont demandés par les véhicules sortants. L'idée consiste à consolider et synchroniser les flux amont et aval tout en garantissant un minimum de stock temporaire. Cela nécessite une bonne planification du chargement et de déchargement des véhicules aux quais (docks), un routage interne efficace dans le cross-dock pour les conteneurs en tenant compte de plusieurs contraintes liées à la capacité des véhicules ainsi que leurs heures d'arrivée et de départs souhaités, les types des produits et leurs destinations, etc.

La contribution de cette thèse est d'étudier la planification des opérations dans trois types de cross-docks : Road-Road, Road-Rail et Rail-Road (Route-Route, Route-Rail et Rail-Route) dans le contexte de l'Internet Physique. Nous proposons différentes approches de résolution (Modèles mathématiques, simulations, systèmes multi-agents, couplages simulation-optimisation) appliquées sur les trois variantes de cross-docks. Les deux

chapitres suivants abordent le passage des cross-docks classiques vers les PI-hubs. En effet, avant de passer à la configuration PI du cross-dock (le PI-hub), il est nécessaire d'étudier d'abord la planification des opérations dans un cross-dock classique en proposant une modélisation mathématique MILP du problème et des méta-heuristiques : la recherche tabou et le recuit simulé (Tabu Search et Simulated Annealing) qui seront évaluées par rapport à la méthode intuitive utilisée par les opérateurs du cross-dock. Puis, ce cross-dock sera évalué par rapport au PI-cross-dock (nommé aussi : PI-hub, la version PI du cross-dock) via une étude par simulation. Dans les deux derniers chapitres, nous nous concentrons sur les cross-docks de type Road-Rail et les Rail-Road PI-hubs en proposant d'abord, en plus d'un modèle mathématique MILP, un modèle multi-agent pour le Road-Rail PI-hub qui permet de réagir en cas de perturbations pour générer des planifications alternatives. Nous proposons aussi des approches multi-objectives lexicographiques pour le Rail-Road PI-hub, à savoir un modèle mathématique multi-objectifs MO-MIP, et deux méta-heuristiques hybrides MO-VNSSA et MO-VNSTS.

Ce manuscrit est constitué de cinq chapitres :

Le premier chapitre est consacré à la présentation de l'état de l'art sur les cross-docks classiques et le passage vers l'Internet Physique (les PI-hubs). Il est divisé en 4 grandes parties. La première est consacrée à la présentation de l'état de l'art des problèmes de planification des cross-docks classiques. Ensuite, la deuxième partie présente le concept de l'Internet Physique et la définition des notions de bases nécessaires à la compréhension de cette thèse. La troisième partie est consacrée à l'interconnectivité ainsi que les notions de durabilité et résilience dans la chaîne logistique. La dernière partie présente la littérature sur les PI-cross-docks dans le contexte du PI.

Le deuxième chapitre présente le problème de planification des véhicules et le transfert et stockage des produits dans un cross-dock de type Road-Road. Tout d'abord, un modèle mathématique linéaire mixte (MILP) est proposé pour la planification des véhicules dans un Road-Road cross-dock possédant plusieurs zones de stockage temporaires et gérant différents types de produits. L'objectif est de trouver une planification de début/fin de déchargement/chargement des véhicules ainsi que les produits à stocker temporairement et ceux à transférer directement vers les véhicules sortants tout en minimisant le retard des véhicules dans les quais pour qu'ils partent le plus tôt possible et libérer les quais pour

les prochains véhicules. Le modèle est résolu et validé sur le solveur IBM CPLEX. Puisque ce dernier ne trouve les solutions optimales que pour les petites instances, une heuristique (H) et trois méta-heuristiques : Random Insertion Local Search (LS), Simulated Annealing (SA) et Tabu Search (TS) sont proposées pour résoudre ce problème de planification. L'heuristique (H) et la Simulated Annealing (SA) sont ensuite adaptées pour résoudre des instances de la littérature. Ce chapitre se termine par une étude de simulation pour évaluer l'apport des algorithmes proposés par rapport à la méthode intuitive utilisée par les opérateurs du cross-dock.

Dans le troisième chapitre nous étudions l'apport de l'Internet Physique sur la performance du cross-dock classique (Road-Road) et le PI-hub en considérant différents scénarios pour tester leur robustesse face à des différentes distributions de probabilité des arrivées des véhicules entrants et sortants. Ensuite, nous proposons une approche de couplage simulation-optimisation qui permet de générer des solutions robustes. En effet, l'optimisation est faite par la méta-heuristique Tabu Search (TS) dont la qualité des solutions est comparée au modèle mathématique MILP présenté dans le deuxième chapitre. Par la suite, les solutions de la TS sont simulées en considérant des perturbations dans les moyens de transport internes (Chariots élévateurs pour le cross-dock classique et les convoyeurs dans le cas du PI-hub).

Le quatrième chapitre vise à étudier le problème de planification des véhicules et le groupage des PI-conteneurs dans la section Road-Rail du Road-Rail cross-dock qui est conçue pour le transfert des PI-conteneurs depuis les véhicules vers les wagons du train. Un modèle multi-agent (MAS) est proposé pour gérer les interactions entre les différentes composantes (agents) du Road-Rail cross-dock : trains, camions, etc. L'utilisation d'un MAS permet en particulier de garantir une réactivité face aux perturbations inattendues (retard des véhicules, indisponibilités des quais, etc). Pour obtenir des solutions de qualité (planifications des véhicules et groupage des conteneurs), trois méta-heuristiques hybrides et parallèles sont intégrées dans trois agents de planification du modèle MAS : Variable Neighbourhood Search (VNS), Greedy Randomized Adaptive Search Procedure (GRASP) et Tabu Search (TS) hybridées avec Simulated Annealing (SA) : VNS-SA, GRASP-SA et TS-SA. Afin d'évaluer la qualité des solutions obtenues, un modèle mathématique linéaire mixte (MILP) est proposé. Le MILP est validé et résolu avec le solveur IBM CPLEX. La réactivité du modèle MAS est évaluée ensuite dans un environnement

avec des perturbations. Les expérimentations numériques obtenues permettent de montrer la capacité du modèle MAS proposé à réagir face aux perturbations inattendues en proposant des solutions alternatives. Les résultats obtenus sont validés par une analyse de variance (ANOVA).

L'objectif du cinquième chapitre est d'étudier la durabilité dans le cas du cross-dock Rail-Road à travers des approches multi-objectives. Pour résoudre ce problème, nous proposons plusieurs approches multi-objectives lexicographiques. Tout d'abord, nous proposons un modèle mathématique multi-objectifs MO-MIP pour minimiser deux objectifs contradictoires, à savoir la consommation énergétique des PI-conveyors et le coût d'utilisations des véhicules d'expédition pour chaque destination. Le modèle proposé est résolu sur CPLEX en utilisant la programmation lexicographique. Ensuite, deux méta-heuristiques hybrides et multi-objectives sont développées, à savoir MO-VNSSA et MO-VNSTS. La stabilité et la convergence de ces méthodes sont ensuite analysées.

Ce manuscrit se termine par une conclusion générale sur les travaux réalisés durant cette thèse et la proposition de nouvelles perspectives de recherche.

Chapitre 1

Du cross-docking vers l'Internet Physique : Etat de l'art

Ce chapitre présente le concept de l'Internet Physique (PI) et des définitions nécessaires à la compréhension de cette thèse, puis, l'état de l'art des problèmes liés au PI au niveau de la chaîne logistique globale ainsi qu'au niveau des cross-docks classiques et des PI-cross-docks (PI-hubs).

1.1 Introduction

De nos jours, la synchronisation et l'interconnectivité de la chaîne logistique deviennent un défi majeur pour les entreprises spécialisées en logistique et distribution afin d'améliorer leur efficacité et efficience. L'objectif est de réduire le coût économique global et d'augmenter la productivité en tenant compte des aspects sociaux et environnementaux. Ce chapitre présente l'état de l'art de l'Internet Physique (PI ou π) et plus spécifiquement les problèmes d'optimisation de la chaîne logistique globale et les problèmes de planification interne des cross-docks. Ce nouveau paradigme (PI) a été introduit par Montreuil et al. (2013a). L'idée de l'Internet Physique consiste à créer un réseau logistique global et synchronisé qui peut être utilisé par tous les intervenants de la chaîne logistique globale (fournisseurs, usines de production, distributeurs et clients finaux).

Ce chapitre se compose de cinq sections principales. La section 1.2 est consacrée aux travaux sur les cross-docks classiques ainsi que les différentes classifications dans la littérature des problèmes du cross-docking. Ensuite, les différentes formes et variantes des cross-docks sont détaillées. Cette première partie se termine par un état de l'art détaillé sur les problèmes de planification des véhicules dans les cross-docks à un quai entrant et un quai sortant et les cross-docks à quais multiples. La section 1.3 est dédiée à la présentation du concept et des définitions des éléments clés de l'Internet Physique. Puis, la section 1.4 présente l'état de l'art de l'Internet Physique dans la chaîne logistique globale ainsi que les notions de durabilité et résilience. La section 1.5, les cross-docks dans le contexte de l'Internet Physique sont abordés en commençant par les Rail-Road PI-hubs et puis les Road-Road PI-hubs. Ce chapitre se termine par une synthèse sur les travaux liés aux cross-docks et aux PI-hubs pour présenter brièvement les contributions de cette thèse et introduire les chapitres suivants.

1.2 Les cross-docks classiques

Dans cette section, les travaux sur les cross-docks classiques sont présentés ainsi que les différentes méthodes de résolution utilisées pour résoudre les problèmes de planification

des opérations de cross-docking. Les plateformes logistiques ou cross-docks sont considérés comme un point de consolidation dans la chaîne logistique. L'objectif d'une plateforme cross-dock est de faire un routage des produits pour les transférer des véhicules entrants vers les véhicules sortants en passant par un stockage temporaire si nécessaire. La première étape dans un processus de cross-docking est le déchargement des véhicules entrants (par exemple : camions, trains, etc.) provenant de différents fournisseurs. Dans la deuxième étape, les produits déchargés sont ensuite transférés soit vers les rayons des zones de stockage temporaires ou bien triés et ensuite transférés directement vers les véhicules sortants s'ils sont disponibles. Le processus général du cross-docking est illustré dans la figure 1.1 pour un cross-dock de type Road-Road (Transfert entre véhicules).

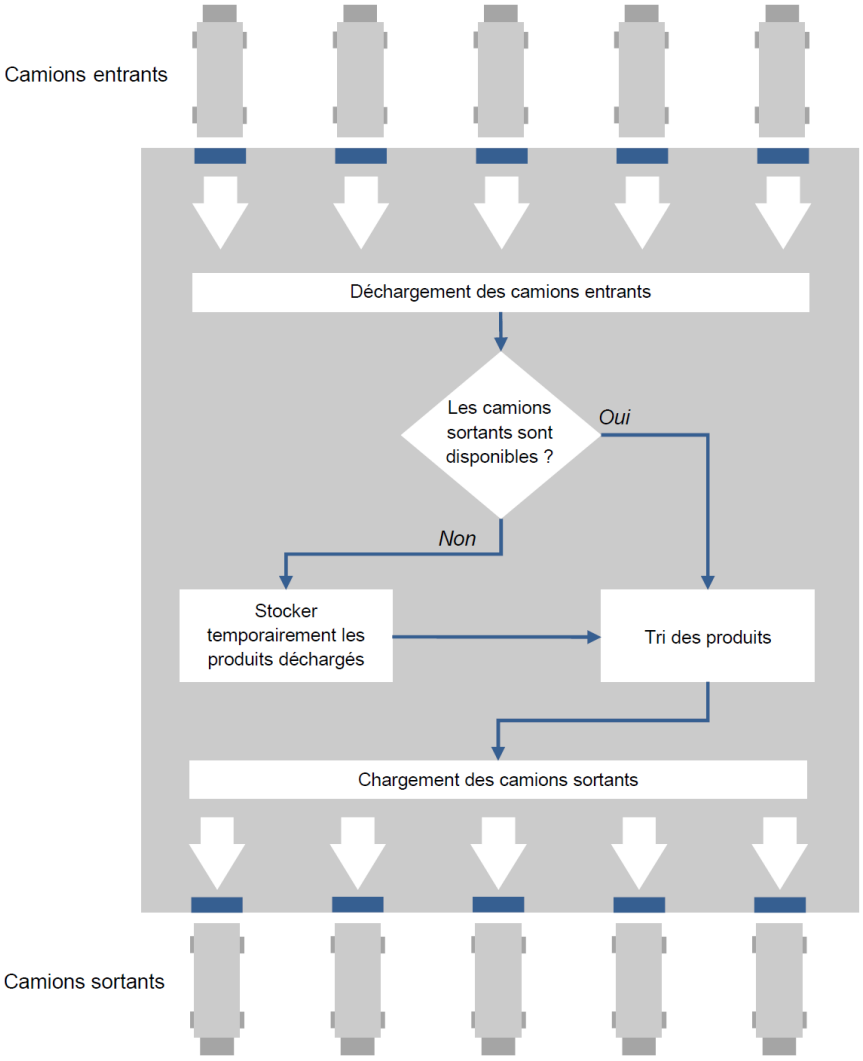

Figure 1.1 – Processus d'un cross-dock classique de type Road-Road

1.2.1 Classification des problèmes du cross-dock classique

Plusieurs revues de littératures ont été réalisées sur les classifications des travaux sur le cross-docking. Par exemple, Boysen and Fliedner (2010) ont présenté une classification des problèmes de planification des véhicules dans les quais. Leur travail traite la répartition des ressources dans le temps pour le transbordement des produits ainsi que pour le déchargement / chargement des véhicules dans les quais d'arrivée et de départ. Ces problèmes ont été classés en fonction de trois éléments : le premier concerne les quais d'arrivée et de départ, représenté par le mode de service (exclusif, mixte et combiné) et le nombre de quais. Le second est lié aux caractéristiques opérationnelles telles que la préemption, l'heure d'arrivée, les délais et le stockage intermédiaire. Enfin, les fonctions objectives telles que le temps de traitement, le temps de séjour des produits et les retards. Un autre état de l'art a été présenté par Van Belle et al. (2012), qui aborde divers problèmes de cross-docking classés sur plusieurs niveaux : stratégique, tactique et opérationnel, y compris les problèmes étudiés dans l'état de l'art sur les modèles mathématiques de Agustina et al. (2010). Stephan and Boysen (2011) ont examiné la littérature et ont présenté une classification détaillée des problèmes du cross-docking.

Walha et al. (2014b) a présenté un état de l'art pour le cross-docking sous incertitude. La classification des recherches abordées dans leur travail est basée sur le type d'incertitude (incertitude interne ou externe) ainsi que sur les caractéristiques physiques (forme, nombre de quais, etc.), opérationnelles (préemption, mode de service, etc.) et de configuration de flux (arrivée, départ, interchangeabilité des produits, etc.). Les auteurs ont abordé également le problème du cross-docking dans le contexte de l'Internet Physique (PI-hub) en présentant la conception et l'installation du PI-hub et en détaillant les différents processus internes d'un PI-hub.

Une revue de littérature récente a été effectuée par Ladier and Alpan (2016a) en analysant la littérature sur le problème de la planification et de l'affectation des véhicules dans les cross-docks, en comparant les différents travaux dans la littérature à l'aide d'une classification des problèmes couvrant les trois niveaux stratégique, tactique et opérationnel. Les auteurs ont divisé le niveau opérationnel en cinq catégories : premièrement, les problèmes d'affectation des véhicules aux quais de réception ou d'expédition. La deuxième catégorie est liée aux problèmes de séquencement des véhicules qui consistent à trouver

l'ordre de passage des véhicules au niveau des quais. La troisième catégorie concerne les problèmes de planification des véhicules aux quais en les attribuant sur un horizon de planification. La quatrième catégorie est liée aux problèmes de séquencement des véhicules. Et enfin, les problèmes de planification des véhicules qui sont utilisés dans les cas de quai à quai. Ces deux dernières catégories ne considèrent pas la dimension spatiale qui détermine à quel quai le véhicule est attribué.

Enfin, afin d'évaluer la solution, qui consiste à attribuer les quais aux véhicules, les problèmes de planification des véhicules utilisent généralement une fonction objective liée au temps (Boysen and Fliedner, 2010).

1.2.2 Les différentes formes des cross-docks

Avant la mise en place d'une plateforme de cross-docking, il est important de prendre en compte les décisions stratégiques telles que la forme et la conception du cross-dock. Plusieurs auteurs ont étudié les formes possibles d'un cross-dock, par exemple, dans Kapetanios et al. (2009) les auteurs ont comparé les performances des cross-docks en fonction du nombre de quais de réception et d'expédition. Bartholdi and Gue (2004) ont évalué les différentes formes des cross-docks (I, L, T, U, H, E, X, etc.) et ont proposé un dimensionnement du cross-dock en fonction du nombre de quais. Par exemple, afin de maximiser les performances dans un cross-dock en forme de « I », le nombre approprié de quais doit être inférieur à 150. Bartholdi and Gue (2004) ont évalué également la performance du cross-dock en utilisant la durée totale du trajet entre les quais.

1.2.3 Les cross-docks à un quai d'entrée et un quai de sortie

Le problème de planification des véhicules dans un cross-dock avec un quai de réception et un quai d'expédition a été abordé par de nombreux auteurs dans la littérature. Yu and Egbelu (2008) ont suggéré un modèle mathématique linéaire mixte (MILP) dont l'objectif est la planification et le séquencement des véhicules entrants et sortants tout en minimisant le makespan. Les auteurs ont également développé une heuristique pour minimiser le nombre de produits dans la zone de stockage située devant le quai d'expédition. Dans leur modèle MILP, les produits sont interchangeables et tous les véhicules entrants

et sortants sont considérés disponibles et prêts à l'instant $t = 0$. De plus, Keshtzari et al. (2016) ont suggéré un nouveau modèle mathématique MILP et ont comparé ses performances à celles du modèle MILP de Yu and Egbelu (2008) pour résoudre les petites instances. Les auteurs ont également proposé une optimisation par essaim particulaire (PSO) hybridée avec le recuit simulé (SA) pour les grandes instances.

Plusieurs travaux ont porté aussi sur les méthodes de résolution des problèmes de planification des véhicules dans les cross-docks, par exemple, Vahdani and Zandieh (2010) ont travaillé sur le même modèle MILP de Yu and Egbelu (2008) et ont présenté cinq méta-heuristiques en les comparant à celle proposée par Yu and Egbelu (2008). Un algorithme génétique de recherche tabou hybride (HGATS) a été proposé par Shiguemoto et al. (2014) pour la planification des véhicules dans les quais afin de réduire le makespan. Le HGATS proposé a été testé sur de grandes instances, ainsi que sur un groupe de petites instances de Yu and Egbelu (2008). Boloori-Arabani et al. (2011) ont suggéré cinq méta-heuristiques pour résoudre le modèle MILP présenté par Yu and Egbelu (2008) et ont comparé la sensibilité des algorithmes aux tailles du problème. Khalili-Damghani et al. (2017) ont présenté un problème d'ordonnancement multi-périodes des véhicules sur plusieurs quais et un algorithme génétique pour le résoudre. Deux algorithmes hybrides d'évolution différentielle ont été suggérés par Liao et al. (2012) pour le séquencement des véhicules entrants et sortants. De plus, Golshahi-Roudbaneh et al. (2017) ont proposé deux heuristiques permettant de résoudre le modèle MILP de Yu and Egbelu (2008) ainsi que quatre méta-heuristiques et une méta-heuristique hybride SA-PSO (recuit simulé - optimisation par essaim de particules). Les auteurs ont également comparé la convergence des heuristiques entre elles. Madani-Isfahani et al. (2014) ont développé un modèle MILP pour plusieurs cross-docks où chaque cross-dock a un quai de réception et un quai d'expédition. Les auteurs ont proposé aussi deux méta-heuristiques pour résoudre le problème.

D'autres travaux ont pris en compte la possibilité de pannes des véhicules, par exemple, Amini and Tavakkoli-Moghaddam (2016) ont présenté un problème d'ordonnancement des véhicules en prenant en compte les pannes des véhicules entrants et sortants. Le nombre de pannes suit une distribution de Poisson. Les auteurs ont proposé des méta-heuristiques multi-objectifs en comparant leurs performances.

Il est important de mentionner que d'autres auteurs ont formulé le problème du cross-

docking de manière différente, par exemple, Chen and Song (2009) ont proposé un modèle mathématique linéaire mixte (MILP) pour la planification du cross-dock en formulant le problème sous la forme d'un problème flow-shop hybride à deux étages. L'objectif étant de minimiser le makespan. Les auteurs ont également proposé quatre heuristiques pour résoudre le problème.

1.2.4 Les cross-docks à quais multiples

Le problème de la planification des véhicules dans un cross-dock à quais multiples a été abordé par plusieurs chercheurs. De nombreux auteurs ont suggéré des formulations mathématiques pour la planification des véhicules dans un cross-dock à plusieurs quais. Certains travaux ont traité le problème de la planification des véhicules tout en prenant en compte les dates de départ fixes pour les véhicules sortants, c'est-à-dire que les véhicules partants doivent quitter les quais d'expédition à une heure prédéterminée. Par exemple, Konur and Golias (2013) ont étudié l'ordonnancement des véhicules entrants, dont les temps de traitement sont différents, tout en tenant compte de l'incertitude de l'heure d'arrivée des véhicules. Les auteurs ont proposé une heuristique basée sur un algorithme génétique où l'objectif est de minimiser le coût moyen total du service. De plus, Tootkaleh et al. (2016) ont suggéré un modèle MILP et une heuristique pour la planification des véhicules afin de minimiser les coûts de conservation des stocks avec fixation de l'heure de départ des véhicules sortants. Cependant, les produits retardés sont conservés dans la zone de stockage jusqu'à ce qu'un autre véhicule sortant avec la même destination soit disponible. Boysen et al. (2013) ont proposé un modèle MILP pour la planification des véhicules entrants dans un cross-dock où les affectations et les horaires des véhicules sortants sont prédéterminés. Les auteurs ont également proposé deux heuristiques (procédure de décomposition et recuit simulé). Dans Molavi et al. (2018), les auteurs ont présenté un modèle MILP dans lequel les véhicules sortants doivent libérer les quais à une date prédéfinie. Ils ont proposé quatre méta-heuristiques pour résoudre le problème. Dans leur modèle, le stockage temporaire est autorisé sur les quais d'expédition et les produits sont immédiatement transférés aux quais d'expédition après le déchargement des véhicules entrants.

D'autres travaux ont suggéré des modèles mathématiques pour la planification des

arrivées des véhicules sur les quais, sans considérer l'heure de départ des véhicules sortants comme une contrainte difficile. Par exemple, Van Belle et al. (2013) ont présenté un modèle MILP pour un problème de planification des véhicules dans les quais de réception et d'expédition d'un cross-dock à quais multiples. Ils ont également fourni une recherche taboue pour résoudre le problème en utilisant un benchmark d'instances. Cependant, le chargement en parallèle des produits des véhicules entrants est possible. En outre, Assadi and Bagheri (2016) ont suggéré un modèle MILP pour la planification des véhicules entrants et sortants et deux méta-heuristiques pour résoudre les grandes instances. Leur modèle se concentre sur l'optimisation de l'avance et du retard des véhicules sortants. Cependant, les deux travaux, Van Belle et al. (2013) et Assadi and Bagheri (2016) considèrent que le stockage temporaire est situé devant les quais de réception. Une version modifiée des modèles MILP de Yu and Egbelu (2008) et de Van Belle et al. (2013) a été suggérée par Wisittipanich and Hengmeechai (2017) pour la planification des véhicules dans les cross-docks à quais multiples manipulant différents types de produits. Ils ont proposé une optimisation par essaim de particules modifiée pour résoudre le problème. Cependant, dans leur modèle mathématique, les heures d'arrivée et de départ ne sont pas définies par les véhicules. Comme extension du modèle MILP de Konur and Golias (2013), où les auteurs se sont concentrés sur les véhicules entrants dans un concept d'optimisation bi-objectifs et bi-niveaux, Heidari et al. (2018) ont étendu leur modèle d'ordonnancement des véhicules entrants et sortants et ont proposé deux méta-heuristiques. Pourtant, les deux modèles ne prennent pas en compte le stockage temporaire et en plus l'affectation des produits aux véhicules sortants est prédéterminée dans le modèle de Heidari et al. (2018).

Dans le cadre d'une étude de cas d'une entreprise automobile, Serrano et al. (2017) ont présenté un modèle mathématique pour la planification des arrivées des véhicules. Leur modèle considère également un atelier de reconditionnement avec des opérations de manutention telles que le regroupement, le reconditionnement et le transfert direct. Ladier and Alpan (2016b) ont proposé des modèles mathématiques robustes pour la planification des cross-docks avec des fenêtres temporelles tout en tenant compte des incertitudes possibles telles que le temps d'arrivée, le temps de déchargement et le temps de transfert. Dans une extension des travaux de Chen and Lee (2009) et de Chen and Lee (2009), Cota et al. (2016) ont formulé la planification des véhicules dans un cross-

dock sous forme d'un problème flow-shop hybride à deux étages et ont suggéré un modèle MILP indexé dans le temps avec une heuristique pour le résoudre. Dans le contexte de la chaîne logistique alimentaire, Agustina et al. (2014) ont étudié un centre de cross-docking et proposé un modèle MILP intégrant à la fois la planification et le routage des véhicules avec des fenêtres horaires pour les livraisons définies par les clients. Le modèle minimise les coûts des pénalités de l'avancement et du retard, ainsi que les coûts de conservation / transport des produits, tout en appliquant des créneaux horaires en utilisant le concept de zones des clients afin de réduire l'espace de recherche de la solution. Par la suite, le modèle a été validé sur des instances basées sur des cross-docks réels. Maknoon et al. (2016) ont formulé mathématiquement le problème du transbordement de produits dans un cross-dock afin de minimiser la charge de travail liée au transbordement en minimisant le nombre de doubles manipulations des produits. Comme le modèle ne peut résoudre que des petites instances, une heuristique basée sur la recherche par voisinage variable (VNS) a été suggérée pour résoudre les grandes instances.

Il est important de mentionner que de nombreux travaux se sont concentrés sur le développement des méta-heuristiques pour résoudre le problème de planification des véhicules dans les cross-docks. Par exemple, Dulebenets (2019) a abordé le problème de la planification des véhicules en développant un nouvel algorithme évolutif parallèle à démarrage différé DSPEA (Delayed Start Parallel Evolutionary Algorithm). L'idée est d'échanger les solutions prometteuses entre plusieurs algorithmes évolutifs distincts. Le problème a été formulé sous forme d'un modèle MILP afin de minimiser le coût total de service des véhicules. L'algorithme proposé surpasse cinq autres méta-heuristiques de la littérature sur la planification des cross-docks. Kuo (2013) a proposé une recherche par voisinage variable (VNS) et un recuit simulé (SA) pour résoudre un problème de cross-docking à quais multiples en recherchant les affectations et la séquence optimale des véhicules aux quais de réception et d'expédition. Dans Liao et al. (2014), les auteurs ont proposé cinq méta-heuristiques hybrides (trois optimisations de colonies de fourmis hybrides et deux recuits simulés hybrides). Ils ont comparé les performances de leurs algorithmes où l'objectif est de minimiser le makespan. De plus, Liao et al. (2013) ont suggéré six méta-heuristiques pour l'affectation et le séquencement des véhicules entrants avec des dates de départs fixes et ont comparé les performances des méta-heuristiques entre elles.

Tableau 1.1 – Les travaux réalisés sur la planification des véhicules dans un cross-dock à quais multiples

Articles	Flux des produits	Interchangeabilité des produits	Fonction objective	Stockage temporaire	Types de produits	Méthodes de résolution
Assadi and Bagheri (2016)	Défini par les véhicules entrants et sortants	Oui	Minimiser l'avancement et le retard des véhicules sortants	Aux quais de réception	Plusieurs types	MILP, Differential evolution et Population-based simulated annealing.
Boysen et al. (2013)	Les temps de traitement des véhicules entrants sont prédéfinis	N/A	Minimiser les retards des chargements	N/A	Un seul type	MILP, decomposition procedures et simulated annealing
Heidari et al. (2018)	Matrice de flux de produits prédéterminés	Non	Minimiser le coût total moyen de service	Pas de stockage	Un seul type	MILP, non-dominated sorting genetic algorithm et multi-objective differential evolution
Konur and Golias (2013)	Les temps de traitement des véhicules entrants sont prédéfinis	N/A	Minimiser le coût total moyen de service	N/A	N/A	MILP, Genetic algorithm.
Molavi et al. (2018)	Matrice de flux de produits prédéterminée	Non	Minimiser les retards des chargements et le coût de livraison des chargements en retard	Devant les quais d'expédition	Un seul type	MILP, hybrid genetic algorithm and reduced variable neighborhood search, differential evolution et particle swarm optimization.
Tootkaleh et al. (2016)	Matrice de flux de produits prédéterminés (ainsi que le temps de transfert des produits)	Oui	Minimiser le coût de stockage	Une zone de stockage pour tous les types de produits	Plusieurs types	MILP, Heuristique
Van Belle et al. (2013)	Matrice de flux de produits prédéterminée	Non	Minimiser les retards des véhicules entrants et sortants	Devant les quais de réception	Un seul type	MILP, Tabu search.
Wisittipanich and Hengmeechai (2017)	Défini par les véhicules entrants et sortants	Oui	Minimiser le makespan	Devant les quais de réception	Plusieurs types	MILP, modified particle swarm optimization

Le tableau 1.1 présente les caractéristiques des travaux qui traitent la planification des véhicules dans un cross-dock à quais multiples ainsi que les contraintes des modèles MILP proposés dans la littérature. Dans le tableau 1.1, la deuxième colonne « flux de produits » indique si le routage et le transfert des produits entre les véhicules entrants et les véhicules sortants est prédéterminé ou bien c'est une variable de décision à déterminer. En effet, lorsque la matrice de flux de produits est une variable de décision, les produits déchargés/chargés sont définis pour chaque véhicule entrant/sortant. La troisième colonne indique si les produits sont interchangeables. Comme défini dans Ladier and Alpan (2016a), si les produits sont interchangeables, alors chaque produit peut remplacer un autre dans un véhicule sortant, ou bien chaque produit est dédié à une destination spécifique. Toutefois, si les produits ne sont pas interchangeables, chaque produit doit être chargé dans un véhicule sortant spécifique. La quatrième colonne montre la fonction objective du modèle. La cinquième colonne indique si le stockage temporaire est pris en compte dans le cross-dock. Le nombre de types de produits dans le modèle MILP est présenté dans la sixième colonne. La dernière colonne présente les méthodes de résolution utilisées dans chaque travail.

Dans le tableau 1.1, Boysen et al. (2013), Heidari et al. (2018), Molavi et al. (2018) et Van Belle et al. (2013) considèrent un MILP avec un seul type de produits et le flux de produits est défini par une matrice prédéterminée dans Heidari et al. (2018), Molavi et al. (2018), Tootkaleh et al. (2016) et Van Belle et al. (2013). Parmi les travaux présentés dans le tableau1.1, ils existent deux travaux dans lesquels le flux de produits est défini par des véhicules entrants/sortants et manipulant plusieurs produits interchangeables simultanément : Assadi and Bagheri (2016) et Wisittipanich and Hengmeechai (2017). Cependant, dans les deux travaux, le stockage temporaire est placé devant les quais de réception.

1.2.5 La durabilité dans les cross-docks classiques

Dans le contexte du cross-docking classique, la notion de durabilité est souvent ignorée et n'a pas été largement abordée dans la littérature. À notre connaissance, peu de travaux ont été consacrés à la durabilité et à la consommation d'énergie dans le cadre du problème de planification des véhicules dans les cross-docks. Par exemple, Dulebenets

(2018) a abordé la planification des véhicules durables et a présenté un modèle mathématique MILP pour la planification des véhicules entrants et sortants dans un cross-dock à quais multiples. En considérant un ensemble de véhicules entrants et sortants avec leurs trois coûts (coûts de manutention, d'attente et de départ retardé), l'objectif est de trouver une planification pour les véhicules entrants et sortants tout en minimisant le coût total. Le modèle mathématique a été résolu sur CPLEX. Ensuite, un algorithme : Diploid Evolutionary Algorithm a été proposé pour résoudre le problème. L'algorithme proposé surpasse l'Evolutionary Algorithm typique de la littérature sur la planification des véhicules aux quais. Une autre étude liée à la consommation d'énergie Shahram fard and Vahdani (2019) aborde le problème d'affectation et la planification des véhicules dans un centre de cross-docking. Les auteurs ont proposé un modèle mathématique à deux objectifs pour la planification des véhicules entrants et sortants en tenant compte de la consommation d'énergie des chariots élévateurs. Le problème a été ensuite résolu à l'aide de deux méta-heuristiques multi-objectives.

1.3 L'Internet Physique : concept et définitions

1.3.1 Présentation

L'efficacité et la réactivité des chaînes logistiques sont devenues un défi majeur pour les sociétés de logistique et de distribution pour satisfaire les demandes des détaillants et des clients en termes de coût, de qualité et de délai de livraison (Christopher, 2016; Stock et al., 2010). Les clients deviennent de plus en plus exigeants, non seulement en termes de qualité du produit, mais également en ce qui concerne les services de livraison qui doivent être rapides et flexibles. En conséquence, avec l'augmentation des contraintes environnementales, la durabilité de la chaîne logistique est devenue un objectif majeur pour les entreprises de logistique afin d'améliorer leur durabilité économique, sociale et environnementale (Centobelli et al., 2018; Ciccullo et al., 2018; Croom et al., 2018).

Dans ce contexte, l'Internet Physique (PI ou π) a été présenté comme un concept ambitieux visant à transformer le mode de traitement et de transport des produits dans l'ensemble du réseau de la chaîne logistique (Montreuil et al., 2013a). L'Internet Physique

a été défini comme un système logistique mondial et ouvert basé sur l'interconnectivité physique, numérique et opérationnelle via l'encapsulation (Montreuil, 2011). Atteindre la durabilité dans ses trois axes : économique, environnementale et sociale est l'objectif principal du paradigme du PI vers une logistique plus efficace. Le but est de créer un réseau logistique mondial entièrement synchronisé et interconnecté pouvant être utilisé par les différentes composantes de la chaîne logistique : fournisseurs, fabricants et détaillants. Un autre aspect majeur du PI est que les marchandises ou les produits ne sont pas gérés directement mais via des conteneurs modulaires standardisés appelés : PI-containers.

1.3.2 Définitions et notions de base

Trois éléments clés sont nécessaires pour assurer le bon fonctionnement du PI : les PI-containers, les PI-movers et les PI-nodes (Figure 1.2). Les PI-containers sont des conteneurs standardisés. Ils peuvent être gérés et stockés par différentes entreprises. Les PI-movers sont utilisés pour déplacer et manipuler les PI-containers tels que les PI-vehicles (PI-trucks, PI-lift, etc.) et les PI-carriers (PI-wagons, PI-trailers, etc.). Les PI-nodes sont des emplacements pour la réception, le stockage et le transfert des PI-containers entre les PI-movers. Les PI-nodes peuvent être des PI-transits, des PI-switches, des PI-bridges ou des PI-hubs, tels que des Road-Rail PI-hubs et Road-Road PI-hubs (Montreuil, 2011; Sarraj et al., 2014).

1.3.2.1 Les PI-containers

Les PI-containers sont les éléments clés du PI. Ce sont des conteneurs modulaires aux dimensions standardisées basés sur le concept de l'encapsulation. Ils peuvent être gérés et stockés par différentes entreprises. Les PI-containers sont faciles à acheminer vers des centres de distribution en exploitant leurs capacités d'identification, de communication et de raisonnement en temps réel. En plus, les véhicules, les systèmes de manutention, et les transporteurs sont conçus pour manipuler les PI-containers de manière fluide et efficace.

Les PI-containers possèdent de nombreux avantages :

— **Faciles à gérer, stocker et transporter :** Les PI-containers ont des dimensions

standardisées et sont faciles à identifier dans la chaîne logistique.

— **Intelligents et connectés :** Les PI-containers sont faciles à identifier et capables de communiquer entre eux.

— **Ecologiques :** Les PI-containers sont conçues pour être légers, réutilisables et recyclables.

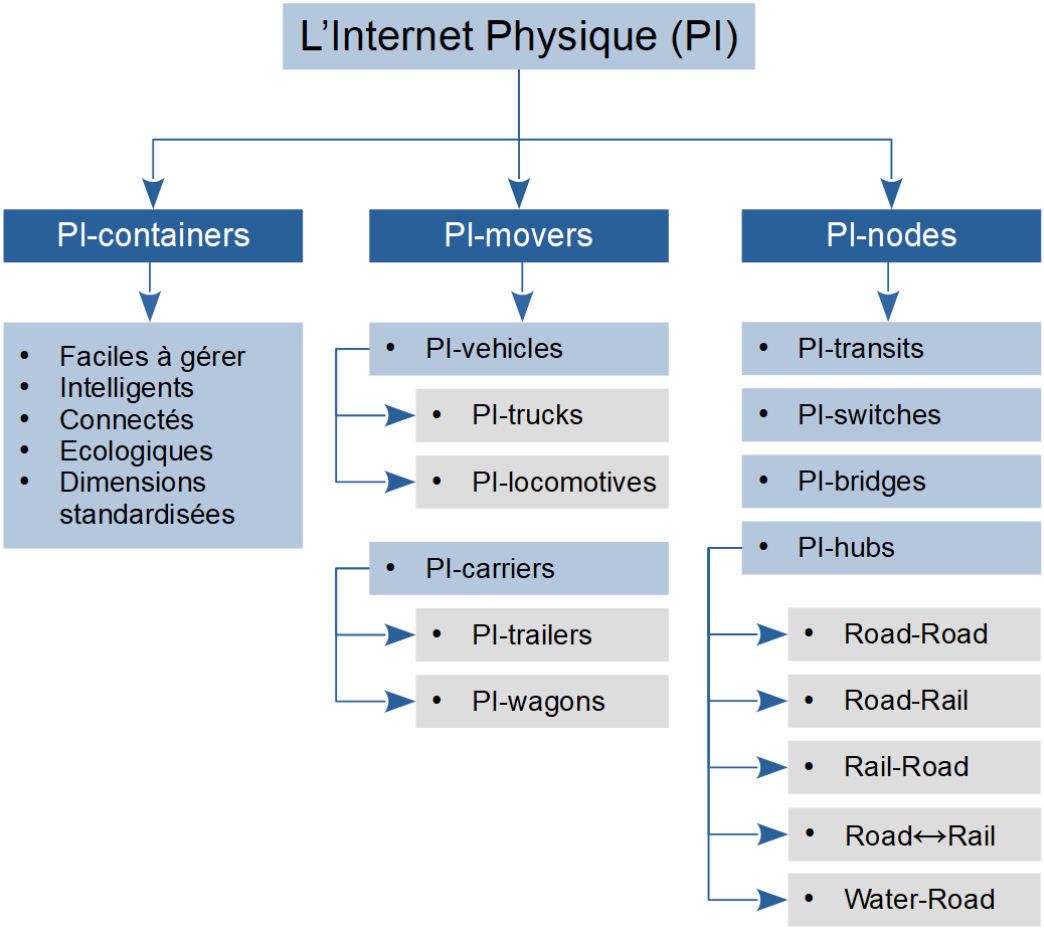

Figure 1.2 – Les éléments clés de l'Internet Physique

Les PI-containers sont développés dans une variété de dimensions modulaires et standardisées qui feront l'objet de normes internationales en constante évolution. Cette modularité dimensionnelle des PI-containers peut être exprimée en hauteur, largeur et profondeur grâce à une combinaison de plusieurs dimensions. Ils doivent être disponibles dans une variété adaptée au poids et aux différentes caractéristiques des charges qu'ils doivent contenir, tout en étant aussi légers que possible. Ils peuvent également avoir des capacités de conditionnement telles que le contrôle de la température, de l'humidité et des vibrations. Les PI-containers devraient être aussi respectueux de l'environnement, conformément aux principes de durabilité (Montreuil, 2011).

1.3.2.2 Les PI-movers

Les PI-movers sont utilisés pour déplacer et manipuler les PI-containers tels que les PI-vehicles et les PI-carriers. Les PI-vehicles sont des véhicules tracteurs et ils incluent les PI-trucks, PI-locomotives, PI-boats, PI-planes, PI-lifts, etc., tandis que les PI-carriers (tels que : les PI-trailers et les PI-wagons) doivent être déplacés par les PI-vehicles.

1.3.2.3 Les PI-nodes

Ce sont des emplacements pour la réception, le stockage et le transfert des PI-containers entre des PI-movers. Les PI-nodes peuvent être des PI-transits, des PI-switches, des PI-bridges ou des PI-hubs, tels que les Road-Road PI-hubs, les Road-Rail PI-hubs et les Water-Road PI-hubs. Parmi les nombreux types de PI-hubs on peut citer :

— **Les Road-Road PI-hubs :** Ce sont des terminaux de cross-docking utilisés pour transférer efficacement les PI-containers entre les véhicules entrants et les véhicules sortants à travers les PI-sorters. Les Road-Road PI-hubs sont composés de PI-sorters connectés aux PI-docks en utilisant des zones de manœuvre qui rangent les PI-containers après le déchargement des véhicules entrants pour être ensuite acheminés dans les PI-sorters puis groupés et chargés dans les véhicules sortants (Meller et al., 2012).

— **Les Rail-Road PI-hubs :** Ils utilisent le même mécanisme que les Road-Road PI-hubs. Toutefois, les PI-containers sont transférés des trains (wagons) vers les véhicules sortants à l'aide des PI-sorters dans la zone de routage puis dans la zone de manœuvre afin de grouper les PI-containers. Ballot et al. (2012) ont présenté une conception fonctionnelle détaillée des Rail-Road PI-hubs.

— **Les Road-Rail PI-hubs :** Contrairement au Rail-Road PI-hubs, les Road-Rail PI-hubs sont utilisés pour le routage et le transfert des PI-containers des véhicules vers les wagons.

— **Les Two-Way Road-Rail PI-hubs :** Ils sont conçus pour transférer en même temps les PI-containers des véhicules vers les wagons et vice-versa.

— **Les Water-Road PI-hubs :** Déstinés au transfert des PI-containers entre bateaux et véhicules dans un terminal portuaire (Montreuil et al., 2013b).

1.4 L'Internet Physique : vers une chaîne logistique interconnectée, durable et résiliente

1.4.1 L'interconnectivité dans la chaîne logistique PI

Au cours des dernières années, de nombreuses études ont été menées dans la littérature du PI utilisant des différentes méthodologies telles que la modélisation, les études de cas et les simulations. Plusieurs études ont abordé la conception et le dimensionnement des conteneurs dans le contexte du PI (Gazzard and Montreuil, 2015; Landschützer et al., 2015; Lin et al., 2014). L'hyper-connectivité de la chaîne logistique dans le contexte du PI a été abordée par de nombreux auteurs dans la littérature. Crainic and Montreuil (2016) ont introduit l'idée d'une logistique urbaine hyperconnectée et ses concepts principaux ainsi que des systèmes de transport qui sont beaucoup plus efficaces et durables. Sohrabi (2016) a proposé des modèles d'optimisation pour les systèmes de distribution hyperconnectés tandis que d'autres études telles que celles de Marcotte et al. (2015) se sont concentrées sur les opérations de production. Naccache (2016); Pan et al. (2015); Yang (2016) ont abordé le contrôle et la gestion des stocks dans une chaîne logistique hyperconnectée, tandis que d'autres études se sont concentrées sur la conception d'installations du PI telles que des centres de Road-Road cross-docks (Montreuil et al., 2013c) et les Road-Rail PI-hubs (Ballot et al., 2012). Qiao et al. (2019) ont développé un modèle de tarification dynamique et multi-périodes afin d'étudier l'influence des futures demandes sur la décision de tarification courante. Pan et al. (2017); Treiblmaier et al. (2016) ont présenté une classification et une analyse détaillées des travaux relatifs au PI. La multimodalité dans les transports consiste à utiliser au moins deux modes de transport différents. Le transport synchro-modale a été présenté comme un nouveau concept qui, outre la multimodalité, envisage des approches réactives en temps réel permettant de gérer les perturbations éventuelles de la chaîne logistique, telles que les retards et le réacheminement des véhicules (Ambra et al., 2019).

Dans les réseaux logistiques classiques, en plus du stockage et la distribution qui sont souvent limités et fixés, l'architecture de ces réseaux est généralement hiérarchique, contrairement aux réseaux logistiques du PI qui utilisent des approches décentralisées qui

assurent la synchronisation et l'interconnectivité de la chaîne logistique. En effet, dans le PI, les ressources de stockage et de transport sont partagées et peuvent être utilisées à la demande de l'utilisateur qui peut être un fournisseur, un hub ou un détaillant. Cette flexibilité du réseau logistique s'est révélée supérieure à celle du problème classique de gestion des inventaires (Yang et al., 2017a) même en cas de perturbations (Yang et al., 2017b). L'impact positif de la configuration du PI a également été observé du point de vue social et environnemental (Fazili et al., 2017).

1.4.2 La durabilité dans la chaîne logistique

Au cours de ces dernières années, l'amélioration de la durabilité de la chaîne logistique est devenue un objectif prioritaire pour les entreprises logistiques en prenant en compte les aspects environnementaux, économiques et sociaux tout en réduisant les coûts totaux et en optimisant les opérations globales de la chaîne logistique. La durabilité d'une chaîne logistique est abordée à travers les interactions entre les différentes composantes de la chaîne logistique (Fournisseurs, usines de production, détaillants, etc.) afin de générer des gains environnementaux, économiques et sociaux pour un seul composant ou pour l'ensemble de la chaîne logistique (Taylor and Vachon, 2018). Récemment, de nombreuses revues de littérature ont été effectuées sur le sujet de la durabilité des chaînes logistiques. La littérature existante a été limitée aux aspects individuels de la durabilité de la chaîne logistique au lieu d'aborder les dimensions globales qui devraient prendre en compte d'autres aspects tels que les dimensions sociales et économiques et la gestion de la chaîne logistique verte (Winter and Knemeyer, 2013).

