Titre: Title:	Modèle d'optimisation pour le transbordement de palettes dans des conteneurs
Auteur: Author:	Noam Mssellati
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Mssellati, N. (2025). Modèle d'optimisation pour le transbordement de palettes dans des conteneurs [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/70098/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/70098/
Directeurs de recherche: Advisors:	Daniel Aloise
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Modèle d'optimisation pour le transbordement de palettes dans des conteneurs

NOAM MSSELLATI
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Octobre 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Modèle d'optimisation pour le transbordement de palettes dans des conteneurs

présenté par **Noam MSSELLATI**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Martine BELLAÏCHE, présidente
Daniel ALOISE, membre et directeur de recherche
Alain HERTZ, membre

DÉDICACE

À mes parents et mon frère pour leur soutien inégalé
À tous les professeurs qui ont pu contribuer à ma passion des sciences
En particulier à Prof. Gary Cohen et Prof. Gilles Bitton
À mes amis rencontrés à chacune des étapes du parcours
À D.ieu pour la bénédiction et la sagesse

REMERCIEMENTS

Je souhaite exprimer ma profonde gratitude à mon directeur de recherche pour sa confiance et son accompagnement tout au long de ce projet.

Un hommage particulier à François Soumis pour sa créativité, son inspiration et sa sagesse. Qu'il repose en paix.

Je remercie également Alain Hertz pour la chaleur, l'enthousiasme et la rigueur avec lesquels il a transmis son savoir, essentiel à ce travail.

Je tiens à remercier MITACS pour le financement de ce projet ainsi que l'entreprise partenaire pour l'accueil, les perspectives enrichissantes et l'opportunité exceptionnelle de travailler sur un problème concret aux enjeux réels. Ma reconnaissance va à ENSTA Paris et à Polytechnique Montréal pour les opportunités de mobilité qui m'ont énormément appris, tant sur moi-même que sur le monde.

Je remercie le GERAD pour avoir mis à disposition un environnement scientifique riche et stimulant.

Ma gratitude va aussi à Simon Boivin pour le suivi et la coordination du projet, ainsi qu'à Luke, Ciarran et Luc pour leur implication précieuse.

Un grand merci à Dov pour la confrontation d'idées stimulantes et le soutien moral constant.

RÉSUMÉ

Le transbordement est une opération logistique consistant à transférer des marchandises d'un mode de transport à un autre en minimisant les besoins de stockage intermédiaire et les délais. Cette pratique contribue à l'optimisation des coûts d'expédition et est souvent nécessaire dans le cadre d'exportations de marchandises en vrac non conditionné. Les problématiques associées au transbordement présentent des similitudes avec celles du *cross-docking* ou du transbordement intermodal. Alors que la littérature existante se concentre principalement sur l'ordonnancement des opérations et les décisions stratégiques, peu de travaux abordent la planification opérationnelle à l'échelle du centre logistique. La présente étude vise à optimiser les coûts opérationnels d'une installation de transbordement de palettes, sous l'hypothèse d'un ordre fixe d'arrivée des envois, en intégrant des contraintes spécifiques de mélange ainsi que des limitations de capacité de stockage. La méthode proposée repose sur une approche en deux étapes. La première, une énumération adaptative des variables de décision à partir d'un graphe de ressources adapté, où chaque sous ensemble de nœuds représente un plan de consolidation pour une séquence donnée d'arrivage de lots. La seconde, la résolution d'un programme linéaire en nombres entiers mixtes (MILP) intégrant ces variables dans un modèle afin de sélectionner le sous-ensemble optimal de ces plans de consolidation. Cette approche est évaluée par comparaison avec la pratique industrielle de référence Fisrt In - First Out (FIFO). En moyenne, notre approche permet une réduction de la congestion de 21 %, atteignant jusqu'à 36 % dans le meilleur des cas. Les expérimentations numériques montrent que la méthode permet d'obtenir des solutions économiquement performantes dans des temps de calcul compatibles avec une utilisation opérationnelle.

ABSTRACT

Transloading is a supply chain operation that involves transferring goods from one mode of transport to another with minimum storage and delay. It helps optimize shipment costs and is often necessary when cargo is exported. The challenges in transloading are similar to cross-docking or transshipment. While previous research has focused mainly on scheduling and strategic decision-making, few studies address facility operational planning. This work optimizes the operational costs of a pallet transloading facility with a fixed shipment arrival order, taking into consideration special mixing constraints and storage capacity limitations. A two-step resolution approach is adopted. First, we enumerate decision variables, from a customized resource graph, representing each a consolidation plan for a sequence of batch arrivals. Then, we solve a Mixed-Integer Linear Program (MILP) that integrates these variables into a single optimization model that searches for the best subset of consolidation plans. Our approach is compared against the industry-standard First In, First Out (FIFO) strategy. On average, our approach achieves a 21% congestion reduction, reaching up to 36% in the best case. Computational experiments demonstrate that it achieves cost-efficient solutions within practical resolution times.

TABLE DES MATIÈRES

DÉDICACE iii

REMERCIEMENTS iv

RÉSUMÉ v

ABSTRACT vi

LISTE DES TABLEAUX ix

LISTE DES FIGURES x

LISTE DES SIGLES ET ABRÉVIATIONS xi

LISTE DES ANNEXES xii

CHAPITRE 1 INTRODUCTION 1
1.1 Énoncé du problème 2
1.2 Questions de recherche 5
1.3 Objectifs de recherche 6
1.4 Plan du mémoire 6

CHAPITRE 2 REVUE DE LITTÉRATURE 7
2.1 Concepts de base 7
2.1.1 Algorithme de recherche : Backtracking, Largeur et Profondeur . . . 7
2.1.2 Programmation Linéaire en nombre entier 9
2.1.3 Heuristiques 10
2.2 Travaux connexes 12

CHAPITRE 3 MÉTHODOLOGIE 16
3.1 Hypothèses 16
3.2 Référence opérationnelle : FIFO 18
3.3 Formulation 24
3.3.1 Modélisation 24
3.3.2 Remarques 28
3.3.3 Taille du modèle 30
3.4 Énumération de sous-ensembles 31