Les concepts internes et externes de la chaîne logistique ont un impact important sur la performance en matière de durabilité. En effet, les pratiques internes de la chaîne logistique ont un effet majeur sur la durabilité environnementale. La coordination externe et la communication au sein de la chaîne logistique ont un impact direct sur la durabilité, non seulement du point de vue environnemental, mais également pour l'aspect social (Jadhav et al., 2018). Une revue de littérature récente sur ce sujet Koberg and Longoni (2019) a révélé que la configuration des composantes de la chaîne logistique ainsi que les mécanismes relationnels sont les éléments clés de la durabilité dans la chaîne logistique globale. Le premier élément, qui est la configuration, consiste en la liaison entre les diffé-

rents acteurs du réseau de la chaîne logistique. Le mécanisme relationnel consiste à gérer les relations entre les acteurs de la chaîne logistique.

Afin d'intégrer la durabilité dans les décisions de planification de la chaîne logistique, Mota et al. (2015) ont proposé un modèle mathématique multi-objectif prenant en compte les trois aspects de la durabilité en proposant des indicateurs d'évaluation pour chacun des trois aspects. En effet, l'aspect économique consiste à prendre en compte le coût total qui comprend les coûts fixes, les coûts des matières premières, les coûts de transport et les ressources humaines. L'aspect environnemental est évalué à travers l'impact environnemental des différentes activités de la chaîne logistique telles que la production et le transport. L'aspect social est défini par un indicateur de bénéfice social. Le modèle proposé a ensuite été validé sur une étude de cas réelle. Kong et al. (2018) ont abordé le concept du juste à temps (JIT) dans la gestion de la chaîne logistique et ont intégré l'impact environnemental et économique au problème de la planification des lots en proposant un modèle mathématique minimisant plusieurs objectifs à la fois : avancement et retard, gaspillage de ressources et émissions dans l'environnement. Le problème a ensuite été résolu à l'aide d'un algorithme à temps polynomial.

L'industrie agro-alimentaire a un effet important sur la durabilité des chaînes logistiques, notamment du point de vue environnemental. Dans ce contexte, Demartini et al. (2018) ont analysé la durabilité des chaînes logistiques des boissons pour déterminer les meilleures pratiques en matière de durabilité ainsi que les indicateurs de performance clés les plus liés à la durabilité. Guo et al. (2017) ont proposé une approche basée sur un algorithme génétique hybride pour examiner les effets de la coordination entre la production et le transport sur la durabilité de la chaîne logistique, en prenant en compte plusieurs caractéristiques telles que les heures de départ et les modes de transport. Les résultats ont montré que la coordination entre la production et le transport "vert" avait un impact important sur la durabilité de la chaîne logistique globale.

1.4.3 La résilience de la chaîne logistique

De manière générale, la résilience d'un système est définie par sa capacité à revenir à son état d'origine ou à passer à un nouvel état plus désirable après avoir été perturbé (Bhamra et al., 2011). Dans le contexte de la durabilité, de nombreux travaux ont abordé

la résilience de la chaîne logistique à différents types de perturbations. Afin d'examiner la résilience et la robustesse d'une chaîne logistique, différentes approches telles que celles basées sur la simulation et la programmation mathématique peuvent être utilisées. On peut trouver de nombreuses études utilisant la simulation pour examiner la durabilité et la résilience. Par exemple, Ivanov (2018) a analysé les effets de la propagation des perturbations dans la chaîne logistique en utilisant la simulation à événements discrets visant à améliorer la durabilité. Plusieurs facteurs de durabilité, tels que la stratégie d'approvisionnement unique ou la réduction du stock, ont des impacts différents sur la durabilité. Yang et al. (2017a) ont proposé un modèle d'inventaire géré par le fournisseur et ont développé aussi un modèle d'optimisation basé sur la simulation pour contrôler au mieux le stock. Les auteurs ont comparé les performances du modèle d'inventaire hiérarchique classique au modèle basé sur le PI. Leurs résultats expérimentaux montrent que la performance du modèle de gestion du stock basé sur le PI dépasse celle des modèles classiques grâce à sa flexibilité et à sa capacité à partager de manière dynamique les ressources de stockage et de transport entre les hubs.

À l'aide d'un modèle de simulation, Yang et al. (2017b) ont évalué la robustesse des modèles d'inventaire dans un réseau interconnecté basé sur le PI. Les auteurs ont constaté que le modèle basé sur le PI fonctionne mieux que le modèle classique en cas de perturbations dans les hubs et les usines de production. Dans une étude par simulation multi-agents, Nouiri et al. (2018) ont proposé deux modèles de simulation multi-agents pour évaluer les performances de la chaîne logistique PI par rapport à la chaîne logistique classique en évaluant le coût de transport, le coût de stockage, le coût d'utilisation des ressources ainsi que le délai de livraison. Plusieurs scénarios ont été réalisés en faisant varier le nombre de véhicules et leurs capacités, les stratégies de demande des détaillants et le niveau de perturbations. Les résultats ont montré que la chaîne logistique PI est plus efficace que la chaîne logistique classique. Les coûts de possession, de transport, d'utilisation moyenne des véhicules et des retards moyens sont améliorés par rapport à la chaîne logistique classique. Fazili et al. (2017) ont comparé les performances du PI à un système logistique conventionnel à travers une simulation de Monte-Carlo. Leurs résultats montrent que, d'un point de vue social, le PI diminue le temps de conduite et réduit aussi la distance parcourue ainsi que la circulation des véhicules et donc leur impact environnemental. D'autres chercheurs ont utilisé la programmation stochastique pour

résoudre le problème de la durabilité. Par exemple, Jabbarzadeh et al. (2018) ont utilisé une approche en deux phases consistant à combiner une méthode de regroupement de type c-means floue pour évaluer la durabilité et un modèle de programmation stochastique pour maintenir la performance de durabilité de la chaîne logistique en cas de perturbations aléatoires.

1.5 Les PI-cross-docks

Dans cette section, les problèmes liés aux différents types de PI-cross-docks les plus traités dans la littérature sont présentés : à savoir les Rail-Road PI-hubs et les Road-Road PI-hubs. Ces problèmes sont abordés en utilisant plusieurs approches qui seront également détaillées. Comparés aux opérations classiques des cross-docks, les PI-cross-docks fonctionnent dans des temps de chargement/déchargement plus courts. Les PI-cross-docks fonctionnent à un haut niveau d'automatisation en utilisant des systèmes de manutention innovants au lieu des chariots élévateurs ou les convoyeurs classiques. En plus, les PI-containers ont la capacité de se déplacer automatiquement sur les PI-sorters composés de plusieurs PI-conveyors. Chaque PI-conveyor peut transporter un PI-container dans quatre directions (Montreuil et al., 2013c).

Dans la suite de ce chapitre, on se concentre sur les travaux liés aux deux premières catégories des PI-cross-docks (section 1.3.2.3 de ce chapitre) qui sont les seules abordées dans la littérature du PI, à savoir les Rail-Road et les Road-Road PI-hubs.

1.5.1 Les Rail-Road PI-hubs

De nombreux chercheurs ont abordé le problème du Rail-Road PI-hub (Figure 1.3). Par exemple, Ballot et al. (2012) ont proposé une conception fonctionnelle du Rail-Road PI-hub. Les auteurs décrivent le flux des PI-containers, des véhicules et des trains dans le PI-hub. En plus, leur travail présente également plusieurs indicateurs de performance (KPI) afin d'évaluer la performance du Rail-Road PI-hub du point de vue des clients (tels que : le temps de traitement des trains, les emplacements vides dans les trains et les véhicules, etc.). Et du point de vue des opérateurs du Rail-Road PI-hub (par exemple :

le nombre de wagons, le nombre de quai de chargement / déchargement, le nombre de PI-containers traités par période, etc.).

D'autres études se sont davantage concentrées sur le regroupement des PI-containers dans les Rail-Road PI-hub ; par exemple, Pach et al. (2014) ont proposé différentes stratégies de regroupement pour le chargement de PI-containers dans les véhicules. Leur étude a montré l'effet du regroupement des PI-containers sur la réduction du temps d'évacuation dans le Rail-Road PI-hub. En outre, ils ont également spécifié la stratégie à utiliser dans chaque situation différente pour éviter de bloquer les PI-docks (PI-quais), ce qui entraine des retards de chargement des véhicules. Sallez et al. (2016) ont étudié la notion d'activité (activeness) des PI-containers en les considérant comme des produits autonomes capables de communiquer, d'apprendre et d'identifier leur état. Le concept d'activité des PI-containers a été évalué par le biais d'une étude de simulation sur un Rail-Road PI-hub.

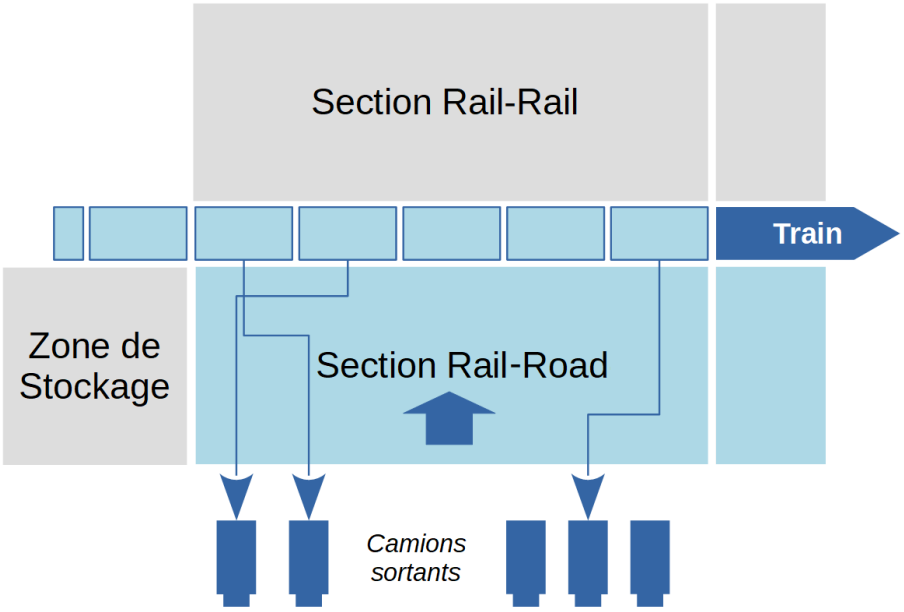

Figure 1.3 – Le Rail-Road PI-hub

Les architectures de contrôle hybrides dans les Rail-Road PI-hubs ont été abordées par plusieurs auteurs dans la littérature. Dans une étude par simulation multi-agents, Sallez et al. (2015) ont développé une approche réactive dans une architecture de contrôle hybride pour l'acheminement des PI-containers dans le Rail-Road PI-hub cross-dock. La robustesse de l'approche réactive proposée a été évaluée par le temps d'évacuation moyen des PI-containers tout en prenant en compte des éventuelles perturbations internes et externes. Vo et al. (2018) ont proposé une architecture hybride ORCA (architecture de

contrôle optimisé et réactif ou Optimized and Reactive Control Architecture) pour le Rail-Road PI-hub. La performance de leur approche a été évaluée à travers une étude de simulation dans laquelle ils comparent différentes stratégies de routage. Afin de mettre en évidence les avantages environnementaux du PI, Sun et al. (2018) ont étudié le réseau de transport intermodal Rail-Road à travers une simulation multi-agents et ont montré son impact sur la réduction de la consommation de carburant.

Plusieurs auteurs ont abordé le problème d'affectation des véhicules aux PI-docks dans les Rail-Road PI-hubs. Le problème concerne la section de déchargement des trains (la section Rail-Road) du PI-hub utilisée pour transférer les PI-containers des wagons vers les véhicules. Par exemple, Walha et al. (2014a) ont suggéré une formulation mathématique et une heuristique pour trouver la meilleure affectation des véhicules aux quais tout en minimisant la distance interne parcourue par les PI-containers des wagons vers les véhicules. Plus récemment, les mêmes auteurs Walha et al. (2016) ont proposé une méta-heuristique de recuit simulé permettant de résoudre le problème d'affectation des véhicules aux PI-docks afin de minimiser la distance parcourue par les PI-containers et le nombre de véhicules utilisés. Les auteurs ont également proposé une approche multi-agents qui génère des solutions considérant l'aspect dynamique et prenant en compte le taux de remplissage et la distance parcourue.

1.5.2 Les Road-Road PI-hubs

Contrairement aux Rail-Road PI-hubs, les Road-Road PI-hubs n'ont pas été assez abordés dans la littérature. L'objectif des Road-Road PI-hubs est de transférer de manière efficace et durable les PI-containers des véhicules vers d'autre véhicules (Figure 1.4). Montreuil et al. (2013c) ont proposé une conception fonctionnelle détaillée du Road-Road PI-hub ainsi que des indicateurs de performance (KPI) pour évaluer la performance du point de vu des clients ainsi que les performances au niveau interne du Road-Road PI-hub. Les auteurs ont utilisé la simulation pour évaluer leur conception et les performances du PI-hub. Ils ont également présenté un modèle pour la zone de PI-switch.

Afin de récapituler les travaux de recherche liés aux PI-cross-docks, le tableau 1.2 présente ces travaux ainsi que les approches et les méthodes de résolution utilisées.

On constate dans le tableau 1.2 que les Road-Rail et Road-Road PI-hubs n'ont pas été abordés dans la littérature du PI. La plupart des contributions dans la littérature proposent des modèles multi-agent afin de garantir une réactivité face aux perturbations, ce qui permet de trouver des solutions alternatives (Figure 1.5).

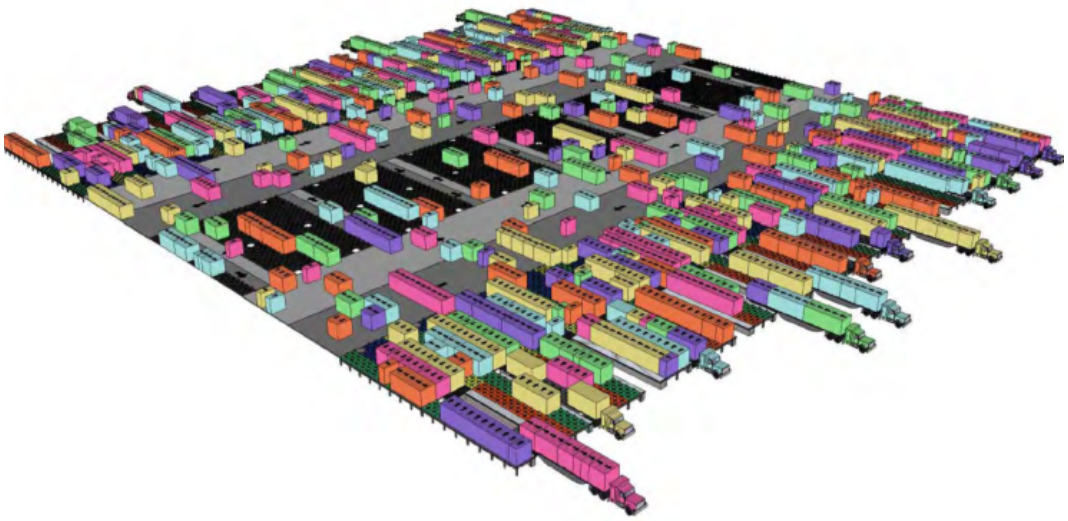

Figure 1.4 – Le Road-Road PI-hub Montreuil et al. (2013c)

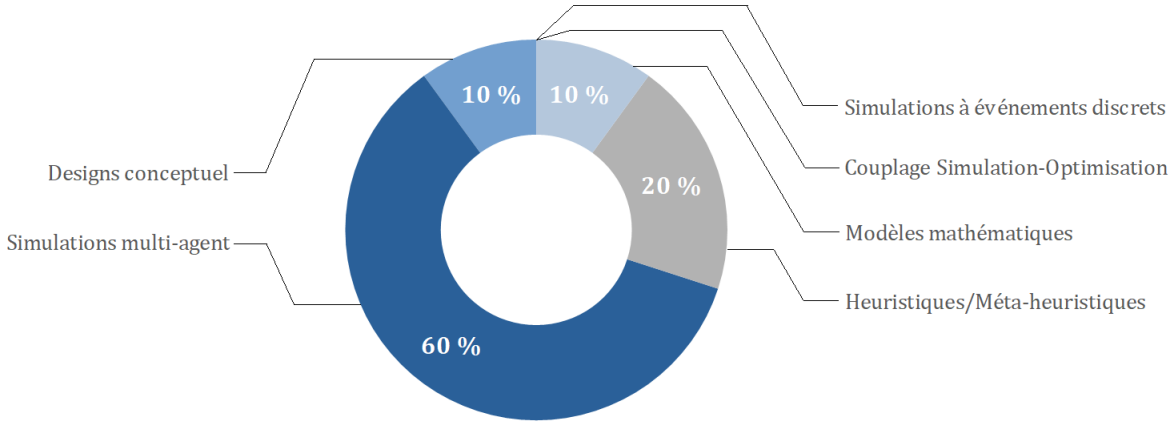

Figure 1.5 – Approches de résolution des problèmes du PI-cross-dock dans la littérature

La plupart des travaux traitant les problèmes liés aux PI-cross-docks se sont concentrés principalement sur les problèmes du Rail-Road PI-hubs, proposent des solutions à ces problèmes en utilisant principalement la simulation multi-agents et s'intéressent très peu aux autres méthodes d'optimisation telles que : la modélisation mathématique, les méta-heuristiques, la simulation à événements discrets, le couplage Simulation-Optimisation, etc. Ces approches peuvent être efficacement intégrées avec la simulation multi-agents.

Tableau 1.2 – Travaux sur les PI-cross-docks et les approches de résolution utilisées

Auteurs	**Travaux**	**Approches de résolution**
Ballot et al. (2012)	Design fonctionnel et KPIs pour le Road-Rail PI-hub	Design conceptuel et fonctionnel
Montreuil et al. (2013c)	Design fonctionnel et indicateurs de performance pour le Road-Road PI-hub	Design conceptuel et fonctionnel
Pach et al. (2014)	Stratégies de groupement des PI-containers et leur impact sur le chargement des véhicules	Simulation multi-agent
Sallez et al. (2015)	Architecture de contrôle hybride pour le routage des PI-containers	Simulation multi-agent
Sallez et al. (2016)	Activité des PI-containers appliqué sur une étude de cas dans un Rail-Road PI-hub	Simulation multi-agent
Sun et al. (2018)	Transport intermodal dans un réseau Rail-Road	Simulation multi-agent
Vo et al. (2018)	Architecture hybride de contrôle optimisé et réactif pour le Rail-Road PI-hub	Simulation multi-agent
Walha et al. (2014a)	Allocation des PI-containers aux quais et l'affectation des véhicules dans un Rail-Road PI-hub	Modèle mathématique MILP, Heuristique
Walha et al. (2016)	Approche active et réactive pour le problème d'allocation dans un Rail-Road PI-hub	Heuristique, Simulated Annealing, approche réactive multi-agent

1.6 Synthèse

A travers cette analyse de l'état de l'art, nous avons constaté que dans le cas des cross-docks classiques, il n'y a pas, à notre connaissance, des modèles mathématiques pour la planification des véhicules aux cross-docks qui prennent en compte plusieurs contraintes opérationnelles en même temps, telles que : Les fenêtres horaires des véhicules entrants/sortants, la gestion de plusieurs types de produits à la fois, plusieurs zones de stockage dédiées pour stocker chaque type de produits. Pour cela, dans le chapitre 2, un modèle mathématique MILP prenant en compte toutes ces contraintes à la fois sera proposé avec deux méta-heuristiques. Ainsi, afin d'évaluer les apports et les avantages de l'Internet Physique et évaluer le passage des cross-docks classiques vers les PI-hubs,

nous proposons dans le chapitre 3 deux modèles de simulation et un couplage simulation-optimisation pour générer des solutions robustes aux éventuelles pannes au niveau des ressources de transport internes.

Concernant la littérature des PI-cross-docks (ou PI-hubs) nous avons constaté que les problèmes liés au Road-Rail PI-hubs sont presque négligés, puisque la plupart des travaux dans la littérature du PI se concentrent soit sur les Rail-Road PI-hubs, soit sur la chaîne logistique globale de l'Internet Physique. Dans cette thèse, nous allons proposer différentes approches pour aborder les problèmes d'optimisation des opérations dans les Road-Rail PI-hubs. Dans le chapitre 4, nous proposons un modèle multi-agents MAS comportant trois méta-heuristiques hybrides et parallèles et un modèle MILP pour valider les solutions générées par le MAS. Le modèle MAS sera évalué également dans un environnement dynamique considérant des perturbations internes au niveau des quais de réception.

Enfin, dans le dernier chapitre nous abordons deux thématiques à la fois, qui ont été négligées dans la littérature des PI-cross-docks, à savoir : la durabilité des PI-hubs et l'optimisation multi-objective. Comme cas d'étude, nous prenons comme PI-hub le Rail-Road. Nous présentons ainsi plusieurs approches multi-objectives : un modèle mathématique par programmation lexicographique (MO-MIP) et deux méta-heuristiques hybrides (MO-VNSSA et MO-VNSTS).

1.7 Conclusion

Dans ce chapitre, l'état de l'art sur l'Internet Physique a été présenté ainsi que les notions de durabilité et la résilience des chaînes logistiques classiques et PI. Ensuite, la deuxième partie s'est focalisée sur les cross-docks qui représentent l'élément clé pour la distribution dans une chaîne logistique. D'abord, la classification de la littérature des cross-docks classiques a été suffisamment détaillée. Puis, les problèmes liés au PI-cross-docks ont été présentés et analysés en commençant d'abord par les Rail-Road PI-hubs, puis, les Road-Rail PI-hubs. Ce chapitre s'est terminé par une synthèse sur la limite des travaux abordés dans la littérature des PI-cross-docks et une présentation des contributions de cette thèse.

Chapitre 2

Planification des véhicules dans un cross-dock Road-Road classique à quais multiples

---

Ce chapitre sera consacré à l'étude du cross-dock classique avant d'entamer les PI-cross-docks (PI-hubs) dans les chapitres suivants. Ainsi, affin d'étudier le passage vers l'Internet Physique (PI), nous proposons d'abord un modèle MILP d'un problème de cross-docking inspiré d'un partenaire industriel contenant plusieurs zones de stockage. Il s'agit de la planification des véhicules entrants/sortants ainsi que la gestion du flux des produits. L'étude réalisée dans ce chapitre permettra d'évaluer l'apport du PI sur la planification des opérations de cross-docking dans les chapitres suivants.

2.1 Introduction

Avant d'aborder les cross-docks dans le contexte de l'Internet Physique, il est nécessaire d'étudier d'abord les cross-docks classiques et évaluer leurs performances. Ce chapitre aborde le problème de planification des véhicules dans un cross-dock classique à quais multiples. La suite de ce chapitre est structurée en plusieurs parties. Tout d'abord, le contexte du problème est présenté. Ensuite, nous proposons un modèle mathématique MILP pour optimiser la planification, le stockage, l'affectation et le séquencement des véhicules sur les quais de réception et d'expédition pour un problème inspiré d'un cross-dock à quais multiples d'un partenaire industriel disposant de plusieurs zones de stockage temporaires. Puis, une heuristique et deux méta-heuristiques : Tabu Search (TS) et Simulated Annealing (SA) sont proposées pour le problème de planification des véhicules dans un cross-dock à quais multiples avec plusieurs zones de stockage temporaires. L'heuristique de construction et la SA sont adaptées pour la résolution d'un benchmark d'instances de la littérature. Ce chapitre se termine par la présentation de la mise en œuvre des méthodes de résolution proposées dans le cross-dock étudié et montre les avantages industriels de l'approche proposée en la comparant à la méthode manuelle précédemment utilisée par les opérateurs du cross-dock à travers la simulation à évènements discrets.

2.2 Présentation de la problématique

Le problème étudié est rencontré par notre partenaire industriel qui souhaite améliorer la planification des véhicules aux quais de réception et d'expédition ainsi que le routage des produits entre les quais et les zones de stockage dans le cross-dock sous les contraintes suivantes :

— Le mode de service pour les quais de réception et d'expédition est exclusif.
— Le cross-dock étudié est composé de plusieurs zones de stockage dédiées et placées dans le centre du cross-dock, qui gèrent divers types de produits interchangeables, ce qui implique que les demandes des produits sont définies pour chaque camions et non par une matrice de transfère (Figure 2.1).
— Les opérations menées dans le cross-dock sont les suivantes : déchargement des

véhicules, transfert direct ou stockage de marchandises dans des zones de stockage et chargement des véhicules.

Le but est d'optimiser les opérations du cross-docking dans l'objectif de minimiser les retards des véhicules entrants et sortants. Avant l'étude, les opérateurs des quais au cross-dock affectaient les véhicules qui arrivaient manuellement et aléatoirement, en fonction du nombre de véhicules en attente ainsi que de la disponibilité des quais, ce qui entraînaient des décisions non optimales et des retards supplémentaires pour les véhicules. L'objectif de cette étude est de fournir des algorithmes au gestionnaire du cross-dock et aux opérateurs des quais pour gérer efficacement l'affectation et la planification des véhicules, ainsi que le routage et le stockage des flux de produits dans le cross-dock. L'objectif défini par le manager du cross-dock est de minimiser les retards des véhicules entrants / sortants afin de minimiser les coûts des pénalités appliquées par les clients et les fournisseurs. Les algorithmes développés aideront les opérateurs des quais à répondre aux questions opérationnelles suivantes :

— À quel quai le véhicule doit être déchargé / chargé ?
— À quelle heure ? Pour combien de temps ?
— Quels sont les produits à transférer directement ?
— Dans quel véhicule sortant doivent-ils être chargés ?
— Quels sont les produits à stocker temporairement ?

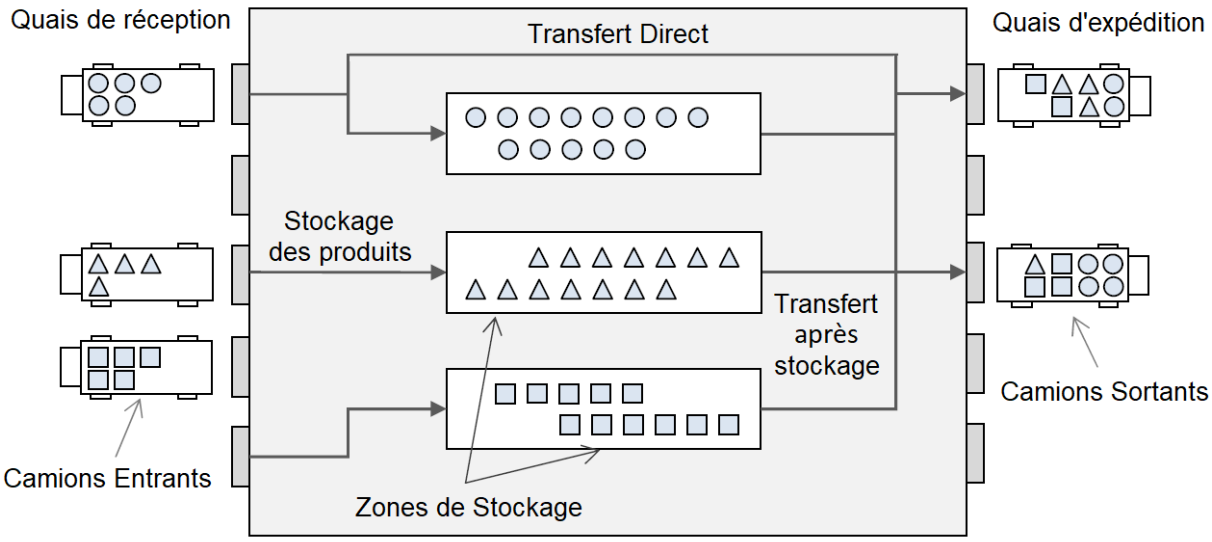

Figure 2.1 – Processus et flux des produits dans le cross-dock étudié

Les hypothèses qui traduisent plus précisément les contraintes gérées par notre partenaire industriel sont les suivantes :

- Plusieurs quais de réception/expédition sont pris en compte, et le mode de service des quais est exclusif, chaque quai étant destiné à recevoir ou expédier des produits.
- Le modèle MILP proposé prend en compte le délai de transbordement des produits. La durée de transbordement dans le cross-dock dépend de deux paramètres :
  - La quantité de produits à transférer.
  - La distance entre les quais de réception/expédition et les zones de stockage.
- Les véhicules entrants et sortants ne sont pas disponibles à l'instant zéro.
- Les heures d'arrivée et de départ prévues sont définies par les véhicules entrants et sortants.
- Plusieurs zones de stockage dédiées sont considérées en fonction du nombre de types de produits gérés par le cross-dock.
- Si les véhicules sortants ne sont pas encore disponibles, les produits reçus sont stockés.
- Les produits sont interchangeables et peuvent être transférés de manière flexible vers tout véhicule sortant pouvant prendre ce type de produit.
- Le temps de chargement/déchargement des produits dépend de la quantité à charger/décharger et le temps de transfert des produits dépend du nombre de produits et de la distance parcourue par les produits à l'intérieur du cross-dock.
- Les ressources de transport considérées dans ce modèle sont des ressources mobiles (telles que des chariots élévateurs, des véhicules autoguidés AGV, etc.). Il existe une ressource de transport entre chaque quai entrant/sortant ou entre chaque quai et une zone de stockage.
- Si les deux véhicules entrants et sortants sont disponibles en même temps, les produits sont transférés directement sans passer par les zones de stockage.
- Pour les clients, les véhicules sortants peuvent demander différentes quantités de tous types de produits ou seulement certains types spécifiques.
- Les produits sont expédiés sur des palettes standardisées (un produit par palette).
- Les véhicules sortants peuvent demander différents types de produits avec des quantités différentes.

— Il n'y a pas de préemption pour les activités de déchargement ou de chargement. Avant de quitter le quai, le véhicule doit terminer le déchargement ou chargement de tous ses produits.

2.3 Modèle mathématique

Dans cette section, nous proposons un modèle MILP pour le problème d'ordonnancement et d'affectation des véhicules dans un cross-dock à quais multiples avec plusieurs zones de stockage. L'objectif est de minimiser le retard total des véhicules entrants et sortants (Figure 2.3). En minimisant les retards des véhicules, le temps de transfert des produits est également minimisé. Les heures d'arrivée et de départ souhaitées des véhicules entrants et sortants sont considérées comme des données d'entrée du modèle, y compris les distances entre les quais de réception/expédition et entre les quais et les zones de stockage. La quantité de produits à décharger par les véhicules entrants et la demande de produits pour chacun des véhicules sortants sont également considérées comme paramètres d'entrée.

En plus de l'affectation et du séquencement des véhicules, la quantité de produits qui doit être transférée des véhicules entrants aux véhicules sortants ou via des zones de stockage est considérée comme une variable de décision dans ce modèle. La variable de décision α_{ijp} est la quantité de produits de type p à transférer directement (sans stockage) du véhicule entrant i au véhicule sortant j. La variable de décision β_{ijp} est la quantité de produits de type p qui sera déchargée du véhicule entrant i, puis stockée dans la zone de stockage des produits p, puis transférée de la zone de stockage p au véhicule sortant j. Les variables α_{ijp} et β_{ijp} sont utilisées pour calculer l'heure de fin (e_i) du déchargement du véhicule entrant i et l'heure de fin du chargement (f_j) du véhicule sortant j. Les principaux paramètres d'entrée et les variables de sortie du problème sont mis en évidence dans la figure 2.2. Le modèle comprend 10 variables principales suffisantes pour représenter une solution (Figure 2.2). Les variables restantes ne sont que des variables auxiliaires qui aident à calculer les variables principales.

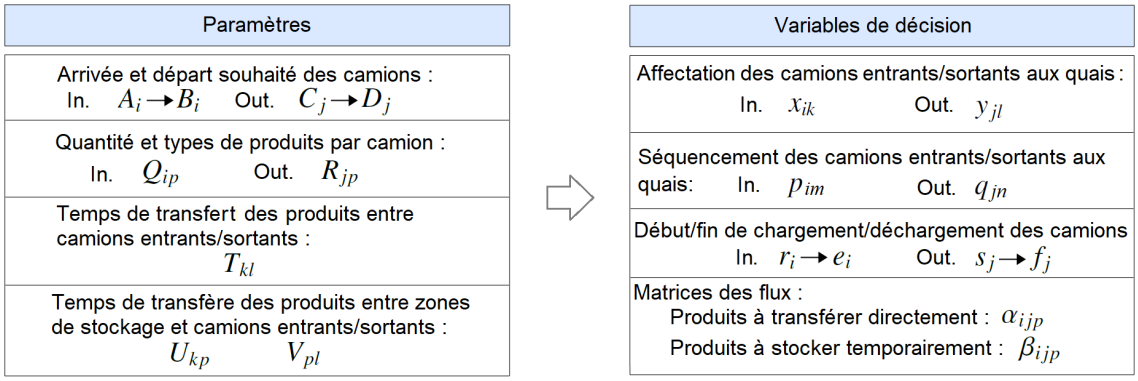

Figure 2.2 – Paramètres et variables de décision principales du modèle proposé (MILP)

2.3.1 Paramètres

N_1 Nombre de véhicules entrants

N_2 Nombre de véhicules sortants

M_1 Nombre de quais de réception

M_2 Nombre de quais d'expédition

P Nombre de types de produits

i, m Indices des véhicules entrants : $i, m = 1, ..., N_1$

j, n Indices des véhicules sortants : $j, n = 1, ..., N_2$

k Indices des quais de réception : $k = 1, ..., M_1$

l Indices des quais d'expédition : $l = 1, ..., M_2$

p Indices des types de produits (et des zones de stockage) : $p = 1, ..., P$

T_{kl} Temps pour transférer un produit d'un quai de réception k au véhicule sortant demandant le produit au quai d'expédition l

U_{kp} Temps nécessaire pour transférer un produit d'un quai de réception k vers la zone de stockage correspondante p

V_{pl} Temps de transfert d'un produit d'une zone de stockage p vers le véhicule sortant pouvant prendre le produit au quai d'expédition l

Q_{ip} Quantité de produits de type p à décharger par un véhicule entrant i

R_{jp} Quantité de produits de type p demandée par le véhicule sortant j

A_i Heure d'arrivée du véhicule entrant i

B_i Heure de départ prévue du véhicule entrant i

C_j Heure d'arrivée du véhicule sortant j

D_j Heure de départ prévue du véhicule sortant j

L Temps pour charger ou décharger un produit

E Temps de changement des véhicules au niveau des quais

M Un grand nombre positif

2.3.2 Variables de décision

2.3.2.1 Variables continues

Les variables continues suivantes sont prises en compte pour calculer le temps et les retards :

r_i Heure de début du déchargement du véhicule entrant i

e_i Heure de fin du déchargement du véhicule entrant i

s_j Heure de début du chargement du véhicule sortant j

f_j Heure de fin du chargement du véhicule sortant j

d_i Heure de fin du transfert des produits directement du véhicule entrant i

g_i Heure de fin de stockage des produits du véhicule entrant i

Étant donné que les zones de stockage sont au centre du cross-dock (Figure 2.3), les produits ne peuvent pas être stockés devant les quais entrants ou sortants. Dans ce modèle mathématique, de nouvelles variables de décision sont intégrées pour gérer le processus de stockage et de nouvelles contraintes sont mises en œuvre pour le contrôle des flux des produits.

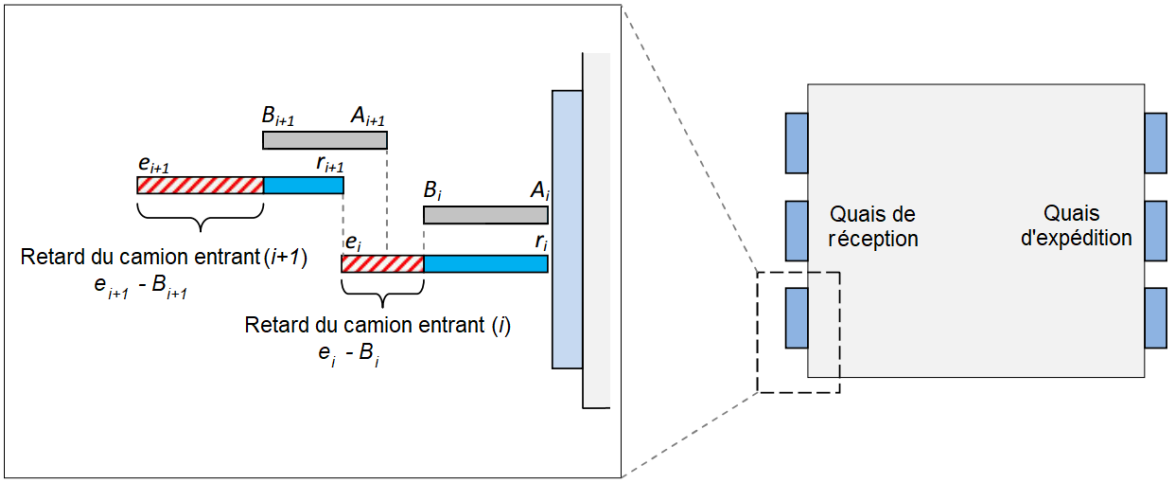

Figure 2.3 – Exemple de retard des véhicules entrants i et $i+1$ sur le même quai

2.3.2.2 Variables entières

La quantité de produits transférés/stockés est calculée à l'aide des variables entières suivantes (matrices de flux) :

α_{ijp} Quantité de produits de type p à transférer directement du véhicule entrant i au véhicule sortant j

β_{ijp} Quantité de produits de type p stockée par le véhicule entrant i et transférée ultérieurement vers un véhicule sortant j de la zone de stockage p

Les variables entières suivantes calculent les quantités de produits transférées entre les véhicules et les zones de stockage, y compris les quais d'expédition/réception et les zones de stockage depuis lesquelles les produits sont transférés :

h_{ijkl} Nombre de produits à transférer du véhicule entrant i au véhicule sortant j si le véhicule i est affecté au quai k et que le véhicule j est affecté au quai l

λ_{ikp} Nombre de produits à transférer du véhicule entrant i à la zone de stockage p si le véhicule entrant i est affecté au quai k

μ_{ijlp} Nombre de produits à transférer de la zone de stockage p au véhicule sortant j si le véhicule sortant j est affecté au quai l

2.3.2.3 Variables binaires

$$v_{ij} = \begin{cases} 1 & \text{S'il y a un transfert entre le véhicule entrant } i \text{ et le véhicule sortant } j \ (\alpha_{ijp} > 0) \\ 0 & \text{Otherwise} \end{cases}$$

$$w_{ip} = \begin{cases} 1 & \text{Si un produit a été transféré du véhicule entrant } i \text{ à la zone de stockage } p \\ 0 & \text{Otherwise} \end{cases}$$

$$\tau_{ijp} = \begin{cases} 1 & \text{Si un produit a été transféré du véhicule entrant } i \text{ à travers la zone de} \\ & \text{stockage } p \text{ vers le véhicule sortant} j \\ 0 & \text{Otherwise} \end{cases}$$

$$x_{ik} = \begin{cases} 1 & \text{Si le véhicule entrant } i \text{ est affecté au quai de réception} k \\ 0 & \text{Otherwise} \end{cases}$$

$$y_{jl} = \begin{cases} 1 & \text{Si le véhicule entrant } i \text{ est affecté au quai d'expédition} l \\ 0 & \text{Otherwise} \end{cases}$$

$$p_{im} = \begin{cases} 1 & \text{Si les véhicules entrants } i \text{ et } m \text{ sont affectés au même quai de réception et} \\ & i \text{ précède } m \\ 0 & \text{Otherwise} \end{cases}$$

$$q_{jn} = \begin{cases} 1 & \text{Si les véhicules sortants } j \text{ et } n \text{ sont affectés au même quai d'expédition et} \\ & j \text{ précède } n \\ 0 & \text{Otherwise} \end{cases}$$

$$z_{ijkl} = \begin{cases} 1 & \text{Si le véhicule entrant } i \text{ est affecté au quai } k \text{ et le véhicule sortant } j \text{ est} \\ & \text{affecté au quai } l \text{ et } v_{ij} = 1 \\ 0 & \text{Otherwise} \end{cases}$$

$$\delta_{ikp} = \begin{cases} 1 & \text{Si le véhicule entrant } i \text{ est affecté au quai } k \text{ et } w_{ip} = 1 \\ 0 & \text{Otherwise} \end{cases}$$

$$\gamma_{ijlp} = \begin{cases} 1 & \text{Si le véhicule sortant } j \text{ est affecté au quai } l \text{ et } \tau_{ijp} = 1 \\ 0 & \text{Otherwise} \end{cases}$$

$$\sigma_i = \begin{cases} 1 & \text{Si le véhicule entrant } i \text{ a stocké au moins un produit} \\ 0 & \text{Otherwise} \end{cases}$$

2.3.3 Fonction objective

Dans cette formulation mathématique, l'objectif est de minimiser le retard total des véhicules entrants et sortants. Ce retard est représenté par la différence entre l'heure de fin du déchargement (ou du chargement) et l'heure de départ prévue.

$$\text{Minimiser :} \qquad \sum_{i=1}^{N_1}(e_i - B_i) + \sum_{j=1}^{N_2}(f_j - D_j) \tag{2.1}$$

2.3.4 Contraintes

$$\sum_{j=1}^{N_2} \alpha_{ijp} + \sum_{j=1}^{N_2} \beta_{ijp} = Q_{ip} \qquad (\forall i = 1...N_1, \forall p = 1...P) \tag{2.2}$$

$$\sum_{i=1}^{N_1} \alpha_{ijp} + \sum_{i=1}^{N_1} \beta_{ijp} = R_{jp} \qquad (\forall j = 1...N_2, \forall p = 1...P) \tag{2.3}$$

$$v_{ij} \leq M \sum_{p=1}^{P} \alpha_{ijp} \qquad (\forall i = 1...N_1, \forall j = 1...N_2) \tag{2.4}$$

$$w_{ip} \leq M \sum_{j=1}^{N_2} \beta_{ijp} \qquad (\forall i = 1...N_1, \forall p = 1...P) \tag{2.5}$$

$$\tau_{ijp} \leq M\beta_{ijp} \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall p = 1...P) \tag{2.6}$$

$$\sum_{k=1}^{M_1} \sum_{l=1}^{M_2} h_{ijkl} = \sum_{p=1}^{P} \alpha_{ijp} \qquad (\forall i = 1...N_1, \forall j = 1...N_2) \tag{2.7}$$

$$\sum_{k=1}^{M_1} \lambda_{ikp} = \sum_{j=1}^{N_2} \beta_{ijp} \qquad (\forall i = 1...N_1, \forall p = 1...P) \tag{2.8}$$

$$\sum_{l=1}^{M_2} \mu_{ijlp} = \beta_{ijp} \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall p = 1...P) \tag{2.9}$$

$$h_{ijkl} \leq z_{ijkl} M \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall k = 1...M_1, \forall l = 1...M_2) \tag{2.10}$$

$$\lambda_{ikp} \leq \delta_{ikp} M \qquad (\forall i = 1...N_1, \forall k = 1...M_1, \forall p = 1...P) \tag{2.11}$$

$$\mu_{ijlp} \leq \gamma_{ijlp} M \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall l = 1...M_2, \forall p = 1...P) \tag{2.12}$$

$$\sum_{k=1}^{M_1} \delta_{ikp} = w_{ip} \qquad (\forall i = 1...N_1, \forall p = 1...P) \tag{2.13}$$

$$\sum_{l=1}^{M_2} \gamma_{ijlp} = \tau_{ijp} \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall p = 1...P) \tag{2.14}$$

$$\delta_{ikp} \leq x_{ik} \qquad (\forall i = 1...N_1, \forall k = 1...M_1, \forall p = 1...P) \tag{2.15}$$

$$\gamma_{ijlp} \leq y_{jl} \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall l = 1...M_2, \forall p = 1...P) \tag{2.16}$$

$$\sum_{k=1}^{M_1} x_{ik} = 1 \qquad (\forall i = 1...N_1) \tag{2.17}$$

$$\sum_{l=1}^{M_2} y_{jl} = 1 \qquad (\forall j = 1...N_2) \tag{2.18}$$

$$\sum_{k=1}^{M_1} \sum_{l=1}^{M_2} z_{ijkl} = v_{ij} \qquad (\forall i = 1...N_1, \forall j = 1...N_2) \tag{2.19}$$

$$z_{ijkl} \leq x_{ik} \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall k = 1...M_1, \forall l = 1...M_2) \tag{2.20}$$

$$z_{ijkl} \leq y_{jl} \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall k = 1...M_1, \forall l = 1...M_2) \tag{2.21}$$

$$x_{ik} + x_{mk} - 1 \leq p_{im} + p_{mi} \qquad (\forall i, m = 1...N_1, i \neq m, \forall k = 1...M_1) \tag{2.22}$$

$$p_{im} + p_{mi} \leq 1 \qquad (\forall i, m = 1...N_1) \tag{2.23}$$

$$y_{jl} + y_{nl} - 1 \leq q_{jn} + q_{nj} \qquad (\forall j, n = 1...N_2, j \neq n, \forall l = 1...M_2) \tag{2.24}$$

$$q_{jn} + q_{nj} \leq 1 \qquad (\forall j, n = 1...N_2) \tag{2.25}$$

$$\sum_{m=1}^{N_1} p_{im} \leq 1 \qquad (\forall i = 1...N_1) \tag{2.26}$$

$$\sum_{n=1}^{N_2} q_{jn} \leq 1 \qquad (\forall j = 1...N_2) \tag{2.27}$$

$$r_i \geq A_i \qquad (\forall i = 1...N_1) \tag{2.28}$$

$$r_m \geq e_i + E - M(1 - p_{im}) \qquad (\forall i, m = 1...N_1) \tag{2.29}$$

$$v_{ij} \leq (e_i - s_j)M \qquad (\forall i = 1...N_1, \forall j = 1...N_2) \tag{2.30}$$

$$d_i \geq r_i + L\sum_{j=1}^{N_2}\sum_{p=1}^{P}\alpha_{ijp} + L\sum_{p=1}^{P}\alpha_{ijp} + 2\sum_{k=1}^{M_1}\sum_{l=1}^{M_2}T_{kl}h_{ijkl} \qquad (\forall i = 1...N_1, \forall j = 1...N_2) \tag{2.31}$$

$$g_i \geq d_i + 2L\sum_{j=1}^{N_2}\sum_{p=1}^{P}\beta_{ijp} + 2\sum_{k=1}^{M_1}\sum_{p=1}^{P}U_{kp}\lambda_{ikp} - M(1 - \sigma_i) \qquad (\forall i = 1...N_1) \tag{2.32}$$

$$\sigma_i \leq M\sum_{j=1}^{N_2}\sum_{p=1}^{P}\beta_{ijp} \qquad (\forall i = 1...N_1) \tag{2.33}$$

$$e_i \geq r_i + L\sum_{j=1}^{N_2}\sum_{p=1}^{P}\alpha_{ijp} + L\sum_{p=1}^{P}\alpha_{ijp} + 2\sum_{k=1}^{M_1}\sum_{l=1}^{M_2}T_{kl}h_{ijkl} + 2L\sum_{j=1}^{N_2}\sum_{p=1}^{P}\beta_{ijp} + 2\sum_{k=1}^{M_1}\sum_{p=1}^{P}U_{kp}\lambda_{ikp} \qquad (\forall i = 1...N_1, \forall j = 1...N_2) \tag{2.34}$$

$$s_j \geq C_j \qquad (\forall j = 1...N_2) \tag{2.35}$$

$$s_n \geq f_j + E - M(1 - q_{jn}) \qquad (\forall j, n = 1...N_2) \tag{2.36}$$

$$s_j \geq r_i - M(1 - v_{ij}) \qquad (\forall i = 1...N_1, \forall j = 1...N_2) \tag{2.37}$$

$$f_j \geq r_i + L\sum_{p=1}^{P} \alpha_{ijp} + L\sum_{i=1}^{N_1}\sum_{p=1}^{P} \alpha_{ijp} + L\sum_{i=1}^{N_1}\sum_{p=1}^{P} \beta_{ijp} + 2\sum_{k=1}^{M_1}\sum_{l=1}^{M_2} T_{kl} h_{ijkl} - M(1 - v_{ij}) \tag{2.38}$$
$$(\forall i = 1...N_1, \forall j = 1...N_2)$$

$$f_j \geq s_j + L\left(\sum_{i=1}^{N_1}\sum_{p=1}^{P} \alpha_{ijp} + \sum_{i=1}^{N_1}\sum_{p=1}^{P} \beta_{ijp}\right) \qquad (\forall j = 1...N_2) \tag{2.39}$$

$$f_j \geq s_j + L\sum_{p=1}^{P} \beta_{ijp} + L\sum_{i=1}^{N_1}\sum_{p=1}^{P} \beta_{ijp} + 2\sum_{l=1}^{M_2} V_{lp}\mu_{ijlp} \qquad (\forall i = 1...N_1, \forall j = 1...N_2, \forall p = 1...P) \tag{2.40}$$

$$\begin{aligned} &r_i \geq 0, e_i \geq 0, t_i \geq 0, d_i \geq 0, g_i \geq 0, s_j \geq 0, f_j \geq 0, u_j \geq 0, \alpha_{ijp} \geq 0, \beta_{ijp} \geq 0, \\ &h_{ijkl} \geq 0, \lambda_{ikp} \geq 0, \mu_{ijlp} \geq 0, v_{ij}, w_{ip}, \tau_{ijp}, x_{ik}, y_{jl}, p_{im}, q_{jn}, z_{ijkl}, \delta_{ikp}, \gamma_{ijlp}, \sigma_i \in \{0, 1\} \\ &(\forall i = 1...N_1, \forall j = 1...N_2, \forall k = 1...M_1, \forall l = 1...M_2, \forall p = 1...P) \end{aligned} \tag{2.41}$$

2.3.5 Contraintes de flux

Les contraintes (2.2) et (2.3) garantissent l'équilibrage entre les flux entrants et les flux sortants. Les contraintes (2.4), (2.5) et (2.6) assurent que les variables binaires v_{ij}, w_{ip} et τ_{ijp} sont égales à zéro lorsque le flux de produit est nul. Les contraintes (2.7), (2.8) et (2.9) calculent les valeurs de h_{ijkl}, λ_{ikp} et μ_{ijlp} nécessaires pour connaître le flux direct entre deux véhicules α_{ijp} (ou le flux entre les quais et les différentes zones de stockage β_{ijp}) ainsi que l'affectation de ces véhicules entrants/sortants aux quais de réception/expédition. Les contraintes (2.10), (2.11) et (2.12) assurent que les variables binaires z_{ijkl}, δ_{ikp} et γ_{ijlp} sont égales à zéro lorsque les variables de flux/affectation h_{ijkl}, λ_{ikp} et μ_{ijlp} sont nulles.