3.4.1 Réglage des hyper-paramètres 34
3.5 Logique de résolution finale 36

CHAPITRE 4 RÉSULTATS EXPÉRIMENTAUX 38
4.1 Scénarios testés 38
4.2 Analyse des résultats sur une instance type 40
4.2.1 Évaluation du nombre maximal de variables énumérées 40
4.2.2 Évaluation de l'énumération des variables complémentaires 41
4.2.3 Illustration d'une solution pour l'instance type 45
4.3 Analyse comparative sur les scénarios 45

CHAPITRE 5 CONCLUSION 49
5.1 Limitations de la solution proposée 49
5.2 Améliorations futures 50

RÉFÉRENCES 51

ANNEXES 55

LISTE DES TABLEAUX

Tableau 4.1 Caractéristiques de l'instance choisie pour illustration 40
Tableau A.1 Paramètres d'instances pour le scénario 1 (Uniforme) 55
Tableau A.2 Paramètres d'instances pour scénario 2 (Exact) 55
Tableau A.3 Paramètres d'instances pour le scénario 3 (Gaussien) 55
Tableau A.4 Paramètres d'instances pour le scénario 4 (Bloc) 55
Tableau A.5 Paramètres d'instances pour le scénario 5 (Bimodal) 56
Tableau A.6 Paramètres d'instances pour le scénario 6 (Séquentiel) 56

LISTE DES FIGURES

Figure 1.1 Exemple de configuration de déchargement à partir d'un wagon-trémie représentant une seule ligne. source : Vasiliev *et al.* [1] 2
Figure 1.2 Cartographie des processus et cadre des travaux 3
Figure 2.1 Comparaison de l'espace des solutions réalisables ayant les mêmes contraintes pour un problème linéaire continu (gauche) et en nombre entier (droite) 10
Figure 3.1 Comparaison de l'hypothèse d'échange de lots *batch swapping* . À gauche, nous illustrons une consolidation sans échange de lots, tandis qu'à droite, l'échange de lots est appliqué. 17
Figure 3.2 Exemples de séquences de consolidation selon la règle FIFO. Haut : exemple avec mise en stockage intermédiaire Bas : exemple avec apparition consécutive des lots 19
Figure 3.3 Illustration graphique de l'infériorité en stockage du FIFO 20
Figure 3.4 Exemple illustrant les mouvements lors de l'application de différentes stratégies de consolidation pour des apparitions non consécutives de lots 22
Figure 3.5 Exemple illustrant les mouvements lors de l'application de différentes stratégies de consolidation pour des apparitions consécutives de lots . 23
Figure 3.6 Représentation de ℓ_{qs} pour les sous-ensembles q_1 et q_2 26
Figure 3.7 Représentation de la matrice des contraintes 3.6 de partitionnement . 28
Figure 3.8 Exemple de représentation de solution dans la formulation `TCP`. . . . 28
Figure 3.9 Processus de résolution de bout en bout 37
Figure 4.1 Décomposition du temps pour la résolution de l'instance 4.1 avec construction standard 41
Figure 4.2 Résultats d'optimisation comparé à FIFO pour l'instance type en fonction du nombre de variables et de la complémentarité 42
Figure 4.3 Décomposition du temps pour la résolution de l'instance type avec construction complémentaire 43
Figure 4.4 Temps total et de résolution du modèle `TCP` en fonction du nombre de variables (standard et complémentaire) 44
Figure 4.5 Représentation du stockage et des coûts de mouvement par étapes pour l'instance 4.1 46
Figure 4.6 Réductions de coût obtenues par notre méthode pour chacun des scénarios testés 46

LISTE DES SIGLES ET ABRÉVIATIONS

NPC	Nombre de Palettes par Conteneur
BFS	Breadth First Search
DFS	Depth First Search
LIFO	Last In - First Out
FIFO	First In - First Out
PLNE	Programmation Linéaire en Nombre Entiers
LP	Linear Programming
IP	Integer Programming
ILP	Integer Linear Programming
MILP	Mixed Integer Linear Programming
BandB	Branch and Bound
GRASP	Greedy Randomized Adaptive Search Procedure
RCL	Restricted Candidate List
ASRS	Automatic Storage and Retrieval System
CLP	Container Loading Problem
DAG	Directed Acyclic Graph

LISTE DES ANNEXES

Annexe A Détail des scénarios 55

CHAPITRE 1 INTRODUCTION

Le domaine de la logistique est aujourd'hui confronté à des exigences croissantes en matière d'efficacité et de durabilité, portées par l'intensification des échanges internationaux, les impératifs liés au réchauffement climatique et l'essor de l'automatisation. Dans ce contexte, l'optimisation des opérations logistiques constitue un levier stratégique pour réduire les coûts, améliorer la performance et limiter l'empreinte environnementale. Le projet présenté s'inscrit dans cette dynamique et est commandité et financé par un partenaire du secteur privé dans le cadre du programme MITACS, favorisant une collaboration étroite entre le milieu académique et l'industrie. Cette synergie permet d'aborder des problématiques réelles et complexes tout en mobilisant des approches scientifiques rigoureuses. Lorsque des marchandises sont manipulées pour être transférées d'un mode de transport à un autre, il est question de transbordement. L'objectif global du projet est d'optimiser le fonctionnement d'un centre logistique de transbordement à l'interface entre le rail et le transport maritime. Ce centre reçoit en entrée des matières premières granulaires et doit produire, en sortie, des conteneurs prêts pour l'exportation. Les opérations incluent un stockage intermédiaire et sont soumises à des contraintes strictes de mélange des produits au sein des conteneurs. Ces contraintes, combinées aux volumes manipulés, rendent la planification particulièrement complexe. Les problèmes ainsi formulés relèvent de l'étude d'un grand nombre de scénarios, ce qui ouvre la voie à l'application de méthodes avancées d'optimisation discrète et combinatoire. Ces approches permettent d'explorer un large espace de solutions afin d'identifier les configurations opérationnelles les plus efficaces. Le présent travail vise ainsi à développer et évaluer des modèles d'optimisation adaptés à ce contexte spécifique, tout en garantissant leur applicabilité dans un environnement industriel réel.

1.1 Énoncé du problème

L'objectif principal du transbordement est de minimiser le temps nécessaire pour décharger, reconditionner et recharger les marchandises. Les matériaux concernés peuvent inclure différents types de marchandise sèche en vrac, tels que les minerais, les produits chimiques ou encore les céréales, avec un conditionnement final visé, dans notre cas, dans des conteneurs. Comme ce mémoire se concentre sur les opérations à l'intérieur de l'entrepôt, il est indifférent qu'un conteneur soit chargé sur un camion ou directement manipulé par une grue.