2.3.5.1 Contraintes d'affectation et de séquencement

Les contraintes (2.13) à (2.16) définissent les valeurs des variables binaires δ_{ikp} et γ_{ijlp} à l'aide des variables binaires des zones de stockage w_{ip} et τ_{ijp} et les variables d'affectation des véhicules entrants/sortants x_{ik} et y_{jl}. Les contraintes (2.17) et (2.18) garantissent que chacun des véhicules entrants/sortants est affecté à un quai de réception/expédition. Les

contraintes (2.19), (2.20) et (2.21) calculent la valeur de z_{ijkl} en utilisant l'affectation des véhicules aux quais (x_{ik}, y_{jl}) et la variable binaire à flux direct v_{ij}. Les contraintes (2.22) à (2.27) calculent la séquence des véhicules sur les quais entrants/sortants (p_{im} et q_{jn}) en utilisant les affectations des véhicules aux quais (x_{ik}, y_{jl}).

2.3.6 Contraintes de planification

Les contraintes (2.28) et (2.35) garantissent que les véhicules entrants/sortants commencent à décharger / charger (r_i et s_j) une fois arrivés aux quais. Les contraintes (2.29) et (2.36) garantissent que, pour le même quai, l'heure de début du déchargement / chargement d'un véhicule entrant / sortant (r_i et s_j) s'effectue après la fin du déchargement / chargement (e_i et f_j) en plus du temps de changement E. Dans les contraintes (2.30), si un véhicule sortant a commencé à charger des produits avant la fin du déchargement du véhicule entrant, les produits sont transférés directement ($v_{ij} = 1$) au véhicule sortant au lieu de les stocker. Les contraintes (2.31) définissent la variable d_i qui représente la fin du transfert des produits directement des véhicules entrants aux véhicules sortants. Dans les contraintes (2.32), étant donné que la priorité (dans le cas des véhicules entrants) concerne le transfert direct des produits, le stockage de la quantité restante commence juste après la fin du transfert direct du produit : $g_i \geq d_i +$ *temps_de_stockage*. Pour les véhicules entrants, dans les deux opérations (transfert direct et stockage), une ressource de transport est utilisée puisque le stockage des produits et le transfert direct des produits restants sont deux phases différentes, elles sont exécutées en séquence et la priorité est donnée aux produits à transférer directement. Les contraintes (2.33) calculent la valeur de la variable binaire σ_i qui détermine s'il existe ou non un stockage pour le véhicule entrant i. Les contraintes (2.34) calculent l'heure de fin du déchargement e_i, qui inclut la durée du transfert direct et la durée du stockage. Dans les contraintes (2.37), s'il y a un transfert direct de produits entre le véhicule entrant i et le véhicule sortant j, le véhicule j commence le chargement immédiatement après l'heure de début r_i du déchargement du véhicule i. Les contraintes (2.38) calculent l'heure de fin f_j pour les produits transférés directement. es contraintes (2.39) garantissent que chaque véhicule sortant j quitte le quai après avoir chargé tous les produits transférés directement et via des zones de stockage. Les contraintes (2.40) calculent l'heure de fin f_j du chargement des véhicules sortants

à partir des zones de stockage. Les contraintes (2.41) garantissent que les variables de décision sont positives ou binaires.

2.4 Proposition de deux méta-heuristiques

Dans cette section, étant donné que le problème de planification des véhicules dans un cross-dock est NP-hard Boysen and Fliedner (2010), une heuristique, une recherche locale par insertion aléatoire (LS) et deux méta-heuristiques : la recherche taboue (TS) et le recuit simulé. (SA) sont proposés pour résoudre le problème considéré dans ce chapitre. L'heuristique est utilisée pour trouver une solution initiale, qui est améliorée par la suite à l'aide de la recherche locale par insertion aléatoire et des deux méta-heuristiques (Figure 2.4).

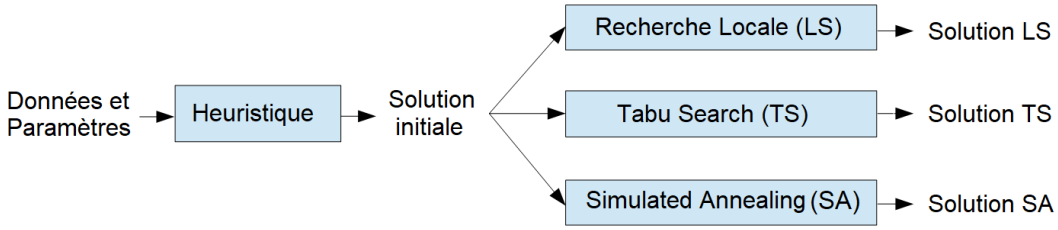

Figure 2.4 – Processus de résolution

2.4.1 Heuristique

Le processus de résolution de l'heuristique commence par une solution initiale générée à l'aide de l'heuristique suggérée (Algorithme 2.1). Tout d'abord, chaque véhicules entrants est affecté au premier quai de réception disponible de 1 à M_1, à partir du quai 1. Pour les véhicules sortants, ils sont affectés au premier quai d'expédition disponible de 1 à M_2. Ensuite, l'heuristique calcule les séquences des véhicules sur chacun des quais de réception/expédition. Après l'affectation des véhicules, l'étape suivante consiste à calculer le flux de produits : la quantité de produits à transférer d'un véhicule entrant i à un véhicule sortant j (ou à travers les zones de stockage), en affectant les produits arrivés des premiers véhicules entrants aux premiers véhicules sortants demandant le même produit. Si un véhicule sortant demande un produit déchargé par le véhicule entrant, la

quantité demandée est transférée directement au quai d'expédition. Dans le cas contraire, le produit est stocké dans l'une des zones de stockage.

Algorithme 2.1 : Algorithme de l'heuristique proposée

Entrée : données du problème ($A_i, B_i, C_j, D_j, Q_{ip}, R_{jp}, T_{kl}, U_{kp}$ *and* V_{pl})
Affecter tous les véhicules entrants aux quais de réception de 1 à M_1 :
Initialiser l'indice des véhicules entrants $k = 0$
Pour chaque véhicule entrant i **Faire**
  Si $k == M_1$ **Alors**
    $k = 0$
  Fin Si
  $x_{ik} = 1$
  $k++$
Fin Pour
Affecter tous les véhicules sortants aux quais d'expédition de 1 à $M - 2$:
Initialiser l'indice des véhicules sortants $l = 0$
Pour chaque véhicule sortant j **Faire**
  Si $l == M_2$ **Alors**
    $l = 0$
  Fin Si
    $y_{jl} = 1$
    $l++$
Fin Pour
Affecter les produits aux zones de stockage ou aux véhicules sortants :
Pour chaque véhicule entrant i **Faire**
  $p =$ type du produit à décharger du véhicule entrant
  Pour chaque véhicule sortant j **Faire**
    Si le véhicule j est arrivé au quai d'expédition et demande le produit déchargé **Alors**
      Transférer le produit directement du véhicule entrant i au véhicule j
    Fin Si
  Fin Pour
  Si Ils restent des produits à décharger **Alors**
    Stocker la quantité restante dans la zone correspondante au produit p
  Fin Si
Fin Pour
Calculer le début et la fin du déchargement des véhicules entrants (r_i et e_i)
Calculer le début et la fin du chargement des véhicules sortants (s_j et f_j)
Calculer le retard des véhicules entrants (t_i)
Calculer le retard des véhicules sortants (u_j)
Sortie : le retard total des véhicules entrants et sortants

L'étape suivante consiste à calculer l'heure de début et de fin du déchargement/ chargement des véhicules. Pour les véhicules entrants, si le véhicule i arrive avant que le véhicule $i-1$ termine le déchargement de tous ses produits, l'heure de début r_i pour le déchargement du véhicule i arrive juste après le déchargement de tous les produits du véhicule entrant précédent $i-1$ (Figure 2.3). Les véhicules entrants quittent le quai à l'instant e_i juste après le déchargement du dernier produit dans le véhicule. Le véhicule sortant j commence le chargement à l'instant s_j qui vient juste après f_{j-1} la fin du chargement du véhicule précédent $j-1$.

Les véhicules sortants quittent les quais après le chargement du dernier produit demandé. Lorsqu'un véhicule sortant j arrive à un quai et qu'un autre véhicule sortant est entrain de charger, le véhicule j doit attendre jusqu'à l'instant f_{j-1} (l'heure de fin du chargement du véhicule $j-1$). La dernière étape de l'heuristique consiste à calculer la fonction objective, qui est la somme des retards des véhicules entrants t_i et des véhicules sortants u_j en utilisant l'heure de fin du chargement/déchargement calculée précédemment et les heures de départ prévues. La description de l'heuristique est présentée dans l'algorithme 2.1.

2.4.1.1 Recherche locale par insertion aléatoire (LS)

L'objectif de cette recherche locale par insertion aléatoire (Algorithme 2.2) est d'améliorer la solution initiale générée par l'heuristique. La recherche locale commence par le chargement des données d'entrée et des affectations initiales générées par l'heuristique. Ensuite, l'algorithme calcule la fonction objective pour initialiser la variable BestSolution. À chaque itération, l'algorithme sélectionne un véhicule entrant aléatoire i (ou un véhicule sortant j) et le déplace vers un autre quai de réception (ou d'expédition). Un exemple de recherche locale est présenté dans la figure 2.5. Il montre l'affectation des véhicules entrants/sortants aux quais de réception/d'expédition, le véhicule sortant 3 est déplacé du quai d'expédition 1 au quai d'expédition 5.

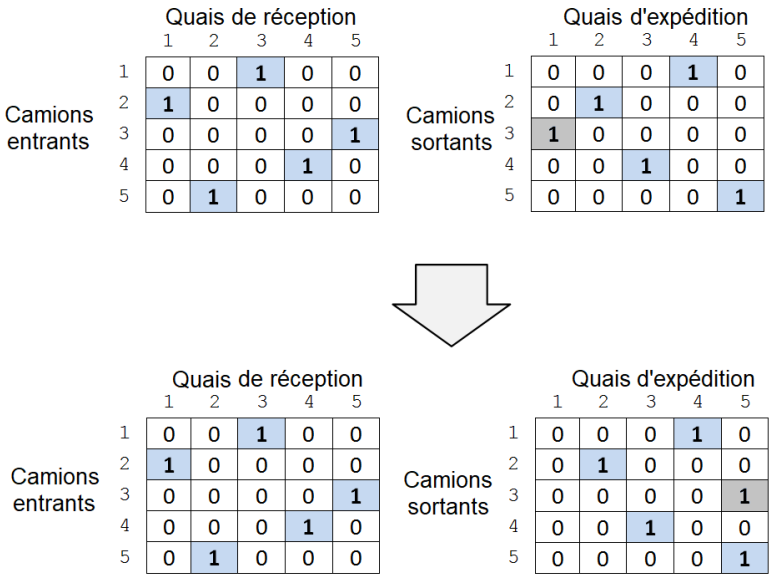

Figure 2.5 – Exemple d'un mouvement de recherche locale

L'étape suivante consiste à calculer la nouvelle valeur de la fonction objective correspondante à la nouvelle affectation. Si l'objectif est amélioré, cette solution devient la meilleure solution. Si le critère d'arrêt est atteint, la meilleure solution trouvée est retournée. Sinon, un nouveau véhicule entrant/sortant est sélectionné pour une nouvelle itération. Si aucune meilleure solution n'a été trouvée après un nombre maximal d'itérations (ou pour une durée limitée de temps), l'algorithme s'arrête et la meilleure solution trouvée est retournée.

Algorithme 2.2 : Algorithme de la recherche locale par insertion aléatoire (LS) proposé

Entrée : données du problème ($A_i, B_i, C_j, D_j, Q_{ip}, R_{jp}, T_{kl}, U_{kp}$ *and* V_{pl})
Entrée : la solution initiale générée par l'heuristique (*InitialSolution*)
Initialiser *BestSolution* = *InitialSolution*
CurrentIteration = 0
Tant Que (*CurrentIteration* < *MaxIterations*) et (*TimeLimit* non atteint) **Faire**
  Choisir aléatoirement un véhicule entrant i (ou véhicule sortant j)
  Déplacer le véhicule entrant i (ou véhicule sortant j) de son quai de réception initial k (ou quai d'expédition l) vers un quai différent $k2$ (ou $l2$) :
  Si le véhicule sélectionné est un véhicule entrant **Alors**
    $x_{ik} = 0$
    $x_{ik2} = 1$
  Sinon
    $y_{jl} = 0$
    $y_{jl2} = 1$
  Fin Si
  Calculer la fonction objective (retard total) de la nouvelle solution (*NewSolution*)
  Si *BestSolution* > *NewSolution* **Alors**
    BestSolution = *NewSolution*
  CurrentIteration = 0 (Réinitialiser les itérations)
  Sinon
    Déplacer le véhicule entrant i (ou véhicule sortant j) vers son quai initial
  Si le véhicule sélectionné est un véhicule entrant **Alors**
      $x_{ik} = 1$
      $x_{ik2} = 0$
    Sinon
      $y_{jl} = 1$
      $y_{jl2} = 0$
    Fin Si
  CurrentIteration ++
  Fin Si
Fin Tant Que
Sortie : *BestSolution*

2.4.2 Tabu Search (TS)

La recherche taboue (ou Tabu Search : TS) est une méta-heuristique qui a été introduite par Glover (1989). À chaque itération, le mécanisme de la TS évite les mouvements

de recherche locale qui se trouvent dans une tabu list qui est mise à jour en ajoutant les mouvements non améliorants. Un critère d'aspiration peut être défini pour améliorer le processus de recherche. Pour le problème considéré dans ce chapitre, le critère d'aspiration est présenté comme un écart entre la nouvelle solution trouvée lors d'une itération et la solution actuelle :

$$Deviation = 100 \times \left(\frac{NewSolution - CurrentSolution}{CurrentSolution} \right)$$

Au lieu de commencer par une solution initiale aléatoire, la TS classique est améliorée en utilisant la solution fournie par l'heuristique comme solution initiale.

L'algorithme du TS est composé de trois étapes (Algorithme 2.3). La première étape consiste à générer une solution initiale en utilisant l'heuristique. La deuxième étape consiste à générer une solution dans le voisinage de la solution actuelle en sélectionnant un véhicule entrant ou sortant de façon aléatoire, puis en l'affectant à un autre quai. Lors du changement de quai, les mouvements de recherche locale qui sont dans la tabu list sont évités. Dans la troisième et dernière étape, si la nouvelle solution satisfait le critère d'aspiration, la solution est considérée comme la meilleure solution actuelle. Sinon, la tabu list est mise à jour en ajoutant le nouveau mouvement. Ces trois étapes sont répétées jusqu'à ce qu'un critère d'arrêt soit satisfait. Pour le TS proposée, l'algorithme s'arrête si aucune meilleure solution n'a été trouvée après un nombre maximal d'itérations (ou après le dépassement de la limite de temps).

2.4.3 Simulated Annealing (SA)

Le recuit simulé (ou Simulated Annealing : SA) est une méta-heuristique de recherche locale capable d'explorer de nouvelles solutions en évitant les minima locaux Kirkpatrick et al. (1983). Dans la première étape, l'algorithme SA génère une solution initiale avec l'heuristique constructive précédemment présentée (2.1).

À chaque itération, un véhicule aléatoire entrant ou sortant est sélectionné pour être déplacé vers un autre quai. Ensuite, un nombre aléatoire r est généré ($0 \leq r \leq 1$). Dans la deuxième étape, une solution de voisinage est générée en affectant un véhicule aléatoire entrant ou sortant à un autre quai. Pour évaluer la nouvelle solution, l'algorithme

Algorithme 2.3 : Algorithme du tabu search (TS) proposé

Entrée : données du problème ($A_i, B_i, C_j, D_j, Q_{ip}, R_{jp}, T_{kl}, U_{kp}$ *and* V_{pl})
Entrée : la solution initiale générée par l'heuristique (*InitialSolution*)
Initialiser *BestSolution* = *InitialSolution*
CurrentIteration = 0
Initialiser la tabu list : *TabuList* = $\emptyset$
Tant Que (*CurrentIteration* < *MaxIterations*) et (*TimeLimit* non atteint) **Faire**
  Choisir aléatoirement un véhicule entrant i (ou véhicule sortant j)
  Déplacer le véhicule entrant i (ou véhicule sortant j) de son quai de réception initial k (ou quai d'expédition l) vers un quai différent $k2$ (or $l2$) :
  Le mouvement de recherche locale ne doit pas être dans la tabu list
  Si le véhicule sélectionné est un véhicule entrant **Alors**
    $x_{ik} = 0$
    $x_{ik2} = 1$
  Sinon
    $y_{jl} = 0$
    $y_{jl2} = 1$
  Fin Si
  Calculer la fonction objective (retard total) de la nouvelle solution (*NewSolution*)
  Vérifier si le critère d'aspiration est satisfait
  Si le critère d'aspiration est satisfait **Alors**
    Accepter la nouvelle solution
    Si *BestSolution* > *NewSolution* **Alors**
      BestSolution = *NewSolution*
      CurrentIteration = 0 (Réinitialiser les itérations)
    Sinon
      CurrentIteration ++
    Fin Si
  Sinon
  Déplacer le véhicule entrant i (ou véhicule sortant j) vers son quai initial
  Si le véhicule sélectionné est un véhicule entrant **Alors**
    $x_{ik} = 1$
    $x_{ik2} = 0$
  Sinon
    $y_{jl} = 1$
    $y_{jl2} = 0$
  Fin Si
  Mettre à jour la tabu list
  Ajouter le mouvement de recherche locale dans la tabu list
  CurrentIteration ++
  Fin Si
Fin Tant Que
Sortie : *BestSolution*

SA utilise un paramètre de température fictif T qui est diminué après chaque itération. Au début de l'algorithme, de nouvelles solutions sont acceptées, même avec des écarts importants. Au fur et à mesure que le paramètre T diminue, l'algorithme devient plus sélectif et n'accepte plus les solutions à grande déviation.

L'étape suivante consiste à décider, selon le paramètre de température T, soit de considérer la nouvelle solution comme meilleure solution actuelle, soit de passer à la

Algorithme 2.4 : Algorithme du Simulated Annealing (SA) proposé

Entrée : données du problème ($A_i, B_i, C_j, D_j, Q_{ip}, R_{jp}, T_{kl}, U_{kp}$ *and* V_{pl})
Entrée : la solution initiale générée par l'heuristique (*InitialSolution*)
Initialiser *BestSolution* = *InitialSolution*
Initialiser $T = 0$
Tant Que ($T > \epsilon$) et (*TimeLimit* non atteint) **Faire**
r = nombre aléatoire ($0 \leq r \leq 1$)
  Choisir aléatoirement un véhicule entrant i (ou véhicule sortant j)
  Déplacer le véhicule entrant i (ou véhicule sortant j) de son quai de réception initial k (ou quai d'expédition l) vers un quai différent $k2$ (or $l2$) :
  Si le véhicule sélectionné est un véhicule entrant **Alors**
    $x_{ik} = 0$
    $x_{ik2} = 1$
  Sinon
    $y_{jl} = 0$
    $y_{jl2} = 1$
  Fin Si
  Calculer la fonction objective (retard total) de la nouvelle solution (*NewSolution*)
  Si $r \leq exp(CurrentSolution - NewSolution)/T$ **Alors**
    Accepter la nouvelle solution
    Si *BestSolution* > *NewSolution* **Alors**
      BestSolution = *NewSolution*
    Fin Si
  Sinon
    Déplacer le véhicule entrant i (ou véhicule sortant j) vers son quai initial
    Si le véhicule sélectionné est un véhicule entrant **Alors**
      $x_{ik} = 1$
      $x_{ik2} = 0$
    Sinon
      $y_{jl} = 1$
      $y_{jl2} = 0$
    Fin Si
  Fin Si
  Diminuer T
Fin Tant Que
Sortie : *BestSolution*

prochaine itération, en utilisant la condition suivante :

$$r \leq exp(CurrentSolution - NewSolution)/T$$

L'algorithme continue jusqu'à ce que le critère d'arrêt soit satisfait ($T > \epsilon$ ou la limite de temps est dépassée). Le processus de la méta-heuristique SA adaptée est présenté dans (Algorithme 2.4).

2.5 Résultats expérimentaux

Cette section présente les résultats expérimentaux après avoir testé les méthodes de résolution proposées sur un ensemble d'instances générées. Aucun benchmark d'instances n'a été trouvé dans la littérature avec les mêmes contraintes et les mêmes données d'entrée/sortie, en particulier pour les contraintes de stockage. Toutefois, les méta-heuristiques proposées ont été adaptées pour tester les performances sur un benchmark d'instances de la littérature. Les résultats obtenus sur le benchmark d'instance sont présentés dans la section suivante.

Toutes les instances générées ont été testées à l'aide du solveur CPLEX (version 12.2). L'heuristique, la recherche locale et les deux méta-heuristiques (TS et SA) sont implémentées en C++ et tout les tests ont été réalisés sur un PC doté d'un processeur Intel$^{(R)}$ Core$^{(TM)}$ i3 de 2.40 GHz et 4GB de RAM. Les résultats obtenus avec l'heuristique, la recherche locale à insertion aléatoire et les deux méta-heuristiques (TS et SA) sont comparés à ceux obtenus par CPLEX.

La méta-heuristique TS et l'insertion aléatoire LS sont testées avec 200 000 itérations pour toutes les instances. Après 200 000 itérations, les algorithmes s'arrêtent si aucune nouvelle solution n'est trouvée. Une limite de temps de 60 secondes est également définie pour LS, TS et SA. Pour la SA, le critère d'arrêt est défini par le paramètre de température T (initialement : $T = 1{,}5$) qui est diminué de 3×10^{-5} à chaque itération. Ces valeurs ont été fixées après un réglage des paramètres. Les valeurs données sont celles qui assurent le meilleur compromis entre la performance et le temps de calcul. Afin d'évaluer la stabilité des méthodes de résolution proposées, la recherche locale LS et les deux méta-heuristiques (TS et SA) sont testées en 5 répétitions pour chacune des 30 instances ($5 \times 30 = 150$ tests) et la valeur moyenne de la fonction objective est présentée.

2.5.1 Instances

Afin de tester les méthodes proposées, 30 instances ont été générées en s'inspirant du cross-dock étudié et en combinant les paramètres suivants :

— Le nombre de quais de réception et d'expédition.

— Le nombre de véhicules entrants et sortants.

— Le nombre de types de produits gérés par le cross-dock.

— La synchronisation entre les véhicules entrants et sortants.

Tableau 2.1 – Paramètres des instances générées pour tester les algorithmes proposés

Catégorie	**Quais (Réception + Expédition)**	**véhicules**	**Types de produits**	**Sync.**
1	3	6 à 24	3 ou 5	Oui
	5	10 à 40	5 ou 10	Oui
	10	20 à 40	5 ou 10	Oui
2	3	6 à 24	3 ou 5	N/A
	5	10 à 40	5 ou 10	N/A
	10	20 à 40	5 ou 10	N/A

Les paramètres des instances sont résumés dans le tableau 2.1. Ils sont divisés en deux catégories de 15 instances. Les heures d'arrivée des véhicules entrants et sortants dans les 15 premières instances (la première catégorie) sont plus synchronisées que les 15 instances restantes (la deuxième catégorie). Pour les deux catégories, les capacités des véhicules sont considérées égales.

Chacune des deux catégories contient 3 ensembles de 5 instances. Le première groupe d'instances est généré pour un cross-dock avec 3 quais de réception et 3 quais d'expédition dans lesquels le nombre total de véhicules varie entre 6 et 24 véhicules (véhicules entrants et sortants), qui traitent 3 ou 5 types de produits différents. Le deuxième groupe contient 5 instances pour un cross-dock qui gère 5 ou 10 types de produits différents pour un cross-dock avec 5 quais de réception et 5 quais d'expédition et le nombre total de véhicules allant de 10 à 40. Dans le troisième groupe, il y a 5 cas de cross-dock avec 10 quais de réception et 10 quais d'expédition dans lesquels le nombre de véhicules varie entre 20 et 40 véhicules avec 5 ou 10 types de produits différents.

Pour toutes les instances, le nombre total de produits déchargés par les véhicules entrants dans le cross-dock est égal à la demande totale des véhicules sortants. La quantité de produits transportés par tous les véhicules entrants est égale à la capacité des véhicules. Cependant, les demandes des véhicules sortants pour chaque produit demandé restent aléatoires pour toutes les instances.

2.5.2 Analyse des résultats expérimentaux

2.5.2.1 Heuristique et CPLEX

Une limite de temps d'une heure (3600s) est appliquée à CPLEX pour toutes les instances. Puisque CPLEX prend beaucoup de temps pour trouver la solution optimale, la meilleure solution trouvée (lorsqu'elle existe) au bout d'une heure de temps est présentée.

Les mêmes instances ont été testées avec l'heuristique, la recherche locale à insertion aléatoire (LS) et les deux méta-heuristiques (TS et SA). Dans les trois premières colonnes du tableau 2.2, D représente le nombre de quais de réception et d'expédition. T est le nombre total de véhicules entrants et sortants. P est le nombre de produits gérés par le cross-dock. Les colonnes 5 et 10 du tableau 3 indiquent les résultats obtenus avec CPLEX et le temps de calcul respectivement (avec une limite de temps de 3600s). Les colonnes 6 et 11 du tableau 2.2 présentent les résultats de l'heuristique et son temps de calcul. La déviation de la fonction objective entre l'heuristique proposée et CPLEX est présentée dans la colonne 10 du tableau 2.3. Par exemple, l'écart entre le résultat heuristique H_i et le résultat CPLEX C_i pour l'instance i est calculé comme suit :

$$Dev(i) = 100 \times \left(\frac{H_i - C_i}{C_i}\right)$$

Comme on peut le voir dans la colonne 10 (tableau 2.3), l'heuristique est capable de trouver des solutions pour toutes les instances avec un écart moyen total de 18,12%. En raison de la complexité du modèle mathématique proposé, CPLEX a trouvé la solution optimale pour seulement 5 instances et des solutions non optimales pour 4 instances. Pour le reste des instances, CPLEX ne trouve aucune solution. L'heuristique proposée semble être très efficace en termes de temps de calcul avec une moyenne de 0,00136 secondes. Les solutions générées par l'heuristique sont considérées comme des solutions initiales.

Tableau 2.2 – Fonctions objectives et temps de calcul de LS, TS, SA et CPLEX

Instances				Fonction Objective					Temps de calcul (s)				
D	**P**	**T**	**Sync.**	**CPLEX**	**H**	**LS**	**TS**	**SA**	**CPLEX**	**H**	**LS**	**TS**	**SA**
3	3	6	Oui	835	835	835	835	835	2,68	0,00022	1,28	1,45	0,33
3	3	12	Oui	2110**	2399	2222,2	2042	2042	3600	0,00052	5,41	8,94	1,38
3	3	24	Oui	*	5726	4723,2	4575	4509	*	0,00122	28,88	44,23	7,16
3	5	10	Oui	2743**	3092	2523,4	2492	2492	3600	0,00042	3,82	4,30	0,97
3	5	20	Oui	*	8234	5981,2	5480,8	5414	*	0,00101	19,11	38,21	4,76
5	5	10	Oui	1529	2351	1994,4	1529	1529	102,79	0,00042	3,98	4,73	0,99
5	5	20	Oui	*	5010	3784	4022,6	3769	*	0,00118	19,87	25,17	4,93
5	5	30	Oui	*	6911	5348,4	6099,2	5088,8	*	0,00203	55,06	60,00	13,79
5	10	20	Oui	*	7773	4813,8	5026,4	4578,4	*	0,00106	22,43	33,73	5,67
5	10	40	Oui	*	26303,3	13883,5	17056,96	12708,74	*	0,00355	60,00	60,00	31,05
10	5	20	Oui	*	4809	3494,2	4159,8	3485,2	*	0,00089	20,84	37,14	5,20
10	5	30	Oui	*	6225,5	5158,1	6126,1	5196,2	*	0,00179	57,37	57,07	14,28
10	5	40	Oui	*	11340	7349,8	11150,8	7575,6	*	0,00359	60,00	60,00	30,13
10	10	20	Oui	*	5731	3567,2	3894,4	3304,8	*	0,00113	23,94	38,94	5,94
10	10	40	Oui	*	16745	8307,8	12170	7688,4	*	0,00336	60,00	60,00	32,80
3	3	6	N/A	1184	1231	1231	1228	1228	2,61	0,00019	1,25	1,38	0,32
3	3	12	N/A	2495**	2800	2450	2432	2432	3600	0,00043	5,28	7,93	1,33
3	3	24	N/A	*	6023	5461,6	5543,8	5379,2	*	0,00108	28,61	48,08	7,15
3	5	10	N/A	2621	3039	2820,8	2696	2696	14,24	0,00033	3,81	4,83	0,96
3	5	20	N/A	*	8279	7399	7070,6	6863,4	*	0,0008	19,28	32,07	4,84
5	5	10	N/A	2500	3331	2694	2645,6	2610,8	15,26	0,00027	3,95	6,23	0,99
5	5	20	N/A	*	5738	4950,6	5337,4	4887,6	*	0,00083	19,80	40,73	4,94
5	5	30	N/A	*	17028	16001	16462	15990,6	*	0,00157	55,44	60,00	13,89
5	10	20	N/A	*	6965,54	5929,788	6268,4	5929,86	*	0,00085	23,05	30,41	5,77
5	10	40	N/A	*	17170	14979,98	16465,1	14845,24	*	0,00307	60,00	60,00	32,38
10	5	20	N/A	*	8708,5	6583	7276,1	6487,8	*	0,00072	20,77	38,96	5,24
10	5	30	N/A	*	10364,5	7020,8	9136,3	7226,5	*	0,00144	57,94	60,00	14,46
10	5	40	N/A	*	15587,5	10217,9	14038,9	10629,4	*	0,00265	60,00	60,00	30,56
10	10	20	N/A	7785**	9152	7372	8308,4	7360,6	3600	0,00097	23,96	36,05	6,07
10	10	40	N/A	*	19456	15582,6	19031,8	16273,8	*	0,00313	60,00	60,00	33,47
Temps de calcul moyen (s) :									1615.3	0.00136	29.50	36.02	10.73

* CPLEX ne trouve aucune solution pour ces instances.

** Meilleures solutions trouvées par CPLEX dans 3600s (Non optimales).

2.5.2.2 Résultats de la recherche locale (LS) et des deux méta-heuristiques (TS et SA)

L'étape suivante consiste à améliorer les solutions initiales générées par l'heuristique à l'aide de la recherche locale à insertion aléatoire (LS) et des deux méta-heuristiques (TS et SA).

Le tableau 2.2 présente les résultats obtenus pour chacune des méta-heuristiques. Les résultats d'insertion aléatoire LS sont présentés dans la colonne 7 et le temps de calcul, dans la colonne 12, avec une moyenne de 29,50 secondes. Pour la TS, le temps de calcul est présenté dans la colonne 13 avec une moyenne de 36,02 secondes. Comme présenté dans la colonne 14, le temps de calcul de SA est plus court, avec une moyenne de 10,73 secondes.

2.5.2.3 Comparaison entre les méta-heuristiques proposées

Comme présenté dans le tableau 2.3, l'insertion aléatoire LS et les méta-heuristiques (TS et SA) ont amélioré la solution initiale de l'heuristique. L'écart par rapport aux résultats de CPLEX est réduit de 18,12% à 4,44% pour l'insertion aléatoire LS et de 18,12% à 0,47% pour la TS. Au cours des expérimentations, même si la solution initiale est améliorée, le point faible de la recherche locale à insertion aléatoire (LS) est qu'elle peut être bloquée dans des minima locaux, ce qui empêche d'explorer de nouveaux espaces de solutions. La SA a permis de réduire l'écart de 18,12% à -1,04% en un temps de calcul plus court.

Tableau 2.3 – Comparaison de LS, TS at SA avec les résultats de CPLEX

Instances				**Fonction Objective**					**Dev. (%) vs. CPLEX**			
D	**P**	**T**	**Sync.**	**CPLEX**	**H**	**LS**	**TS**	**SA**	**H**	**LS**	**TS**	**SA**
3	3	6	Oui	835	835	835	835	835	0,00%	0,00%	0,00%	0,00%
3	3	12	Oui	2110**	2399	2222,2	2042	2042	13,70%	5,32%	−3,22%	−3,22%
3	3	24	Oui	*	5726	4723,2	4575	4509	*	*	*	*
3	5	10	Oui	2743**	3092	2523,4	2492	2492	12,72%	−8,01%	−9,15%	−9,15%
3	5	20	Oui	*	8234	5981,2	5480,8	5414	*	*	*	*
5	5	10	Oui	1529	2351	1994,4	1529	1529	53,76%	30,44%	0,00%	0,00%
5	5	20	Oui	*	5010	3784	4022,6	3769	*	*	*	*
5	5	30	Oui	*	6911	5348,4	6099,2	5088,8	*	*	*	*
5	10	20	Oui	*	7773	4813,8	5026,4	4578,4	*	*	*	*
5	10	40	Oui	*	26303,3	13883,5	17056,96	12708,74	*	*	*	*
10	5	20	Oui	*	4809	3494,2	4159,8	3485,2	*	*	*	*
10	5	30	Oui	*	6225,5	5158,1	6126,1	5196,2	*	*	*	*
10	5	40	Oui	*	11340	7349,8	11150,8	7575,6	*	*	*	*
10	10	20	Oui	*	5731	3567,2	3894,4	3304,8	*	*	*	*
10	10	40	Oui	*	16745	8307,8	12170	7688,4	*	*	*	*
3	3	6	N/A	1184	1231	1231	1228	1228	3,97%	3,97%	3,72%	3,72%
3	3	12	N/A	2495**	2800	2450	2432	2432	12,22%	−1,80%	−2,53%	−2,53%
3	3	24	N/A	*	6023	5461,6	5543,8	5379,2	*	*	*	*
3	5	10	N/A	2621	3039	2820,8	2696	2696	15,95%	7,62%	2,86%	2,86%
3	5	20	N/A	*	8279	7399	7070,6	6863,4	*	*	*	*
5	5	10	N/A	2500	3331	2694	2645,6	2610,8	33,24%	7,76%	5,82%	4,43%
5	5	20	N/A	*	5738	4950,6	5337,4	4887,6	*	*	*	*
5	5	30	N/A	*	17028	16001	16462	15990,6	*	*	*	*
5	10	20	N/A	*	6965,54	5929,788	6268,4	5929,86	*	*	*	*
5	10	40	N/A	*	17170	14979,98	16465,1	14845,24	*	*	*	*
10	5	20	N/A	*	8708,5	6583	7276,1	6487,8	*	*	*	*
10	5	30	N/A	*	10364,5	7020,8	9136,3	7226,5	*	*	*	*
10	5	40	N/A	*	15587,5	10217,9	14038,9	10629,4	*	*	*	*
10	10	20	N/A	7785**	9152	7372	8308,4	7360,6	17,56%	−5,31%	6,72%	−5,45%
10	10	40	N/A	*	19456	15582,6	19031,8	16273,8	*	*	*	*
Déviation moyenne (%) :									18.12%	4.44%	0.47%	−1.04%

* CPLEX ne trouve aucune solution pour ces instances.

** Meilleures solutions trouvées par CPLEX dans 3600s (Non optimales).

2.6 Comparaison avec des instances benchmark de la littérature

Afin de prouver l'efficacité des méta-heuristiques proposées, nous les avons comparées à d'autres de la littérature. Comme nous n'avons pas trouvé exactement le même problème dans la littérature, l'heuristique proposée et l'algorithme SA présenté précédemment ont été légèrement adaptées pour résoudre le premier groupe d'instances benchmark utilisé dans les travaux de Van Belle et al. (2013) ($270 \times 3 = 810$ instances). Les résultats de l'algorithme SA proposé sont comparés à la Tabu Search (TS1) présentée dans leur travail (noté TS1 (VB) dans la suite). Comme décrit dans Van Belle et al. (2013), les instances de benchmark ont été générées en combinant trois paramètres :

— Nombre de véhicules entrants et sortants.
— Routage de flux (matrice de flux).
— Fenêtres horaires (heure d'arrivée et de départ des véhicules entrants et sortants).

La fonction objective de leur modèle mathématique était de minimiser les retards totaux et le temps de transfert des produits à l'intérieur du cross-dock présentés sous forme d'une fonction multi-objectifs à deux facteurs de pondération (w_1 et w_2). Plus de détails sur les instances benchmark peuvent être trouvé dans Van Belle et al. (2013).

Afin de pouvoir tester notre algorithme SA, le modèle MILP proposé est adapté aux instances de référence en ajustant certaines contraintes et variables (types de produits, matrices de stockage et de flux). La quantité de produits déchargés par les véhicules des fournisseurs (Q_{ip}), les demandes des clients (R_{jp}) et les matrices de flux (α_{ijp} et β_{ijp}) sont remplacées par la matrice de flux d'entrée f_{ij} de Van Belle et al. (2013), notre modèle est donc réduit pour correspondre au mieux à leur modèle.

Avant de tester la SA proposée sur ces instances, la température initiale est fixée à $T = 1.5$ mais elle est diminuée de 1×10^{-6} à chaque itération. Afin de tester la stabilité de la méta-heuristique SA, chacune de ces instances est testée avec 5 réplications, soit un total de 4050 tests ($5 \times 270 \times 3 = 4050$). La moyenne, le minimum, le maximum et l'écart type sont présentés pour les 5 réplications de chacune de ces instances.

Les résultats des 810 instances sont présentés dans l'Annexe A. Les trois premières colonnes présentent les valeurs (L : faible, M : moyen et H : élevé) des trois paramètres précédents respectivement. Dans la quatrième colonne, chacune des valeurs présente la valeur moyenne de l'objectif pour 10 instances obtenues avec TS1 (VB) dans Van Belle et al. (2013). La cinquième colonne présente l'objectif moyen pour 10 instances pour notre méta-heuristique SA proposée (la moyenne de 5 réplications). Les colonnes 6, 7 et 8 donnent respectivement le minimum, le maximum et l'écart type pour les 5 réplications. La colonne 9 indique l'écart entre la SA et la TS1 (VB). Les deux dernières colonnes présentent le temps de calcul de SA et TS1 (VB). Les résultats détaillés sont présentés dans l'Annexe A.

Comme présenté dans le tableau A.1 de l'Annexe A, pour la première combinaison de facteurs de pondération ($w_1 = 1$ et $w_2 = 2$), l'écart moyen est de 0,303% par rapport à TS1 (VB) avec un temps de calcul moyen de 2,727s. Dans la deuxième combinaison (tableau A.2 de l'Annexe A) des facteurs de pondération ($w_1 = 1$ et $w_2 = 5$), l'écart moyen est de 0,028% avec un temps de calcul moyen de 2,637 secondes. Pour la troisième combinaison de facteurs de pondération ($w_1 = 1$ et $w_2 = 10$), la méta-heuristique SA proposée donne de meilleurs résultats avec un écart moyen de -0,035% dans une durée de calcul moyenne de 2,461s (tableau A.3 de l'Annexe A). L'écart moyen total entre le SA proposé et le TS1 (VB) pour toutes les instances est de 0,099%.

2.6.1 Analyse statistique

Même si les résultats obtenus valident l'efficacité et la stabilité de la méta-heuristique SA proposée en termes de convergence et de temps de calcul en générant des solutions avec un écart moyen total de 0,099% (Figure 4.8), nous calculons l'intervalle de confiance (CI) des écarts entre SA et TS1 (VB1) comme suit :

$$CI = [I_{MIN}; I_{MAX}] = \left[\overline{X} - Z\,\frac{\sigma}{\sqrt{n}};\ \overline{X} + Z\,\frac{\sigma}{\sqrt{n}}\right]$$

Où : $X = 0{,}099\%$ est la moyenne totale des écarts entre SA et TS1 (VB). $n = 81$ est le nombre d'instances (810 instances regroupées par groupes de 10 instances). $\sigma = 0.003$ est l'écart type de tous les écarts. Enfin, la valeur Z dépend de l'intervalle de confiance

(CI) choisi. On peut voir que les intervalles de confiance sur le tableau 2.4 confirment l'efficacité de la méta-heuristique SA proposée.

Tableau 2.4 – Comparaison des intervalles de confiance

$CI(\%)$	Z	I_{MIN}	I_{MAX}
95%	1,96	0,09835%	0,09965%
99%	2,576	0,09814%	0,09986%

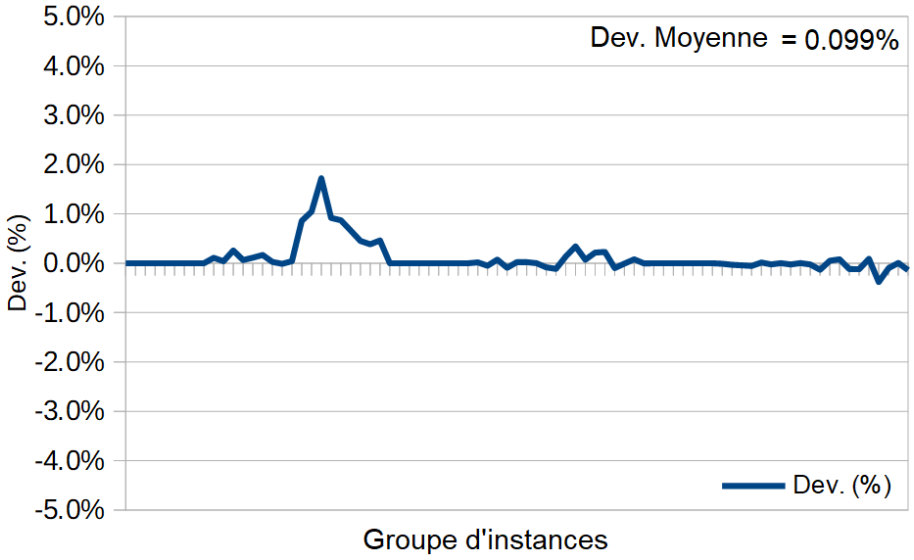

Figure 2.6 – Déviations entre SA et TS1(VB) pour les 810 instances (groupées par 10 instances)

La validation des algorithmes proposés sur le cross-dock étudié est présentée avec un exemple illustratif détaillé dans l'Annexe B. Cet exemple présente la solution de l'instance 6 (obtenue par l'algorithme SA) en la comparant à l'une des réplications de la simulation.

2.7 Conclusion

Ce chapitre a été dédié à l'étude des performances des cross-docks classiques avant de passer au PI-cross-docks (PI-hubs) dans les chapitres suivants. Tout d'abord, un modèle mathématique a été proposé pour la planification des véhicules dans un cross-dock à quais multiples et plusieurs zones de stockage gérant plusieurs types de produits. Des méta-heuristiques ont été proposées en évaluant leurs performances sur des instances de la littérature. L'algorithme SA a permet de minimiser le retard par -38,86% par rapport à la méthode d'affectation manuelle. Les algorithmes proposés vont assister le manager du cross-dock et les opérateurs des quais dans la planification des véhicules et le routage des

produits. Le travail présenté dans ce deuxième chapitre a fait l'objet d'une publication dans une revue internationale : Chargui et al. (2020).

Les données d'entrée des modèles présentés dans ce chapitre sont déterministes. Le but du chapitre suivant sera d'étudier le passage du cross-dock classique au PI-hub via des études de robustesses par simulation en utilisant le modèle MILP présenté dans ce chapitre et aussi par couplage simulation-optimisation en intégrant le TS, présenté également dans ce chapitre, avec un simulateur de perturbations qui permettra de générer des solutions robustes.

Chapitre 3

Du cross-dock classique au PI-hub : Etude de robustesse

Ce chapitre est consacré au passage du cross-dock classique au PI-hub via une étude par simulation pour évaluer la robustesse des deux cross-docks à travers plusieurs scénarios. Ensuite, une approche de couplage simulation-optimisation est proposée pour générer des solutions robustes face aux éventuelles perturbations au niveau des ressources de transport internes.

3.1 Introduction

Le but de ce chapitre est d'évaluer l'apport de l'Internet Physique sur le cross-dock classique étudié dans le chapitre 2 à travers deux approches différentes : Simulation à événements discrets et le couplage Simulation-Optimisation.

La première partie de ce chapitre présente un ensemble de scénarios de simulation afin de comparer les performances du cross-dock avec celles du PI-hub et d'évaluer la contribution de la mise en œuvre de l'Internet Physique en cross-docking. Deux modèles de simulation sont proposés pour comparer les performances du cross dock classique à celles du PI-hub sous le même flux de produits et le même intervalle entre les arrivées des véhicules entrants et sortants. Plusieurs scénarios sont présentés pour tester la robustesse en modifiant la synchronisation et la distribution du temps entre les arrivées. Afin de comparer les performances du cross-dock à celles du PI-hub, quelques indicateurs de performance (KPI) sont pris en compte, tels que le temps d'attente, le nombre de véhicules en attente, l'utilisation des ressources et la durée de cycle des produits.

Le modèle proposé dans le chapitre 2 fournit des solutions optimales dans un environnement déterministe qui ne seront plus valables en cas de perturbation. Pour répondre à ce problème, dans la deuxième partie de ce chapitre, nous proposons une approche de couplage simulation-optimisation. La méta-heuristique TS (Tabu Search), proposée dans le chapitre précèdent, sera combinée avec un simulateur de perturbations pour résoudre le problème de planification des véhicules pour le cross-dock classique et le PI-hub. L'objectif est de minimiser le retard total des véhicules entrants et sortants. Étant donné que dans la réalité, les ressources de transport internes peuvent être non disponibles à cause des pannes, un mécanisme de test de robustesse est présenté pour évaluer la robustesse des solutions de TS en générant des pannes de ressources de transport internes avec des probabilités différentes pour le cross-dock classique et le PI-hub. L'impact de la défaillance des ressources de transport internes sur les performances du cross-dock et du PI-hub est comparé.

3.2 Etude de la robustesse du cross-dock classique et du PI-hub

3.2.1 Vue d'ensemble sur le cross-dock classique

Dans ce modèle de simulation, le cross-dock étudié dans le chapitre précèdent est considéré comme un cas d'étude. Le centre de cross-docking a 5 quais de réception et 5 quais d'expédition permettant de gérer 5 types différents d'appareils électroménagers : (types 1, 2, 3, 4 et 5) provenant de cinq fournisseurs différents.

Le modèle considère une zone de stockage intermédiaire (figure 3.1) séparée en 5 zones (une zone de stockage pour chacun des cinq produits). Pour les ressources, 5 chariots élévateurs sont pris en compte (l'utilisation des chariots élévateurs atteint 70% pour une journée avec un nombre d'arrivées élevé : 50 véhicules entrants et 50 véhicules sortants). Pendant la simulation, les chariots élévateurs sont affectés dynamiquement aux quais de réception/expédition et aux zones de stockage. Par exemple, si un chariot élévateur est non utilisé, il sera automatiquement affecté au quai ou à la zone de stockage la plus proche.

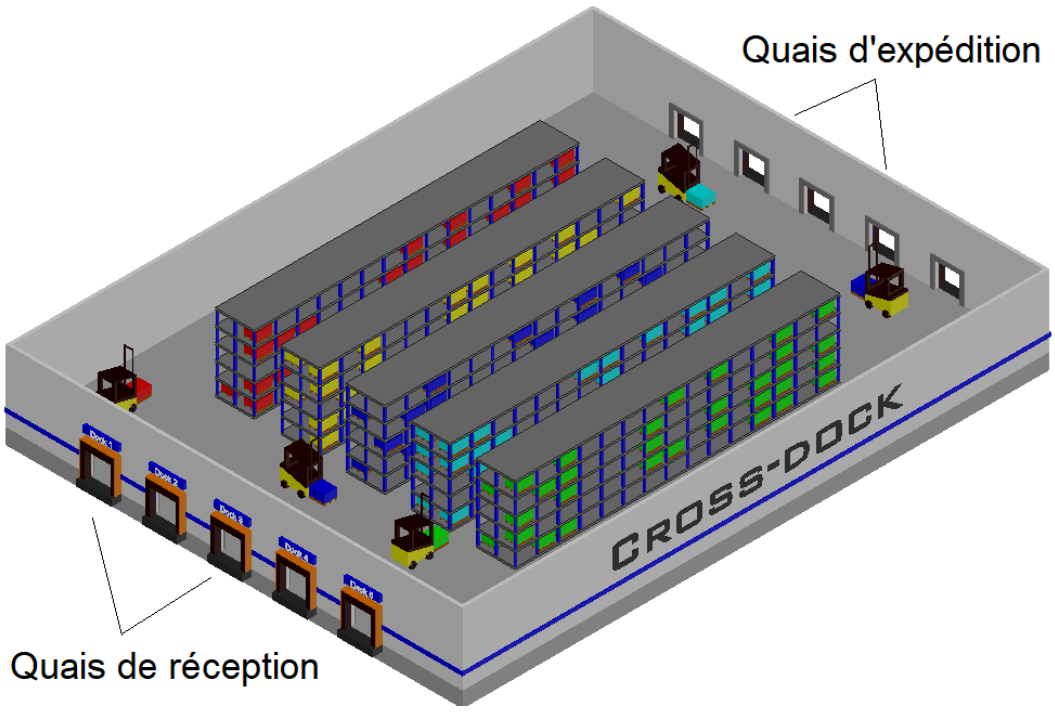

Figure 3.1 – Vue d'ensemble sur le cross-dock classique

La quantité de produits transportés par un véhicule entrant est égale à 50, ce qui correspond à la capacité totale du véhicule pour ce modèle de simulation. Les produits sont déchargés à l'aide des chariots élévateurs disponibles, puis transférés vers les quais sortants si un véhicule le demande ou vers les zones de stockage en attendant l'arrivée des véhicules sortants.

Pour les quais sortants, les véhicules arrivent sur chaque quai avec une demande aléatoire de produits différents. Afin de respecter la règle FIFO, les chariots élévateurs doivent d'abord récupérer les produits de l'entrepôt. Toutefois, si la quantité de produits dans la zone de stockage n'est pas suffisante et qu'un véhicule décharge ce type de produit sur les quais de réception, la quantité demandée sera récupérée directement à partir de ce véhicule sans passer par la zone de stockage.

Ce modèle de simulation prend également en compte les pauses des conducteurs des chariots élévateurs (20 minutes de pause toutes les 4h). Les autres paramètres de simulation sont détaillés dans le tableau 3.1.

Tableau 3.1 – Paramètres de la simulation

Paramètres		**Valeurs**
Nombre de quais	Quais de réception	5
	Quais d'expédition	5
Nombre des types de produits		5
Capacité de véhicules		50
Nombre de produits demandés par les véhicules sortants		Aléatoire
Temps moyen de chargement et de déchargement		20 s
La vitesse moyenne utilisée pour les chariots élévateurs		7 km/h
Surface du cross-dock		6000 m^2
Nombre de véhicules servis en 24h (journée normale)	Quais de réception	25
	Quais d'expédition	25
Nombre de véhicules servis en 24h (journée chargée)	Quais de réception	50
	Quais d'expédition	50
Durée des pauses des opérateurs de chariots élévateurs		20 min
Temps entre le début de deux pauses successives		4 h
Temps entre chaque deux arrivées avec synchronisation (minutes)	Journée normale	TRIA(230,290,350)
	Journée chargée	TRIA(115,145,175)
Temps entre chaque deux arrivées sans synchronisation (minutes)	Journée normale	EXPO(120)
	Journée chargée	EXPO(60)

3.2.2 Vue d'ensemble sur le PI-hub

Pour le modèle de simulation du cross-dock PI-hub, les durées entre les arrivées, le nombre de quais et la surface du cross-dock ainsi que les distances entre les quais et les zones de stockage sont identiques à celles du modèle du cross-dock classique pour tous les scénarios. Toutefois, au lieu d'utiliser des chariots élévateurs, on considère un chargement et un déchargement automatisés via des PI-docks connectés à la zone de stockage en utilisant un réseau flexible de PI-conveyors (vitesse moyenne de 10 km/h). De plus, un système de stockage et de récupération automatisé (AS / RS : Automated Storage and Retrieval System) dans la zone de stockage est connecté au PI-sorter dans la zone de tri. La figure 3.2 présente une vue d'ensemble sur le PI-hub.

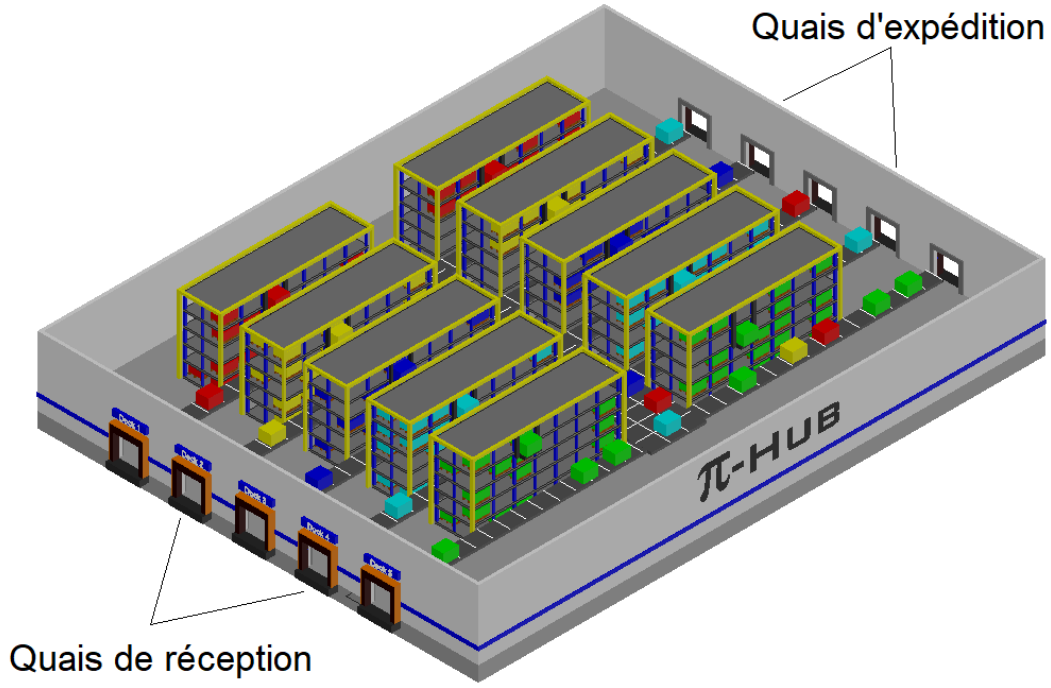

Figure 3.2 – Vue d'ensemble sur le PI-hub (PI-cross-dock)

3.2.3 Présentation des scénarios de simulation

Les deux modèles de simulation sont développés à l'aide du logiciel ARENA Simulation (version 13.50) Kelton et al. (2000). Les modèles de simulation ont été exécutés pendant 24 heures, avec 10 réplications pour chaque scénario. Deux distributions pour le temps entre chaque deux arrivées de véhicules entrants et sortants sont prises en compte : TRIA (Min, Mode, Max) pour les scénarios synchronisés et EXPO (moyenne) pour les scénarios

non synchronisés. Les valeurs générées par la distribution triangulaire sont davantage contrôlées à l'aide des paramètres Min, Max et Mode, qui retournent des valeurs proches du paramètre Mode. Pour la distribution exponentielle, les valeurs générées sont moins contrôlées car elles prennent uniquement le paramètre Moyenne.

Tableau 3.2 – Distribution des temps entre chaque deux arrivées

	Temps entre chaque deux arrivées (en minutes)			
	TRIA			**EXPO**
	Min	**Mode**	**Max**	**Moyenne**
Journée chargée	115	145	175	60
Journée normale	230	290	350	120

Pour chaque distribution de temps entre deux arrivées, 2 scénarios sont envisagés pour chaque distribution : le premier considère une journée normale dans le cross-dock avec 25 véhicules entrants et 25 véhicules sortants ; le second scénario considère une journée chargée avec 50 véhicules entrants et 50 véhicules sortants. Tous les paramètres de distribution utilisés dans ces scénarios sont résumés dans le tableau 3.2.

Dans ce travail, quatre indicateurs de performance sont pris en compte. Premièrement, le temps d'attente moyen et maximal passé par les véhicules entrants et sortants aux quais pour le déchargement/l'expédition de produits. Pour les véhicules entrants, le temps d'attente est généralement lié à la disponibilité des chariots élévateurs. Pour les véhicules sortants, le temps d'attente dépend de la disponibilité des chariots élévateurs et de la disponibilité des produits qui est liée aux retards des véhicules entrants. Le deuxième indicateur de performance est le nombre de véhicules entrants et sortants en attente, ce qui est un indicateur important pour la conception des cross-docks, en particulier pour estimer la surface des zones d'attente dans les quais. Le troisième indicateur de performance correspond au temps de cycle passé par un produit dans le cross-dock. Le dernier indicateur de performance est le taux d'utilisation des ressources, qui comprend les chariots élévateurs pour les cross-docks et des convoyeurs pour le PI-hub.

3.2.4 Analyse des résultats de la simulation

Plusieurs indicateurs de performances sont pris en considération pour évaluer la robustesse des deux cross-docks, à savoir :

— Le temps d'attente des véhicules aux quais de réception/expédition
— Le nombre de véhicules en attente dans les quais de réception/expédition
— Le temps de cycle des produits dans le cross-dock
— Le taux d'utilisation des ressources de transport internes

3.2.4.1 Le temps d'attente des véhicules

Le temps d'attente pris en compte dans cette étude comprend le temps passé par le véhicule pour décharger ou charger les produits et le temps d'attente aux quais. Le tableau 3.3 montre l'écart entre le temps d'attente moyen et maximal dans le cross-dock et le PI-hub. Pour chaque scénario, l'écart entre le PI-hub et le cross-dock est présenté à la troisième colonne. La dernière colonne présente la moyenne totale des valeurs de la déviation moyenne. La déviation est améliorée en particulier pour le scénario non synchronisé des journées chargées qui montre la robustesse du réseau de PI-conveyors en cas de perturbation par rapport aux chariots élévateurs partagés avec les quais entrants et sortants et la zone de stockage.

Tableau 3.3 – Distribution des temps entre chaque deux arrivées

Synchronisation des véhicules			**Temps d'attente dans les quais**				
			Quais de réception		**Quais d'expédition**		**Moyenne**
			Moyenne	**Max**	**Moyenne**	**Max**	**Totale**
Sync,	**Journée chargée**	**Cross-dock**	42,7	120,0	83,9	339,8	
		PI-hub	16,5	98,3	16,5	323,8	
		Dev, (%)	-61,4%	-18,1%	-80,3%	-4,7%	-70,9%
	Journée normale	**Cross-dock**	36,4	178,5	65,1	373,8	
		PI-hub	16,5	161,2	16,5	345,2	
		Dev, (%)	-54,7%	-9,7%	-74,6%	-7,7%	-64,7%
Non-Sync,	**Journée chargée**	**Cross-dock**	154,4	414,0	358,5	991,2	
		PI-hub	19,3	174,2	30,2	489,8	
		Dev, (%)	-87,5%	-57,9%	-91,6%	-50,6%	-89,6%
	Journée normale	**Cross-dock**	51,1	331,0	126,4	862,4	
		PI-hub	17,7	297,3	21,2	796,5	
		Dev, (%)	-65,3%	-10,2%	-83,3%	-7,6%	-74,3%
						Moyenne	**-74,8%**

Le graphe suivant (Figure 3.3) résume les écarts moyens du temps d'attente entre le PI-hub et le cross-dock :

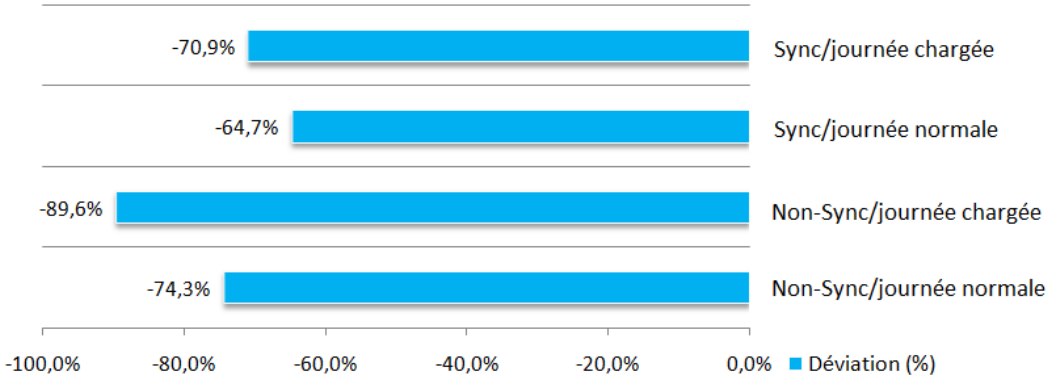

Figure 3.3 – Déviation Moyenne des temps d'attente des véhicules aux quais

3.2.4.2 Le nombre de véhicules en attente

Le tableau 3.4 présente le nombre moyen et maximal de véhicules en attente aux quais d'arrivée et de départ pour tous les scénarios présentés précédemment. Comme le montre le tableau 3.4, le nombre de véhicules entrants et sortants en attente aux quais a été considérablement réduit dans la configuration du PI-hub avec une déviation total moyen de -42,13%, en particulier pendant les journées chargées et lorsque les heures entre les arrivées des véhicules ne sont pas synchronisées.

Tableau 3.4 – Nombre de véhicules en attente dans les quais de réception/expédition

Synchronisation des véhicules			Nombre de véhicules en attente dans les quais				
			Quais de réception		Quais d'expédition		Moyenne
			Moyenne	Max	Moyenne	Max	Totale
Sync.	Journée chargée	Cross-dock	0,3	0,8	1	2,1	
		PI-hub	0,1	0,7	1	2	
		Dev. (%)	-60,50 %	-16,40 %	-2,00 %	-6,50 %	-31,25 %
	Journée normale	Cross-dock	0,1	0,7	1	1,6	
		PI-hub	0,1	0,6	1	1,6	
		Dev. (%)	-53,90 %	-8,50 %	0,00 %	-3,70 %	-26,95 %
Non-Sync.	Journée chargée	Cross-dock	1,3	3,6	4,4	7,6	
		PI-hub	0,2	1,9	2,2	4,9	
		Dev. (%)	-84,60 %	-47,70 %	-50,50 %	-34,90 %	-67,55 %
	Journée normale	Cross-dock	0,2	1,5	1,9	3,4	
		PI-hub	0,1	1,4	1,5	3,1	
		Dev. (%)	-65,30 %	-10,00 %	-20,20 %	-7,10 %	-42,75 %
						Moyenne	**-42,13%**

Le graphe dans la figure 3.4 résume les écarts moyens entre le PI-hub et le cross-dock :

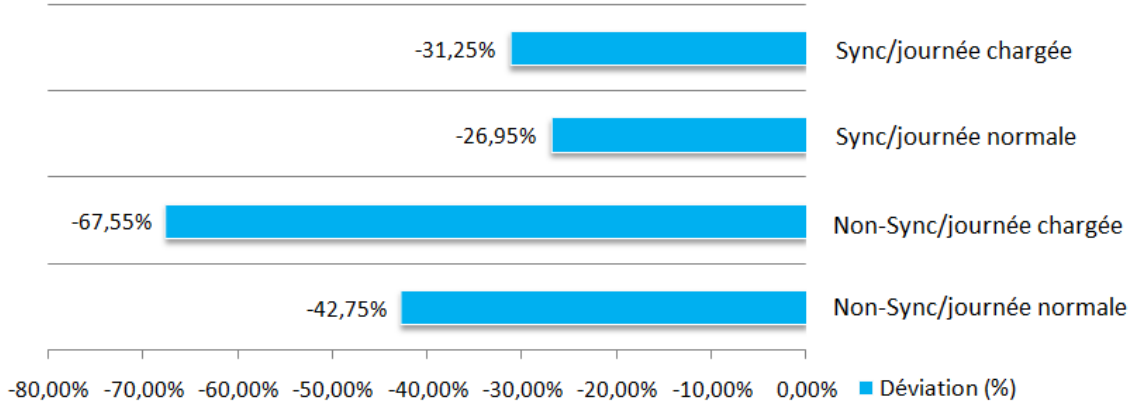

Figure 3.4 – Déviation moyenne du nombre de véhicules en attente dans les quais

3.2.4.3 Le temps de cycle des produits

La durée totale du temps de cycle dans le cross-dock commence lorsque le véhicule arrive au quai et se termine lorsque le produit quitte le cross-dock sur un véhicule sortant. Dans le tableau 3.5, comme pour les indicateurs de performance précédents, la durée totale a été considérablement améliorée, en particulier dans le cas où les heures d'arrivée ne sont pas synchronisées entre les véhicules entrants / sortants au cours d'une journée chargée au cross-dock.

Tableau 3.5 – Le temps de cycle des produits

Synchronisation des véhicules			**Temps de cycle**	
			Moyenne	**Max**
Sync.	**Journée chargée**	**Cross-dock**	0,3	0,8
		PI-hub	0,1	0,7
		Dev. (%)	-60,50 %	-16,40 %
	Journée normale	**Cross-dock**	0,1	5
		PI-hub	0,1	0,6
		Dev. (%)	-53,90 %	-8,50 %
Non-Sync.	**Journée chargée**	**Cross-dock**	1,3	3,6
		PI-hub	0,2	1,9
		Dev. (%)	-84,60 %	-47,70 %
	Journée normale	**Cross-dock**	0,2	1,5
		PI-hub	0,1	1,4
		Dev. (%)	-65,30 %	-10,00 %
			Moyenne	**-42,13%**

L'écart moyen de la déviation des temps de cycles est présenté dans la figure 3.5.

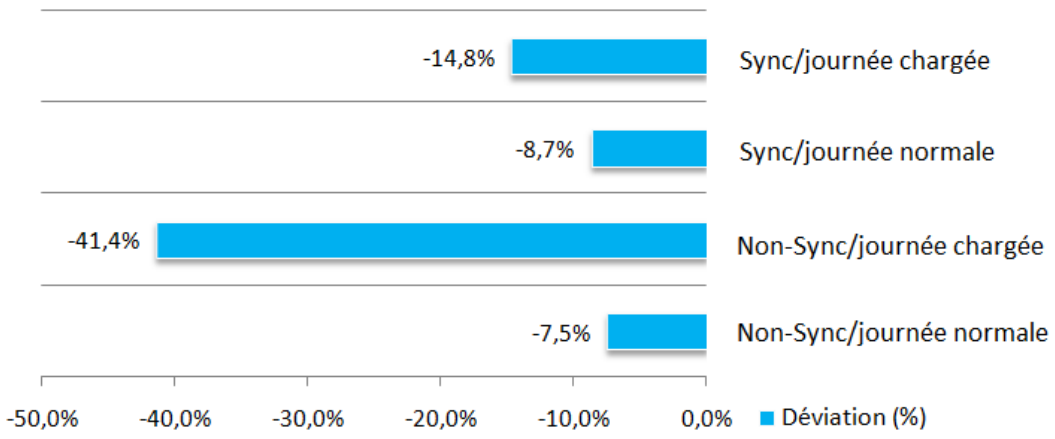

Figure 3.5 – Déviation du temps de cycle des produits

3.2.5 Le taux d'utilisation des ressources

Comme indiqué dans le tableau 3.6, l'utilisation des chariots élévateurs augmente en cas de non synchronisation dans les temps entre arrivées, en particulier lorsque le nombre de véhicules augmente dans les journées chargées. L'utilisation des convoyeurs reste presque la même dans tous les scénarios. La dernière colonne indique l'écart moyen entre l'utilisation des chariots élévateurs et du système AS/RS.

Tableau 3.6 – Taux d'utilisation des ressources dans les deux cross-docks

Synchronisation des véhicules			Taux d'utilisation des ressources			Dev. (%)
			Chariots élévateurs	Convoyeurs	AS/RS	
Sync.	Journée chargée	Cross-dock	71,90 %	-	-	-18.89%
		PI-hub	-	0,10 %	58.32%	
	Journée normale	Cross-dock	40,10 %	-	-	-19.38%
		PI-hub	-	0,00 %	32.33%	
Sync.	Journée chargée	Cross-dock	89,80 %	-	-	-53.28%
		PI-hub	-	0,10 %	41.96%	
	Journée normale	Cross-dock	50,00 %	-	-	-22.26%
		PI-hub	-	0,10 %	38.87%	
					Moyenne	**-28.45%**

L'écart moyen de l'utilisation des ressources est illustré dans la figure 3.6 pour les différents scénarios. Les résultats obtenus montrent clairement que, sous la même charge de travail et sous le même flux de produits, le taux d'utilisation des ressources dans le PI-hub reste largement inférieur à celui du cross-dock classique. Ceci est principalement dû aux avantages de l'automatisation et à la capacité de traitement importante des convoyeurs et des systèmes AS/RS.

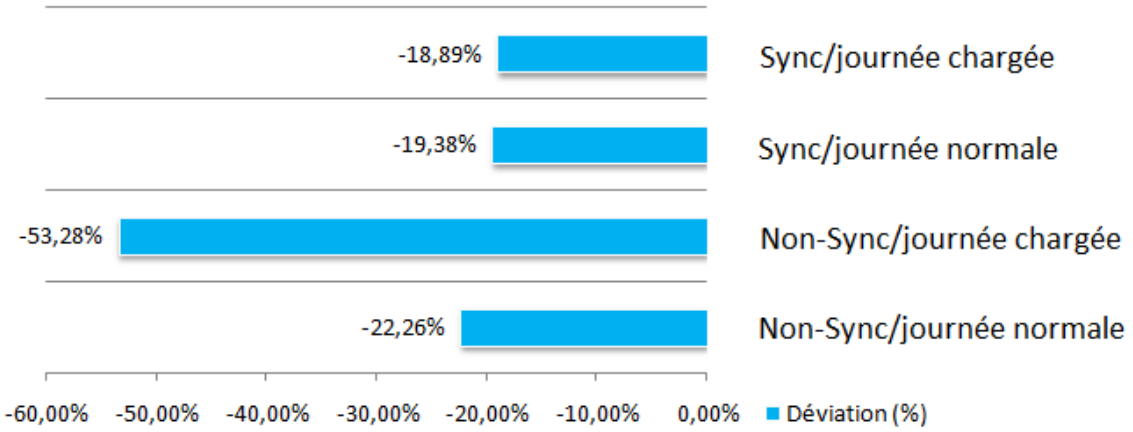

Figure 3.6 – Déviation des taux d'utilisation des ressources

3.3 Couplage simulation-optimisation : Etude de robustesse

L'objectif de cette section est de proposer un mécanisme de couplage simulation-optimisation à travers l'intégration d'un simulateur avec la méta-heuristique Tabu Search (TS) proposée dans le chapitre 2.

Dans un contexte industriel réel du cross-dock classique, les ressources de transport manuel (chariots élévateurs dans le cas du cross-dock étudié) peuvent subir des pannes. Les chariots élévateurs étant l'unique ressource de manutention dans le cross-dock étudié, leur défaillance peut ralentir les quatre processus de cross-docking : déchargement, stockage, tri et chargement des conteneurs.

Pour le PI-hub, la panne d'un PI-conveyors déconnectera le quai correspondant du PI-sorter. Cependant, en cas de défaillance du PI-sorter, l'ensemble du réseau de PI-conveyors est désactivé. Cela est dû à la conception du cross-dock étudié, car les rayonnages de stockage occupent une grande surface du cross-dock, ce qui laisse au PI-sorter une zone de tri très limitée.

Les pannes des ressources de transport internes consistent à désactiver (avec une probabilité P) les chariots élévateurs dans le cross-dock classique, ainsi que les PI-conveyors et le PI-sorter dans le PI-hub.

3.3.1 Indicateur de performance

Afin de comparer les performances du cross-dock classique et du PI-hub en cas de perturbations, le principal indicateur de performance (KPI) considéré dans cette étude est le retard total des véhicules entrants et sortants, calculé par la différence entre l'heure de fin du déchargement/chargement et l'heure de départ prévue (Fonction objective présentée dans le chapitre 2) :

$$\text{Minimiser :} \quad \sum_{i=1}^{N_1}(e_i - B_i) + \sum_{j=1}^{N_2}(f_j - D_j)$$

3.3.2 Approches de résolution : Simulateur intégré dans la méta-heuristique (TS)

Le processus de résolution proposé comprend trois étapes principales (Figure 3.7). La première étape consiste à générer une solution initiale en utilisant l'heuristique (proposée dans le chapitre 2). Dans la deuxième étape, la méta-heuristique Tabu Search (proposée également dans le chapitre 2) est utilisée pour améliorer la solution initiale à la fois pour le cross-dock classique et pour le PI-hub. La troisième étape consiste à tester la robustesse de la solution de la Tabu Search en générant des pannes des ressources de transport.

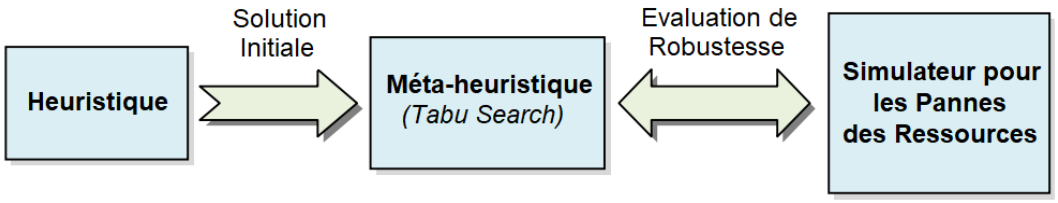

Figure 3.7 – Les différentes étapes du processus d'évaluation de la robustesse des solutions TS

3.3.3 Mécanisme d'évaluation de robustesse

L'objectif de cette section est de tester la robustesse des solutions générées par la méta-heuristique TS pour le cross-dock classique et dans le PI-hub (Figure 3.8). Dans un contexte industriel réel, les ressources de transport manuelles ou automatisées peuvent subir des pannes.

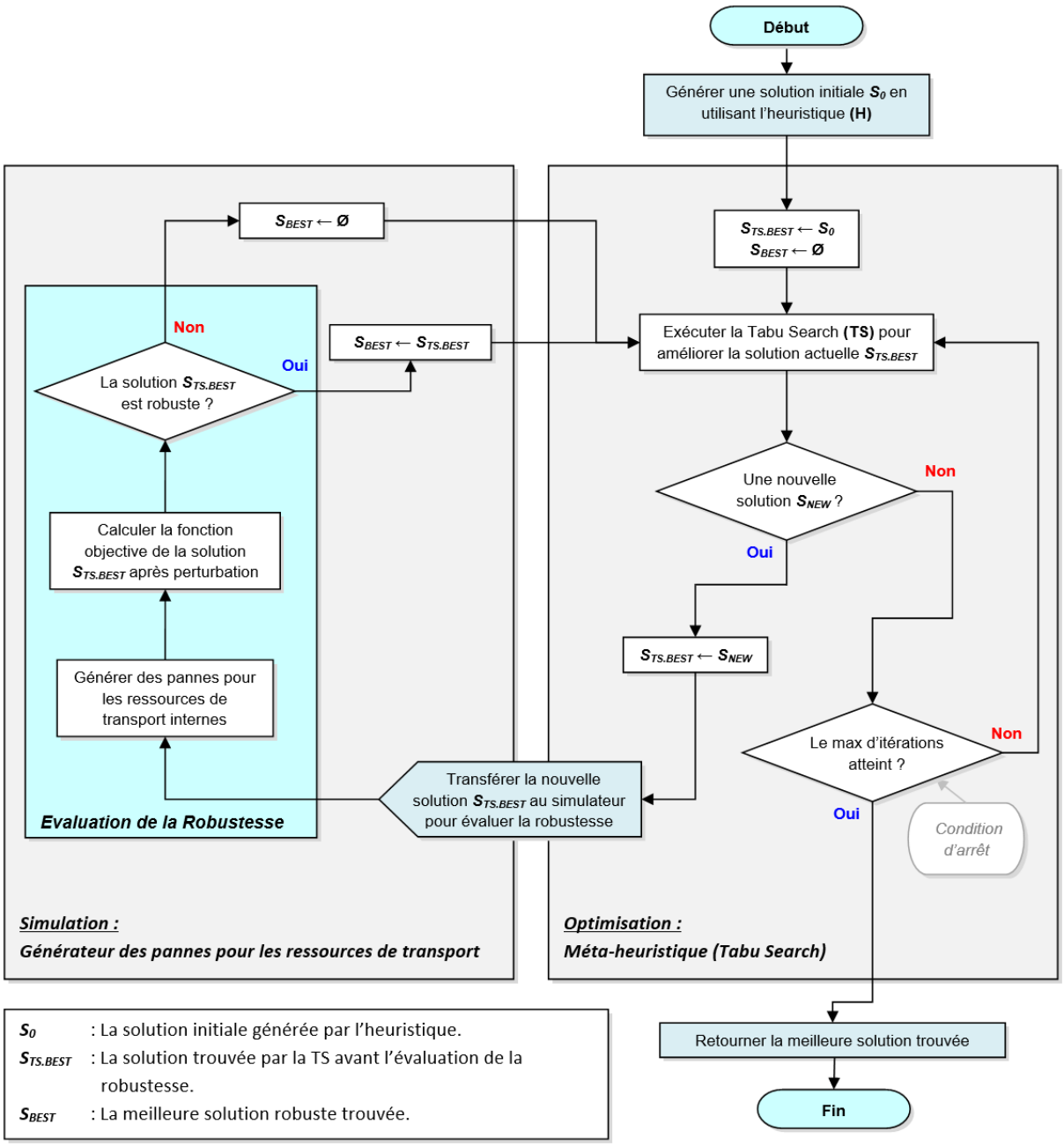

Figure 3.8 – Mécanisme du couplage simulation-optimisation pour la robustesse de la TS

Le mécanisme de robustesse proposé commence par exécuter l'heuristique pour obtenir une solution initiale (S_0) qui sera améliorée ultérieurement par la méta-heuristique TS. Les solutions fournies par la TS ne tiennent pas compte des éventuelles pannes de ressources. Afin de vérifier la robustesse des solutions de la TS, chaque nouvelle solution trouvée par la Tabu Search sera vérifiée par le simulateur de perturbation qui génère des pannes internes des ressources de transport avec une probabilité P, puis calcule la fonction objective après les perturbations. Si la solution est robuste, elle sera considérée comme la nouvelle solution actuelle, sinon, la méta-heuristique lancera une nouvelle

recherche jusqu'à ce que le nombre maximal d'itérations soit atteint.

Plusieurs études dans la littérature ont proposé des indicateurs d'évaluation de la robustesse ; par exemple, Ladier and Alpan (2016b) ont suggéré trois indicateurs de robustesse : le temps de transfert, le temps de déchargement et l'heure d'arrivée du véhicule. Dans cette étude, un critère de robustesse doit être défini pour décider si une solution est robuste ou non. Ainsi, l'écart entre les fonctions objectives avant et après la perturbation est considéré comme le critère de robustesse. En effet, moins la fonction objective est affectée, plus la solution est considérée comme robuste. Par conséquent, les solutions de la TS sont validées si l'écart est inférieur à l'indicateur de robustesse RC :

$$RC = 100 \times \left(\frac{S_P - S_{TS.BEST}}{S_{TS.BEST}} \right)$$

Où :

— S_p : est la fonction objective de la solution après la perturbation.
— $S_{TS.BEST}$: est la meilleure solution actuellement trouvée par la Tabu Search avant d'effectuer le test de robustesse.

3.3.4 Résultats expérimentaux

La solution du TS $S_{TS.BEST}$ est considérée comme une nouvelle solution robuste si la déviation entre S_p et $S_{TS.BEST}$ est inférieure au critère de robustesse RC.

L'heuristique, la méta-heuristique TS et le simulateur des perturbations des ressources sont implémentés en C++ et les tests sont effectués sur un PC doté d'un processeur Intel$^{(R)}$ Core$^{(TM)}$ i3 de 2.40 GHz et 4GB de RAM. Pour le critère d'arrêt, la recherche locale du TS se termine si aucune nouvelle solution n'est trouvée après 200 000 itérations. Ce nombre d'itérations a été fixé après un réglage des paramètres. Pour le cross-dock et le PI-hub, chaque probabilité de défaillance est testée sur 10 réplications et la valeur moyenne de la fonction objective est présentée.

Afin de visualiser l'impact des différentes probabilités de perturbation, une valeur pour l'indicateur de robustesse RC doit être définie. Après avoir effectué plusieurs tests avec différentes valeurs de RC, sa valeur a été fixée à 15%, ce qui signifie que toute nouvelle

Tableau 3.7 – Le retard des véhicules entrants/sortants sans perturbations

		CPLEX (Optimale)	Heuristique	TS
Cross-dock	**Objectif**	1529	2351	1529
	Temps (s)	102,79	0,000	5,9
PI-hub	**Objectif**	990	1363	990
	Temps (s)	91,67	0,000	5,66

Tableau 3.8 – Déviations moyennes pour chaque probabilité de perturbation (10 réplications)

P*			**TS** (P = 0%)	**TS** (P > 0%)	**Gap (%)**
Faible (20%)	**Cross-dock**	**Objectif**	1529	1627,90	6,5%
		Temps (s)	5,9	7,70	—
	PI-hub	**Objective**	990	1052,80	6,3%
		Temps (s)	5,66	10,59	—
Moyenne (25%)	**Cross-dock**	**Objectif**	1529	1629,30	6,6%
		Temps (s)	5,9	9,13	—
	PI-hub	**Objective**	990	1056,10	6,7%
		Temps (s)	5,66	11,82	—
Forte (30%)	**Cross-dock**	**Objectif**	1529	1635,70	7,0%
		Temps (s)	5,9	7,32	—
	PI-hub	**Objective**	990	1058,00	6,9%
		Temps (s)	5,66	8,75	—
Très forte (40%)	**Cross-dock**	**Objectif**	1529	1820,70	19,1%
		Temps (s)	5,9	8,53	—
	PI-hub	**Objective**	990	1307,50	32,1%
		Temps (s)	5,66	13,80	—

P* = Probabilité de la panne.

TS** = Tabu Search.

solution du TS ($S_{TS.BEST}$) ne peut pas être considérée comme robuste si l'écart entre la solution après perturbation S_p et $S_{TS.BEST}$ est supérieur à RC. L'objectif est de tester la robustesse de chaque nouvelle solution $S_{TS.BEST}$. Par conséquent, même si la valeur de la fonction objective dans S_p est inférieure à $S_{TS.BEST}$ (la solution actuellement la plus robuste), S_p ne peut pas être considérée comme la nouvelle meilleure solution car SP est la solution perturbée de $S_{TS.BEST}$. Le tableau 3.7 présente la valeur de la fonction objective sans perturbation et le temps de calcul obtenu sur IBM CPLEX Solver (valeur optimale), l'heuristique et le TS. Les résultats des tests de robustesse sont résumés dans le tableau 3.8.

Une analyse de variance à un facteur (ANOVA) est réalisée pour évaluer l'impact des différentes probabilités de perturbations sur les performances du cross-dock et du PI-hub. Après exécution du test ANOVA, les résultats obtenus sont les suivants : p-value = 0,0329 et rapport $F = 6{,}59$. Comme la valeur de $p \leq 0,05$, la différence entre l'écart moyen entre le cross-dock et le PI-hub est statistiquement significative, ce qui signifie que l'impact des perturbations sur le retard des véhicules est très différent entre le cross-dock et le PI-hub pour cette étude de cas.

Les résultats expérimentaux sont présentés dans le tableau 3.8. La dernière colonne montre l'écart entre la fonction objective avant et après les perturbations. Comme on peut le constater, pour les probabilités de défaillance faibles, moyennes et élevées, l'écart entre la qualité de la solution de la TS avant et après la perturbation est faible. Cependant, pour la probabilité de panne très forte, la déviation devient plus importante dans les deux cas. Dans ce dernier cas, l'écart pour le PI-hub est plus important que celui du cross-dock. En effet, lorsque les ressources de transport cross-dock (chariots élévateurs) sont en panne, le reste des ressources continue de fonctionner, ce qui est identique pour les PI-conveyors du PI-hub. Cependant, si le PI-sorter s'arrête, tout le système de transport est désactivé.

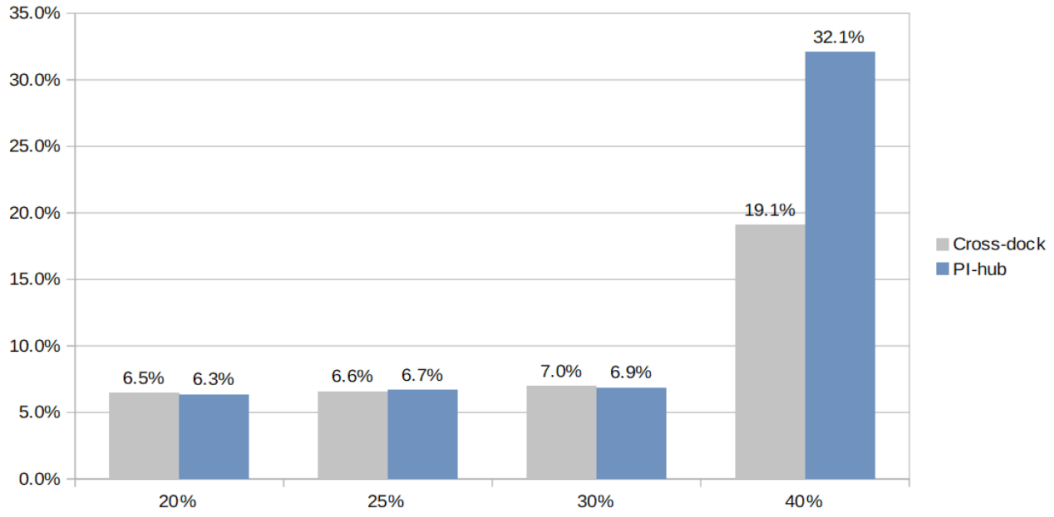

Figure 3.9 – Déviations Moyennes pour les différentes valeurs des probabilités de pannes

La figure 3.9 montre que, lorsque les perturbations deviennent plus importantes, le cross-dock classique est plus robuste que le PI-hub, sous les hypothèses du cross-dock étudié et par rapport aux perturbations internes étudiées. Il s'agit là d'un problème à

prendre en compte dans les futures études de la communauté de recherche sur l'Internet Physique.

3.4 Conclusion

Dans ce chapitre, nous avons examiné le passage du cross-dock classique au PI-hub en deux parties. Dans la première, la simulation du cross-dock et du PI-hub avec divers scénarios a montré que la mise en œuvre de l'Internet Physique a un impact positif sur les performances des cross-docks en réduisant notamment le temps d'attente des véhicules entrants et sortants, le temps total passé par les produits dans le cross-dock ainsi que le taux d'utilisation des ressources de transport internes. Les résultats de la simulation montrent la robustesse de la configuration de PI-hub, en particulier en cas de mauvaise synchronisation entre les véhicules entrants et sortants ou d'une augmentation du flux de produits. Dans la deuxième partie, nous avons proposé une approche de couplage simulation-optimisation en combinant le simulateur avec la méta-heuristique (TS) pour générer des solutions robustes en cas de perturbations au niveau des ressources de transport internes. Les pannes des ressources de transport ont eu un impact différent sur les performances du cross-dock et du PI-hub. En effet, dans les deux cas, l'écart des solutions de la TS n'était pas important pour les probabilités de perturbations faibles, moyennes et élevées. Cependant, pour le PI-hub, lorsque les pannes sont devenues plus importantes, la qualité des solutions de la TS a été considérablement réduite et nous avons constaté que les pannes au niveau du PI-sorter dans le PI-hub ont un effet important sur la qualité des solutions obtenues. Le travail présenté dans ce chapitre a fait l'objet de deux publications dans deux livres : Chargui et al. (2018b) et Chargui et al. (2019e).

Les approches de résolution proposées dans ce chapitre sont de nature pro-actives et permettent de générer des solutions robustes conçue a être faisables en cas de perturbation. Cependant, en cas de perturbations imprévues (qui n'ont pas été prise en considération dans le couplage simulation-optimisation), la solution ne sera plus valable. Le but du prochain chapitre et de proposer un modèle multi-agents réactif en temps-réel à ce type de perturbations imprévues.

Chapitre 4

Planification durable et réactive des opérations dans un cross-dock Road-Rail PI-hub

---

Affin d'assurer un niveau de robustesse en temps-réel pour face aux perturbations imprévues, ce chapitre sera consacré à l'étude du cross-dock Road-Rail PI-hub en proposant un modèle multi-agent MAS pour la planification des véhicules et le groupement des PI-containers. Le MAS est validé par un modèle MILP dans le cas statique. La robustesse du MAS est également évaluée dans un environnement dynamique sous perturbations.