Le point d'entrée de notre processus modélisé est la composition du train. Celui-ci est constitué de wagons spécialisés appelés wagons-trémies (entonnoir), qui permettent un déchargement aisé des matières premières par le bas, comme illustré à la Figure 1.1 issue de Vasiliev *et al.* [1]. Étant donné que le déchargement de chaque wagon nécessite plusieurs équipements, la ligne de déchargement dispose d'un nombre limité de points d'entrée, déterminant le nombre maximal de wagons pouvant être déchargés simultanément. Le processus commence par la mise en place des wagons devant leurs lignes de déchargement respectives, où un opérateur fixe un tuyau d'aspiration au bas de chaque wagon. Le matériau est ensuite transféré à travers différentes étapes.

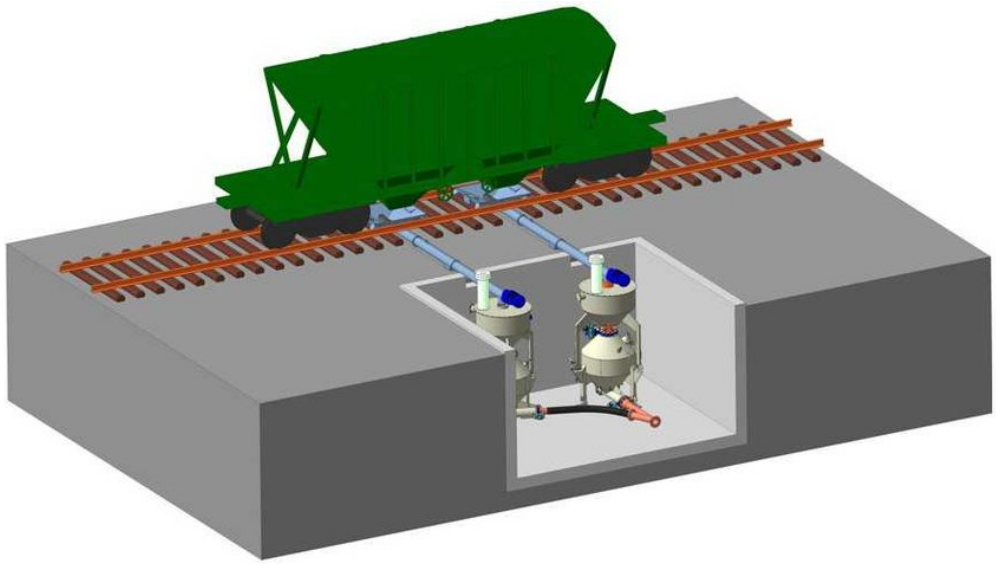

FIGURE 1.1 Exemple de configuration de déchargement à partir d'un wagon-trémie représentant une seule ligne. source : Vasiliev *et al.* [1]

Du placement du train jusqu'au chargement en conteneur, quatre phases principales sont distinguées : déchargement, palettisation, transfert et chargement, comme illustré à la Figure 1.2. Un ensemble d'équipements, incluant système d'aspiration et silos, est utilisé pour

transporter et stocker temporairement le matériau. Selon certaines spécifications, la marchandise peut être placée en gros sacs (bulk bag) ou conditionnée en plus petits sacs qui seront ensuite disposés sur palettes en contrôlant leur poids. Lorsque de petits sacs sont utilisés, le rôle du palettiseur est de les disposer et les placer sur des palettes. Ensuite, chaque palette est emballée et acheminée via un convoyeur vers un chargeur automatique de conteneurs. Dans notre cas, tous les conteneurs sortants sont de dimensions uniformes, ce qui signifie que le nombre de palettes nécessaires pour remplir un conteneur est fixe, noté Nombre de Palettes par Conteneur (NPC). Une fois la totalité du matériau d'un wagon déchargée et palettisée, le train est déplacé et de nouveaux wagons sont positionnés devant les lignes de déchargement pour répéter le processus. Cette séquence d'opérations définit une étape de déchargement, répétée jusqu'à ce que le train soit entièrement vidé.

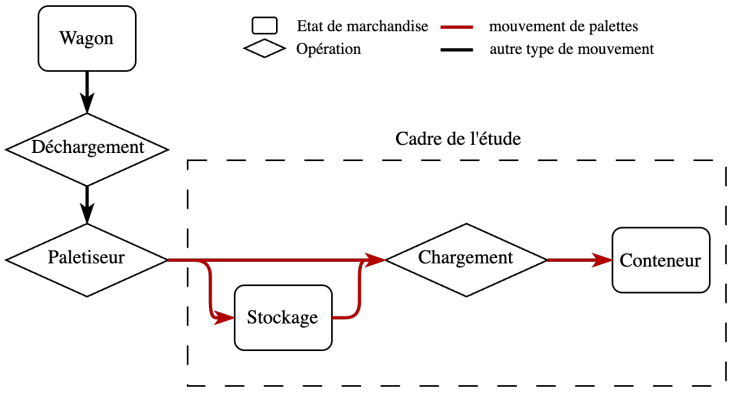

FIGURE 1.2 Cartographie des processus et cadre des travaux

La quantité de marchandise contenue dans un wagon ne correspond pas toujours à un nombre entier de conteneurs pleins, ce qui entraîne la production de palettes restantes ne permettant pas de remplir un conteneur au complet. Le nombre de ces palettes restantes est compris entre 0 et NPC - 1. Comme l'expédition d'un conteneur partiellement rempli est généralement peu rentable, ces palettes doivent être conservées jusqu'à pouvoir être combinées avec d'autres pour constituer un conteneur complet, ce qui engendre des coûts supplémentaires de stockage et de manutention.

Par ailleurs, plusieurs contraintes opérationnelles doivent être prises en compte lors de la gestion de ces palettes, justifiant la nécessité d'un système d'aide à la décision plus sophistiqué. Les marchandises transportées par train sont distinguées à deux niveaux : *produit* et *lot* (batch), chaque wagon transportant un produit donné et identifié par un numéro de lot, conformément aux exigences de traçabilité et de compatibilité de l'industrie. Nous supposons qu'un même produit peut apparaître plusieurs fois dans la composition du train, chaque oc-

currence correspondant à un lot distinct transporté dans un wagon spécifique. Dans notre cas, qui concerne des marchandises en vrac, un remaniement excessif des palettes lors du chargement des conteneurs, à l'arrivée, n'est pas souhaitable. Le respect de cette disposition simplifie la composition des conteneurs et rend le processus de déchargement plus efficace pour le destinataire. Ces contraintes influencent directement la manière dont les palettes restantes peuvent être regroupées pour former des conteneurs complets. Premièrement, chaque conteneur doit contenir un seul type de produit ; ainsi, lors de la consolidation, seules les palettes futures portant exactement le même produit peuvent être considérées. Deuxièmement, pour un produit donné, un conteneur peut inclure au maximum deux lots différents.

Nous imposons également une limite au nombre maximal de palettes pouvant être stockées simultanément dans le centre logistique. En pratique, cette contrainte est dictée par la disponibilité physique des emplacements de stockage, dont l'extension est coûteuse. Plus la capacité de stockage est grande, plus les coûts associés sont élevés, tant en investissement initial dans un système automatisé de stockage et de récupération (Automatic Storage and Retrieval System (ASRS)) — qui croît avec le nombre d'unités de stockage — qu'en infrastructure, car une capacité plus importante requiert un bâtiment plus vaste, augmentant les coûts de construction. À l'inverse, une sous-utilisation de la capacité réduit le retour sur investissement de l'ASRS, affectant négativement la rentabilité globale.

L'objectif principal que nous abordons est de soutenir la planification opérationnelle en se concentrant sur le temps, métrique essentielle de performance dans les opérations de transbordement. Réduire le temps d'immobilisation des actifs, tels que wagons et conteneurs, est crucial, car ces derniers sont partagés entre plusieurs acteurs interdépendants. Un transbordement rapide permet de traiter davantage de trains sur une période donnée, augmentant ainsi la capacité de traitement et facilitant la circulation des marchandises dans le réseau. Cela améliore non seulement le débit de l'entrepôt, mais aussi sa résilience en lui permettant d'absorber les aléas sans générer de retards en cascade. Enfin, minimiser le temps de transbordement est nécessaire pour respecter des délais de livraison stricts et garantir la fiabilité de la logistique en aval.

La dimension temporelle du problème couvre plusieurs composantes, incluant les mouvements de train, les opérations de palettisation et la capacité des convoyeurs, entre autres. Des contraintes temporelles supplémentaires proviennent des caractéristiques propres aux produits. Par exemple, la densité d'un produit influence significativement le débit du système d'aspiration ; ainsi, deux produits de volume identique peuvent nécessiter des durées de déchargement différentes. Étant donné que notre problème est formulé à l'échelle de la palette, nous ne considérons pas explicitement ces processus en amont et supposons qu'ils

ne constituent pas des variables décisionnelles du problème. En revanche, notre objectif temporel principal est de réduire la congestion du système, mesurée par le nombre total de mouvements de palettes. Une opération de consolidation excessivement complexe nuit non seulement à l'efficacité temporelle, mais aussi à la logique de planification, augmentant le risque de retards.

1.2 Questions de recherche

Le partenaire industrielle nous ayant accompagné dans le cadre de projet est une entreprise effectuant des opérations logistiques de différentes natures. Son activité laisse place plusieurs opportunités d'optimisation et elle avait déjà travaillé sur des projets similaires. Le transbordement constitue une composante majeure du métier de l'entreprise à l'origine du besoin, notamment le chargement de marchandises en vrac dans des conteneurs. Jusqu'à présent, une grande partie des opérations reposait sur une forte implication humaine. Dans la perspective de concevoir un centre de transbordement doté de convoyeurs et d'un système de stockage automatisé, la question d'une prise de décision plus flexible et de plus haut niveau se pose. En préparation de cette analyse, un modèle de simulation a été développé afin d'identifier les goulots d'étranglement dans le processus de chargement. Parallèlement, une base de règles a été formalisée dans la simulation, dont la règle principale adoptée est le FIFO. Cette règle, couramment appliquée dans les opérations logistiques, consiste à traiter les palettes ou les lots dans l'ordre exact de leur arrivée. Dans notre contexte particulier, cela signifie que toute palette restante d'un produit est regroupée avec le lot suivant du même produit présent sur le train.

À l'issue de l'étude, il a été constaté que les contraintes liées au mélange des produits et de leurs lots représentent un enjeu majeur, tant pour les clients que du point de vue de l'efficacité opérationnelle. Par ailleurs, la mise en place d'espaces de stockage automatisés implique un investissement considérable, ainsi une analyse de leur prise de décisions s'avérait nécessaire.

La nature combinatoire des décisions de consolidation a influencé notre approche d'optimisation du problème. La question initiale consiste à déterminer s'il est possible de dépasser les règles classiques, telles que le FIFO, afin d'identifier des stratégies de gestion plus performantes et mieux adaptées aux contraintes opérationnelles.

L'objectif est de déterminer dans quelle mesure les opérations peuvent être améliorées, tout en respectant les contraintes pratiques, telles que l'interdiction de mélanger plusieurs lots d'un même produit et la limitation du nombre de lots par conteneur.

1.3 Objectifs de recherche

Les objectifs de la recherche sont multiples. Étant en présence d'un problème réel venant d'un industriel, le but étant, dans un premier temps, de comprendre et cadrer les enjeux. Une fois cette étape importante réalisée, les hypothèses, la modélisation et l'approche de résolution se doivent d'être en accord avec les enjeux du commendataire. Les objectifs se résument donc à :

- Évaluer et caractériser la stratégie actuelle FIFO
- Modéliser le problème en prenant en compte les espaces de stockages, les mouvements et le temps des opérations
- Développer une méthode de résolution, exact ou heuristique, réaliste en taille d'instance et en temps de résolution

1.4 Plan du mémoire

Afin de traiter les problématiques énoncées ci-dessus, la suite de ce mémoire est organisée comme suit. La revue de littérature recontextualise les fondements théoriques de l'approche et permet de situer le problème étudié à travers des problématiques similaires dans la littérature existante. La partie méthodologie présente d'abord une analyse des opérations selon la logique FIFO, puis décrit la formulation MILP proposée ainsi qu'une méthode heuristique de construction des variables du modèle. La section « Résultats » expose les expérimentations numériques et leur analyse. Enfin, la conclusion discute les travaux réalisés, met en évidence leurs contributions et propose des pistes pour de futures recherches.