4.1 Introduction

L'objectif de ce chapitre est d'étudier la planification durable des véhicules et le groupement des PI-containers dans les Road-Rail PI-hubs. Le but est de minimiser la somme pondérée du nombre de wagons utilisé, la distance interne parcourue par les PI-containers entre les véhicules et les wagons, ainsi que les retards des véhicules. Un modèle multi-agents efficace et réactif (MAS) est proposé pour la résolution de la planification des véhicules et du groupement des PI-containers. Pour assurer l'efficacité du modèle MAS et améliorer la qualité de chacune de ses solutions, trois méta-heuristiques hybrides parallèles sont intégrées dans trois agents de planification. Ensuite, un modèle de programmation linéaire mixte (MILP) est proposé pour évaluer les performances du MAS. Enfin, le MAS est également évalué sous des perturbations internes. Les résultats obtenus montrent la capacité du MAS à générer des solutions alternatives durables en replanifiant les véhicules en cas de perturbations.

4.2 Présentation du problème

Dans ce chapitre, nous examinons le problème de la planification des véhicules dans un Road-Rail PI-hub ainsi que l'affectation et le groupement des PI-containers. Comme le montre la figure 4.1, en plus de la section Rail-Rail, le Road-Rail PI-hub est composé de deux sections principales : la section Rail-Road et la section Road-Rail.

Ce chapitre se concentre sur la deuxième section : le Road-Rail PI-hub (Section 2 de la figure 4.1), qui gère le transfert des PI-containers des véhicules aux wagons. Initialement, le train est composé de nombreux wagons vides. Chaque ensemble de cinq wagons est considéré comme un seul bloc. Les wagons seront chargés par les PI-containers déchargés des véhicules.

Une fois que le train arrive au Road-Rail PI-hub, le premier bloc de cinq wagons est affecté à la section PI-sorter du Rail-Road. Cette section est dédiée à la réception des PI-containers du train et leur chargement dans les véhicules sortants. Dans l'étape suivante, le bloc de cinq wagons se déplace vers la section Road-Rail pour charger des PI-containers à partir des véhicules vers les wagons. Les PI-containers ayant les mêmes

destinations doivent être chargés dans les mêmes wagons consécutifs afin de faciliter le déchargement des PI-containers. Une fois le premier bloc de wagons est chargé avec des PI-containers, le bloc suivant est affecté à la section de déchargement Rail-Road.

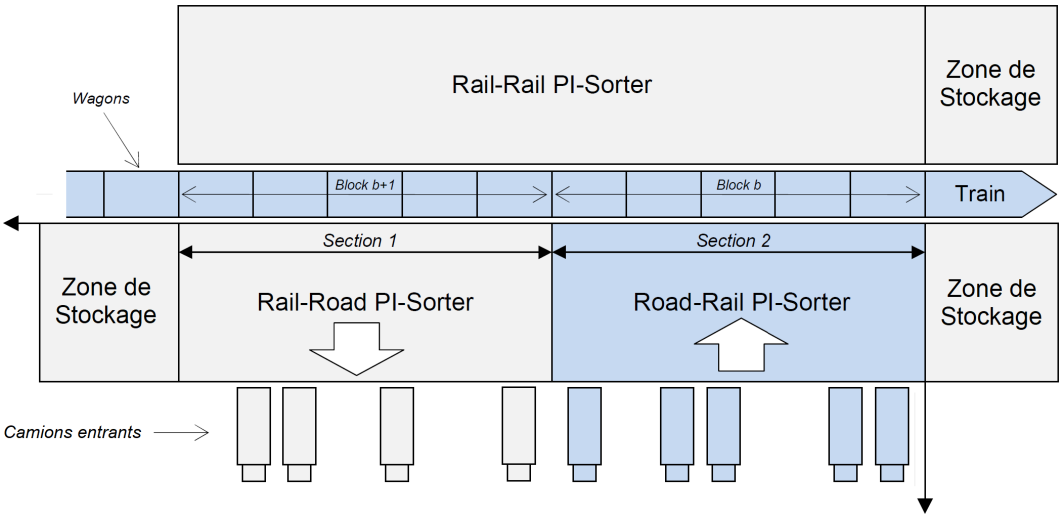

Figure 4.1 – Le Road-Rail PI-hub (section 2)

Les hypothèses principales considérées dans ce problème sont les suivantes :

— Les véhicules entrants peuvent décharger des PI-containers de différentes longueurs et chacun de ces PI-containers possède une destination spécifique.
— Les PI-containers ayant la même destination doivent être chargés dans des wagons consécutifs.
— Chaque wagon du train ne doit charger que des PI-containers ayant la même destination.
— Il ne doit pas y avoir de wagons vides entre deux wagons utilisés.

Les modèles proposés dans ce chapitre (MAS et MILP) traitent la durabilité à travers la minimisation du nombre de wagons utilisés ainsi que par la réduction de la distance parcourue par les PI-containers dans la zone Road-Rail, ce qui réduira l'énergie électrique utilisée par les PI-conveyors pour transférer les PI-containers, ainsi que le retard de déchargement des véhicules. Les modèles proposés traitent deux parmi les trois piliers du développement durable, à savoir les piliers économique et environnemental. Le concept du PI considère le pilier social comme un élément important, en particulier la santé des chauffeurs routiers qui passent un temps important sur la route Montreuil (2011). D'un point de vue économique, les modèles proposés réduisent le coût total en utilisant un nombre minimal de wagons et en minimisant le retard des véhicules. D'un point de

vue environnemental, les deux modèles réduisent non seulement la distance parcourue par les PI-containers, ce qui permet de minimiser l'énergie utilisée par les PI-conveyors, mais minimisent aussi le nombre de wagons utilisés, ce qui réduit les émissions de CO_2 de la locomotive du train. Dans les deux modèles, la principale contrainte est que les PI-containers avec la même destination doivent être chargés et regroupés dans des wagons consécutifs tout en minimisant la fonction objective. Le but est de regrouper les PI-containers tout en conservant ceux ayant la même destination dans des wagons consécutifs. Les deux modèles proposés sont ensuite évalués sur les mêmes instances. Enfin, afin de prendre en compte l'aspect dynamique, le MAS est évalué face aux perturbations internes du PI-hub.

4.3 Modèles proposés

Afin de résoudre le problème étudié, nous proposons un modèle multi-agents (MAS) dont l'objectif est de minimiser le nombre de wagons utilisés, la distance parcourue par les PI-containers des véhicules vers les wagons du train ainsi que le retard des véhicules sur les quais, tout en respectant plusieurs contraintes liées à la capacité des wagons et aux destinations des PI-containers. Afin d'évaluer la qualité des solutions fournies par le MAS, un modèle mathématique MILP est proposé.

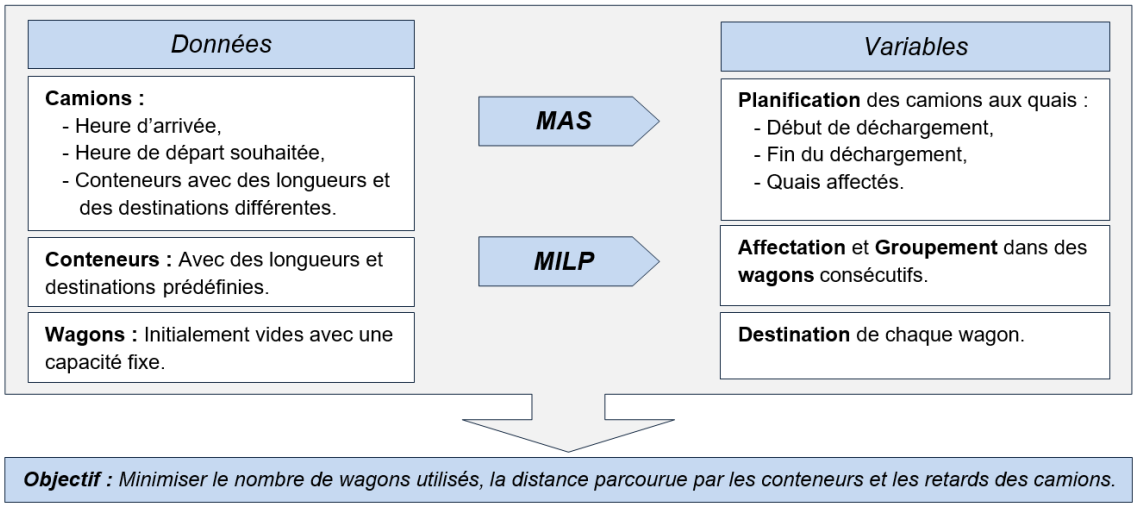

Figure 4.2 – Les principales données et variables des deux modèles proposés (MAS et MILP)

La figure 4.2 montre les principaux données et variables des modèles proposés. Il est important de mentionner que ce n'est pas pratique de résoudre le problème étudié en

utilisant le MILP en raison des temps de calcul qui sont très élevés, en particulier pour les grandes instances, en plus il est difficile d'utiliser le MILP dans un environnement perturbé.

4.3.1 Modèle mathématique : MILP

Cette section présente la formulation mathématique proposée du problème de planification des véhicules et de regroupement des PI-containers dans un Road-Rail sous la forme d'un modèle de programmation linéaire en nombres entiers mixte (MILP). Le problème est constitué de N PI-containers de longueurs différentes L_i, qui sont initialement placés dans H véhicules. Un véhicule peut décharger des PI-containers avec différentes destinations parmi les D destinations considérées dans le modèle. Chaque véhicule doit être affecté à l'un des K quais de la section Road-Rail pour le déchargement. Les PI-containers déchargés doivent être groupés dans un ensemble de wagons consécutifs, qui doit avoir le nombre de wagons le plus faible possible. Les véhicules doivent être affectés aux quais afin de minimiser les retards et la distance totale parcourue par les PI-containers vers les wagons.

Le modèle mathématique proposé a deux objectifs principaux pour le problème :

— La planification des véhicules sur les quais.
— Le groupement et l'affectation des PI-containers aux wagons.

4.3.1.1 Données et paramètres

N Nombre total de PI-containers
K Nombre de quais
D Nombre de destinations
W Nombre total de wagons dans le train
H Nombre de véhicules
Q Capacité du wagon
i Indices des PI-containers
k Indices des quais
d Indices des destinations

$w, w1, w2$ Indices des wagons

h Indices des véhicules

E_h Heure d'arrivée du véhicule h

F_h Heure de départ prévu du véhicule h

J_h Temps de traitement du véhicule h

V Temps de changement des véhicules dans un quai

P_k Position du centre du quai k à partir de l'axe de droite de la zone Road-Rail

R_w Position du centre du wagon w à partir de l'axe de droite de la zone Road-Rail

L_i Longueur du PI-containers i

$$A_{hi} = \begin{cases} 1 & \text{Si le PI-container } i \text{ est dans le véhicule } h \\ 0 & \text{Sinon} \end{cases}$$

$$S_{di} = \begin{cases} 1 & \text{Si } d \text{ est la destination du PI-container } i \\ 0 & \text{Sinon} \end{cases}$$

α Facteur de pondération pour le nombre de wagons utilisés

β Facteur de pondération pour la distance totale parcourue par les PI-containers

γ Facteur de pondération pour le retard des véhicules

M Un grand nombre positif

4.3.1.2 Variables de décision

Variables binaires

$$x_{iw} = \begin{cases} 1 & \text{Si le PI-container } i \text{ est affecté au wagon } w \\ 0 & \text{Sinon} \end{cases}$$

$$y_{hk} = \begin{cases} 1 & \text{Si le véhicule } h \text{ est affecté au quais } k \\ 0 & \text{Sinon} \end{cases}$$

$$u_w = \begin{cases} 1 & \text{Si le wagon } w \text{ est utilisé} \\ 0 & \text{Sinon} \end{cases}$$

$$e_{wd} = \begin{cases} 1 & \text{Si } d \text{ est la destination du wagon } w \\ 0 & \text{Sinon} \end{cases}$$

$$g_{h_1h_2} = \begin{cases} 1 & \text{Si les véhicules } h1 \text{ et } h_2 \text{ sont affesctés au même} \\ & \text{quais et } h_1 \text{ est un prédécesseur de } h_2 \\ 0 & \text{Sinon} \end{cases}$$

Variables continues

r_h Début du déchargement du véhicule h

s_h Fin du déchargement du véhicule h

f_h Retard du véhicule h

d_{iw} Distance parcourue par le PI-container i vers le wagon w

Fonction objective

L'objectif du modèle MILP est de minimiser la somme pondérée du nombre de wagons utilisés, la distance interne totale parcourue par les PI-containers et le retard total des véhicules.

$$\text{Minimiser :} \qquad \alpha \sum_{w=1}^{W} u_w + \beta \sum_{i=1}^{N} \sum_{w=1}^{W} d_{iw} + \gamma \sum_{h=1}^{H} f_h \tag{4.1}$$

Où α, β et γ sont les facteurs de pondération du nombre de wagons utilisés, de la distance totale parcourue par les PI-containers et du retard total des véhicules respectivement ($\alpha + \beta + \gamma = 1$). Il est important de mentionner que ces trois objectifs ont trois unités différentes. Le retard est exprimé en minutes, la distance en mètres tandis que le premier objectif est calculé comme un nombre non dimensionnel. De plus, les valeurs des facteurs de pondération α, β et γ peuvent être définies par l'utilisateur en fonction de la priorité et de l'importance de chaque objectif.

Contraintes

$$\sum_{w=1}^{W} x_{iw} = 1 \qquad (\forall i = 1...N) \tag{4.2}$$

$$\sum_{i=1}^{N} x_{iw} L_i \leq Q \qquad (\forall w = 1...W) \tag{4.3}$$

$$x_{iw} + x_{jw} \leq \sum_{d=1}^{D} S_{di} S_{dj} + 1 \qquad (\forall i, j = 1...N, \forall w = 1...W : i \neq j) \tag{4.4}$$

$$x_{iw} \leq u_w \qquad (\forall i = 1...N, \forall w = 1...W) \tag{4.5}$$

$$e_{wd} \leq S_{di} + 1 - x_{iw} \qquad (\forall i = 1...N, \forall w = 1...W, \forall d = 1...D) \tag{4.6}$$

$$u_w = \sum_{d=1}^{D} e_{wd} \qquad (\forall w = 1...W) \tag{4.7}$$

$$\begin{aligned} |w_1 - w_2| + 1 \leq \sum_{w=1}^{W} e_{wd} + M(2 - (e_{w_1 d} + e_{w_2 d})) \\ (\forall d = 1...D, \forall w_1, w_2 = 1...W, w_1 \neq w_2) \end{aligned} \tag{4.8}$$

$$|w_1 - w_2| + 1 \leq \sum_{w=1}^{W} u_w + M(2 - (u_{w_1} + u_{w_2})) \qquad (\forall w_1, w_2 = 1...W, w_1 \neq w_2) \tag{4.9}$$

$$u_1 = 1 \tag{4.10}$$

$$\begin{aligned} d_{iw} \geq |P_k - R_w| - M(2 - (A_{hi} y_{hk} + x_{iw})) \\ (\forall i = 1...N, \forall w = 1...W, \forall k = 1...K, \forall h = 1...H) \end{aligned} \tag{4.11}$$

$$\sum_{k=1}^{K} y_{hk} = 1 \qquad (\forall h = 1...H) \tag{4.12}$$

$$y_{h_1 k} + y_{h_2 k} - 1 \leq g_{h_1 h_2} + g_{h_2 h_1} \qquad (\forall k = 1...K, \forall h_1, h_2 = 1...H, h_1 \neq h_2) \tag{4.13}$$

$$g_{h_1 h_2} + g_{h_2 h_1} \leq 1 \qquad (\forall h_1, h_2 = 1...H) \tag{4.14}$$

$$r_h \geq E_h \qquad (\forall h = 1...H) \tag{4.15}$$

$$r_{h_2} \geq s_{h_1} + V - M(1 - g_{h_1 h_2}) \qquad (\forall h_1, h_2 = 1...H) \tag{4.16}$$

$$s_h \geq r_h + J_h \qquad (\forall h = 1...H) \tag{4.17}$$

$$f_h \geq s_h - F_h \qquad (\forall h = 1...H) \tag{4.18}$$

$$x_{iw}, y_{hk}, u_w, e_{wd}, g_{h_1h_2} \in \{0,1\} \quad (\forall i = 1...N,\ \forall w = 1...W, \forall k = 1...K, \forall h, h_1, h_2 = 1...H, \forall d = 1...D) \tag{4.19}$$

$$r_h \geq 0, s_h \geq 0, f_h \geq 0, d_{iw} \geq 0 \qquad (\forall i = 1...N,\ \forall w = 1...W,\ \forall h = 1...H) \tag{4.20}$$

Contraintes de capacité et de groupement

Dans les contraintes (4.2), chaque PI-container i déchargé du véhicule h doit être affecté à un seul wagon w. Les contraintes (4.3) garantissent que la capacité totale des wagons Q ne soit pas dépassée par la longueur totale des PI-containers attribués. Les contraintes (4.4) garantissent que deux PI-containers i et j avec des destinations différentes ne peuvent pas être chargés dans le même wagon. Dans les contraintes (4.5), si un PI-container est affecté à un wagon, ce wagon doit être utilisé ($u_w = 1$). Les contraintes (4.6) définissent la destination d'un wagon si un PI-container lui est attribué. Les contraintes (4.7) garantissent que si un wagon est utilisé, une destination doit être affectée à ce wagon. Dans les contraintes (4.8), les wagons ayant la même destination doivent être consécutifs. Les contraintes (4.9) garantissent que tous les wagons utilisés doivent être consécutifs afin d'éviter des wagons vides entre ceux qui sont utilisés (Figure 4.3). Les contraintes (4.10) garantissent que les wagons sont utilisés en commençant par le premier wagon.

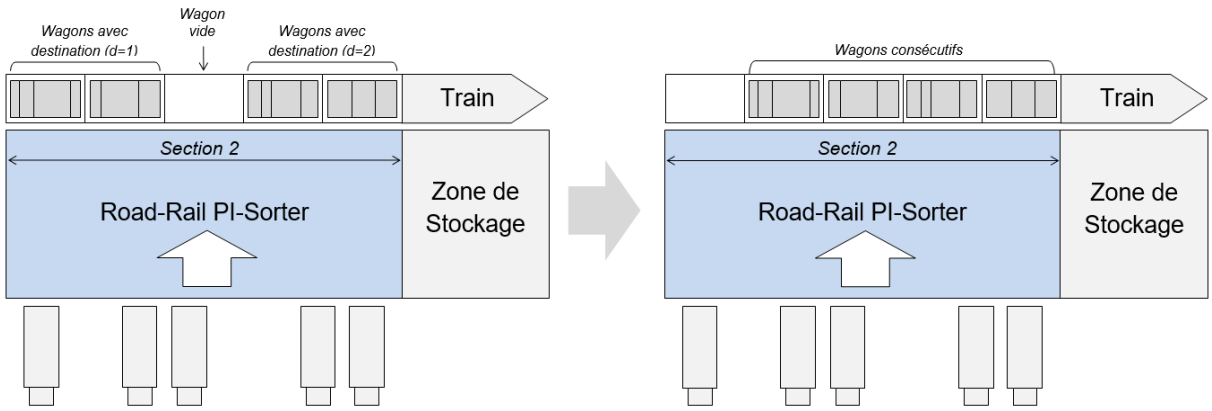

Figure 4.3 – Les wagons utilisés doivent être consécutifs

Contrainte de distance

Les contraintes (4.11) calculent la distance parcourue par un PI-container du quai au wagon.

Contraintes de planification des véhicules

Dans les contraintes (4.12), chacun des véhicules doit être affecté à un seul quai. Les contraintes (4.13) et (4.14) gèrent l'affectation et le séquencement des véhicules sur les quais. Les contraintes (4.15) assurent que chaque véhicule commence à décharger les conteneurs une fois arrivés au PI-hub. Dans les contraintes (4.16), si le véhicule h_1 précède le véhicule h_2 au même quai, alors h_2 commence le déchargement après la fin du déchargement du véhicule h_1. Les contraintes (4.17) calculent l'heure de fin du déchargement d'un véhicule. Les contraintes (4.18) calculent le retard d'un véhicule. Les contraintes (4.19) garantissent que les variables sont binaires. Les contraintes (4.20) assurent que les variables continues sont positives. Dans les deux contraintes (4.8) et (4.9), w_1 et w_2 sont utilisés comme indices allant de 1 à W. L'idée est de limiter la différence entre les indices (valeur absolue) afin de garder tous les wagons utilisés consécutives dans les contraintes (4.8) et assurer que les wagons ont la même destination dans les contraintes (4.9).

La valeur minimale du paramètre M est :

$$M \geq max(W, 5\times\ WagonLength,\ PlanningHorizon)$$

Où W est le nombre total de wagons dans le train et $5\times$ *WagonLength* est la longueur de la section Road-Rail pouvant traiter 5 wagons à la fois (dans cette étude, *WagonLength* $= 20m$ est la longueur totale d'un wagon et la longueur utile du wagon est de $18m$). En effet, dans les contraintes (4.8) et (4.9), la valeur minimale de M est le nombre total des wagons W. Cependant, dans la contrainte (4.11), la valeur minimale de M est la longueur totale des cinq wagons (longueur de la section Road-Rail). Dans la contrainte (4.16), la valeur minimale de M est l'horizon de planification de l'instance. Il est important de mentionner que la même unité de longueur (mètre m dans cette étude) doit être utilisée dans tous les paramètres de longueur (P_k, R_w, L_i et Q) et dans la variable de distance d_{iw}.

4.3.2 Modèle multi-agents : MAS

Dans cette section, nous présentons une description détaillée de l'architecture et des composantes du MAS proposé. La figure 4.4 illustre les différentes composantes du MAS. Le modèle est composé de H agents véhicules, un agent train, un agent synchroniseur et trois agents de planification parallèles (VNS-SA, GRASP-SA et TS-SA).

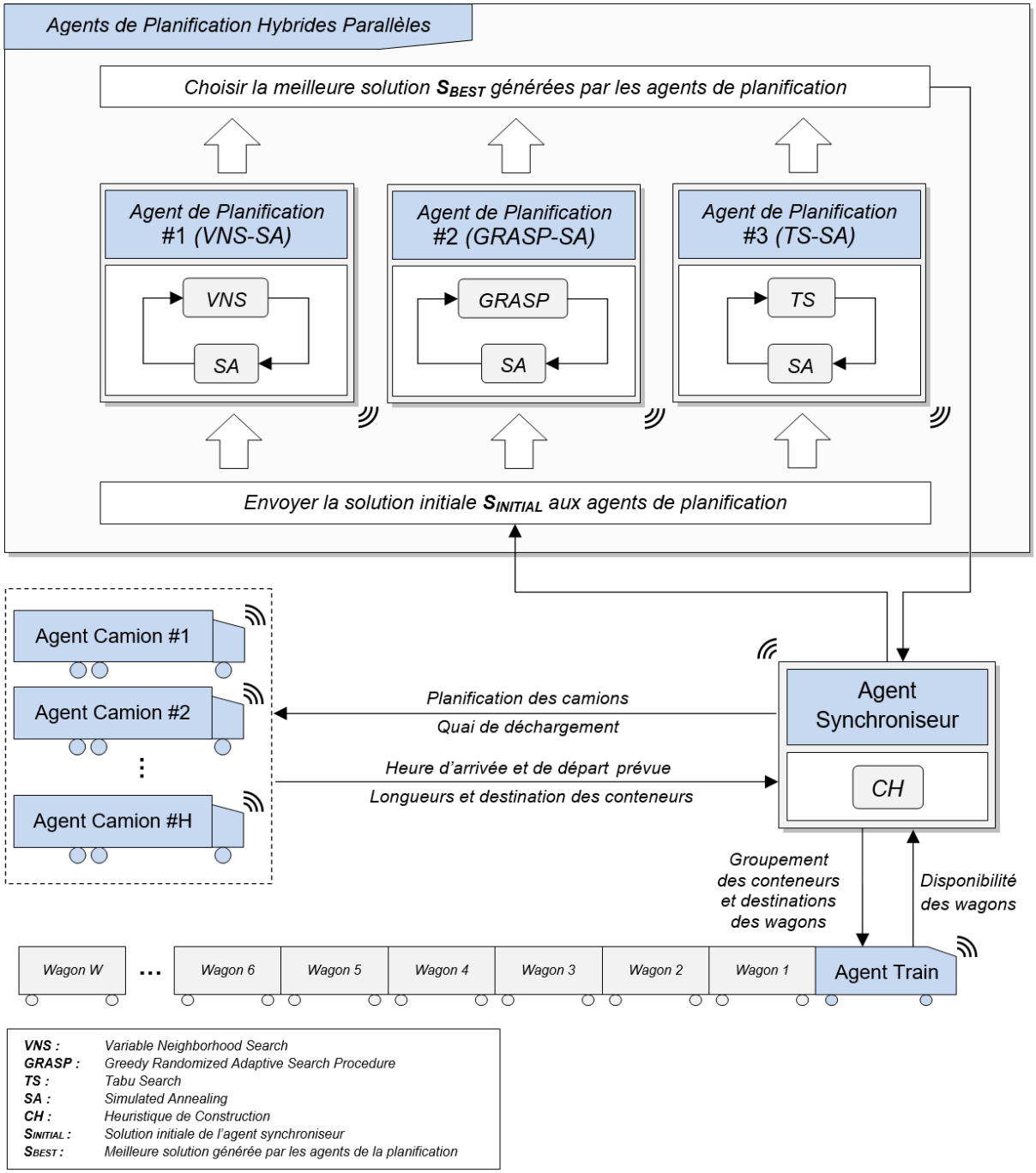

Figure 4.4 – Architecture du modèle MAS proposé

Le diagramme de séquence de la figure 4.5 résume la communication et la négociation entre les différents agents du système MAS proposé, qui communiquent dans les étapes suivantes :

Étape 1 : Les agents de véhicules envoient leurs heures d'arrivée et de départ prévues, ainsi que les destinations et les longueurs des PI-containers à l'agent Synchroniseur.

Étape 2 : L'agent train envoie la disponibilité de ses wagons à l'agent Synchroniseur.

Étape 3 : l'agent Synchroniseur exécute l'heuristique de construction (CH) pour générer une solution initiale $S_{INITIAL}$.

Étape 4 : L'agent synchroniseur envoie la solution initiale $S_{INITIAL}$ aux trois agents de planification parallèles hybrides.

Étape 5 : Chacun des trois agents de planification exécute une méta-heuristique hybride différente pour améliorer la solution initiale.

Étape 6 : La meilleure solution S_{BEST} trouvée par les trois agents de planification est envoyée à l'agent de synchronisation.

Étape 7 : l'agent synchroniseur envoie la planification (heure de début / fin du déchargement et le quai attribué) vers les agents véhicules ainsi que les groupements de PI-containers et les destinations des wagons vers les agents trains.

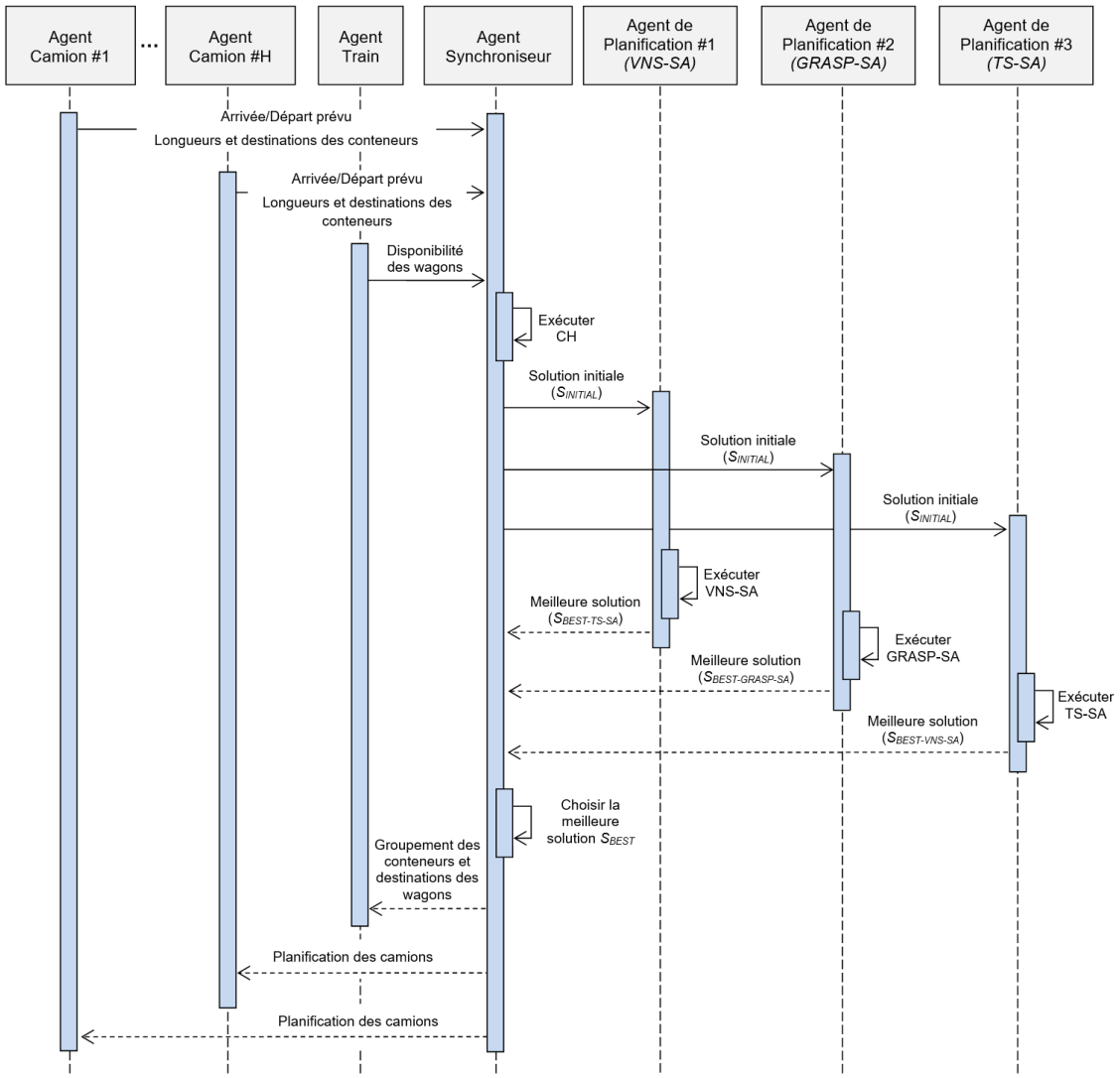

Figure 4.5 – Diagramme de séquence du modèle MAS proposé

Fonction objective

Comme mentionné précédemment, l'objectif consiste à minimiser :

— Le nombre de wagons utilisés.

— La distance totale parcourue par les PI-containers des quais aux wagons.

— Le retard des véhicules.

Après chaque mouvement de recherche locale (Insertion, Permutation ou Insertion+ Permutation) sur les PI-containers et les véhicules dans les trois agents de planification, la fonction objective est calculée comme suit : Le premier objectif, qui est le nombre de wagons utilisés, est calculé sur la base de l'affectation des PI-containers aux wagons. Ensuite, la distance totale parcourue par les PI-containers est calculée en utilisant la position de leurs véhicules aux quais et la position moyenne du wagon auquel le PI-container est affecté. Le dernier objectif est présenté comme l'écart entre l'heure de la fin du déchargement et l'heure de départ prévue des véhicules.

Le MAS est composé des agents suivants qui communiquent entre eux :

Agent véhicule

Avant leur planification sur les quais, les agents de véhicules envoient à l'agent synchroniseur la destination et la longueur de leurs PI-containers ainsi que leur heure d'arrivée et l'heure de départ prévue. Ils attendent alors leur planification aux quais.

Agent train

L'agent train envoie le nombre de ses wagons disponibles et leurs capacités à l'agent synchroniseur et attend que les PI-containers soient groupés dans les wagons.

Agent synchroniseur

L'objectif de l'agent synchroniseur est de recevoir les demandes des agents véhicules et de la disponibilité des wagons de l'agent de train, puis il exécute une heuristique de construction (CH) pour trouver un groupement initial de PI-containers dans les wagons ainsi qu'une planification initiale des véhicules aux quais. Ensuite, cette solution initiale est envoyée aux trois agents de planification pour l'améliorer. Une description du pseudo-code de l'heuristique CH est présentée dans l'algorithme 4.1.

Comme indiqué dans l'algorithme 4.1, dans la première étape, l'agent synchroniseur

commence par trier tous les PI-containers dans un ordre décroissant de leur longueur et en les regroupant par destination. L'objectif est d'utiliser le moins de wagons possible. On suppose qu'il y a un nombre suffisant de wagons disponibles dans le train. Ensuite, il trouve la position moyenne $AvPos[h]$ du véhicule en utilisant la position de ses PI-containers dans les wagons. Pour chaque véhicule, l'algorithme affecte le véhicule au quai disponible le plus proche de sa position moyenne $AvPos[h]$ et calcule l'heure de début/fin de déchargement du véhicule.

Algorithme 4.1 : Heuristique de construction de la solution initiale de l'agent synchroniseur

```
1  Recevoir les longueurs et destinations des PI-containers des agents véhicules et la disponibilité
     des wagons de l'agent train
2  Trier les PI-containers dans l'ordre décroissant de longueurs et les grouper par destination
3  Entrée : L[i], longueur du PI-container (i)
4  Entrée : AvL[w] longueur disponible dans chaque wagon (w)
5  (Initialiser par AvL[w] = Q : Capacité des wagons)
6  Entrée : AvPos[h] position Moyenne du véhicule (h)
7  Entrée : SumDist = 0 (Variable temporaire pour calculer AvPos[h])
8  Pour chaque destination d Faire
9     Pour chaque PI-container (i) de destination (d) Faire
10       Pour chaque wagon (w) Faire
11          Si wagon (w) vide Alors
12             Affecter la destination (d) au wagon (w)
13          Fin Si
14          Si L[i] ≤ AvL[w] Et (i) et (w) ont la même destination Alors
15             Affecter le PI-container (i) au wagon (w)
16             AvL[w] = AvL[w] − L[i]
17             Sortir (Choisir le prochain PI-container)
18          Fin Si
19       Fin Pour
20    Fin Pour
21 Fin Pour
22 Pour chaque véhicule (h) Faire
23    SumDist = 0
24    Pour chaque PI-container (i) Faire
25       Si PI-container (i) est dans le véhicule (h) Alors
26          SumDist = SumDist+ Position du wagon contenant (i)
27       Fin Si
28    Fin Pour
29    AvPos[h] = SumDist / Nombre des PI-containers dans le véhicule (h)
30    Affecter le véhicule (h) au quai le plus proche à AvPos[h]
31    Calculer le début/fin de déchargement du véhicule (h)
32 Fin Pour
33 Calculer la Fonction Objective
34 Sortie : S_INITIAL
```

Agents de planification

Il existe trois agents de planification dans l'architecture du modèle MAS proposé. Chacun exécute une méta-heuristique hybride différente : Variable Neighbourhood Search

(VNS) Mladenović and Hansen (1997), Greedy Randomized Adaptive Search Procedure (GRASP) Feo and Resende (1995) et Tabu Search (TS) Glover (1989). Chacune de ces trois méta-heuristiques est hybridée avec le Simulated Annealing (SA) Kirkpatrick et al. (1983), ce qui donne trois méta-heuristiques hybrides (VNS-SA, GRASP-SA et TS-SA).

Le SA est combiné avec trois méta-heuristiques différentes (VNS, GRASP et TS) afin de profiter des avantages et des fonctionnalités de chaque algorithme et d'établir un équilibre entre la diversification et l'intensification car la convergence de le SA est rapide. Le VNS est choisie pour sa capacité à changer de manière dynamique le voisinage de la recherche pour éviter les optima locaux. Le GRASP est sélectionnée pour sa fonctionnalité de démarrages multiples combinés avec la recherche locale. Le GRASP aide aussi à sortir des optima locaux. Le TS est choisie pour sa capacité à utiliser la mémoire lors de l'exploration des solutions du voisinage. En effet, les mouvements de recherche locale effectués sont stockés dans une *tabu list*, ce qui donne à l'algorithme la possibilité d'éviter la revisite des solutions déjà rencontrées. Enfin, pour le SA, lorsque le paramètre de température $Tmpr$ est élevé, il a tendance à accepter des solutions détériorées. Ensuite, à mesure que la température $Tmpr$ diminue, l'algorithme devient très sélectif sur les nouvelles solutions générées. A travers des tests et des expériences, nous observons que chacune des trois méta-heuristiques (VNS-SA, GRASP-SA et TS-SA) a des forces et des faiblesses selon la configuration des instances, ce qui est la raison principale pour les exécuter toutes les trois en parallèle.

Dans les trois agents de planification, VNS, GRASP et TS sont utilisés pour le séquencement des destinations des wagons et le regroupement des PI-containers, puis, en raison de sa convergence rapide, le SA (voir l'algorithme 4.2) est utilisée pour trouver la meilleure planification des véhicules aux quais pour chaque combinaison de séquence des destinations des wagons. Ces trois agents de planification sont concurrents et s'exécutent en parallèle pour améliorer la solution initiale envoyée par l'agent synchroniseur. Dans les trois algorithmes, trois mouvements de recherche locale sont pris en compte : Insertion, Permutation et Insertion+Permutation. Le mouvement d'insertion consiste à sélectionner un véhicule au hasard, puis à l'insérer dans un autre quai. L'échange consiste en l'échange des quais de deux véhicules différents. Enfin, dans le mouvement Insertion+Permutation, un véhicule est aléatoirement sélectionné, inséré dans un autre quai puis échangé avec un autre véhicule.

Algorithme 4.2 : Simulated Annealing pour la planification des véhicules

Entrée : $Tmpr$, S, S_{BEST}, S_{NEW} (S_{NEW} est la nouvelle solution contenant le séquencement des destinations des wagons dans chaque nouvelle itération)
Tant Que $Tmpr > \epsilon$ **Faire**
  Effectuer un mouvement de recherche locale sur la planification des véhicules :
  Insertion, Permutation ou Insertion+Permutation
  (Probabilité de chaque mouvement = 1/3)
  r = Nombre aléatoire ($0 \leq r \leq 1$)
  Si $r < Exp((S - S_{NEW})/Tmpr)$ **Alors**
    $S = S_{NEW}$
    Si $S_{BEST} > S_{NEW}$ **Alors**
      $S_{BEST} = S_{NEW}$
    Sinon
      Annuler le mouvement de recherche locale
      Diminuer $Tmpr$
    Fin Si
  Fin Si
    Diminuer $Tmpr$
  Fin Si
Fin Tant Que
Réinitialiser $Tmpr$ à sa valeur initiale
Sortie : S, S_{BEST}

Agent de planification #1 : (VNS-SA)

Le premier agent de planification utilise la méta-heuristique hybride VNS-SA. Comme présenté dans l'algorithme 4.3, l'agent de planification #1 commence par charger la solution initiale envoyée par l'agent synchroniseur. Ensuite, un ensemble de structures de voisinage NbN est généré pour le séquencement des destinations des wagons.

Algorithme 4.3 : Algorithme du VNS-SA proposé

Entrée : solution $S_{INITIAL}$ reçue de l'agent synchroniseur
Initialiser : $MaxGlobalIt$, $Tmpr$, NbN, $MaxShuffleIt$
Initialiser : $S_{BEST} = S = S_{INITIAL}$, $MaxVNSIt = MaxGlobalIt/NbN$
Générer NbN structures de voisinage
$GlobalIteration = 0$
Pour chaque structure de voisinage **Faire**
  Tant Que $GlobalIteration < MaxVNSIt$ **Faire**
    Effectuer un mouvement de recherche locale sur le séquencement des destinations des wagons : Insertion, Permutation ou Insertion+Permutation
    (Probabilité de chaque mouvement = 1/3)
    Exécuter : SA_Algorithm() pour améliorer la planification des véhicules (Voir Algorithme 4.2)
    Mélanger le séquencement des PI-containers après chaque $MaxShuffleIt$ itérations
    $GlobalIteration++$
  Fin Tant Que
Fin Pour
Sortie : S_{BEST}

Pour chaque structure de voisinage et à chaque itération, l'algorithme 4.3 effectue un mouvement de recherche locale (Insertion, Permutation et Insertion+Permutation) avec

une probabilité de 1/3 pour chaque mouvement, puis exécute la Simulated Annealing : *SA_Algorithm()* (Algorithme 4.3) pour améliorer les affectations des véhicules sur les quais.

Agent de planification #2 : (GRASP-SA)

Pour le deuxième agent de planification, la GRASP est utilisé pour la séquence des destinations des wagons. Comme présenté dans l'algorithme 4.4, l'agent de planification GRASP-SA commence par charger la solution initiale envoyée par l'agent synchroniseur. Ensuite, pour chaque itération globale, l'un des trois opérateurs de recherche locale est effectué, suivi de *SA_Algorithm()* pour la planification des véhicules. Cependant, si le nombre maximum d'itérations locales $LocalMaxIt$ est atteint, une nouvelle séquence de destinations du wagon est générée.

Algorithme 4.4 : Algorithme du GRASP-SA proposé

Entrée : solution $S_{INITIAL}$ reçue de l'agent synchroniseur
Initialiser : $MaxGlobalIt$, $Tmpr$, $MaxShuffleIt$
Initialiser : $S_{BEST} = S = S_{INITIAL}$
$GlobalIteration = 0$, $LocalIteration = 0$
Tant Que $GlobalIteration < MaxGlobalIt$ **Faire**
  Si $LocalIteration == LocalMaxIt$ **Alors**
    Générer aléatoirement un nouveau séquencement des destinations des wagons
    $LocalIteration = 0$
  Fin Si
  Effectuer un mouvement de recherche locale sur le séquencement des destinations des wagons :
  Insertion, Permutation ou Insertion+Permutation (Probabilité de chaque mouvement = 1/3)
  Exécuter : SA_Algorithm() pour améliorer la planification des véhicules (Voir Algorithme 4.2)
  Mélanger le séquencement des PI-containers après chaque $MaxShuffleIt$ itérations
  $LocalIteration$ ++
  $GlobalIteration$ ++
Fin Tant Que
Sortie : S_{BEST}

Agent de planification #3 : (TS-SA)

Dans le troisième et dernier agent de planification, le séquencement des destinations des wagons est géré à l'aide de la méta-heuristique Tabu Search. Comme présenté dans l'algorithme 4.5, à chaque itération, l'un des trois mouvements de recherche locale est effectué. Si le déplacement est dans la *tabu list*, il sera annulé, sinon, il sera ajouté à la *tabu list*. Ensuite, le *SA_Algorithm()* est appelé pour améliorer la planification des véhicules.

Algorithme 4.5 : Algorithme du TS-SA proposé

Entrée : solution $S_{INITIAL}$ reçue de l'agent synchroniseur
Initialiser : $MaxGlobalIt$, $Tmpr$, $MaxShuffleIt$
Initialiser : $S_{BEST} = S = S_{INITIAL}$
Initialiser : la *tabu list* : $TL = \emptyset$
$GlobalIteration = 0$
Tant Que $GlobalIteration < MaxGlobalIt$ **Faire**
  Effectuer un mouvement de recherche locale sur le séquencement des destinations des wagons :
  Insertion, Permutation ou Insertion+Permutation (Probabilité de chaque mouvement = 1/3)
  Si le mouvement de recherche locale est dans la TL **Alors**
    Annuler le mouvement de recherche locale
  Sinon
    Mettre à jour la tabu list TL
  Fin Si
  Exécuter : SA_Algorithm() pour améliorer la planification des véhicules (Voir Algorithme 4.2)
  Mélanger le séquencement des PI-containers après chaque $MaxShuffleIt$ itérations
  $GlobalIteration$ ++
Fin Tant Que
Retourner : S_{BEST}

4.4 Résultats expérimentaux

4.4.1 Implémentation et instances

Pour valider le MAS et le MILP proposés, les trois méta-heuristiques hybrides (VNS-SA, GRASP-SA et TS-SA) sont développées en Java. Les agents sont créés et implémentés dans la plate-forme JADE (Java Agent DEvelopment framework) et tous les tests sont exécutées sur un PC doté d'un processeur Intel$^{(R)}$ Core$^{(TM)}$ i3 de 2.40 GHz et 4GB de RAM. Le modèle MILP est implémenté dans IBM ILOG CPLEX Solver (version 12.8). Pour évaluer la stabilité du MAS, les tests sont effectués en 5 réplications pour chacune des 34 instances et la valeur moyenne de la fonction objective est présentée. Une limite de temps de 3600s est définie pour CPLEX. La valeur du facteur de pondération α, β et γ peut être définie par l'utilisateur. Dans cette étude, ces facteurs de pondération sont fixés à $\alpha = 0.995$, $\beta = 0.003$ et $\gamma = 0.002$ afin de garantir l'utilisation d'un nombre minimal de wagons tout en minimisant les distances totales parcourues par les PI-containers ainsi que les retards des véhicules.

Afin d'évaluer les deux modèles proposés, plusieurs instances sont générées aléatoirement en modifiant le nombre de PI-containers et leurs longueurs L_i, le nombre de

destinations D et le nombre de véhicules H. La variation de ces paramètres est présentée dans le tableau 4.1.

Tableau 4.1 – Variation des paramètres dans les instances générées

Paramètres	**Valeurs**
Wagons (W)	Le train est composé de 30 wagons vides dans toutes les instances.
PI-containers (N)	20 instances avec 10 PI-containers et 14 instances avec 15 PI-containers.
Longueurs des PI-containers (L_i) en mètres (m)	Les longueurs sont sélectionnées à partir des valeurs suivantes : {1.2, 2.4, 3.6, 4.8, 6, 12}.
Destinations (D)	Le nombre de quais est sélectionné à partir des valeurs suivantes : {2, 3, 4, 5, 6}.
véhicules (H)	Le nombre de véhicules est sélectionné dans les valeurs suivantes : {2, 3, 4, 5}
Longueur utile des véhicules	$13.5m$
Nombre des quais dans le Road-Rail	28
Longueur utile d'un wagon (Q)	$18m$
Longueur totale d'un wagon	$20m$

Pour les paramètres des agents de planification dans le MAS, le nombre total d'itérations $MaxGlobalIt$ est fixé à 250 pour les instances 1 à 20 et à 1000 pour les instances restantes. La température initiale $Tmpr$ du SA est fixée à 1 et elle est diminuée de 10^{-3} pour les instances de 1 à 20 et de $7,5 \times 10^{-4}$ pour les autres d'instances. Pour l'agent de planification VNS-SA, le nombre de structures de voisinage varie entre 1 à 3 en fonction de la taille des instances. $ShuffleMaxIt$ est fixé à 10. Enfin, le nombre d'itérations $MaxLocalIt$ dans l'agent de planification GRASP-SA est fixé à 50 pour les instances 1 à 20 et à 150 pour les instances restantes.

Afin de garantir de bonnes performances, la définition des valeurs des paramètres dans les algorithmes est très importante. Ces valeurs sont définies pour cette étude de cas après un réglage des paramètres. Ils garantissent le meilleur compromis entre la performance et la qualité des solutions. Ces paramètres ont été définis après avoir effectué des tests sur plusieurs petites instances en définissant un paramètre, puis en évaluant son impact sur la solution. Par exemple, les paramètres d'itérations ont été définis après l'observation de la convergence des algorithmes. Ainsi, une fois la valeur de la fonction objective devient stable, la valeur du paramètre est fixée.

4.4.2 Résultats du MILP et du MAS

Les résultats obtenus sont présentés dans le tableau 4.2. Les valeurs optimales sont présentées en gras. Les paramètres des instances (N : nombre de PI-containers, D : nombre de destinations et H : nombre de véhicules) sont présentés dans les quatre premières colonnes. Les résultats du MILP sur le solveur CPLEX sont présentés dans les colonnes 5 à 9, qui indiquent respectivement le nombre de wagons utilisés, la distance totale parcourue par les PI-containers, le retard des véhicules, la fonction objective et le temps de calcul du MILP.

Tableau 4.2 – Comparaison entre le MILP et le MAS

Instances				**MILP**					**MAS**					**MAS vs MILP**
#	**N**	**D**	**H**	**Wag**	**Dist**	**Retard**	**Obj**	**Temps (s)**	**Wag**	**Dist**	**Retard**	**Obj**	**Temps (s)**	
1	10	2	2	2	64,28	0,0	**2,183**	1,42	2	64,28	0,0	**2,183**	1,38	**0,00%**
2	10	3	2	3	127,14	0,0	**3,366**	4,05	3	127,14	0,0	**3,366**	1,41	**0,00%**
3	10	4	2	4	144,28	0,0	**4,413**	8,34	4	144,28	0,0	**4,413**	1,72	**0,00%**
4	10	5	2	5	180,71	0,0	**5,517**	25,67	5	180,71	0,0	**5,517**	1,33	**0,00%**
5	10	6	2	6	200,00	0,0	**6,570**	34,89	6	200,00	0,0	**6,570**	1,38	**0,00%**
6	10	2	3	2	83,57	0,0	**2,241**	1,16	2	83,57	0,0	**2,241**	1,48	**0,00%**
7	10	3	3	3	124,29	0,0	**3,358**	5,89	3	125,29	0,0	3,361	1,49	0,09%
8	10	4	3	4	142,86	0,0	**4,409**	43,33	4	144,00	0,0	4,412	1,39	0,08%
9	10	5	3	5	200,71	0,0	**5,577**	29,00	5	200,71	0,0	**5,577**	1,28	**0,00%**
10	10	6	3	6	141,43	0,0	**6,394**	48,03	6	142,29	0,0	6,397	1,42	0,04%
11	10	2	4	2	80,71	0,0	**2,232**	1,33	2	80,71	0,0	**2,232**	1,50	**0,00%**
12	10	3	4	3	120,00	0,0	**3,345**	23,97	3	120,00	0,0	**3,345**	1,41	**0,00%**
13	10	4	4	4	100,71	0,0	**4,282**	32,17	4	101,71	0,0	4,285	1,39	0,07%
14	10	5	4	5	160,00	0,0	**5,455**	536,34	5	160,00	0,0	**5,455**	1,47	**0,00%**
15	10	6	4	6	122,14	0,0	**6,336**	169,27	6	124,71	0,0	6,344	1,42	0,12%
16	10	2	5	3	103,57	0,0	**3,296**	48,16	3	105,43	0,0	3,301	1,38	0,17%
17	10	3	5	4	101,43	0,0	**4,284**	34,33	4	110,57	1,0	4,314	1,44	0,69%
18	10	4	5	4	120,71	0,0	**4,342**	42,33	4	125,00	0,0	4,355	1,35	0,30%
19	10	5	5	5	123,57	0,0	**5,346**	425,45	5	137,43	0,0	5,387	1,39	0,78%
20	10	6	5	6	102,86	0,0	**6,279**	199,94	6	114,14	0,0	6,312	3,43	0,54%
21	15	2	3	3	162,50	0,0	**3,473**	7,09	3	165,64	0,0	3,482	3,02	0,27%
22	15	3	3	4	221,79	0,0	**4,645**	123,25	4	223,21	0,0	4,650	3,18	0,09%
23	15	4	3	4	224,64	0,0	**4,654**	39,77	4	225,64	0,0	4,657	3,19	0,06%
24	15	5	3	5	302,50	0,0	**5,883**	324,38	5	302,50	0,0	**5,883**	3,15	**0,00%**
25	15	6	3	6	263,21	0,0	**6,760**	122,95	6	263,79	0,0	6,761	3,15	0,03%
26	15	2	4	4	141,79	0,0	**4,405**	295,63	4	170,93	0,0	4,493	3,25	1,98%
27	15	3	4	3	180,36	0,0	**3,526**	274,25	3	180,36	0,0	**3,526**	3,28	**0,00%**
28	15	4	4	4	205,36	0,0	**4,596**	163,48	4	210,50	0,0	4,611	3,37	0,34%
29	15	5	4	5	285,36	0,0	**5,831**	591,11	5	286,07	2,0	5,837	3,31	0,11%
30	15	6	4	6	246,07	0,0	**6,708**	946,58	6	250,64	1,0	6,724	3,27	0,23%
31	15	2	5	4	161,07	0,0	**4,463**	1751,31	4	213,64	0,0	4,621	3,28	3,53%
32	15	3	5	4	223,93	0,0	**4,652**	1636,95	4	237,21	1,0	4,694	3,39	0,90%
33	15	4	5	5	242,50	0,0	5,703*	3600,00	5	247,79	2,0	5,722	3,43	0,35%
34	15	5	5	**	**	**	**	**	5	258,36	0	5,750	3,52	–
Moyenne					**163,82**	**0,00**		**351,27s**		**168,79**	**0,21**		**2,20s**	**0,33%**

* Meilleures solutions trouvées par CPLEX dans 3600s (Non optimales),

** CPLEX ne trouve aucune solution pour ces instances,

Comme on peut le voir dans le tableau 4.2 et la figure 4.6, à mesure que la taille des instances augmente, le temps de calcul dans le modèle MILP devient considérablement élevé. CPLEX a trouvé les solutions optimales pour 32 instances, à l'exception de l'instance 33 pour laquelle le délai de 3600s est dépassé. Pour l'instance 34, CPLEX ne trouve aucune solution. Dans le tableau 4.2, les colonnes 10 à 14 indiquent respectivement le nombre de wagons utilisés, la distance totale parcourue par les PI-containers, le retard des véhicules, la fonction objective et le temps de calcul pour le MAS. La dernière colonne montre la déviation entre la fonction objective du MILP et du MAS. L'écart est calculé comme suit (cette formule sert également à calculer la déviation dans le tableau 4.4) :

$$Dev(i) = 100 \times \left(\frac{Objectif_{MAS} - Objectif_{MILP}}{Objectif_{MILP}} \right)$$

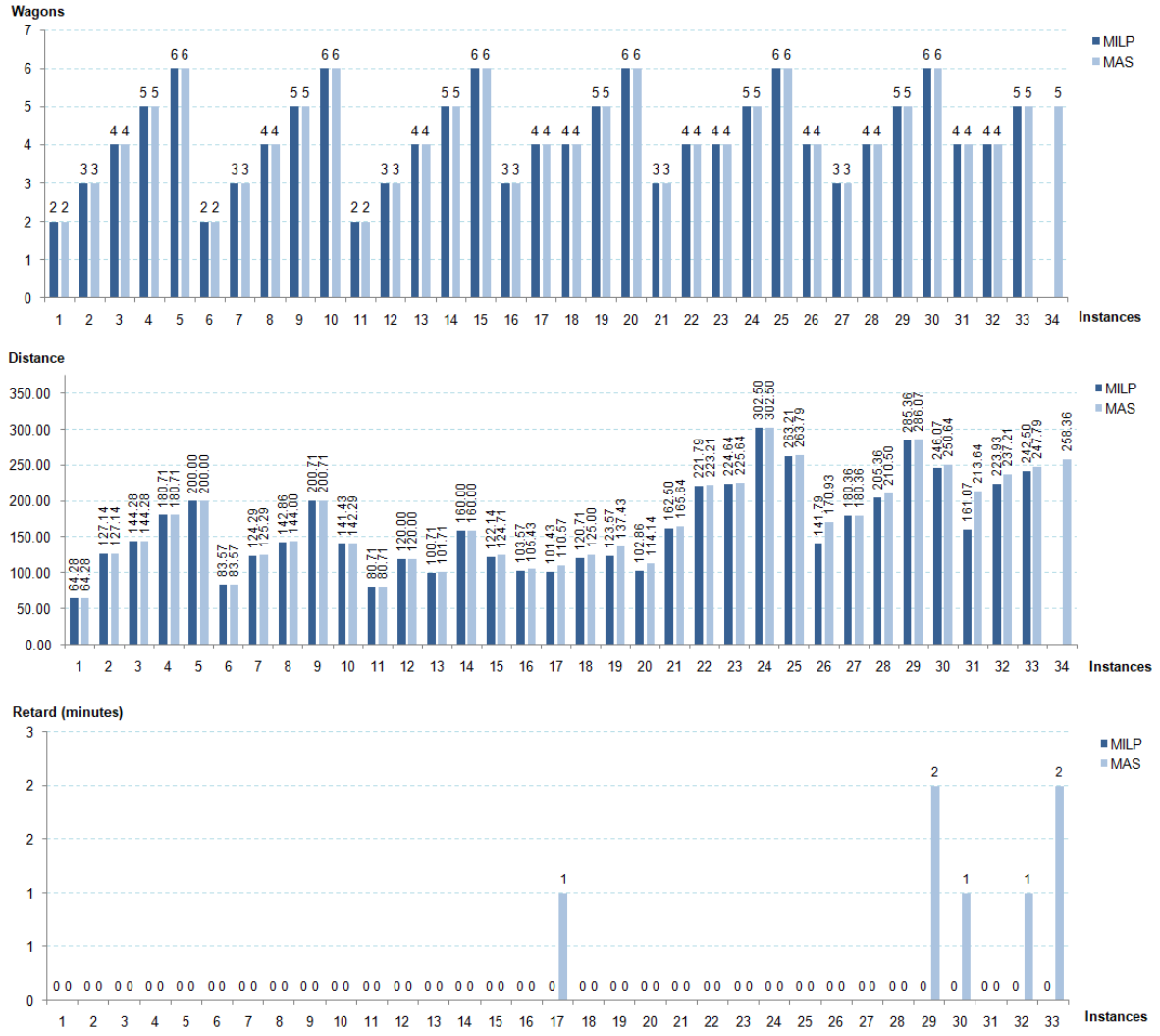

Figure 4.6 – Fonction objective (wagons, distances et retards) dans le MILP et le MAS

Comme on peut le voir dans la dernière colonne (Tableau 4.2), le MAS a trouvé la solution optimale (wagons, distances et retard des véhicules) pour 12 instances en un temps de calcul beaucoup plus court (2,20s) par rapport au solveur CPLEX (351,27s). L'écart moyen total de la fonction objective entre le MILP et le MAS est de 0,33%, ce qui valide les performances de l'architecture MAS proposée également en termes de qualité des solutions. Afin de pousser les méta-heuristiques à leurs limites, plusieurs grandes instances sont ajoutées. Les résultats obtenus sont présentés dans le tableau 4.3.

Tableau 4.3 – Résultats du MAS pour les grandes instances

Instances				MAS				
#	N	D	H	Wag	Dist	Tard	Obj	Temps (s)
35	30	8	8	8	539,29	1,00	9,580	4,51
36	30	15	12	15	512,43	0,00	16,462	5,13
37	40	10	20	11	830,28	0,00	13,436	7,13
38	50	15	25	17	1143,28	1,00	20,347	9,35

Comme le montre la figure 4.7, lorsque le nombre de destinations augmente, nous constatons une augmentation significative des distances parcourues par les PI-containers.

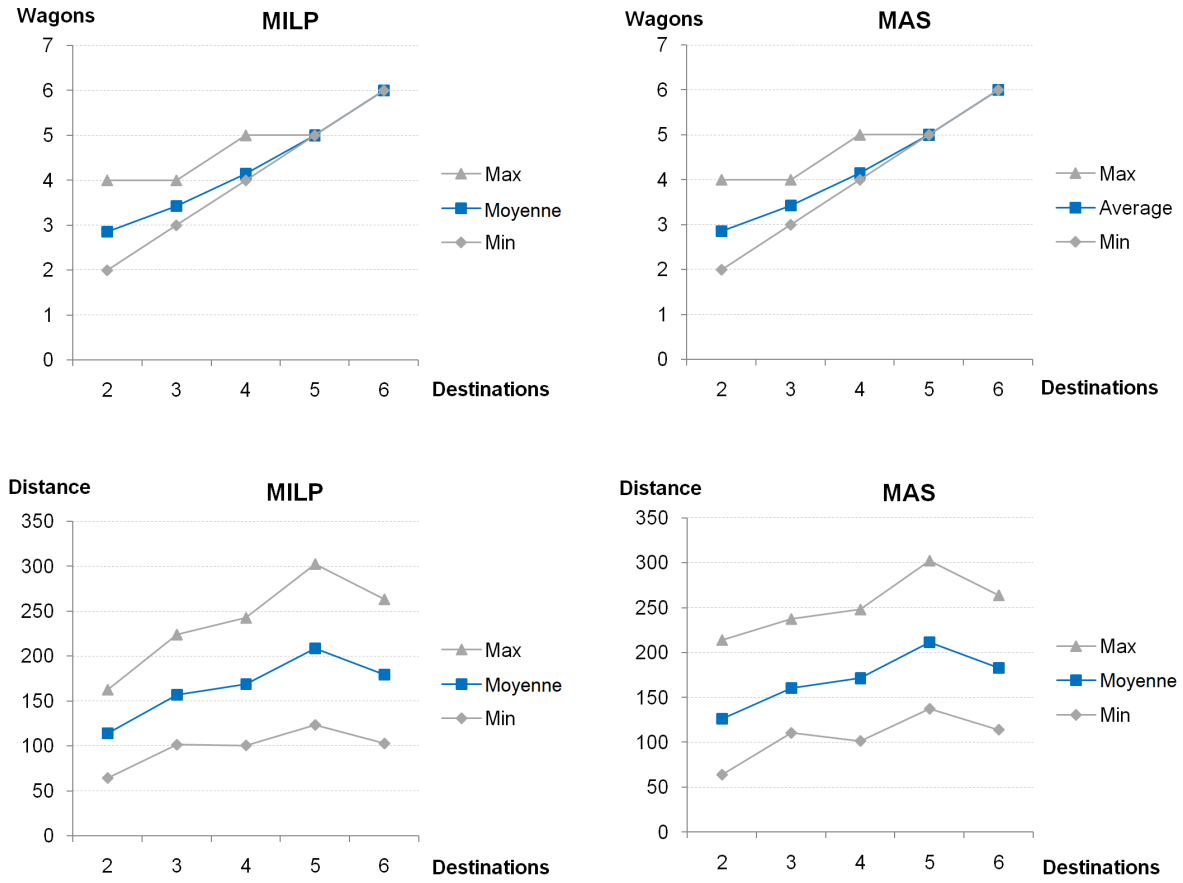

Figure 4.7 – Impact du nombre de destinations sur les wagons utilisés et sur les distances pour le MILP et le MAS

De plus, nous observons que si le nombre de destinations augmente, le nombre de wagons utilisés augmente également. Par exemple, dans les instances 1 à 5, tout en conservant le même nombre de PI-containers, lorsque le nombre de destinations passe de 2 à 6, la distance passe de 64,28 à 200,00 et le nombre de wagons utilisés augmente également. Ces résultats peuvent être expliqués par le fait que, lorsque les PI-containers ont des destinations différentes, on utilisera davantage de wagons, car les PI-containers avec la même destination ne peuvent pas être chargés dans le même wagon, et les PI-containers doivent parcourir des distances supplémentaires pour atteindre leurs wagons.

Afin de confirmer la performance du MAS, un test de comparaison est effectué sur des instances benchmark de la littérature. Etant donné que nous ne pouvions pas trouver d'instances benchmark correspondantes parfaitement à notre problème, nous avons légèrement adapté les trois méta-heuristiques pour résoudre le premier groupe d'instances benchmark ($27 \times 10 = 270$ instances) du problème de cross-docking à quais multiples présenté dans Van Belle et al. (2013). Comme on peut le voir dans la figure 4.8, les trois méta-heuristiques ont fourni des solutions avec des écarts réduits par rapport aux résultats du Tabu Search (TS1) de Van Belle et al. (2013). Les résultats détaillés de l'écart entre les méta-heuristiques du MAS et la Tabu Search proposé par Van Belle et al. (2013) sont présentés dans l'annexe C.

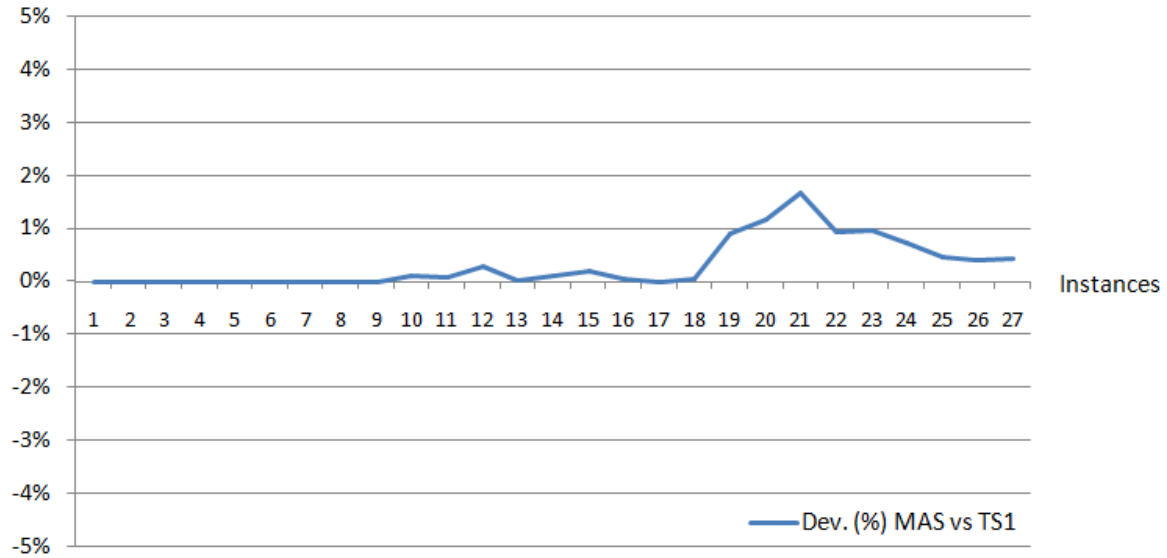

Figure 4.8 – Déviation entre le MAS et le TS1 de Van Belle et al. (2013)

4.5 Evaluation de la robustesse du MAS

Dans cette section, la robustesse du MAS est évaluée sous des perturbations. Dans un contexte réel du PI, différentes perturbations peuvent survenir au cours de la journée. Ces perturbations peuvent être séparées en perturbations internes, telles que l'indisponibilité des quais, des wagons, ou d'un convoyeur, et en perturbations externes liées notamment aux retards des véhicules et des trains. Dans cette section, nous considérons la panne des quais, qui est une perturbation interne pour évaluer la réactivité du MAS.

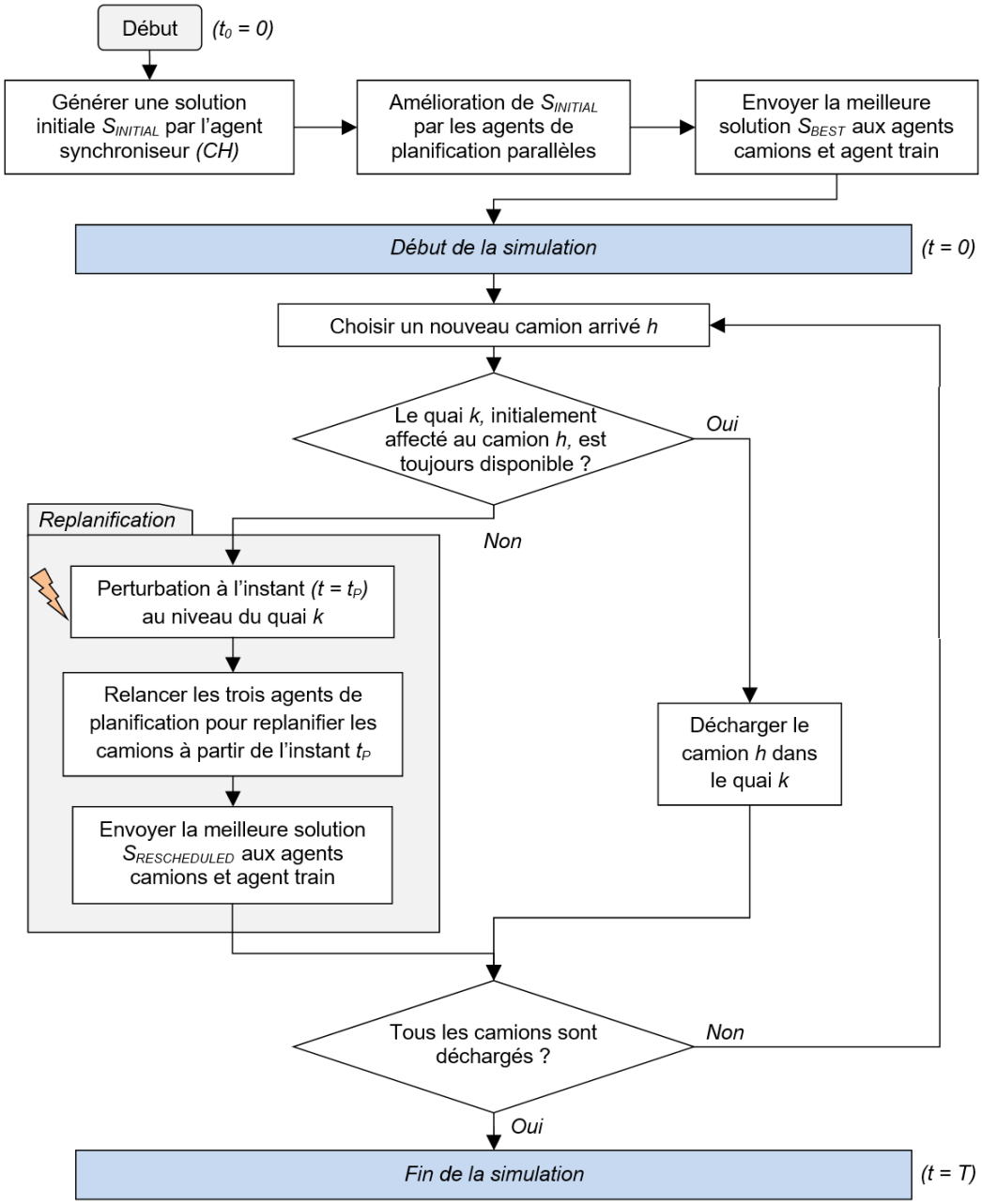

Figure 4.9 – Processus de replanification dans le MAS sous perturbations

L'objectif de cette section est d'évaluer la réactivité et la robustesse du modèle MAS proposé face aux perturbations. Comme illustré dans l'organigramme de la figure 4.9, une solution (S_{BEST}) est générée au début de la journée à l'aide des trois agents de planification hybrides parallèles présentés précédemment. Cette planification est ensuite envoyée aux agents véhicules et train via l'agent synchroniseur. La simulation commence à l'instant zéro ($t_0 = 0$) et chaque véhicule se rend au quai qui lui a été initialement attribué pour le déchargement. Ensuite, deux cas peuvent se produire : le quai est disponible pour le déchargement ou le quai n'est pas disponible pendant un certain temps.

Dans le premier cas, si le quai est toujours disponible, le véhicule sera déchargé sur ce quai. Cependant, dans le second cas, le MAS fait appel aux trois agents de planification pour trouver une nouvelle affectation pour les véhicules à partir de l'instant de perturbation ($t = t_p$). Ensuite, la nouvelle planification est envoyée aux agents véhicules et à l'agent train.

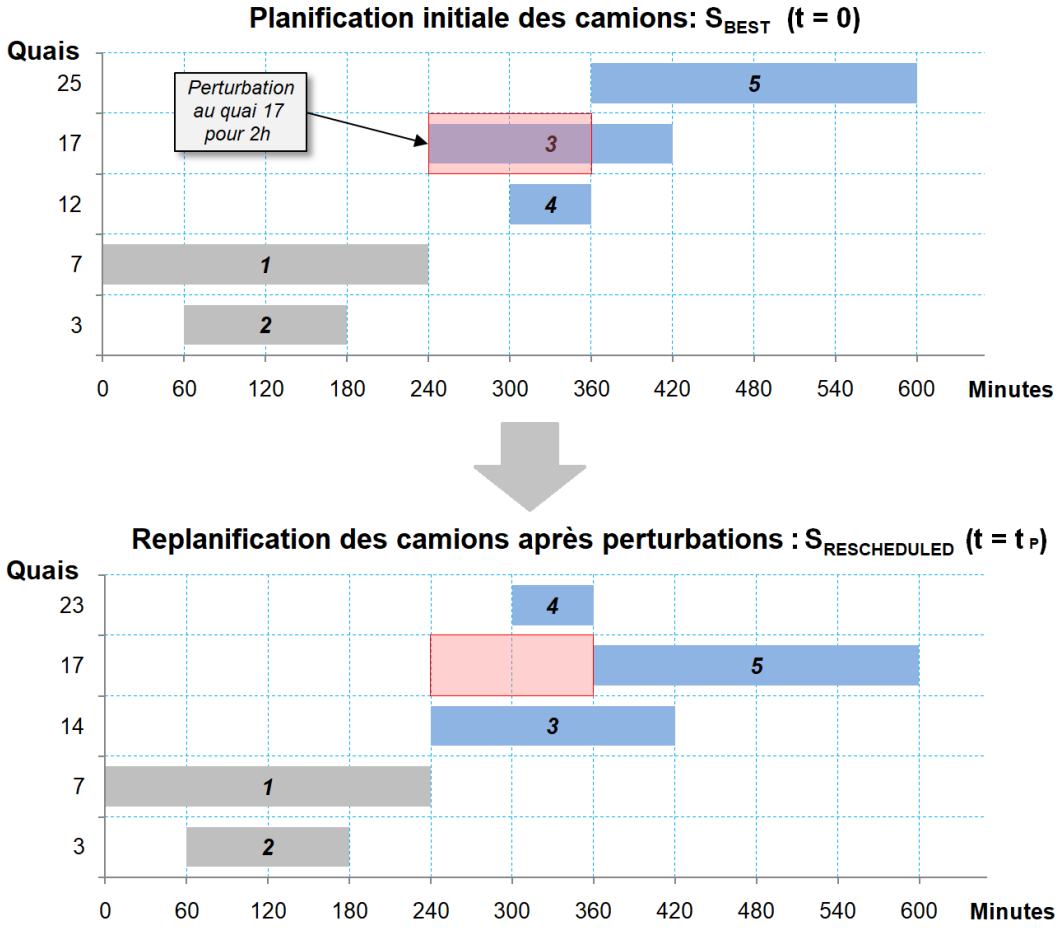

Figure 4.10 – Un exemple illustratif pour la replanification des véhicules après une panne de 2 heures au niveau du quai 17

Comme le montre l'exemple illustratif de la figure 4.10, la perturbation se produit au quai 17 sur lequel le véhicule 3 a été initialement affecté. Ce quai sera indisponible pendant $2h$ de panne. À la fin de la panne, le quai pourra être programmé pour les autres véhicules entrants (par exemple, le véhicule 5 de la figure 4.10). Afin d'éviter tout retard supplémentaire pour le véhicule 3, un réordonnancement est effectué à l'aide des trois agents de planification après avoir été notifiés par l'agent synchroniseur de l'indisponibilité temporaire du dock 17. Une fois la nouvelle planification est disponible, elle est envoyée par l'agent synchroniseur aux véhicules restants.

Tableau 4.4 – Comparaison entre la planification au début de la journée (S_{BEST}) et la replanification à la fin de la journée ($S_{RESCHEDULED}$) dans le modèle MAS

Instances				**S_{BEST}**					**$S_{RESCHEDULED}$**					**Dev. (%)**
#	**N**	**D**	**H**	**Wag**	**Dist**	**Retard**	**Obj**	**Temps (s)**	**Wag**	**Dist**	**Retard**	**Obj**	**Temps (s)**	
1	10	2	2	2	64,28	0,0	2,183	1,38	2	94,00	0,0	2,272	0,82	4,08%
2	10	3	2	3	127,14	0,0	3,366	1,41	3	193,43	0,0	3,565	0,85	5,91%
3	10	4	2	4	144,28	0,0	4,413	1,72	4	207,14	0,0	4,601	0,81	4,27%
4	10	5	2	5	180,71	0,0	5,517	1,33	5	198,43	0,0	5,570	0,90	0,96%
5	10	6	2	6	200,00	0,0	6,570	1,38	6	208,57	0,0	6,596	0,86	0,39%
6	10	2	3	2	83,57	0,0	2,241	1,48	2	153,28	0,0	2,450	0,88	9,33%
7	10	3	3	3	125,29	0,0	3,361	1,49	3	211,86	0,0	3,621	0,92	7,73%
8	10	4	3	4	144,00	0,0	4,412	1,39	4	175,57	0,0	4,507	0,90	2,15%
9	10	5	3	5	200,71	0,0	5,577	1,28	5	222,57	0,0	5,643	0,93	1,18%
10	10	6	3	6	142,29	0,0	6,397	1,42	6	204,43	0,0	6,583	0,88	2,91%
11	10	2	4	2	80,71	0,0	2,232	1,50	2	217,14	0,0	2,641	1,08	18,34%
12	10	3	4	3	120,00	0,0	3,345	1,41	3	225,43	0,0	3,661	0,92	9,46%
13	10	4	4	4	101,71	0,0	4,285	1,39	4	232,71	0,0	4,678	1,15	9,17%
14	10	5	4	5	160,00	0,0	5,455	1,47	5	177,57	0,0	5,508	0,96	0,97%
15	10	6	4	6	124,71	0,0	6,344	1,42	6	181,57	0,0	6,515	1,07	2,69%
16	10	2	5	3	105,43	0,0	3,301	1,38	3	204,43	0,0	3,598	1,01	9,00%
17	10	3	5	4	110,57	1,0	4,314	1,44	4	200,71	13,0	4,608	1,24	6,83%
18	10	4	5	4	125,00	0,0	4,355	1,35	4	210,85	0,0	4,613	1,12	5,91%
19	10	5	5	5	137,43	0,0	5,387	1,39	5	199,71	1,0	5,576	0,91	3,51%
20	10	6	5	6	114,14	0,0	6,312	3,43	6	148,29	0,0	6,415	1,34	1,62%
21	15	2	3	3	165,64	0,0	3,482	3,02	3	316,92	0,0	3,936	2,68	13,03%
22	15	3	3	4	223,21	0,0	4,650	3,18	4	299,07	0,0	4,877	2,79	4,89%
23	15	4	3	4	225,64	0,0	4,657	3,19	4	405,35	0,0	5,196	2,71	11,58%
24	15	5	3	5	302,50	0,0	5,883	3,15	5	322,64	0,0	5,943	2,72	1,03%
25	15	6	3	6	263,79	0,0	6,761	3,15	6	308,64	0,0	6,896	2,95	1,99%
26	15	2	4	4	170,93	0,0	4,493	3,25	4	347,64	0,0	5,023	2,97	11,80%
27	15	3	4	3	180,36	0,0	3,526	3,28	3	358,07	0,0	4,059	3,11	15,12%
28	15	4	4	4	210,50	0,0	4,611	3,37	4	325,07	0,0	4,955	3,00	7,45%
29	15	5	4	5	286,07	2,0	5,837	3,31	5	337,36	0,0	5,987	3,59	2,57%
30	15	6	4	6	250,64	1,0	6,724	3,27	6	374,93	0,0	7,095	3,22	5,52%
31	15	2	5	4	213,64	0,0	4,621	3,28	4	351,35	0,0	5,034	3,49	8,94%
32	15	3	5	4	237,21	1,0	4,694	3,39	4	372,92	50,0	5,199	3,30	10,76%
33	15	4	5	5	247,79	2,0	5,722	3,43	5	356,93	1,0	6,048	3,78	5,69%
Moyenne					**168,79**	**0,21**	**4,698**	**2,20s**		**252,87**	**1,97**	**4,954**	**1,81s**	**6,27%**

Les simulations sont effectuées sur un horizon temporel de 10 heures en 5 réplications. La valeur moyenne est présentée pour chaque objectif. À chaque réplication, une pertur-

bation est générée aléatoirement sur un quai pour une durée de 1 à 3 heures. Les résultats détaillés de la simulation pour le MAS en l'absence de perturbations sont présentés dans le tableau 4.4 à partir des colonnes 5 à 9. Pour le scénario avec perturbations, les résultats sont présentés dans les colonnes 10 à 14. L'écart de la fonction objective entre les deux solutions S_{BEST} et $S_{RESCHEDULED}$ sont présentés dans la dernière colonne.

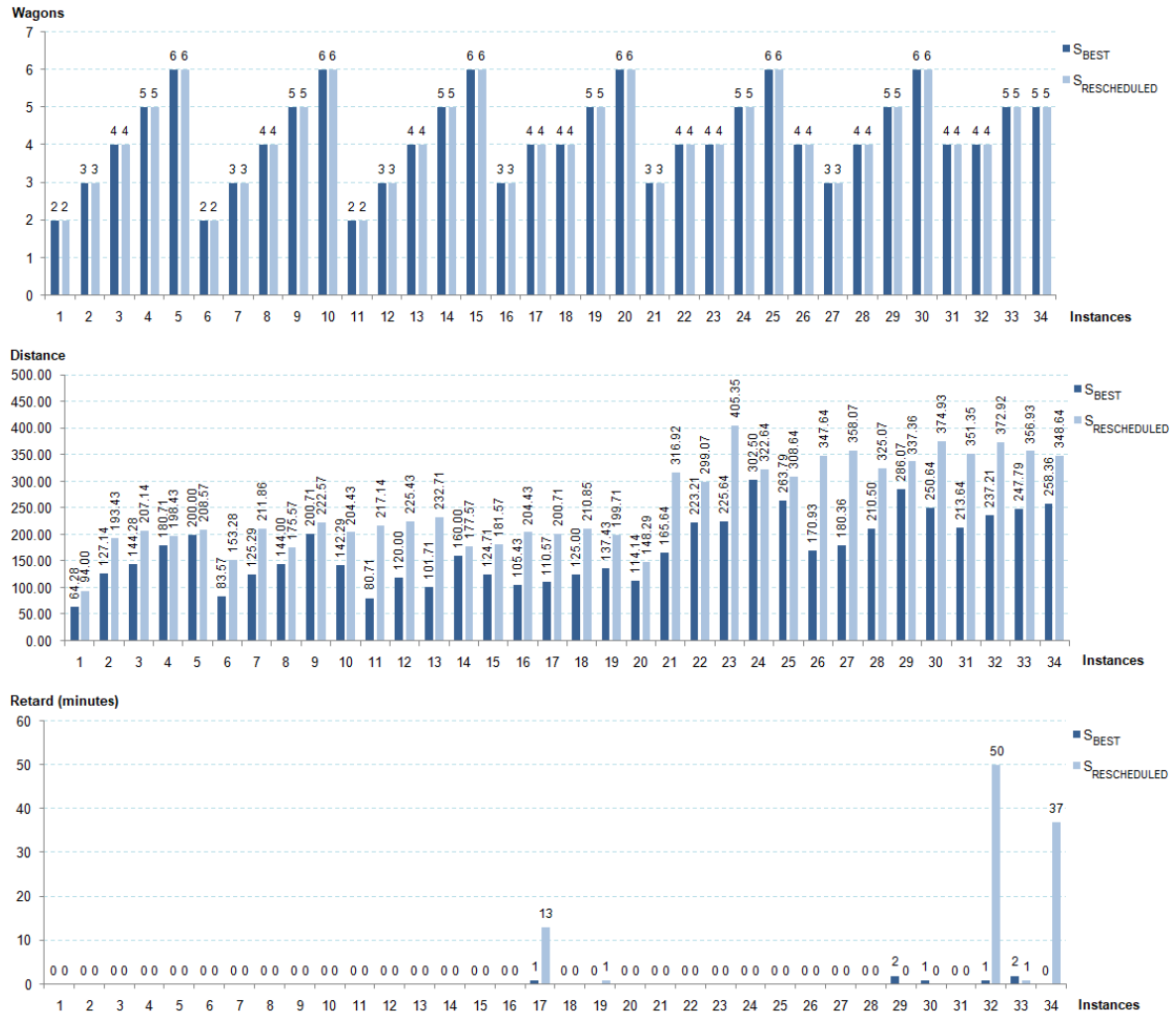

Figure 4.11 – Fonctions objectives (wagons, distance et retard) dans le MAS avec et sans perturbations

Comme on peut le voir dans tableau 4.4 et dans la figure 4.11, les perturbations n'ont pas d'impact sur le nombre de wagons utilisés qui sont restés les mêmes pour toutes les instances. Cependant, la distance moyenne a considérablement augmenté, passant de 171,42 à 255,68. Les retards des véhicules sont également restés stables, à quelques exceptions (instances 17 et 32). Les valeurs moyennes dans la dernière ligne du tableau 4.4 sont calculées pour toutes les instances 1 à 34. L'écart moyen total de la fonction objective est de 6,26%. Le temps de calcul indiqué dans la colonne 14 du tableau 4.4 est le temps

passé par les agents de planification parallèles à trouver la nouvelle replanification des véhicules restants après la perturbation.

La figure 4.12 présente les fonctions objectives des solutions replanifiées pour 5 réplications pour chaque instance. Afin d'évaluer l'impact des perturbations sur la qualité de la solution (fonction objective) fournie par le MAS, nous effectuons une analyse de variance ANOVA à un facteur pour chacune des 34 instances après avoir tester la normalité pour les 5 réplications de chaque instance en utilisant le test de Kolmogorov-Smirnov. Les p-value ANOVA obtenues pour chaque instance sont présentées dans la figure 4.13. Comme on peut le constater, les perturbations ont un impact significatif sur la solution initiale puisque p-value $< 0,05$ pour la plupart des instances (sauf pour 9 instances). Après des tests supplémentaires, il semble que la qualité des solutions commence à être affectée lorsque l'indisponibilité des quais augmente considérablement.

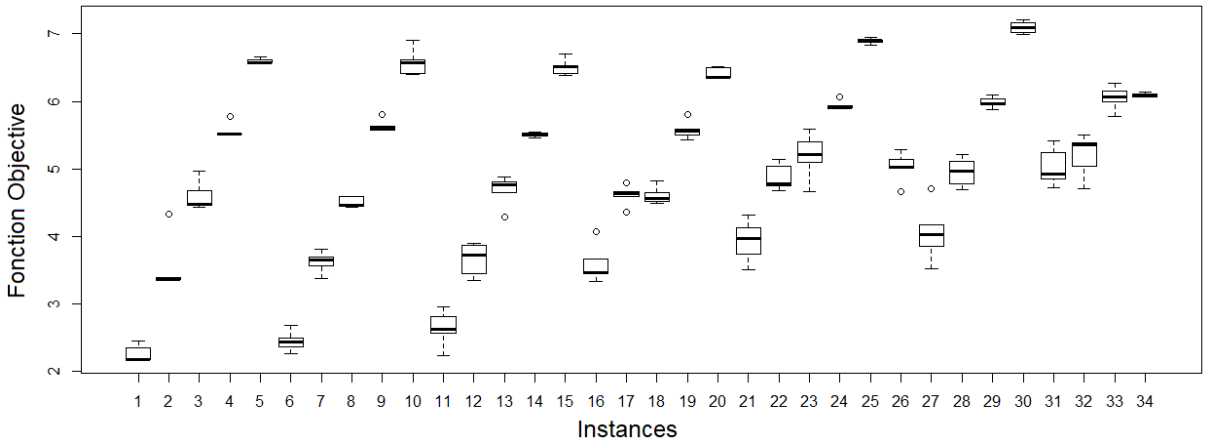

Figure 4.12 – Fonctions objectives dans les 5 réplications de chaque instance

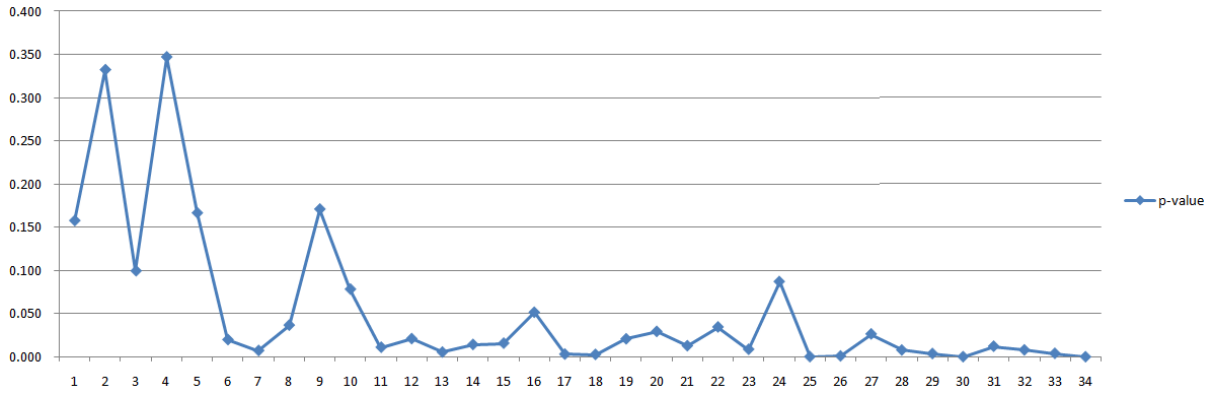

Figure 4.13 – Les p-values de l'ANOVA pour chacune des instances

4.6 Conclusion

Dans ce chapitre, un modèle multi-agents a été proposé pour la planification des véhicules et l'affectation et le regroupement des PI-containers dans un Road-Rail PI-hub. Afin de maintenir des solutions de qualité, l'architecture du MAS proposé exécute trois agents de planification parallèles utilisant trois méta-heuristiques hybrides différentes (VNS-SA, GRASP-SA et TS-SA). Pour assurer la durabilité, l'objectif du modèle a été conçu pour optimiser une somme pondérée de trois objectifs : minimiser le nombre de wagons utilisés, la distance totale parcourue par les PI-containers des quais aux wagons, en plus du retard total des véhicules. Les performances du MAS proposé dans un environnement statique ont ensuite été évaluées à l'aide d'un modèle mathématique linéaire en nombres entiers mixtes (MILP). Les résultats valident les performances du MAS. Enfin, la réactivité du MAS a été testée face aux perturbations internes. Les résultats ont montré que l'écart total moyen était de 6,27% par rapport à la planification initial au début de la journée, ce qui illustre la capacité du MAS à réagir efficacement dans un environnement perturbé. Ceci a été validé par une analyse ANOVA. Les approches de résolution présentées dans ce chapitre ont fait l'objet d'une publication dans une revue internationale (Chargui et al., 2019c).

L'objective du prochain chapitre est de mettre en évidence la durabilité et la consommation énergétique dans le Rail-Road PI-hub à travers des modèles multi-objectifs.

Chapitre 5

Planification multi-objective et durable des opérations dans un cross-dock Rail-Road PI-hub

---

Malgré le fait que le modèle MAS présenté dans le chapitre précédent permet d'assurer une réactivité en temps-réel face aux perturbations imprévues, cela n'assure pas directement la durabilité dans le PI-hub. Ainsi, en intégrant la consommation énergétique dans la fonction objective, les modèles présentés dans ce chapitre permettront d'assurer la durabilité en minimisant à la fois le coût d'utilisation des véhicules ainsi que la consommation énergétique des PI-conveyors à travers plusieurs approches multi-objectives qui n'ont pas été abordées dans la littérature du Rail-Road PI-hub.

5.1 Introduction

Ce chapitre est consacré à l'étude du Rail-Road cross-dock PI-hub. Un modèle mathématique multi-objectif MO-MIP est proposé pour la planification durable des véhicules et le groupement des PI-containers. Deux fonctions objectives sont considérées. La première fonction consiste à minimiser le coût d'utilisation des véhicules sortants pour chaque destination, tandis que le deuxième objectif est de minimiser la consommation d'énergie des PI-conveyors. Le modèle MO-MIP proposé est résolu d'abord sur CPLEX en utilisant la programmation lexicographique. Puis, une heuristique de construction et deux métaheuristiques multi-objectives hybrides sont proposées : MO-VNSSA (Multi-Objective Variable Neighborhood Search - Simulated Annealing) et MO-VNSTS (Multi-Objective Variable Neighborhood Search - Tabu Search). Les résultats obtenus sont ensuite analysés.

5.2 Présentation du problème

Cette section présente une description du processus de cross-docking dans le Rail-Road PI-hub avec ses différentes fonctionnalités. Comme mentionné dans le chapitre 1, le Rail-Road PI-hub est un type de PI-hub utilisé pour transférer des PI-containers du train vers les véhicules sortants. Il est composé principalement d'un PI-sorter et de deux zones de manœuvre devant le train et les quais de chargement (Figure 5.1).

Le processus de cross-docking commence par le déchargement des PI-containers des wagons. Ensuite, les PI-containers sont acheminés vers les quais d'expédition, regroupés par destination, puis chargés dans les véhicules sortants.

Dans le contexte de la planification des cross-docks, les objectifs et les contraintes liés à la durabilité n'ont pas été largement abordés dans la littérature. A notre connaissance, les auteurs dans Dulebenets (2018) et Shahram fard and Vahdani (2019) sont les seules à prendre en compte les contraintes de durabilité ou de consommation d'énergie dans les terminaux de cross-docking classiques. Par conséquent, dans le contexte du PI, il est toujours nécessaire de mener une recherche en tenant compte des objectifs et des contraintes durables pour les terminaux de cross-docking Rail-Road.