CHAPITRE 2 REVUE DE LITTÉRATURE

Il est essentiel, dans tout travail scientifique, de replacer l'étude dans son contexte conceptuel. La première partie de cette section présente le cadre théorique, qui expose les fondements et principes soutenant la méthode utilisée. La seconde se concentre sur la revue de littérature, en mettant en lumière les recherches antérieures portant sur des problématiques similaires.

2.1 Concepts de base

Un problème d'optimisation peut être abordé de différentes manières selon sa nature et ses caractéristiques. Dans le cas d'un problème appliqué, un travail de modélisation est nécessaire afin de traduire la complexité du système réel dans un cadre mathématique, ce qui suppose l'adoption d'hypothèses simplificatrices. Les méthodes de résolution varient en fonction du type de variables et de contraintes, des propriétés de la fonction objectif et de la présence d'incertitudes. Ainsi, un problème fortement incertain se prête davantage à des approches par simulation ou à des techniques d'optimisation stochastiques. À l'inverse, un problème déterministe pourra être résolu par des méthodes formelles telles que l'algorithme du simplexe ou l'exploration systématique de l'espace des solutions. Dans la pratique, ces approches sont souvent mises en œuvre à l'aide de solveurs, programmes spécialisés dans la résolution de modèles d'optimisation.

Toutes les méthodes ne garantissent pas l'optimalité : les heuristiques fournissent rapidement des solutions de bonne qualité, particulièrement utiles pour les problèmes complexes ou de grande taille. Dans la suite, nous présentons les fondements théoriques nécessaires à la compréhension des méthodes de résolution utilisées dans ce travail. Formellement, un problème d'optimisation se définit comme suit :

$$\min_{x} \quad f_0(x)$$

$$\text{pour} \quad f_i(x) \leq b_i, \quad i = 1, \ldots, m.$$

Dans la suite, nous allons considérer que les problèmes traités sont tous des problèmes de minimisation, mais la théorie pour la maximisation est similaire.

2.1.1 Algorithme de recherche : Backtracking, Largeur et Profondeur

La recherche en largeur (Breadth Fisrt Search (BFS)) consiste à explorer les nœuds et arêtes d'un graphe en parcourant successivement les niveaux à partir d'un nœud origine, aussi appelé source s [2] . L'objectif est de déterminer les distances et les liaisons entre le nœud source

et tous les autres nœuds atteignables depuis celui-ci. Elle est dite « en largeur » car elle visite tous les nœuds situés à une distance k de la source avant de passer à ceux situés à une distance k+1. Ce principe définit une frontière séparant les nœuds déjà découverts de ceux qui ne le sont pas encore. Les nœuds à explorer sont placés dans une file au fur et à mesure de l'exploration, et l'algorithme se poursuit tant que cette file n'est pas vide. Par ailleurs, BFS conserve en mémoire le moment où chaque sommet est rencontré pour la première fois, ce qui permet de construire un arbre couvrant de racine s, reliant la source à l'ensemble des nœuds atteignables avec une distance minimale. Ceci peut être adapté à un graphe avec des poids sur les arêtes pour déterminer l'arbre avec couverture minimal. Enfin, ses principes inspirent plusieurs algorithmes pour graphes pondérés, tels que l'algorithme de Prim pour l'arbre couvrant minimal et l'algorithme de Dijkstra pour le plus court chemin à partir d'une source unique.

La recherche en profondeur (Deep Fisrt Search (DFS)) consiste à explorer un graphe en s'avançant aussi loin que possible à partir d'un sommet donné avant de revenir sur ses pas. Depuis le sommet initial, on choisit un voisin à explorer, puis on répète l'opération à partir de ce nouveau sommet, en marquant le sommet précédent comme visité. L'opération de retour sur trace, plus communément appelée *backtracking*, s'effectue lorsque tous les voisins du sommet courant ont déjà été explorés. Autrement dit, l'algorithme parcourt entièrement une branche du graphe avant de revenir au dernier embranchement non exploré, et ainsi de suite.

Les sommets explorés sont gérés à l'aide d'une pile, où le sommet courant se trouve au sommet de la structure. Les voisins nouvellement découverts y sont ajoutés, puis dépilés au fur et à mesure de l'exploration ; cette logique correspond au fonctionnement d'une pile LIFO (Last In - First Out (LIFO)).

La recherche en profondeur permet notamment de détecter des cycles, mais constitue aussi la base d'algorithmes plus élaborés comme ceux de Kosaraju ou de Tarjan permettant d'identifier des composantes fortement connexes [3].

Un arbre n'étant qu'un cas particulier de graphe, les algorithmes de parcours comme BFS et DFS peuvent également être utilisés pour explorer un arbre de recherche, notamment dans le cadre de l'exploration systématique de l'espace des solutions d'un problème. En particulier, cela peut s'appliquer à la résolution des Programmation Linéaire en Nombre Entiers (PLNE).

2.1.2 Programmation Linéaire en nombre entier

La programmation linéaire (Linear Programming (LP)) est une formulation d'un problème d'optimisation dans laquelle l'objectif et les contraintes sont linéaires. Sous ces mêmes conditions de linéarité, lorsque certaines variables appartiennent à un ensemble fini, on parle de programmation linéaire en nombres entiers mixtes (Mixed Integer Linear Programming (MILP)), et de programmation linéaire en nombres entiers (Integer Linear Programming (ILP)) lorsque toutes les variables sont entières.

Lorsque toutes les contraintes sont linéaires, un problème d'optimisation peut s'écrire sous la forme :

$$\min_{x \in \mathbb{R}^n} \quad c^\top x \quad \text{s.t.} \quad Ax \leq b, \quad x \geq 0$$

où $A \in \mathbb{R}^{m \times n}$ est une matrice et $b, c \in \mathbb{R}^m$ et $\mathbb{R}^n$ sont des vecteurs. L'ensemble défini par A et b correspond à la région faisable, c'est-à-dire l'ensemble des solutions qui satisfont toutes les contraintes. Cette région est convexe et formée par l'intersection d'hyperplans, chaque hyperplan représentant une contrainte.

L'algorithme du simplexe exploite la convexité de l'espace des solutions (polyèdre convexe) et la linéarité de la fonction objectif. Ces propriétés garantissent que tout optimum local est également global et qu'il se situe à un sommet du polyèdre. Le principe du simplexe consiste à se déplacer de sommet en sommet en améliorant l'objectif, jusqu'à atteindre l'optimum.

La partie gauche de la Figure 2.1, représente le domaine réalisable, en jaune, d'un programme linéaire avec des variables continues où les contraintes sont des droites en bleu, définissant les bords du domaine. Dans cette représentation, les solutions explorées par l'algorithme progressent d'un point jaune à un autre avant d'atteindre celui qui maximise l'objectif.

Lorsque certaines variables sont entières, la complexité du problème augmente fortement. En effet, la convexité est perdue et une exploration plus systématique de l'espace des solutions devient nécessaire.

Une représentation de l'espace des solutions réalisables entières défini par les mêmes contraintes dans la Figure 2.1 est représenté à droite. Les solutions réalisables sont désormais constituées d'un ensemble discret de points, et l'optimum ne se situe pas nécessairement sur les frontières de la région faisable. Notons z_{IP} (Integer Program), le minimum recherché sur l'espace de solutions entières inclut dans S noté $S_{\mathbb{Z}}$.

$$\min_{x \in S \subset \mathbb{R}^n} f(x) = z_{LP} \leq z_{IP} = \min_{q \in S_{\mathbb{Z}} \subset \mathbb{Z}^n} f(q),$$

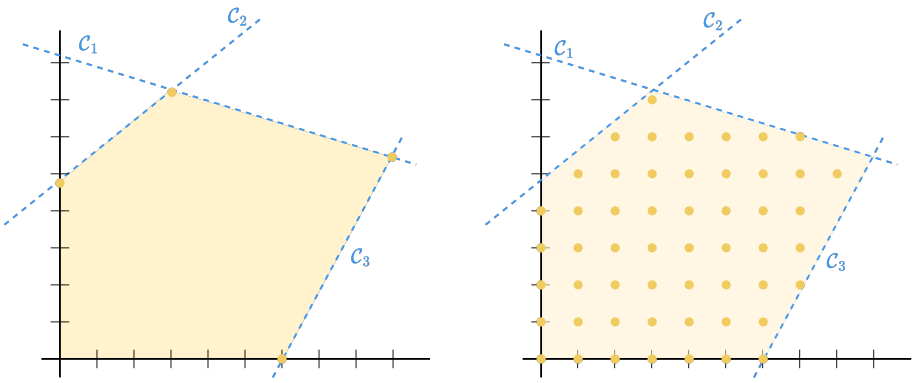

FIGURE 2.1 Comparaison de l'espace des solutions réalisables ayant les mêmes contraintes pour un problème linéaire continu (gauche) et en nombre entier (droite)

où $S_{\mathbb{Z}} = \{q \in \mathbb{Z}^n \mid q \in S\}$.

Ici, z_{LP} désigne la solution de la relaxation linéaire sur le sous-ensemble S. Cette valeur fournit une borne inférieure de z_{IP} sur S. En conservant la meilleure solution entière rencontrée jusqu'alors, il est possible de déterminer si un sous-domaine mérite d'être exploré. Si la borne inférieure fournie par la relaxation linéaire est inférieure à la valeur de la meilleure solution entière connue, le sous-domaine est « branché », c'est-à-dire divisé en sous-domaines plus restreints, chacun correspondant à un choix particulier de bornes sur les variables. Ce processus est répété de manière itérative jusqu'à trouver des solutions entières. En revanche, si la meilleure solution entière connue est déjà inférieure ou égale à la borne fournie par la relaxation linéaire, cela signifie qu'aucune meilleure solution entière ne peut être trouvée dans ce sous-domaine. Celui-ci est alors élagué (*prune*), et l'exploration se poursuit sur d'autres branches issues de la décomposition de l'espace des solutions faisables. Le processus décrit jusqu'à maintenant est la base de l'algorithme du Branch and Bound (BandB).

D'autres algorithmes existent et sont souvent utilisés en combinaison avec le BandB. Le *Branch-and-Cut* intègre au schéma du BandB l'ajout dynamique de contraintes (plans de coupe) afin de resserrer la relaxation linéaire et réduire l'espace de recherche. Le *Branch-and-Price* combine le BandB avec la génération de colonnes, ce qui permet de traiter efficacement des modèles comportant un très grand nombre de variables potentielles.

2.1.3 Heuristiques

Les méthodes exactes, bien qu'elles garantissent l'obtention d'une solution optimale, peuvent s'avérer trop coûteuses en temps de calcul, voire irréalisables, pour les problèmes de grande

taille, complexes ou de nature NP-difficile. Dans ces situations, le recours à des méthodes heuristiques constitue une alternative pertinente.

Une heuristique est une méthode d'optimisation ne garantissant pas l'optimalité, mais visant à fournir, dans un temps raisonnable, une solution réalisable de bonne qualité. Elle repose sur l'exploration de l'espace des solutions à partir de trois composantes fondamentales :

- S : l'espace des solutions possibles
- f : la fonction objectif à optimiser
- N : la ou les structures de voisinage définissant les solutions accessibles depuis une solution donnée

Ces trois éléments constituent les piliers de la conception d'une heuristique et doivent être définis de manière cohérente avec le problème traité.

On distingue deux grandes familles d'heuristiques [4] :

- **Les méthodes de construction** : produisent une solution complète à partir de rien, en ajoutant progressivement des éléments jusqu'à ce qu'une solution faisable soit obtenue.
- **Les méthodes d'amélioration** : partent d'une solution initiale et cherchent à l'améliorer par exploration de son voisinage.

Certaines approches combinent ces deux phases, comme la méthode Greedy Randomized Adaptive Search Procedure (GRASP) [5], qui construit d'abord une solution, puis applique une recherche locale pour l'améliorer. Dans notre étude, nous nous concentrons sur la phase de construction.

Parmi les méthodes de construction, l'approche gloutonne (*greedy*) consiste à ajouter, à chaque étape, l'élément qui améliore le plus la valeur de la fonction objectif f. L'algorithme s'arrête lorsque plus aucun élément ne peut être ajouté sans violer les contraintes. Bien que cette méthode soit rapide et simple à implémenter, elle peut conduire à des solutions sous-optimales, car elle ne tient pas compte des effets à long terme des choix effectués.