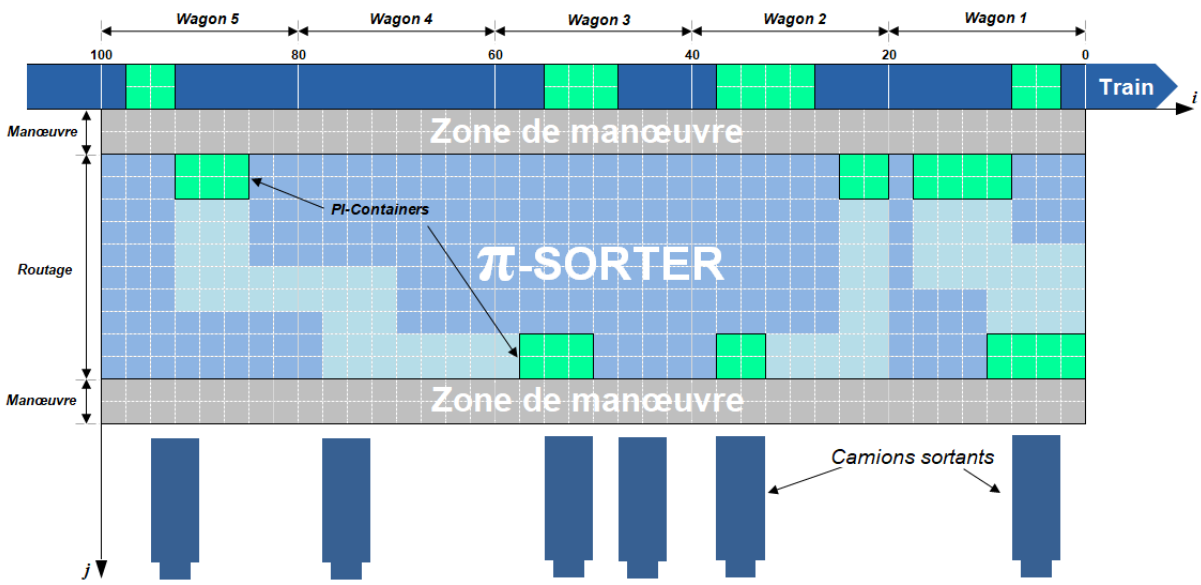

Figure 5.1 – Le Rail-Road PI-hub

Pour la présente étude, dans le contexte de la planification des véhicules dans un cross-dock Rail-Road PI-hub, assurer la durabilité consiste à minimiser l'utilisation de l'énergie consommée par les PI-conveyors pour déplacer les PI-containers entre les véhicules sortants et les wagons du train ainsi que la minimisation du coût des véhicules utilisés pour desservir chaque destination. L'énergie consommée est ensuite calculée en recherchant le nombre d'unités de PI-conveyors parcourus par les PI-containers entre les wagons et les véhicules. Ces deux objectifs sont contradictoires. En effet, minimiser l'énergie de PI-conveyors pourrait conduire à l'utilisation de plusieurs véhicules pour la même destination, ce qui n'est pas un choix optimal, car le coût d'utilisation de véhicules supplémentaires pour la même destination est beaucoup plus coûteux par rapport à l'énergie des PI-conveyors. En outre, les véhicules qui seront utilisés auront un taux de chargement très faible. Pour faire face à ce problème, des approches multi-objectifs sont proposées et détaillées dans les deux sections suivantes : un modèle mathématique multi-objectifs (MO-MIP) et deux méta-heuristiques multi-objectives (MO-VNSSA et MO-VNSTS). Les principales hypothèses considérées pour résoudre ce problème sont les suivantes :

— Il y a deux fonctions objectives à minimiser : F_1 (le coût d'utilisation des véhicules sortants) et F_2 (le coût de la consommation d'énergie pour les PI-conveyors).
— Les deux fonctions objectives sont classées dans un ordre défini par le décideur : par exemple, minimiser F_1 puis F_2 dans cette étude.
— Le train décharge des PI-containers de différentes longueurs et destinations.

— Chaque véhicule sortant ne doit charger que des PI-containers ayant la même destination.

5.3 Modèle mathématique

Cette section présente le modèle mathématique du problème de planification durable des véhicules et le groupement des PI-containers dans un Rail-Road PI-hub. Le problème est formulé sous forme d'un modèle de programmation multi-objectifs (MO-MIP). Le but est de trouver le regroupement des PI-containers ainsi que l'affectation et la planification des véhicules sur les quais. Étant donné les longueurs, les destinations, les positions des PI-containers dans le train et l'ensemble de véhicules sortants disponibles, l'objectif est de trouver le meilleur regroupement des PI-containers dans les véhicules, ainsi que l'affectation et la planification des véhicules aux quais tout en minimisant deux objectifs contradictoires : le coût énergétique lié à l'utilisation de PI-conveyors pour traiter les PI-containers et, en même temps, le coût d'utilisation des véhicules sortants. Les principaux paramètres d'entrée/données et les variables de sortie sont présentés dans la figure 5.2.

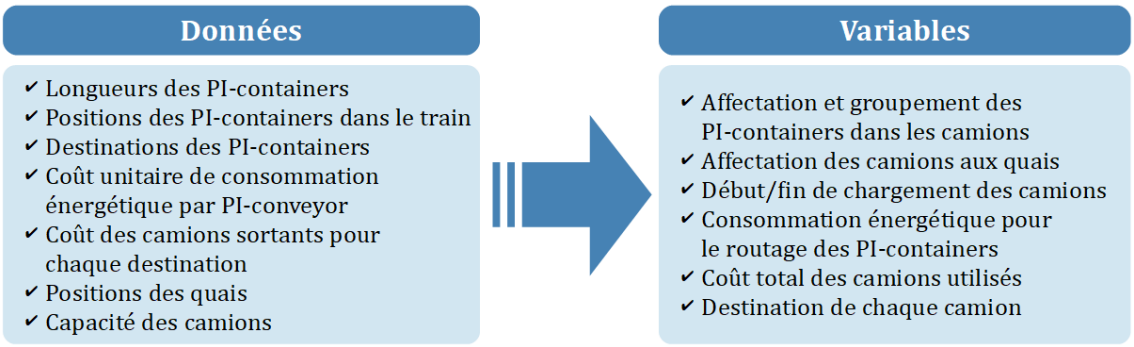

Figure 5.2 – Données et variable du modèle MILP proposé

5.3.1 Données et paramètres du problème

N Nombre total de PI-containers dans le train

K Nombre total de quais d'expédition dans la section Rail-Road

D Nombre total de destinations de PI-containers

H Nombre total de véhicules sortants disponibles

i, j Indices des PI-containers à transférer du train aux véhicules entrants ($i, j = 1...N$)

k Index des quais sortants dans la section Rail-Road ($k = 1...K$)

d Index des destinations des PI-containers à charger dans les véhicules sortants ($d = 1...D$)

h, g Indices des véhicules sortants ($h, g = 1...H$)

C_E Coût de la consommation d'énergie pour une unité de PI-conveyor

C_d^T Coût d'utilisation d'un véhicule sortant pour la destination d

I Temps pour charger un PI-container dans un véhicule sortant

V Temps pour changement de véhicule dans le quai

Y Longueur verticale de la section Rail-Road

P_i Position du coin inférieur gauche des PI-containers dans les wagons du train à partir de l'axe droit de la section Rail-Road

R_k Position du quai sortant k à partir de l'axe de droite de la section Rail-Road

L_i Longueur d'un PI-container à décharger du train

G_{di} Matrice binaire à deux dimensions ($D \times N$) contenant la destination de chaque PI-container, où :

$$G_{di} = \begin{cases} 1 & \text{Si } d \text{ est la destination du PI-container } i \\ 0 & \text{Sinon} \end{cases}$$

Q Capacité des véhicules sortants

M Un grand nombre positif. La valeur minimale de M doit être supérieure à : $M > Max(2\times$ Longueur d'un bloc de 5 wagons $+Y \times Max_{i=1...N}(L_i)$, *Horizon de planification*$)$

5.3.2 Variables de décision

5.3.2.1 Variables binaires

$$x_{hk} = \begin{cases} 1 & \text{Si le véhicule sortant } h \text{ est affecté au quai d'expédition } k \\ 0 & \text{Sinon} \end{cases}$$

$$p_{ih} = \begin{cases} 1 & \text{Si le PI-container } i \text{ est affecté au véhicule sortant } h \\ 0 & \text{Sinon} \end{cases}$$

$$a_{hd} = \begin{cases} 1 & \text{Si } d \text{ est la destination du véhicule sortant } h \\ 0 & \text{Sinon} \end{cases}$$

$$v_h = \begin{cases} 1 & \text{Si le véhicule sortant } h \text{ est utilisé} \\ 0 & \text{Sinon} \end{cases}$$

$$n_{hg} = \begin{cases} 1 & \text{Si les véhicules sortants } h \text{ et } g \text{ sont affectés au même quai d'expédition} \\ & \text{et } h \text{ est un prédécesseur de } g \\ 0 & \text{Sinon} \end{cases}$$

5.3.2.2 Variables entières

z_{ih} La surface (nombre d'unités de PI-conveyors) parcourue par le PI-container i pour arriver à sa destination au véhicule sortant h

5.3.2.3 Variables continues

b_h Heure de début du chargement du véhicule sortant h

q_h Heure de fin du chargement du véhicule sortant h

5.3.3 Fonction objective

La fonction objective de ce modèle MO-MIP consiste à minimiser deux objectifs contradictoires :

$$\begin{aligned} &\text{Minimiser :} \quad F_1 = \sum_{h=1}^{H} \sum_{d=1}^{D} C_d^T a_{hd} \\ &\text{Minimiser :} \quad F_2 = C^E \sum_{i=1}^{N} \sum_{h=1}^{H} z_{ih} \end{aligned} \tag{5.1}$$

Où :

F_1 Le coût total des véhicules utilisés pour toutes les destinations

F_2 Coût total de la consommation d'énergie pour les PI-conveyors

5.3.4 Contraintes

$$\sum_{h=1}^{H} p_{ih} = 1 \qquad (\forall i = 1...N) \tag{5.2}$$

$$\sum_{i=1}^{N} p_{ih} L_i \leq Q \qquad (\forall h = 1...H) \tag{5.3}$$

$$p_{ih} + p_{jh} \leq \sum_{d=1}^{D} G_{di} G_{dj} + 1 \qquad (\forall i, j = 1...N, \forall h = 1...H : i \neq j) \tag{5.4}$$

$$p_{ih} \leq v_h \qquad (\forall i = 1...N, \forall h = 1...H) \tag{5.5}$$

$$a_{hd} \leq G_{di} + 1 - p_{ih} \qquad (\forall i = 1...N, \forall h = 1...H, \forall d = 1...D) \tag{5.6}$$

$$v_h = \sum_{d=1}^{D} a_{hd} \qquad (\forall h = 1...H) \tag{5.7}$$

$$z_{ih} \geq 2|P_i - R_k| + Y L_i - M(2 - (p_{ih} + x_{hk})) \qquad (\forall i = 1...N, \forall k = 1...K, \forall h = 1...H) \tag{5.8}$$

$$\sum_{k=1}^{K} x_{hk} = v_h \qquad (\forall h = 1...H) \tag{5.9}$$

$$x_{hk} + x_{gk} - 1 \leq n_{hg} + n_{gh} \qquad (\forall k = 1...K, \forall h, g = 1...H, h \neq g) \tag{5.10}$$

$$n_{hg} + n_{gh} \leq 1 \qquad (\forall h, g = 1...H) \tag{5.11}$$

$$b_g = Max(0, q_h + V - M(1 - n_{hg})) \qquad (\forall h, g = 1...H) \tag{5.12}$$

$$q_h = b_h + I \sum_{i=1}^{N} p_{ih} \qquad (\forall h = 1...H) \tag{5.13}$$

$$x_{hk}, p_{ih}, a_{hd}, v_h, n_{h,g} \in \{0, 1\} \qquad (\forall i = 1...N, \forall k = 1...K, \forall d = 1...D, \forall h, g = 1...H) \tag{5.14}$$

$$z_{ih} \geq 0, b_h \geq 0, q_h \geq 0 \qquad (\forall i = 1...N, \forall h = 1...H) \tag{5.15}$$

Les contraintes (5.2) vérifient que chaque PI-container est affecté à un seul véhicule sortant. Les contraintes (5.3) garantissent que les PI-containers ne dépassent pas la capacité des véhicules sortants. Les contraintes (5.4) assurent que chaque véhicule sortant, s'il est utilisé, ne peut charger que des PI-containers ayant la même destination. Les contraintes (5.5) garantissent qu'un véhicule est utilisé si au moins un PI-container lui est attribué. Les contraintes (5.6) trouvent la destination d'un véhicule sortant utilisé en fonction des PI-containers qui lui sont affectés. Les contraintes (5.7) garantissent que chaque véhicule utilisé n'a qu'une seule destination. Les contraintes (5.8) calculent le nombre d'unités de PI-conveyors parcourus par un PI-container pour arriver à sa destination dans le véhicule sortant. Les contraintes (5.9) garantissent que chaque véhicule entrant doit être affecté à un seul quai. Les contraintes (5.10) et (5.11) gèrent la relation correcte entre la variable d'affectation (x_{hk}) et la variable de séquencement (p_{ih}). Les contraintes (5.12) calculent le temps de début du chargement des véhicules sortants en tenant compte de l'heure de fin du chargement des véhicules précédents s'ils sont traités au même quai. Les contraintes (5.13) calculent l'heure de fin du chargement des véhicules sortants. Les contraintes (5.14) garantissent que les variables de décision : $x_{hk}, p_{ih}, a_{hd}, v_h, n_{hg}$ sont binaires. Enfin, les contraintes (5.15) garantissent que les variables de décision : z_{ih}, b_h, q_h sont positives.

5.4 Méthodes de résolution proposées

En raison des temps de calcul élevés, la résolution du modèle mathématique n'est pas pratique pour les instances de grande taille. Par conséquent, le développement d'algorithmes ou méta-heuristiques peut fournir des solutions quasi-optimales ou optimales dans des délais de calcul raisonnables. Les méta-heuristiques sont largement utilisées dans la littérature pour résoudre des problèmes d'optimisation multi-objectifs Gandibleux and Ehrgott (2005), Jones and Tamiz (2016), Palacios et al. (2015) et Romero (2014), en utilisant différentes techniques telles que la somme pondérée, la méthode lexicographique, la méthode pondérée min-max, etc. Cette étude est basée sur la méthode de programmation lexicographique, qui consiste à classer les fonctions d'objectives par ordre d'importance Marler and Arora (2004).

Dans la programmation lexicographique, le premier objectif est considéré comme le critère le plus important et il est possible de détériorer les autres objectifs pour améliorer le premier. Le deuxième objectif est considéré comme le plus important après le premier, et ainsi de suite. Le dernier objectif est le moins important parmi tous les critères précédents. Dans cette étude, deux méta-heuristiques hybrides sont proposées : Variable Neighborhood Search hybridé avec Simulated Annealing (MO-VNSSA) et la seconde un VNS hybridé avec le Tabu Search (MO-VNSTS). Le VNS est choisi en raison de sa capacité à modifier dynamiquement le voisinage de recherche. L'hybridation avec Tabu Search donne au VNS la possibilité d'éviter les mouvements de recherche locale déjà effectués. Alors que le SA permet au VNS d'accepter des solutions au début de la recherche, même avec des écarts importants, puis, dans chaque itération la probabilité d'accepter des solutions est réduite et l'algorithme devient très sélectif dans l'adoption de nouvelles solutions.

Tout d'abord, une heuristique de construction (H_0) est proposée pour construire une solution initiale en cherchant : d'une part, la minimisation du coût des véhicules utilisés en regroupant les PI-containers dans les véhicules en tenant compte des contraintes de destination. D'autre part, à trouver une position initiale des véhicules sur les quais en utilisant les positions des PI-containers à décharger du train. L'idée principale est de trouver un regroupement des PI-containers dans les véhicules sortants afin d'optimiser le premier

objectif F_1. La solution générée est alors considérée comme un point de départ pour les deux méta-heuristiques (MO-VNSSA et MO-VNSTS). Dans la suite, une description détaillée des étapes de chaque algorithme est présentée (heuristique de construction H_0, MO-VNSSA et MO-VNSTS).

5.4.1 Heuristique de construction

Les méta-heuristiques proposées étant basées sur le VNS qui nécessite une solution de départ, il est donc nécessaire de créer une solution initiale en tant que point de départ pour les méta-heuristiques hybrides proposées. C'est pour cela, une heuristique de construction (Algorithme 5.1) est proposée pour trouver un groupement initial de PI-containers, puis l'affectation et la planification des véhicules sortants sur les quais.

Algorithme 5.1 : Heuristique de construction (H_0) de la solution initiale

Entrée : données et paramètres du problème
Trier les PI-containers en longueurs décroissantes
Grouper les PI-containers par destination
Entrée : L_i longueur de chaque PI-container (i)
Entrée : A_h longueur disponible dans chaque véhicule (h)
Entrée : $A_h = Q$ capacité initiale du véhicule (h)
Entrée : F_h position moyenne d'un véhicule (h) au quai
Entrée : $Dist = 0$ (Somme des distances pour le véhicule (h))
Pour chaque destination du PI-container **Faire**
  Pour chaque véhicule (h) **Faire**
    Si le véhicule (h) est vide **Alors**
      Attribuer la destination (d) au véhicule (h)
    Fin Si
    Si $L_i \leq A_h$ et le PI-container (i) et le véhicule (h) ont la même destination **Alors**
      Le PI-container (i) est affecté au véhicule (h)
      $A_h = A_h - L_i$
      Passer au PI-container suivant
    Fin Si
  Fin Pour
Fin Pour
Pour chaque véhicule (h) **Faire**
  $Dist = 0$
  Pour chaque PI-container (i) **Faire**
    Si le PI-container (i) est dans le véhicule (h) **Alors**
      Dist = Dist+ Position du PI-container dans le train
    Fin Si
  Fin Pour
  A_h = Dist/ Nombre de PI-containers dans le véhicule (h)
  Affecter le véhicule (h) au quai le plus proche de la position moyenne A_h
  Calculez l'heure de début/fin pour le chargement du véhicule (h)
Fin Pour
Sortir : Solution initiale (S_0)

La première étape de l'heuristique consiste à regrouper les PI-containers dans les véhicules après les avoir triés dans un ordre décroissant en fonction de leur longueur, en tenant compte de la destination de chaque PI-container. Dans la seconde étape, la position moyenne de chaque véhicule est calculée en fonction de la position des PI-containers affectés à ce véhicule. Cette position moyenne sera utilisée ultérieurement pour trouver le quai auquel le véhicule sera affecté au chargement des PI-containers. L'étape suivante consiste à grouper et attribuer les PI-containers aux véhicules par destination. Si un véhicule n'a pas assez d'espace ou a une destination différente, un nouveau véhicule doit être sélectionné. La dernière étape consiste à calculer la planification des véhicules (heure de début/fin du chargement)

5.4.2 Opérateurs de recherche locale

Dans les deux méta-heuristiques (MO-VNSSA et MO-VNSTS), il existe trois opérateurs de recherche locale exécutés à chaque itération interne : Insertion, Permutation et Insertion+Permutation (Figure 5.3).

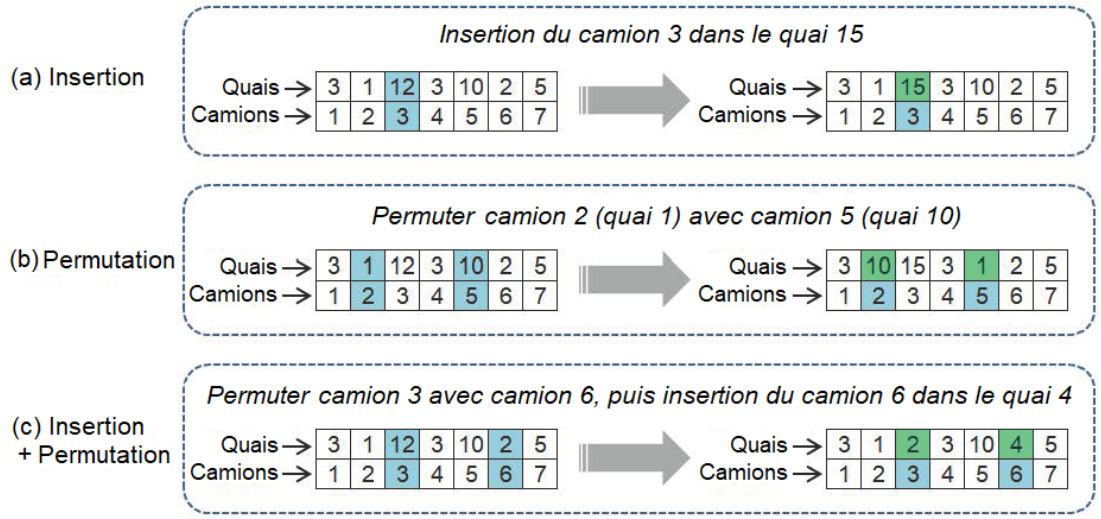

Figure 5.3 – Opérateurs de recherche locale dans MO-VNSSA et MO-VNSTS : Insertion, Permutation et Insertion+Permutation

L'insertion consiste à sélectionner un véhicule aléatoirement dans le voisinage, puis à affecter le véhicule sélectionné à un autre quai. L'opérateur de permutation sélectionne deux véhicules différents et échange leurs quais. Enfin, dans le dernier opérateur (Insertion+Permutation), un véhicule aléatoire est sélectionné, inséré dans un autre quai, puis

échangé avec un autre véhicule choisit aléatoirement. Ces trois opérateurs de voisinage sont exécutés avec des probabilités égales à chaque itération.

5.4.3 MO-VNSSA

L'algorithme MO-VNSSA commence par charger la solution initiale de l'heuristique de construction. Ensuite, un ensemble de structures de voisinage NbN est généré. Pour chaque structure sélectionnée, l'algorithme exécute la recherche locale suivant l'algorithme de Simulated Annealing (SA) pour $MaxVNS$ itérations. A chaque itération, le paramètre de température T est initialisé à T_{Max}. Pendant la résolution, la température T diminue de Δt après chaque mouvement de recherche locale (insertion, permutation et insertion+permutation) avec des probabilités égales pour chaque mouvement (Voir Algorithme 5.2).

Algorithme 5.2 : Algorithme de la méta-heuristique MO-VNSSA

Entrée : Solution initiale S_0 générée par l'heuristique H_0
Entrée : $MaxVNS$, NbN, $MaxShake$
Entrée : T, δt, T_{Max} *// Paramètres de la SA*
Entrée : $S_{BEST} = S_0$ *// Meilleure solution*
Entrée : $S = S_0$ *// Solution actuelle*
Générer NbN structures de voisinage
Pour chaque structure de voisinage **Faire**
  Pour chaque $MaxVNS$ itérations **Faire**
    $T = T_{Max}$ *// Réinitialiser T*
    Tant Que $T > \epsilon$ **Faire** *// est un nombre très petit* $\epsilon = 1 \times 10^{-5}$
      Effectuer un mouvement de recherche locale sur la séquence des véhicules : Insertion, Permutation ou Insertion+Permutation
      Calculer la nouvelle fonction objective de la solution S'
      $r = rand[0, 1]$ *// r est un nombre aléatoire généré dans l'intervalle [0, 1]*
      Si $r < Exp((S \breve{} S')/T)$ **Alors**
        $S = S'$
        Si $S_{BEST} > S'$ **Alors**
          $S_{BEST} = S'$
        Sinon
          $T = T - \Delta t$ *// Réduire T*
        Fin Si
      Sinon
        Annuler le mouvement de recherche locale
        $T = T - \Delta t$ *// Réduire T*
      Fin Si
    Fin Tant Que
  Fin Pour
  Mélanger l'affectation des PI-containers avec même destination entre les véhicules après chaque $MaxShake$ itérations
Fin Pour
Sortir : S_{BEST}

Les nouvelles solutions générées (S_0) sont acceptées comme solution actuelle avec une probabilité p :

$$p = exp\left(\frac{S - S'}{T}\right)$$

Au début, quand T est encore élevée, l'algorithme a tendance à accepter des solutions détériorées. Ensuite, à mesure que la température T diminue, l'algorithme devient très sélectif sur les nouvelles solutions générées (S'). Ce processus est répété pour les $MaxVNS$ itérations et pour chaque structure de voisinage. Un mélange des affectations de PI-containers entre les véhicules ayant la même destination est effectuée après les $MaxShake$ itérations. À chaque itération de la SA, une solution aléatoire est générée dans la structure de voisinage actuelle, la solution générée (S') est acceptée avec la probabilité p en tant que solution actuelle (S). Ensuite, si $S' < S_{BEST}$, S' est considéré comme la meilleure solution actuellement trouvée. La boucle interne de la SA, qui est répétée pour les $MaxVNS$ itérations, s'arrête une fois que $T \leq \epsilon$. Ce processus est répété pour chaque structure de voisinage. L'algorithme s'arrête après avoir exploré toutes les structures de voisinage.

5.4.4 MO-VNSTS

Dans la méta-heuristique MO-VNSTS, le VNS est hybridé avec Tabu Search (TS) qui, contrairement à SA, utilise la mémoire. En effet, à chaque itération de la TS, chaque mouvement effectué est stocké dans une *tabu list* afin que, lors des itérations suivantes, ces mouvements ne soient pas exécutés pendant un certain nombre d'itérations. Ce mécanisme permet à l'algorithme d'apprendre des mouvements passés pour éviter ceux qui génèrent des solutions détériorées. Un critère d'aspiration est défini comme l'écart entre la nouvelle solution trouvée (S') et la solution actuelle (S) afin d'empêcher l'algorithme de se bloquer dans les optima locaux. Comme décrit dans l'algorithme 5.3, le processus de recherche commence par charger la solution initiale à partir de l'heuristique de construction. L'affectation des PI-containers est permutée entre les véhicules ayant la même destination après $MaxShake$ itérations. Ensuite, l'algorithme Tabu Search est exécuté pour $MaxTS$ itérations pour chaque structure de voisinage. A chaque itération, une nouvelle solution

aléatoire est générée dans le voisinage sélectionné (S'). Si le critère d'aspiration est vérifié pour la nouvelle solution, celle-ci est considérée comme solution actuelle ($S = S'$). Si $S' < S_{BEST}$, la nouvelle solution est considérée comme meilleure solution actuelle. L'algorithme se termine après avoir répété ce processus pour toutes les structures de voisinage. Les principales étapes de l'algorithme MO-VNSTS sont présentées dans l'algorithme 5.3.

Algorithme 5.3 : Algorithme de la méta-heuristique MO-VNSTS

Entrée : Solution initiale S_0 générée par l'heuristique H_0
Entrée : $MaxVNS$, NbN, $MaxShake$
Entrée : $TL = \emptyset$, TL_{Size}, TS_{Max} *// Paramètres de la TS*
Entrée : $S_{BEST} = S_0$ *// Meilleure solution*
Entrée : $S = S_0$ *// Solution actuelle*
Générer NbN structures de voisinage
Pour chaque structure de voisinage **Faire**
  Pour chaque $MaxVNS$ itérations Faire
    $TL = \emptyset$ *// Réinitialiser la tabu list*
    Pour chaque TS_{Max} itérations **Faire**
      Effectuer un mouvement de recherche locale sur la séquence des véhicules : Insertion, Permutation ou Insertion+Permutation
      Tant Que mouvement de recherche locale est dans la TL **Faire**
        Générer un nouveau mouvement de recherche locale
      Fin Tant Que
      Calculer la nouvelle fonction objective de la solution S'
      Si S' satisfait le critère d'aspiration **Alors**
        $S = S'$
        Si $S_{BEST} > S'$ **Alors**
          $S_{BEST} = S'$
        Fin Si
      Sinon
        Annuler le mouvement de recherche locale
      Fin Si
      Mettre à jour la TL en ajoutant le nouveau mouvement de recherche locale
    Fin Pour
  Fin Pour
  Mélanger l'affectation des PI-containers avec même destination entre les véhicules après chaque $MaxShake$ itérations
Fin Pour
Sortir : S_{BEST}

5.5 Résultats expérimentaux

Après une brève description de l'implémentation du modèle mathématique et du développement des deux méta-heuristiques, cette section présente les résultats obtenus sur les instances générées aléatoirement suivis d'une analyse.

5.5.1 Implémentation et instances

Cette section est destinée à évaluer la performance des méta-heuristiques proposées en les testant sur plusieurs instances générées aléatoirement. Tout d'abord, le modèle mathématique multi-objectif est implémenté et résolu à l'aide de l'optimisation par programmation lexicographique sur le solveur IBM CPLEX (version 12.9). Un délai d'une heure (3600s) est défini pour le solveur. L'heuristique (H_0) et les deux méta-heuristiques (MO-VNSSA et MO-VNSTS) sont développées en C++ et toutes les expérimentations sont effectuées sur un PC doté d'un CPU Intel$^{(R)}$ Core$^{(TM)}$ i3 avec 4 GB de RAM. Les tests sont effectués en 5 réplications pour chaque instance. La valeur moyenne est présentée. Les algorithmes proposés sont testés sur un ensemble de petites et grandes instances générées aléatoirement en faisant varier plusieurs paramètres. Les valeurs des paramètres sont résumées dans le tableau 5.1. Le réglage des paramètres des méta-heuristiques est présenté dans le tableau 5.2.

Tableau 5.1 – Valeurs des paramètres et des données d'entrée pour les petites instances

Données	Abréviations	Valeurs
Nombre de PI-containers	N	[4, 12]
Nombre de destinations	D	[1, 3]
Nombre de quais	K	15
Longueur des PI-containers	L_i	{1, 2, 3, 4, 5, 10}
Coût de la consommation d'énergie pour un PI-conveyor	C^E	0.5
Coût d'utilisation d'un véhicule pour chaque destination	C_d^T	[200, 800]
Positions des PI-containers dans le train	P_i	[1, 75]
Destination des PI-containers	G_{di}	Aléatoire (Binaire)

Tableau 5.2 – Valeurs des paramètres dans les méta-heuristiques : MO-VNSSA et MO-VNSTS pour les petites instances

Données	Abréviations	Valeurs
Nombre d'itérations pour chaque structure de voisinage	$MaxVNS$	500
Nombre de voisinages générés	NbN	3
Nombre d'itérations avant de permuter les PI-containers entre les véhicules	$MaxShake$	10
Le pas de diminution de la température de la SA	Δt	0.01
Température initiale de la SA	T_{Max}	1
Taille de la *tabu list TL*	TL_{Size}	10
Nombre d'itérations pour la TS	TS_{Max}	100

5.5.2 Résultats

Le tableau 5.3 présente les valeurs des deux fonctions objectives : F_1 (Coût des véhicules utilisés par destination) et F_2 (Consommation énergétique). Les quatre premières

colonnes présentent les paramètres des instances générées. Ensuite, la troisième colonne montre les résultats obtenus après la résolution du modèle mathématique en utilisant la programmation lexicographique sur CPLEX (F_1 puis F_2). Les temps de calcul sont présentés dans la septième colonne. CPLEX a pu trouver la solution optimale pour 23 instances. Dans les quatre dernières instances, CPLEX dépasse la limite de temps fixée à 3600s sans fournir une solution réalisable. Le reste des colonnes présente les résultats des deux méta-heuristiques (MO-VNSSA et MO-VNSTS) qui fournissent des résultats optimaux pour le premier objectif F_1 et des valeurs quasi-optimales pour le deuxième objectif F_2 dans des temps de calcul très courts par rapport à ceux de CPLEX. Les valeurs optimales sont présentées en gras. Afin de mieux évaluer les performances des deux méta-heuristiques, quelque grandes instances supplémentaires sont générées. Quelques paramètres des méta-heuristiques sont modifiés pour ces instances de grande taille ($MaxVNS = 1500$, $\Delta t = 0.002$ et $TS_{Max} = 500$). Les résultats obtenus sont présentés dans le tableau 5.4 et illustrés dans la figure V-5.

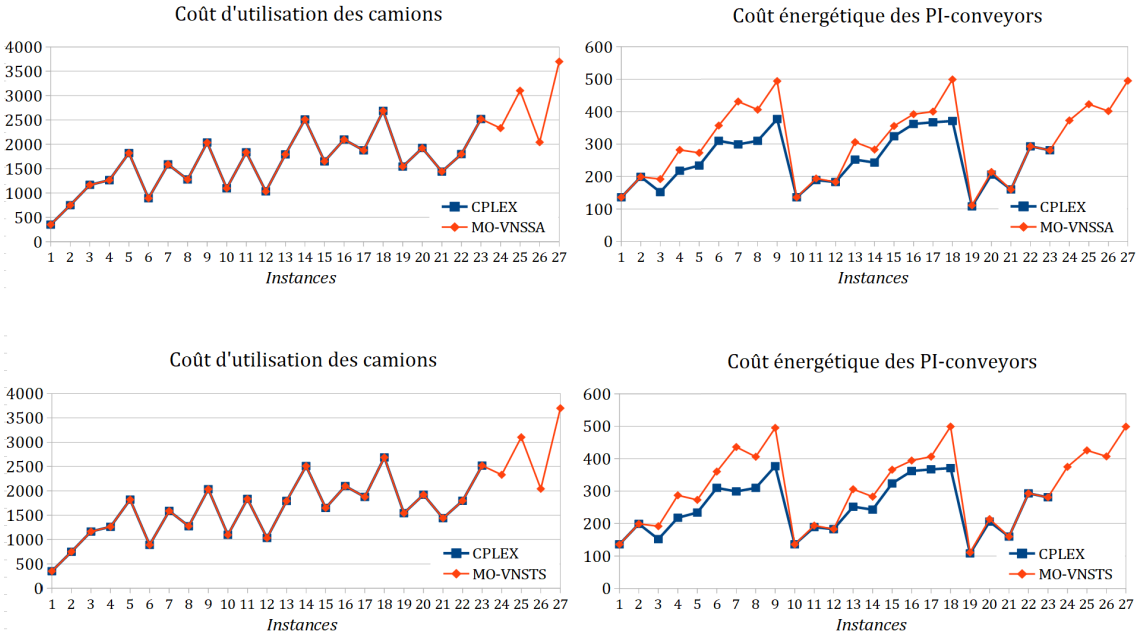

Figure 5.4 – Fonctions objectives pour CPLEX MO-VNSSA et MO-VNSTS pour les deux objectifs F_1 et F_2

Comme on peut le voir sur la figure 5.4, la consommation d'énergie des PI-conveyors, représentée par la fonction objectif F_2, augmente considérablement avec le nombre de PI-containers.

Tableau 5.3 – Résultats expérimentaux pour CPLEX, MO-VNSSA et MO-VNSTS sur les instances générées

Instances				CPLEX			MO-VNSSA			MO-VNSTS		
#	D	N	H	F_1	F_2	Temps (s)	F_1	F_2	Time (s)	F_1	F_2	Temps (s)
1	1	4	4	351	136	0,11	**351**	**136**	0,078	**351**	**136**	0,081
2	1	5	4	750	199	0,42	**750**	**199**	0,087	**750**	**199**	0,091
3	1	6	5	1166	152	0,69	**1166**	192	0,088	**1166**	192	0,081
4	1	7	5	1263	218	11,36	**1263**	282	0,084	**1263**	287	0,094
5	1	8	5	1821	234	2,78	**1821**	273	0,094	**1821**	273	0,100
6	1	9	7	891	310	14,86	**891**	357	0,116	**891**	360,2	0,144
7	1	10	7	1588	299	10,86	**1588**	431	0,134	**1588**	436	0,138
8	1	11	7	1276	310	13,36	**1276**	406	0,122	**1276**	406	0,138
9	1	12	7	2035	377	38,28	**2035**	494	0,140	**2035**	495	0,143
10	2	4	4	1096	136	0,14	**1096**	**136**	0,056	**1096**	**136**	0,078
11	2	5	4	1835	189	0,28	**1835**	194	0,066	**1835**	194	0,078
12	2	6	5	1033	183	0,30	**1033**	**183**	0,081	**1033**	**183**	0,091
13	2	7	5	1792	252	0,52	**1792**	306	0,084	**1792**	306	0,091
14	2	8	5	2510	243	1,69	**2510**	283	0,091	**2510**	283	0,112
15	2	9	7	1650	324	453,09	**1650**	355,8	0,112	**1650**	366	0,125
16	2	10	7	2098	362	277,84	**2098**	392	0,194	**2098**	394,2	0,131
17	2	11	7	1876	367	40,52	**1876**	400	0,147	**1876**	406,2	0,138
18	2	12	7	2688	371	186,89	**2688**	499	0,159	**2688**	499	0,153
19	3	4	4	1542	108	0,16	**1542**	111	0,059	**1542**	111	0,078
20	3	5	4	1923	206	2,09	**1923**	214	0,069	**1923**	214	0,094
21	3	6	5	1440	160	0,25	**1440**	**160**	0,084	**1440**	**160**	0,094
22	3	7	5	1797	293	17,17	**1797**	**293**	0,084	**1797**	**293**	0,094
23	3	8	5	2520	281	164,42	**2520**	**281**	0,091	**2520**	**281**	0,106
24	3	9	7	*	*	*	2331	372,8	0,106	2331	374,8	0,128
25	3	10	7	*	*	*	3102	422,6	0,116	3102	425,6	0,144
26	3	11	7	*	*	*	2042	401,4	0,122	2042	407	0,141
27	3	12	7	*	*	*	3699	495	0,141	3699	498,6	0,150

Tableau 5.4 – Résultats expérimentaux pour MO-VNSSA et MO-VNSTS pour les grandes instances

Instances				MO-VNSSA			MO-VNSTS		
#	D	N	H	F_1	F_2	Time (s)	F_1	F_2	Temps (s)
28	7	20	15	5001	772.6	4.322	5001	750	4.503
29	10	20	15	7338	801.2	4.5872	7338	776	4.7218
30	7	30	20	5081	1279.8	7.4594	5081	1216	7.3938
31	10	30	20	6401	1284.8	7.35	6401	1201	7.4782
32	15	30	20	9529	1220.4	8.1656	9529	1182	8.2032

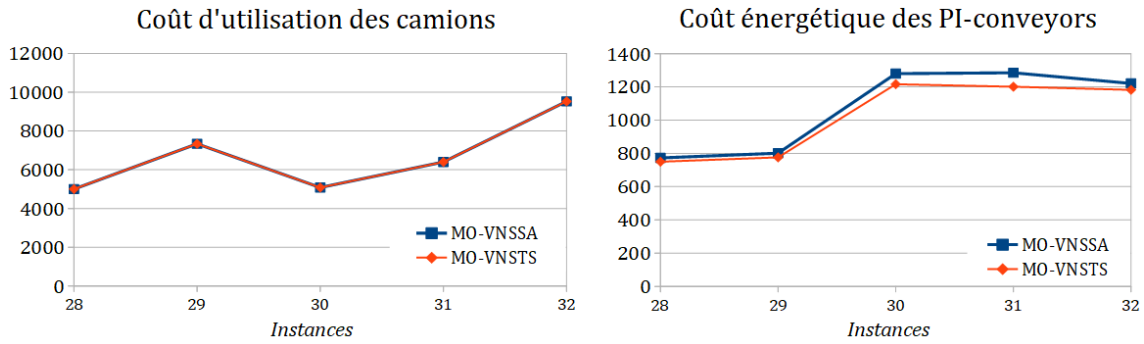

Figure 5.5 – F1 et F2 pour MO-VNSSA et MO-VNSTS dans les grandes instances

5.5.3 Analyse statistique

5.5.3.1 Analyse de variance : ANOVA

Les deux méta-heuristiques MO-VNSSA et MO-VNSTS ont trouvé des valeurs optimales pour la première fonction objective F_1. Afin de montrer la différence non significative pour le deuxième objectif F_2 entre les méta-heuristiques et CPLEX, un test d'hypothèse en utilisant l'analyse de variance ANOVA est effectué pour les deux méta-heuristiques. Le test ANOVA a montré que les deux algorithmes ne diffèrent pas de la valeur optimale de F_2 à 95% de limite de confiance. Les résultats détaillés sont présentés dans le tableau 5.5. Comme la valeur $p > 0,05$, il n'y a pas de différence significative entre les méta-heuristiques et les valeurs optimales de CPLEX.

Tableau 5.5 – Résultats de l'ANOVA pour MO-VNSSA et MO-VNSTS

Meta-Heuristiques	**F**	***p*-Value**
MO-VNSSA	1,623	0,209
MO-VNSTA	0,726	0,196

5.5.3.2 Convergence des méta-heuristiques

Afin d'illustrer la convergence des deux méta-heuristiques, plusieurs tests sont effectués avec différentes valeurs du nombre maximum d'itérations pour chaque structure de voisinage ($MaxVNS$) allant de 1 à 500 itérations. Les résultats obtenus sont présentés dans la figure 5.6, qui montre l'impact de la valeur du nombre d'itérations ($MaxVNS$) sur l'écart moyen total pour le deuxième objectif (F_2).

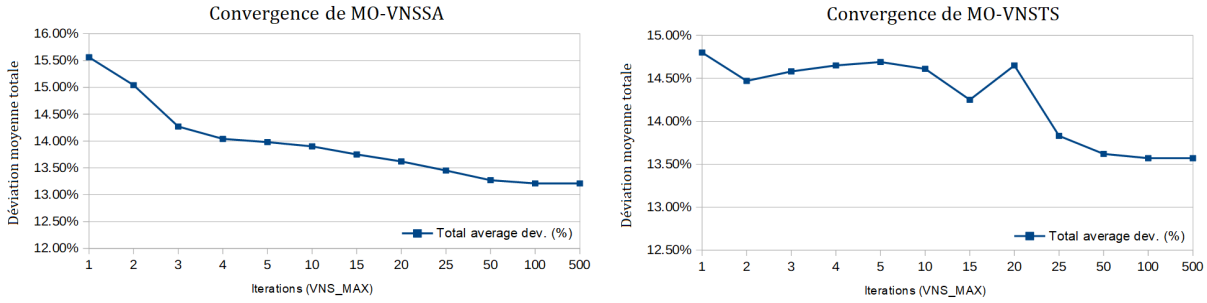

Figure 5.6 – Convergence de MO-VNSSA et MO-VNSTS pour le deuxième objectif F_2

5.5.4 Synthèse

Les résultats obtenus par les deux méta-heuristiques (MO-VNSSA et MO-VNSTS) semblent prometteurs en termes de qualité des solutions obtenues notamment pour le premier objectif F_1 pour lequel la valeur optimale a été atteinte pour toutes les instances et des valeurs quasi-optimales et plusieurs valeurs optimales pour le deuxième objectif F_2. Les résultats semblent également bons en termes de temps de calcul, même pour les instances de grande taille.

Dans le contexte de l'optimisation multi-objective, l'utilisation de la méthode de programmation lexicographique pour le modèle MO-MIP et les deux méta-heuristiques (MO-VNSSA et MO-VNSTS) nécessite une connaissance a priori de l'importance de chacun des objectifs qui repose sur les préférences du manager ou l'utilisateur. Par conséquent, les deux fonctions objectives présentées dans les tableaux 5.3 et 5.4 sont résolues dans un ordre arrangé lexicographique (qui est F_1 puis F_2 dans la présente étude).

Tenant compte de ces éléments, les méthodes proposées ne fournissent que des solutions basées sur les préférences du décideur et ne peuvent pas fournir un ensemble de solutions optimales de Pareto à proposer au manager. De plus, les méta-heuristiques proposées ne sont pas applicables et doivent être adaptées pour trouver une solution alternative en cas de modification de l'importance relative des objectifs.

5.6 Conclusion

Ce chapitre a été dédié à la planification multi-objective et durable des véhicules dans le Rail-Road PI-hub. Le problème a été formulé sous la forme d'un modèle de programmation à objectifs multiples (MO-MIP), qui prend en compte deux objectifs différents. Le premier consiste à minimiser le coût de consommation d'énergie pour l'acheminement des PI-containers à l'aide de PI-conveyors. Le second minimise le coût d'utilisation des véhicules sortants pour chaque destination. Le modèle a ensuite été résolu à l'aide de la programmation lexicographique dans le solveur CPLEX. En raison des temps de calcul considérablement longs, deux méta-heuristiques hybrides multi-objectifs ont été proposées : MO-VNSSA et MO-VNSTS. Le modèle MO-MIP et les deux méta-heuristiques

ont été évalués sur 27 petites instances et 5 grandes instances supplémentaires. CPLEX a trouvé des solutions optimales pour seulement 23 instances. Les résultats obtenus ont montré que les deux méta-heuristiques étaient capables de générer des solutions quasi-optimales et optimales en un temps de calcul très court. Les résultats ont été validés par une analyse ANOVA. Le travail présenté dans ce chapitre a fait l'objet d'une publication dans une revue internationale Chargui et al. (2019b).

Conclusion générale et perspectives

Conclusion générale

Au cours de cette thèse, nous avons abordé les problèmes d'optimisation des opérations de cross-docking dans le contexte de l'Internet Physique (PI). Ce nouveau paradigme, qui a été introduit comme un réseau logistique global et ouvert, vise à améliorer la façon dont les produits sont déplacés et stockés à travers l'interconnectivité physique, digitale et opérationnelle. L'étude menée dans cette thèse s'est focalisée sur les cross-docks (appelés PI-hubs dans le contexte du PI) qui jouent un rôle primordial dans l'efficacité de la chaine logistique.

Dans un premier temps, nous avons élaboré un état de l'art du cross-docking et l'Internet Physique en présentant d'abord les problèmes de planification des opérations dans les cross-docks classiques. Ensuite, nous avons présenté le concept du PI ainsi que les définitions de ses principaux éléments. Puis, les problèmes liés au PI dans la chaine logistique globale et les notions de base de ce concept, à savoir la durabilité et la résilience de la chaine logistique face aux éventuelles perturbations. Enfin, nous avons détaillé les problèmes d'optimisation dans les PI-cross-docks (PI-hubs) et plus spécifiquement les Rail-Road et Road-Road PI-hubs ainsi que les différentes approches de résolution proposées dans la littérature.

Comme première contribution, avant d'étudier les PI-hubs, nous avons abordé le cross-dock classique de type Road-Road en proposant un modèle mathématique pour la planification des véhicules entrants/sortants dans le cross-dock d'un partenaire industriel contenant plusieurs zones de stockage et gérant plusieurs types de produits. Ensuite, deux méta-heuristiques (TS et SA) ont été proposées pour résoudre le problème. La performance et la stabilité de la SA ont été validées sur un benchmark d'instances de la littérature. Les algorithmes proposés sont beaucoup plus performants par rapport à la méthode manuelle utilisée par les opérateurs du cross-dock. Ces algorithmes les aideront dans la planification des véhicules entrants et sortants en plus du routage interne des produits.