Une variante, la construction gloutonne aléatoire, introduit un degré de hasard dans le processus. Au lieu de sélectionner systématiquement le meilleur élément, on définit une liste restreinte de candidats (Restricted Candidate List (RCL)) contenant plusieurs bons choix potentiels selon un critère glouton, puis on en sélectionne un au hasard. Dans le cadre de GRASP, cette technique constitue la première phase, suivie d'une recherche locale visant à améliorer la solution ainsi construite.

Enfin, parfois le choix d'un élément dans l'ensemble de bases va influencer les valeurs des éléments restant dans E. Des méthodes dites adaptatives vont tenir compte de l'évolution de la solution en cours de construction. Ce caractère adaptatif (*adaptive-greedy*) oriente la

sélection des prochains ajouts en fonction de leur impact sur l'objectif, améliorant ainsi la pertinence de la solution finale.

2.2 Travaux connexes

Le transport multimodal est défini comme le transport d'une marchandise de son départ à sa destination finale utilisant au moins 2 modes de transport différent. En fonction des marchandises transportées, plusieurs types d'opérations peuvent être effectuées pour tirer parti du transport multimodal. Bien que les termes « intermodal » et « multimodal » soient souvent utilisés de manière interchangeable, il existe une distinction importante : le transport multimodal peut impliquer la manutention des marchandises elles-mêmes pendant les transferts, tandis que le transport intermodal désigne spécifiquement le déplacement de l'unité d'expédition entière, généralement un conteneur, via plusieurs modes de transport sans manutention du contenu [6]. Similairement, les opérations de transfert entre les différents modes peuvent varier d'appellation. En français, le terme transbordement ne fait pas la différence entre manipulation des marchandises ou du contenant extérieur seulement. En anglais, le terme *transloading* fait spécifiquement référence à l'opération de manipulation des marchandises dans leur transfert d'un mode de transport a l'autre. Le *transloading* est une opération clairement multimodale dans laquelle les marchandises sont déchargées et rechargées dans des conteneurs différents (par exemple, des conteneurs, des palettes, des sacs) [7]. L'objectif principal est de minimiser les étapes intermédiaires en déchargeant, reconditionnant et rechargeant efficacement la cargaison.

En logistique, une catégorisation courante du champ d'un problème consiste à le considérer comme stratégique, tactique ou opérationnel. Le niveau stratégique se rapporte aux décisions à long terme et de haut niveau, le niveau tactique traite de l'horizon moyen terme et de la gestion globale, tandis que le niveau opérationnel concerne les décisions quotidiennes à court terme.

Dans la littérature, les études portant spécifiquement sur le *transloading* sont rares. Celui-ci peut concerner des marchandises liquides ou sèches et est parfois confondu avec des problématiques logistiques connexes, comme le transbordement au sens large (*transshipment*). Les exigences et défis spécifiques d'une installation de transbordement varient selon le type de marchandise traitée. Selon Thomson [8], quatre grandes catégories de marchandises sont couramment transbordées : vrac (sec ou liquide), dimensionnel (ex. bois brut, verre, acier), équipements (ex. machines, véhicules) et marchandises de type entrepôt (ex. produits de consommation, denrées alimentaires). Même au sein d'une seule catégorie, les procédures de manutention peuvent différer de manière significative : par exemple, la gestion de vrac sec

alimentaire impose des contraintes différentes de celles liées aux minerais. Les considérations exposées dans Thomson [8] détaillent également les équipements, caractéristiques de conception et configurations ferroviaires nécessaires pour s'adapter à chaque type de marchandise. Ce degré élevé de variabilité rend difficile l'établissement d'une nomenclature standard, étant fortement dépendant du contexte.

De nombreuses études portent sur les aspects stratégiques des installations de transbordement, tels que l'estimation des coûts et la localisation des sites [9–13]. La planification de la construction et de l'exploitation de ces infrastructures implique des investissements importants. Par exemple, Smith *et al* [14] propose un cadre d'estimation des coûts pour les installations de transbordement, basé sur les volumes projetés et les caractéristiques des marchandises. Une étude qualitative menée par Patel [15] présente un processus de conception pour les terminaux de vrac sec, mettant particulièrement l'accent sur les transferts vrac-à-vrac ne nécessitant pas de reconditionnement particulier. Le transbordement de vrac sec peut également se faire directement dans des navires spécialisés. Dans ce contexte, Cigolini et Rossi [16] utilise une approche par simulation pour déterminer la taille appropriée des navires de transbordement, soulignant l'importance de considérer le transbordement comme un processus de production impliquant manutention, stockage et mélange. En revanche, Dick et Brown [17] se concentre sur le transbordement de liquides et propose des lignes directrices pour le dimensionnement et la configuration des infrastructures ferroviaires.

Seules quelques études traitent spécifiquement des opérations de transbordement. Par exemple, Newdome et Gibson [18] optimisent les flux de charbon dans un réseau comprenant des stockages intermédiaires avec des teneurs en soufre variables, en intégrant des contraintes de mélange afin de ne pas dépasser un seuil fixé. De même, Cheng *et al* [19] étudie le transbordement entre deux modes de transport en vrac (navire à navire). Bien que centrée sur le transport maritime, la structure décisionnelle est analogue : décider de stocker la marchandise ou de la charger directement sur le mode sortant. Contrairement aux approches qui considèrent les flux entrants comme fixes, cette étude met l'accent sur la coordination entre navires entrants et sortants pour maximiser les opportunités de transbordement direct. Cependant, ces travaux se limitent aux transferts entre modes de transport en vrac. À notre connaissance, aucune recherche ne traite des stratégies et de la planification du transbordement lors du passage d'un mode vrac à un mode conteneurisé, ce qui peut nécessiter un reconditionnement sur palettes ou autres unités de manutention intermédiaires.

Transshipment D'autres problématiques décisionnelles présentent néanmoins des structures et défis similaires à ceux du transbordement étudié ici. Une opération clé dans le transport intermodal est le *transshipment*, qui consiste à transférer des conteneurs — unités

d'emballage externes — entre différents modes de transport (ex. du rail vers le navire). Dans le reste de cette section, "transbordement" fera référence à *transshipment*. Ce processus implique une coordination des flux entrants et sortants, des stratégies de stockage temporaire et une planification temporelle, ce qui le rapproche conceptuellement de notre contexte. Par exemple, aux interfaces rail-mer, les conteneurs subissent souvent des délais de stationnement importants dans les terminaux à quai. Vis et De Koster [20] explorent ces opérations, en mettant en lumière les défis de planification rencontrés dans les terminaux à conteneurs.