Par la suite, afin d'évaluer l'apport de l'Internet Physique sur la performance du cross-dock de type Road-Road, nous avons développé deux modèles de simulation à évènements discrets. Le premier modèle concerne le cross-dock classique de type Road-Road tandis que le deuxième modélise le PI-cross-dock (ou PI-hub). La comparaison de la robustesse et la performance des deux cross-docks a été réalisée à travers plusieurs scénarios en comparant de différents indicateurs de performance, à savoir le temps d'attente des véhicules, le nombre de véhicules en attente, le temps de cycle des produits et enfin le taux d'utilisation des ressources. Les résultats obtenus valident la performance du PI-hub par rapport au cross-dock classique en termes d'efficacité et de robustesse face aux variations des arrivées des véhicules entrants et sortants. Dans le même cadre, une approche de couplage simulation-optimisation, en combinant la Tabu Search avec un simulateur de perturbations, a été également proposée pour générer des solutions robustes face aux éventuelles perturbations qui peuvent arriver au niveau des ressources de transport internes.

La contribution suivante s'est focalisée sur le problème de planification durable des véhicules et le groupement des PI-containers dans le Road-Rail PI-hub. Les problèmes d'optimisation liés à ce type de PI-hub n'ont pas été abordés dans la littérature du PI. Pour cela, nous avons proposé un modèle multi-agent (MAS) pour la planification des véhicules entrants et le groupement des PI-containers. Le modèle MAS contient trois agents de planification utilisant trois méta-heuristiques hybrides dont la performance et la stabilité ont été validées par un modèle mathématique MILP et ensuite sur un benchmark d'instances de la littérature. Le MAS a été évalué également dans un environnement dynamique sous perturbations. Les résultats obtenus ont montré la capacité du MAS à réagir face aux perturbations en générant des planifications alternatives en cas de non-disponibilité des quais de déchargement.

Enfin, comme dernière contribution, nous avons abordé le problème de la planification durable des véhicules et le groupement des PI-containers dans un Rail-Road PI-hub. Tout d'abord, nous avons proposé un modèle mathématique multi-objectifs MO-MIP dont le but est de minimiser à la fois la consommation énergétique des PI-conveyors ainsi que le coût d'utilisation des véhicules sortants. Ensuite, après avoir résolu le modèle sur CPLEX en utilisant la programmation multi-objectifs lexicographique, deux méta-heuristiques multi-objectif ont été proposées pour résoudre le problème.

Perspectives

Les travaux réalisés durant cette thèse ont permis d'ouvrir de nombreuses perspectives :

— Prendre en compte d'autres contraintes dans la planification du cross-dock de type Road-Road classique, notamment l'ordonnancement des ressources mobiles de transport interne.
— Tenir compte de plusieurs objectifs à la fois (en plus des retards des véhicules), tels que l'utilisation des ressources, le niveau de stockage, la distance parcourue par les ressources de transport, etc.
— Prendre en compte d'autre types de perturbations, notamment les perturbations externes telles que les retards des véhicules ou des trains, changement de demandes dans les dernières minutes, pannes des véhicules, etc.
— Des contraintes supplémentaires pourraient également être envisagées, notamment celles liées au routage dans les PI-sorters pour gérer les acheminements des PI-containers avec plus de fluidité.
— Développer des approches multi-objectives de types Pareto qui ne nécessitent pas de préférences a priori sur l'importance relative des fonctions objectives. Ces approches basées sur Pareto peuvent être développées à l'aide de méta-heuristiques à population.

References

Agustina, D., Lee, C., and Piplani, R. (2010). A review : mathematical modles for cross docking planning. *International Journal of Engineering Business Management*, 2(2) :47 – 54.

Agustina, D., Lee, C. K. M., and Piplani, R. (2014). Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics*, 152 :29 – 41.

Ambra, T., Caris, A., and Macharis, C. (2019). Towards freight transport system unification : reviewing and combining the advancements in the physical internet and synchromodal transport research. *International Journal of Production Research*, 57(6) :1606–1623.

Amini, A. and Tavakkoli-Moghaddam, R. (2016). A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks. *Computers & Industrial Engineering*, 96 :180 – 191.

Assadi, M. T. and Bagheri, M. (2016). Differential evolution and population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems. *Computers & Industrial Engineering*, 96 :149 – 161.

Ballot, E., Montreuil, B., and Thivierge, C. (2012). Functional design of physical internet facilities : a road-rail hub. In *12th IMHRC Proceedings, Gardanne, France.*

Bartholdi, J. J. and Gue, K. R. (2004). The best shape for a crossdock. *Transportation Science*, 38(2) :235–244.

Bhamra, R., Dani, S., and Burnard, K. (2011). Resilience : the concept, a literature review and future directions. *International Journal of Production Research*, 49(18) :5375–5393.

Boloori-Arabani, A. R., Ghomi, S. M. T. F., and Zandieh, M. (2011). Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert Systems with Applications*, 38(3) :1964 – 1979.

Boysen, N., Briskorn, D., and Tschöke, M. (2013). Truck scheduling in cross-docking terminals with fixed outbound departures. *OR Spectrum*, 35(2) :479–504.

Boysen, N. and Fliedner, M. (2010). Cross dock scheduling : Classification, literature review and research agenda. *Omega*, 38(6) :413 – 422.

Centobelli, P., Cerchione, R., and Esposito, E. (2018). Environmental sustainability and energy-efficient supply chain management : A review of research trends and proposed guidelines. *Energies*, 11(2).

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2018a). A mathematical formulation and tabu search approach for the road-rail assignment problem. In *Proceedings of 5th International Physical Internet Conference (IPIC 2018), Groningen, The Netherlands*.

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2018b). *Simulation for PI-Hub Cross-Docking Robustness*, pages 317–328. Springer International Publishing.

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2019a). A modified hybrid variable neighborhood search for the truck scheduling problem on a multiple-door cross-docking terminal. In *The 7th International Conference on Variable Neighborhood Search, Rabat, Morocco*.

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2019b). Multi-objective sustainable truck scheduling in a rail–road physical internet cross-docking hub considering energy consumption. *Sustainability*, 11(11) :3127.

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2019c). Proposal of a multi-agent model for the sustainable truck scheduling and containers grouping problem in a road-rail physical internet hub. *International Journal of Production Research*, 58(18).

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2019d). A simulation-optimization approach for two-way scheduling/grouping in a road-rail physical internet hub. *IFAC-PapersOnLine*, 52(13) :1644 – 1649. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019, Berlin, Germany.

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2019e). *Tabu Search Robustness for Cross-Dock and PI-Hub Scheduling Under Possible Internal Transportation Breakdowns*, pages 295–307. Springer International Publishing.

Chargui, T., Bekrar, A., Reghioui, M., and Trentesaux, D. (2020). Scheduling trucks and storage operations in a multiple-door cross-docking terminal considering multiple storage zones. *International Journal of Production Research*.

Chargui, T. and Reghioui, M. (2016). Multi-objective sectoring and balancing algorithm. In *2016 3rd International Conference on Logistics Operations Management (GOL)*, pages 1–6. IEEE.

Chargui, T., Reghioui, M., Bekrar, A., and Trentesaux, D. (2016). A simulation model to migrate from cross-docks to pi-hubs. In *The 3rd International Physical Internet Conference (IPIC 2016)*, pages 1–11.

Chen, F. and Lee, C.-Y. (2009). Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193(1) :59 – 72.

Chen, F. and Song, K. (2009). Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers & Operations Research*, 36(6) :2066 – 2073.

Christopher, M. (2016). *Logistics & supply chain management*. Pearson UK.

Ciccullo, F., Pero, M., Caridi, M., Gosling, J., and Purvis, L. (2018). Integrating the environmental and social sustainability pillars into the lean and agile supply chain management paradigms : A literature review and future research directions. *Journal of Cleaner Production*, 172 :2336–2350.

Cota, P. M., Gimenez, B. M. R., Araújo, D. P. M., Nogueira, T. H., Souza, M. C. d., and Ravetti, M. G. (2016). Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre. *Computers & Industrial Engineering*, 95 :135 – 143.

Crainic, T. G. and Montreuil, B. (2016). Physical internet enabled hyperconnected city logistics. *Transportation Research Procedia*, 12 :383 – 398.

Croom, S., Vidal, N., Spetic, W., Marshall, D., and McCarthy, L. (2018). Impact of social sustainability orientation and supply chain practices on operational performance. *International Journal of Operations & Production Management*, 38(12) :2344–2366.

Demartini, M., Pinna, C., Aliakbarian, B., Tonelli, F., and Terzi, S. (2018). Soft drink supply chain sustainability : A case based approach to identify and explain best practices and key performance indicators. *Sustainability*, 10(10).

Dulebenets, M. (2018). A diploid evolutionary algorithm for sustainable truck scheduling at a cross-docking facility. *Sustainability*, 10(5) :1333.

Dulebenets, M. A. (2019). A delayed start parallel evolutionary algorithm for just-in-time truck scheduling at a cross-docking facility. *International Journal of Production Economics*, 212 :236–258.

Fazili, M., Venkatadri, U., Cyrus, P., and Tajbakhsh, M. (2017). Physical internet, conventional and hybrid logistic systems : a routing optimisation-based comparison using the eastern canada road network case study. *International Journal of Production Research*, 55(9) :2703–2730.

Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2) :109–133.

Gandibleux, X. and Ehrgott, M. (2005). 1984-2004 – 20 years of multiobjective metaheuristics. but what about the solution of combinatorial problems with multiple objectives ? In Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E., editors, *Evolutionary Multi-Criterion Optimization*, pages 33–46. Springer Berlin Heidelberg.

Gazzard, N. and Montreuil, B. (2015). A functional design for physical internet modular handling containers. In *Proceedings of 2nd International Physical Internet Conference, Paris, France, July 06-08.*

Glover, F. (1989). Tabu search-part 1. *ORSA Journal on computing*, 1(3) :190–206.

Golshahi-Roudbaneh, A., Hajiaghaei-Keshteli, M., and Paydar, M. M. (2017). Developing a lower bound and strong heuristics for a truck scheduling problem in a cross-docking center. *Knowledge-Based Systems*, 129 :17 – 38.

Guo, F., Liu, Q., Liu, D., and Guo, Z. (2017). On production and green transportation coordination in a sustainable global supply chain. *Sustainability*, 9(11).

Heidari, F., Zegordi, S. H., and Tavakkoli-Moghaddam, R. (2018). Modeling truck scheduling problem at a cross-dock facility through a bi-objective bi-level optimization approach. *Journal of Intelligent Manufacturing*, 29(5) :1155–1170.

Ivanov, D. (2018). Revealing interfaces of supply chain resilience and sustainability : a simulation study. *International Journal of Production Research*, 56(10) :3507–3523.

Jabbarzadeh, A., Fahimnia, B., and Sabouhi, F. (2018). Resilient and sustainable supply chain design : sustainability analysis under disruption risks. *International Journal of Production Research*, 56(17) :5945–5968.

Jadhav, A., Orr, S., and Malik, M. (2018). The role of supply chain orientation in achieving supply chain sustainability. *International Journal of Production Economics*.

Jones, D. and Tamiz, M. (2016). A review of goal programming. In *Multiple Criteria Decision Analysis*, pages 903–926. Springer.

Kapetanios, G., Vrisagotis, V., Pappas, D., Panta, M., and Siassiakos, K. (2009). A mathematical tool for warehousing optimization. In *Proceedings of the 9th WSEAS International Conference on Simulation, Modelling and Optimization*, SMO'09, pages 223–226.

Kelton, W., Sadowski, R., and Sadowski, D. (2000). *Simulation with ARENA*. McGraw Hill.

Keshtzari, M., Naderi, B., and Mehdizadeh, E. (2016). An improved mathematical model and a hybrid metaheuristic for truck scheduling in cross-dock problems. *Computers & Industrial Engineering*, 91 :197 – 204.

Khalili-Damghani, K., Tavana, M., Santos-Arteaga, F. J., and Ghanbarzad-Dashti, M. (2017). A customized genetic algorithm for solving multi-period cross-dock truck scheduling problems. *Measurement*, 108 :101 – 118.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598) :671–680.

Koberg, E. and Longoni, A. (2019). A systematic review of sustainable supply chain management in global supply chains. *Journal of Cleaner Production*, 207 :1084 – 1098.

Kong, L., Li, H., Luo, H., Ding, L., and Zhang, X. (2018). Sustainable performance of just-in-time (JIT) management in time-dependent batch delivery scheduling of precast construction. *Journal of cleaner production*, 193 :684–701.

Konur, D. and Golias, M. M. (2013). Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals : A meta-heuristic approach. *Transportation Research Part E : Logistics and Transportation Review*, 49(1) :71 – 91.

Kuo, Y. (2013). Optimizing truck sequencing and truck dock assignment in a cross docking system. *Expert Systems with Applications*, 40(14) :5532 – 5541.

Ladier, A.-L. and Alpan, G. (2016a). Cross-docking operations : Current research versus industry practice. *Omega*, 62 :145 – 162.

Ladier, A.-L. and Alpan, G. (2016b). Robust cross-dock scheduling with time windows. *Computers & Industrial Engineering*, 99 :16 – 28.

Landschützer, C., Ehrentraut, F., and Jodin, D. (2015). Containers for the physical internet : requirements and engineering design related to FMCG logistics. *Logistics Research*, 8(1) :8.

Liao, T. W., Chang, P. C., Kuo, R. J., and Liao, C.-J. (2014). A comparison of five hybrid metaheuristic algorithms for unrelated parallel-machine scheduling and inbound trucks sequencing in multi-door cross docking systems. *Applied Soft Computing*, 21 :180 – 193.

Liao, T. W., Egbelu, P. J., and Chang, P. C. (2012). Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. *Applied Soft Computing*, 12(11) :3683 – 3697.

Liao, T. W., Egbelu, P. J., and Chang, P. C. (2013). Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations. *International Journal of Production Economics*, 141(1) :212 – 229.

Lin, Y.-H., Meller, R. D., Ellis, K. P., Thomas, L. M., and Lombardi, B. J. (2014). A decomposition-based approach for the selection of standardized modular containers. *International Journal of Production Research*, 52(15) :4660–4672.

Madani-Isfahani, M., Tavakkoli-Moghaddam, R., and Naderi, B. (2014). Multiple cross-docks scheduling using two meta-heuristic algorithms. *Computers & Industrial Engineering*, 74 :129 – 138.

Maknoon, M., Soumis, F., and Baptiste, P. (2016). Optimizing transshipment workloads in less-than-truckload cross-docks. *International Journal of Production Economics*, 179 :90–100.

Marcotte, S., Montreuil, B., and Coelho, L. (2015). Modeling of physical internet enabled interconnected modular production. In *Proceedings of 2nd International Physical Internet Conference, Paris, France.*

Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6) :369–395.

Mbuli, J., Chargui, T., Trentesaux, D., Bekrar, A., and Dailly, T. (2020). Multi-agent system for the reactive fleet maintenance support planning of a fleet of mobile cyber–physical systems. *IET Cyber-Physical Systems : Theory & Applications.*

Meller, R. D., Montreuil, B., Thivierge, C., and Montreuil, Z. (2012). Functional design of physical internet facilities : a road-based transit center. In *12th IMHRC Proceedings, Gardanne, France.*

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11) :1097–1100.

Molavi, D., Shahmardan, A., and Sajadieh, M. S. (2018). Truck scheduling in a cross docking systems with fixed due dates and shipment sorting. *Computers & Industrial Engineering*, 117 :29 – 40.

Montreuil, B. (2011). Toward a physical internet : meeting the global logistics sustainability grand challenge. *Logistics Research*, 3(2) :71–87.

Montreuil, B., Meller, R. D., and Ballot, E. (2013a). Physical internet foundations. In Borangiu, T., Thomas, A., and Trentesaux, D., editors, *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, volume 472 of *Studies in Computational Intelligence*, pages 151–166. Springer Berlin Heidelberg.

Montreuil, B., Meller, R. D., and Ballot, E. (2013b). Towards a physical internet : The impact on logistics facilities and material handling systems design and innovation. In *Proceedings of the International Material Handling Research Colloquium (IMHRC)*, pages 1–23.

Montreuil, B., Meller, R. D., Thivierge, C., and Montreuil, Z. (2013c). Functional design of physical internet facilities : a unimodal road-based crossdocking hub. In *Progress in Material Handling Research. vol. 12, edited by B. Montreuil, A. Carrano, K. Gue, R. de Koster, M. Ogle, and J. Smith, 379–431. USA : MHI, Charlotte, NC.*

Mota, B., Gomes, M. I., Carvalho, A., and Barbosa-Povoa, A. P. (2015). Towards supply chain sustainability : economic, environmental and social design and planning. *Journal of Cleaner Production*, 105 :14–27.

Naccache, S. (2016). Hyperconnected e-commerce distribution sustainability : multi-agent simulation based assessment for e-vendors.

Nouiri, M., Bekrar, A., and Trentesaux, D. (2018). Inventory control under possible delivery perturbations in physical internet supply chain network. In *International Physical Internet Conference : IPIC*.

Pach, C., Sallez, Y., Berger, T., Bonte, T., Trentesaux, D., and Montreuil, B. (2014). Routing management in physical internet crossdocking hubs : Study of grouping strategies for truck loading. In Grabot, B., Vallespir, B., Gomes, S., Bouras, A., and Kiritsis, D., editors, *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*, volume 438 of *IFIP Advances in Information and Communication Technology*, pages 483–490. Springer Berlin Heidelberg.

Palacios, J. J., González-Rodríguez, I., Vela, C. R., and Puente, J. (2015). Swarm lexicographic goal programming for fuzzy open shop scheduling. *Journal of Intelligent Manufacturing*, 26(6) :1201–1215.

Pan, S., Ballot, E., Huang, G. Q., and Montreuil, B. (2017). Physical internet and interconnected logistics services : research and applications. *International Journal of Production Research*, 55(9) :2603–2609.

Pan, S., Nigrelli, M., Ballot, E., Sarraj, R., and Yang, Y. (2015). Perspectives of inventory control models in the physical internet : A simulation study. *Computers & Industrial Engineering*, 84 :122–132.

Qiao, B., Pan, S., and Ballot, E. (2019). Dynamic pricing for carriers in physical internet with peak demand forecasting. In *In 9th IFAC Conference MIM 2019*.

Romero, C. (2014). *Handbook of critical issues in goal programming*. Elsevier.

Sallez, Y., Berger, T., Bonte, T., and Trentesaux, D. (2015). Proposition of a hybrid control architecture for the routing in a physical internet cross-docking hub. *IFAC-PapersOnLine*, 48(3) :1978 – 1983.

Sallez, Y., Pan, S., Montreuil, B., Berger, T., and Ballot, E. (2016). On the activeness of intelligent physical internet containers. *Computers in Industry*, 81 :96 – 104.

Sarraj, R., Ballot, E., Pan, S., and Montreuil, B. (2014). Analogies between internet network and logistics service networks : challenges involved in the interconnection. *Journal of Intelligent Manufacturing*, 25(6) :1207–1219.

Serrano, C., Delorme, X., and Dolgui, A. (2017). Scheduling of truck arrivals, truck departures and shop-floor operation in a cross-dock platform, based on trucks loading plans. *International Journal of Production Economics*, 194 :102 – 112.

Shahram fard, S. and Vahdani, B. (2019). Assignment and scheduling trucks in cross-docking system with energy consumption consideration and trucks queuing. *Journal of Cleaner Production*, 213 :21–41.

Shiguemoto, A. L., Netto, U. S. C., and Bauab, G. H. S. (2014). An efficient hybrid meta-heuristic for a cross-docking system with temporary storage. *International Journal of Production Research*, 52(4) :1231–1239.

Sohrabi, H. (2016). Physical internet enabled hyperconnected distribution assessment.

Stephan, K. and Boysen, N. (2011). Cross-docking. *Journal of Management Control*, 22(1) :129–137.

Stock, J. R., Boyer, S. L., and Harmon, T. (2010). Research opportunities in supply chain management. *Journal of the Academy of Marketing Science*, 38(1) :32–41.

Sun, Y., Zhang, C., Dong, K., and Lang, M. (2018). Multiagent modelling and simulation of a physical internet enabled rail-road intermodal transport system. *Urban Rail Transit*, 4(3) :141–154.

Taylor, K. M. and Vachon, S. (2018). Empirical research on sustainable supply chains : IJPR's contribution and research avenues. *International Journal of Production Research*, 56(1) :950–959.

Tootkaleh, S. R., Ghomi, S. M. T. F., and Sajadieh, M. S. (2016). Cross dock scheduling with fixed outbound trucks departure times under substitution condition. *Computers & Industrial Engineering*, 92 :50 – 56.

Treiblmaier, H., Mirkovski, K., and Lowry, P. (2016). Conceptualizing the physical internet : Literature review, implications and directions for future research. In *11th CSCMP Annual European Research Seminar, Vienna, Austria.*

Vahdani, B. and Zandieh, M. (2010). Scheduling trucks in cross-docking systems : Robust meta-heuristics. *Computers & Industrial Engineering*, 58(1) :12 – 24.

Van Belle, J., Valckenaers, P., Berghe, G. V., and Cattrysse, D. (2013). A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Computers & Industrial Engineering*, 66(4) :818 – 826.

Van Belle, J., Valckenaers, P., and Cattrysse, D. (2012). Cross-docking : State of the art. *Omega*, 40(6) :827 – 846.

Vo, N.-V., Berger, T., Bonte, T., and Sallez, Y. (2018). Control of rail-road PI-hub : The ORCA hybrid control architecture. In Borangiu, T., Trentesaux, D., Thomas, A., and Cardin, O., editors, *Service Orientation in Holonic and Multi-Agent Manufacturing*, volume 762 of *Studies in Computational Intelligence*, pages 291–302. Springer International Publishing.

Walha, F., Bekrar, A., Chaabane, S., and Loukil, T. (2014a). A rail-road PI-hub allocation problems : model and heuristic. In *International Physical Internet Conference : IPIC*.

Walha, F., Bekrar, A., Chaabane, S., and Loukil, T. M. (2016). A rail-road PI-hub allocation problem : Active and reactive approaches. *Computers in Industry*, 81 :138 – 151.

Walha, F., Chaabane, S., Bekrar, A., and Loukil, T. (2014b). The cross docking under uncertainty : State of the art. In *2014 International Conference on Advanced Logistics and Transport (ICALT)*, pages 330–335.

Winter, M. and Knemeyer, A. M. (2013). Exploring the integration of sustainability and supply chain management : Current state and opportunities for future inquiry. *International Journal of Physical Distribution & Logistics Management*, 43(1) :18–38.

Wisittipanich, W. and Hengmeechai, P. (2017). Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization. *Computers & Industrial Engineering*, 113 :793 – 802.

Yang, Y. (2016). Towards more efficient and resilient supply chain management through interconnection of logistics networks.

Yang, Y., Pan, S., and Ballot, E. (2017a). Innovative vendor-managed inventory strategy exploiting interconnected logistics services in the physical internet. *International Journal of Production Research*, 55(9) :2685–2702.

Yang, Y., Pan, S., and Ballot, E. (2017b). Mitigating supply chain disruptions through interconnected logistics services in the physical internet. *International Journal of Production Research*, 55(14) :3970–3983.

Yu, W. and Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, 184(1) :377 – 396.

Annexe A

Cette annexe présente les résultats expérimentaux détaillés obtenus en utilisant l'algorithme SA (proposés dans le chapitre 2) sur le benchmark des 810 instances de Van Belle et al. (2013).

Tableau A.1 – Comparaison entre le SA et le TS1 (VB) de Van Belle et al. (2013) ($w_1 = 1$, $w_2 = 2$)

Type de problème			Fonction Objective					Dev.(%)	Time (s)	
			TS1 (VB)	SA (Moyenne)	SA (Min)	SA (Max)	SA (StdDev)	SA vs. TS1 (VB)	TS1 (VB)	SA
L	L	L	518,300	518,300	518,300	518,300	0,000	0,000%	0,164	0,879
L	L	M	592,072	592,072	592,072	592,072	0,000	0,000%	0,171	0,870
L	L	H	837,478	837,478	837,478	837,478	0,000	0,000%	0,155	0,850
L	M	L	526,391	526,391	526,391	526,391	0,000	0,000%	0,147	0,866
L	M	M	654,059	654,059	654,059	654,059	0,000	0,000%	0,153	0,878
L	M	H	830,229	830,229	830,229	830,229	0,000	0,000%	0,145	0,843
L	H	L	684,285	684,285	684,285	684,285	0,000	0,000%	0,167	0,980
L	H	M	601,945	601,945	601,945	601,945	0,000	0,000%	0,156	0,932
L	H	H	957,740	957,740	957,740	957,740	0,000	0,000%	0,152	0,919
M	L	L	695,459	696,223	695,827	696,777	0,370	0,110%	0,417	2,150
M	L	M	962,622	963,019	962,623	963,801	0,430	0,041%	0,425	2,218
M	L	H	1236,073	1239,278	1238,519	1240,959	0,930	0,259%	0,407	2,636
M	M	L	615,812	616,210	615,812	616,742	0,310	0,065%	0,437	2,451
M	M	M	1004,442	1005,577	1005,062	1006,362	0,500	0,113%	0,411	2,384
M	M	H	1443,384	1445,797	1445,327	1446,339	0,420	0,167%	0,434	2,718
M	H	L	875,440	875,681	875,241	876,041	0,290	0,028%	0,426	2,423
M	H	M	1150,058	1149,918	1149,558	1150,458	0,400	−0,012%	0,421	2,465
M	H	H	1704,644	1705,357	1704,811	1705,750	0,330	0,042%	0,506	2,959
H	L	L	1033,019	1041,887	1039,715	1044,212	1,580	0,858%	0,876	4,230
H	L	M	1197,026	1209,545	1206,774	1211,051	1,630	1,046%	0,846	4,181
H	L	H	1859,446	1891,478	1884,646	1896,779	4,920	1,723%	1,040	4,371
H	M	L	1037,082	1046,603	1045,582	1047,627	0,660	0,918%	1,010	4,875
H	M	M	1151,924	1161,961	1161,107	1163,995	1,060	0,871%	0,996	4,507
H	M	H	1985,397	1998,607	1997,249	2001,743	1,660	0,665%	1,011	5,052
H	H	L	1281,025	1286,813	1285,876	1287,765	0,640	0,452%	1,018	5,204
H	H	M	1712,346	1718,947	1718,403	1720,009	0,550	0,385%	1,139	5,617
H	H	H	2437,626	2448,883	2445,767	2451,032	1,950	0,462%	1,025	5,175
Moyenne :							0,690	0,303%	0,528 s	2,727 s

Tableau A.2 – Comparaison entre le SA et le TS1 (VB) de Van Belle et al. (2013) ($w_1 = 1$, $w_2 = 5$)

Type de problème			Fonction Objective					Dev.(%)	Time (s)	
			TS1 (VB)	SA (Moyenne)	SA (Min)	SA (Max)	SA (StdDev)	SA vs. TS1 (VB)	TS1 (VB)	SA
L	L	L	1076,750	1076,750	1076,750	1076,750	0,000	0,000%	0,165	0,849
L	L	M	1217,282	1217,282	1217,282	1217,282	0,000	0,000%	0,163	0,856
L	L	H	1839,085	1839,083	1839,083	1839,083	0,000	0,000%	0,157	0,837
L	M	L	1014,146	1014,146	1014,146	1014,146	0,000	0,000%	0,149	0,858
L	M	M	1354,048	1354,047	1354,047	1354,047	0,000	0,000%	0,157	0,881
L	M	H	1798,656	1798,656	1798,656	1798,656	0,000	0,000%	0,149	0,852
L	H	L	1387,713	1387,712	1387,712	1387,712	0,000	0,000%	0,168	0,977
L	H	M	1212,213	1212,212	1212,212	1212,212	0,000	0,000%	0,156	0,939
L	H	H	2076,650	2076,651	2076,651	2076,651	0,000	0,000%	0,154	0,918
M	L	L	1345,736	1345,975	1345,735	1346,035	0,120	0,018%	0,447	1,980
M	L	M	1989,142	1988,156	1987,810	1988,242	0,170	−0,050%	0,430	2,199
M	L	H	2680,817	2682,770	2681,217	2684,083	1,150	0,073%	0,431	2,265
M	M	L	1126,519	1125,499	1125,320	1125,920	0,210	−0,091%	0,437	2,259
M	M	M	2088,537	2088,997	2088,537	2089,837	0,460	0,022%	0,425	2,169
M	M	H	3152,726	3153,437	3152,896	3154,018	0,360	0,023%	0,448	2,505
M	H	L	1728,035	1728,074	1728,034	1728,234	0,080	0,002%	0,434	2,327
M	H	M	2399,145	2397,164	2396,944	2397,644	0,290	−0,083%	0,431	2,149
M	H	H	3748,651	3744,408	3743,648	3745,648	0,750	−0,113%	0,491	2,585
H	L	L	2060,984	2063,792	2062,184	2065,163	1,080	0,136%	1,027	4,450
H	L	M	2490,859	2499,367	2497,104	2502,325	1,820	0,342%	0,910	4,208
H	L	H	4075,916	4078,858	4075,540	4083,096	2,500	0,072%	1,084	4,858
H	M	L	2011,945	2016,291	2015,159	2017,828	1,230	0,216%	1,065	5,025
H	M	M	2321,451	2326,777	2325,697	2327,651	0,670	0,229%	1,058	4,692
H	M	H	4367,173	4362,909	4360,380	4364,346	1,420	−0,098%	1,080	5,020
H	H	L	2571,854	2571,677	2570,510	2573,376	0,970	−0,007%	1,023	4,668
H	H	M	3632,164	3635,084	3634,264	3637,064	1,010	0,080%	1,103	5,042
H	H	H	5428,291	5428,068	5424,040	5430,622	2,580	−0,004%	1,020	4,840
Moyenne :							0,625	0,028%	0,547 s	2,637 s

Tableau A.3 – Comparaison entre le SA et le TS1 (VB) de Van Belle et al. (2013) ($w_1 = 1$, $w_2 = 10$)

Type de problème			Fonction Objective					Dev.(%)	Time (s)	
			TS1 (VB)	SA (Moyenne)	SA (Min)	SA (Max)	SA (StdDev)	SA vs. TS1 (VB)	TS1 (VB)	SA
L	L	L	2007,500	2007,501	2007,501	2007,501	0,000	0,000%	0,164	0,837
L	L	M	2258,664	2258,663	2258,663	2258,663	0,000	0,000%	0,167	0,843
L	L	H	3505,452	3505,453	3505,453	3505,453	0,000	0,000%	0,158	0,828
L	M	L	1824,135	1824,136	1824,136	1824,136	0,000	0,000%	0,144	0,848
L	M	M	2516,293	2516,293	2516,293	2516,293	0,000	0,000%	0,158	0,871
L	M	H	3411,034	3411,033	3411,033	3411,033	0,000	0,000%	0,146	0,838
L	H	L	2558,887	2558,888	2558,888	2558,888	0,000	0,000%	0,173	0,972
L	H	M	2228,541	2228,342	2228,342	2228,342	0,000	−0,009%	0,158	0,917
L	H	H	3939,701	3938,499	3938,499	3938,499	0,000	−0,031%	0,173	0,905
M	L	L	2428,650	2427,611	2427,551	2427,851	0,120	−0,043%	0,436	1,791
M	L	M	3695,772	3693,820	3693,610	3693,873	0,110	−0,053%	0,419	1,784
M	L	H	5082,523	5083,240	5082,523	5084,223	0,710	0,014%	0,410	1,932
M	M	L	1972,893	1972,458	1972,377	1972,577	0,100	−0,022%	0,443	2,041
M	M	M	3892,657	3892,736	3892,656	3892,856	0,100	0,002%	0,428	1,931
M	M	H	5995,868	5994,432	5994,392	5994,592	0,080	−0,024%	0,454	2,114
M	H	L	3147,987	3148,087	3147,987	3148,487	0,200	0,003%	0,425	2,005
M	H	M	4476,472	4475,371	4475,271	4475,671	0,150	−0,025%	0,423	2,100
M	H	H	7151,902	7142,456	7142,276	7143,176	0,360	−0,132%	0,501	2,325
H	L	L	3770,312	3772,290	3770,612	3774,112	1,190	0,052%	0,917	3,773
H	L	M	4643,518	4647,250	4645,916	4648,288	1,020	0,080%	0,982	4,196
H	L	H	7748,340	7739,242	7737,135	7742,234	2,000	−0,117%	1,166	4,022
H	M	L	3636,673	3632,297	3630,905	3634,005	1,020	−0,120%	1,148	4,971
H	M	M	4267,180	4271,027	4269,682	4272,888	1,110	0,090%	1,048	4,209
H	M	H	8335,471	8303,794	8301,011	8305,757	1,730	−0,380%	1,051	4,690
H	H	L	4717,865	4713,154	4712,882	4713,440	0,240	−0,100%	1,045	4,641
H	H	M	6829,428	6829,827	6829,627	6830,227	0,230	0,006%	1,160	4,928
H	H	H	10406,981	10392,449	10390,895	10393,995	1,240	−0,140%	0,999	5,136
Moyenne :							0,434	−0,035%	0,552 s	2,461 s

Annexe B

Validation des algorithmes proposés dans le cross-dock étudié : Impact industriel

Comme indiqué précédemment, le problème examiné dans ce chapitre a été suggérée par notre partenaire industriel gérant de plusieurs centres de cross-docks. Cette étude se concentre sur l'un de ces cross-docks (illustrés dans le tableau B.1 et la figure B.1) qui se trouve à proximité d'un grand terminal portuaire.

Tableau B.1 – Exemple d'une journée (24h) dans le cross-dock étudié

Paramètres	**Valeurs**
Nombre de véhicules entrants	15 à 35
Nombre de véhicules sortants	15 à 35
Nombre de quais	$3 \times 8h = 24h$
Surface	6000 m^2
Nombre de quais entrants	5 quais
Nombre de quais sortants	5 quais
Transport interne	Chariots élévateurs
Nombre de types de produits	Plus que 20 types
Nombre de produits par véhicule	30 à 65

Le cross-dock gère les produits électro-ménagers tels que les TVs, les machines à laver, les réfrigérateurs, etc. Une fois déchargés ou chargés, les véhicules doivent retourner au terminal portuaire ou chez les clients dans leur fenêtre horaire prédéfinie. Pour cette raison, l'heure de départ souhaitée est fixée pour tous les véhicules. Pour les véhicules sortants, les heures d'arrivée et de départ sont définies par les détaillants. Le non-respect de ces créneaux horaires pour le terminal portuaire ou pour les détaillants entraînera des pénalités financières.

Les algorithmes proposés (heuristique et SA) seront utilisés par le manager et les opérateurs des quais du cross-dock étudié afin de les assister dans la planification des véhicules sur les quais et de gérer l'acheminement des produits entre les quais de récep-

tion/expédition et les zones de stockage. Cela les aidera à gérer le flux de produits en décidant de transférer les produits directement dans les véhicules sortants ou à travers les zones de stockage.

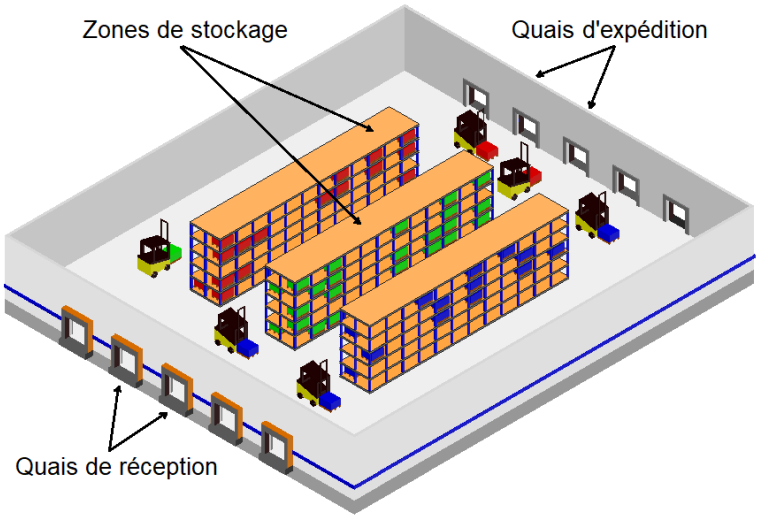

Figure B.1 – Vue d'ensemble du cross-dock étudié

Afin de montrer l'impact industriel et la différence significative entre la méthode manuelle actuellement utilisée par les opérateurs des quais et la planification des véhicules générée par l'algorithme SA proposé, une étude comparative est réalisée en simulant les affectations des véhicules par la méthode manuelle des opérateurs des quais. Avant de mettre en œuvre les algorithmes proposés, les opérateurs de quais affectent les véhicules qui arrivaient de manière aléatoire aux quais suivant la stratégie Premier arrivé, Premier Servi. La simulation consiste à affecter les véhicules à l'arrivée de manière aléatoire à l'un des quais disponibles à l'aide du logiciel de simulation ARENA (version 13.50). Après avoir exécuté la simulation avec 10 réplications pour l'instance 6, les résultats obtenus sont présentés dans le tableau B.2.

Tableau B.2 – Retard des véhicules entrants/sortants pour 10 réplications

	Algorithme	**Méthode manuelle**			
	SA	**Moyenne**	**Min**	**Max**	**StdDev**
Retard des véhicules entrants	786	1164,9	1068	1293	75,00
Retard des véhicules sortants	743	1112,5	1053	1210	50,12
Total	1529	2277,4			

Comme le montre le tableau B.2 et la figure B.2, lorsqu'on utilise la méthode d'affectation manuelle, le retard des véhicules entrants et sortants reste plus élevé que celui obtenu avec l'algorithme SA proposé. L'objectif principal de cette étude est de réduire au

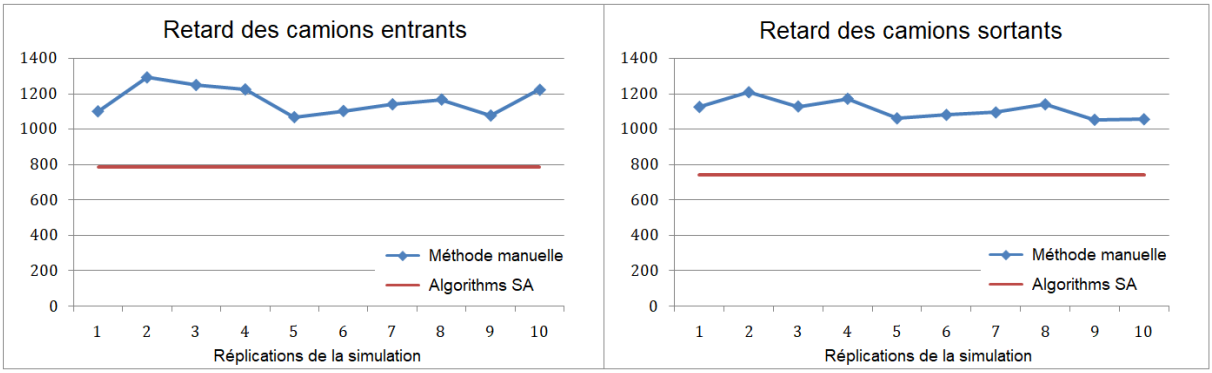

Figure B.2 – Retard des véhicules entrant/sortants pour les 10 réplications

minimum les retards des véhicules ce qui contribuera également à la réduction des coûts des pénalités appliquées par les clients et le terminal portuaire.

Outre la minimisation des retards, les algorithmes proposés présentent divers avantages, notamment le flux de produits à stocker ou à transférer directement. Cela aide les opérateurs des quais à décider exactement, une fois le véhicule arrivé, quels produits à stocker et lesquels à transférer directement aux véhicules sortants. Cependant, maintenir la robustesse des algorithmes proposés nécessite une bonne synchronisation entre le cross-dock, les clients et le terminal portuaire afin d'éviter les perturbations en mettant à jour les données d'entrée des algorithmes proposés, en particulier avec les heures d'arrivée et départ des véhicules entrants/sortants.

Exemple illustratif détaillé

Pour aider à mieux comprendre le modèle proposé, un exemple détaillé est illustré dans cette section en présentant la solution de l'instance 6 et en la comparant à l'une des réplications de la simulation. Le tableau 1 présente les heures d'arrivée et de départ préférées pour les véhicules entrants et sortants. L'heure de départ préférée est fixée à 30 unités de temps après l'heure d'arrivée pour tous les véhicules.

Tableau B.3 – Arrivée et départ préférés des véhicules entrant/sortants

Véhicules entrants	**1**	**2**	**3**	**4**	**5**
Arrivées	12	12	13	37	48
Départs préférés	42	42	43	67	78
Véhicules sortants	**1**	**2**	**3**	**4**	**5**
Arrivées	20	23	34	39	69
Départs préférés	50	53	64	69	99

Les solutions obtenues en utilisant l'algorithme SA proposé et la méthode manuelle actuelle sont détaillées dans le tableau 2. La figure 1 illustre la planification dans les deux approches.

Tableau B.4 – Début et fin de déchargement/chargement des véhicules entrants/sortants

Solution SA	**Véhicule entrants**	**1**	**2**	**3**	**4**	**5**	**Retard total**
	Début déchargement	12	12	13	37	48	786
	Fin déchargement	195	230	189	220	224	
	Véhicules sortants	**1**	**2**	**3**	**4**	**5**	**Retard total**
	Début chargement	20	23	34	39	189	743
	Fin chargement	197	230	188	224	239	
Solution par la méthode manuelle	**Véhicule entrants**	**1**	**2**	**3**	**4**	**5**	**Retard total**
	Début déchargement	12	12	13	37	48	1101
	Fin déchargement	195	182	338	296	362	
	Véhicules sortants	**1**	**2**	**3**	**4**	**5**	**Retard total**
	Début chargement	20	23	34	39	69	1126
	Fin chargement	241	362	338	224	296	

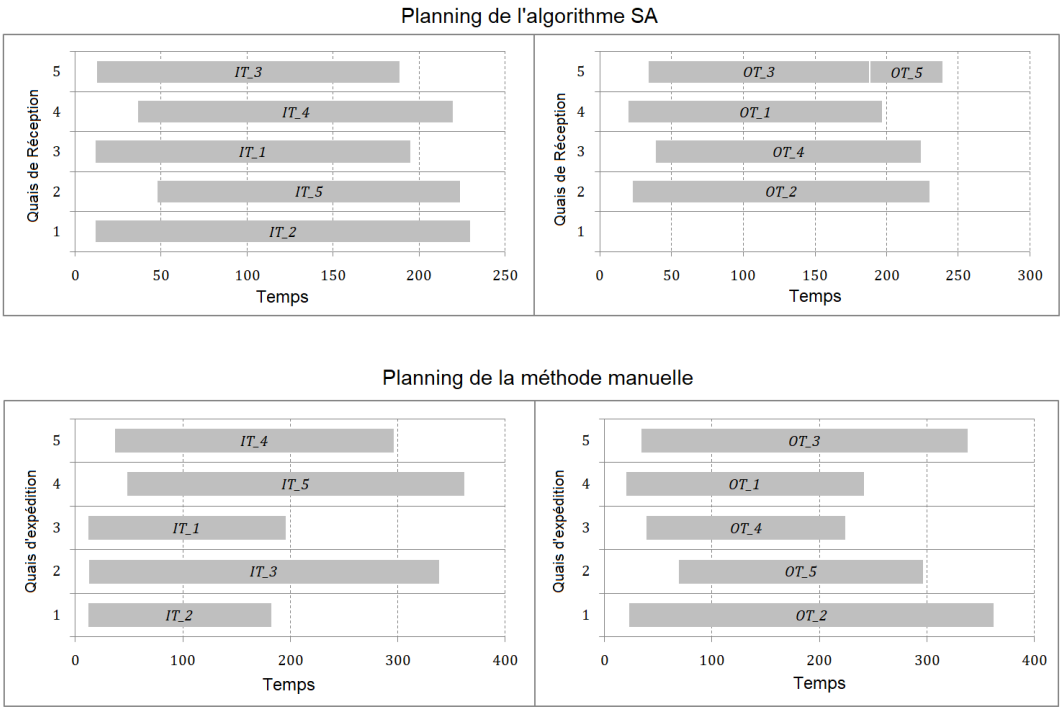

Figure B.3 – Planification des véhicules entrants/sortants par les deux approches

Annexe C

Tableau C.1 – Résultats expérimentaux des trois méta-heuristiques du MAS sur les instances benchmark

Instances			TS1	Temps (s)	MAS	Temps (s)	Dev. (%)
L	L	L	518,300	0,164	518,300	2,599	0,000%
L	L	M	592,072	0,171	592,072	3,810	0,000%
L	L	H	837,478	0,155	837,478	2,633	0,000%
L	M	L	526,391	0,147	526,391	2,049	0,000%
L	M	M	654,059	0,153	654,059	2,648	0,000%
L	M	H	830,229	0,145	830,229	4,057	0,000%
L	H	L	684,285	0,167	684,285	5,353	0,000%
L	H	M	601,945	0,156	601,945	2,192	0,000%
L	H	H	957,740	0,152	957,740	8,816	0,000%
M	L	L	695,459	0,417	696,160	10,445	0,101%
M	L	M	962,622	0,425	963,485	11,406	0,090%
M	L	H	1236,073	0,407	1239,739	8,992	0,297%
M	M	L	615,812	0,437	615,941	11,576	0,021%
M	M	M	1004,442	0,411	1005,520	14,955	0,107%
M	M	H	1443,384	0,434	1446,299	11,138	0,202%
M	H	L	875,440	0,426	875,947	11,003	0,058%
M	H	M	1150,058	0,421	1150,058	11,916	0,000%
M	H	H	1704,644	0,506	1705,685	21,404	0,061%
H	L	L	1033,019	0,876	1042,466	24,987	0,915%
H	L	M	1197,026	0,846	1211,018	19,460	1,169%
H	L	H	1859,446	1,040	1890,713	22,067	1,682%
H	M	L	1037,082	1,010	1046,698	16,313	0,927%
H	M	M	1151,924	0,996	1162,979	23,357	0,960%
H	M	H	1985,397	1,011	1999,832	25,951	0,727%
H	H	L	1281,025	1,018	1286,852	26,828	0,455%
H	H	M	1712,346	1,139	1719,380	26,321	0,411%
H	H	H	2437,626	1,025	2447,986	25,595	0,425%
Moyennes				**0,528**		**13,254**	**0,319%**