Des défis opérationnels importants apparaissent dans les terminaux intermodaux. Les problèmes spécifiques varient selon les acteurs impliqués — transporteurs locaux, exploitants de terminaux ou opérateurs intermodaux [21]. Aux jonctions rail-mer, la prise de décision englobe la gestion des zones de stockage et la planification du chargement des trains. Bien que l'empilage de conteneurs soit moins pertinent pour notre étude, les plans de chargement ferroviaire présentent des similitudes avec les opérations de déchargement. Corry et Kozan [22] ont développés un modèle dynamique d'affectation des conteneurs aux emplacements dans les trains, minimisant la manutention et tenant compte du stockage en cours. Ce travail a été prolongé en 2008 [23] avec un MILP visant à réduire le temps de manutention et la longueur finale du train. Des approches par simulation, comme celle de Benna et Gronalt [24], offrent également des cadres flexibles pour analyser différents scénarios de terminal, incluant les arrivées de camions/trains et les opérations de manutention. Xie et Song [25] élargissent la perspective en intégrant des stratégies de stockage dans les terminaux rail-mer pour minimiser les coûts logistiques totaux, avec prise en compte d'incertitudes via un modèle de programmation dynamique stochastique.

Cross-docking Le cross-docking est une opération logistique où les cargaisons des camions entrants sont directement transférées vers les camions sortants avec un stockage minimal. Cela se déroule dans des installations à fenêtres de temps serrées où l'affectation des quais est optimisée pour réduire le temps de manutention et de stockage [26]. Le cross-docking et le transbordement partagent plusieurs caractéristiques structurelles : gestion des flux entrants et sortants, logique de consolidation, manutention et nécessité de stratégies de stockage efficaces. Cependant, le cross-docking repose fortement sur une planification précise des horaires des camions, ce qui le rend particulièrement sensible aux incertitudes [27]. Ces caractéristiques communes en font une référence utile pour les décisions de transbordement, notamment en matière d'aménagement, de planification des flux et de réactivité du système. Yu et Egbelu [28] modélisent le séquencement des camions pour minimiser le temps de traitement total dans un système à convoyeurs, en proposant des heuristiques pour réduire l'occupation du stockage. Des extensions par Arabani [29] ont testé différentes métaheuristiques pour des scénarios

similaires. Des modèles intégrés combinant affectation des portes de quai, stockage temporaire et planification des cargaisons [30], ainsi que des travaux sur les systèmes automatisés de stockage et de récupération pour produits périssables [31], offrent également des principes de conception pertinents. Par ailleurs, des approches par simulation ont été appliquées pour analyser les flux, la congestion et le débit dans des environnements incertains [32, 33].

Néanmoins, le cross-docking diffère sensiblement de notre contexte en termes de priorités décisionnelles. Si les deux systèmes impliquent la coordination des flux entrants et sortants, le cœur du cross-docking réside généralement dans le séquencement et la planification des camions, souvent sous fortes contraintes temporelles. En revanche, notre problématique met l'accent sur les stratégies de chargement, les contraintes de reconditionnement et la planification des flux du vrac vers le mode conteneurisé. Bien que certains modèles intégrés de cross-docking intègrent des stratégies de stockage et de consolidation, leurs décisions opérationnelles se concentrent sur la coordination des véhicules et des quais, tandis que notre approche cible la gestion de la transformation de la marchandise, la complexité de la manutention et l'adaptation aux exigences spécifiques des produits dans le processus de transbordement.

Container Loading Problem Enfin, certaines contraintes issues de la littérature sur le Probleme de Chargement de Conteneur (Container Loading Problem (CLP)) [34] se retrouvent également dans le contexte du transbordement. Le CLP consiste à déterminer la manière de charger des articles dans des conteneurs, c'est-à-dire à disposer de plus petits éléments dans des unités plus grandes, tout en respectant des contraintes de chargement et en optimisant des objectifs tels que minimiser le nombre de conteneurs ou maximiser la valeur. Par exemple, [35] introduit deux types de contraintes que l'on peut aussi retrouver dans notre problème : la contrainte d'expédition complète et la contrainte d'incompatibilité. La première impose que tous les composants d'un groupe spécifique soient chargés ensemble, tandis que la seconde interdit que certains articles soient chargés simultanément dans le même conteneur.

La revue de littérature montre que, bien que plusieurs domaines connexes comme le transshipment, le cross-docking aient été largement étudiés, aucune recherche ne traite directement des opérations de transbordement du vrac vers des unités conteneurisées nécessitant un reconditionnement intermédiaire. Les travaux existants se concentrent soit sur des flux homogènes et massifs (vrac sec ou liquide), soit sur des systèmes normalisés (conteneurs, cross-docking), sans considérer la complexité spécifique des opérations de manipulation des matériaux. Ainsi, notre étude s'inscrit dans un champ encore peu exploré, en proposant une modélisation pionnière pour analyser et optimiser les décisions opérationnelles propres aux centres logistiques de transbordement de palettes.

CHAPITRE 3 MÉTHODOLOGIE

3.1 Hypothèses

Nous commençons par énoncer les principales hypothèses sur lesquelles repose notre méthode, qui servent à formaliser le cadre décisionnel :

— *Quantité exacte* : la quantité dans chaque wagon est supposée fixe et exacte. Bien que cette hypothèse soit généralement vérifiée, des exceptions peuvent survenir dans certains cas, par exemple en raison d'imprécisions dans les mesures de poids ou d'incertitudes liées à la manutention, entraînant de légères différences avec la quantité réelle à charger. Néanmoins, cette hypothèse nous permet d'exploiter des quantités déterministes sur l'ensemble du train afin de consolider précisément les palettes.

— *Abstraction du stockage* : les emplacements de stockage sont considérés comme non spatialisés et non spécifiques. Cette abstraction permet à la méthode de rester indépendante de l'agencement de l'entrepôt et des équipements, ce qui accroît sa généricité et son adaptabilité.

— *Intégrité des lots* : les palettes restantes issues d'un même lot ne sont jamais séparées. Autoriser une telle séparation élargirait considérablement l'espace des solutions, chaque palette devant alors être traitée comme un élément décisionnel distinct. Préserver l'intégrité des lots réduit non seulement la complexité du calcul, mais reflète également les pratiques opérationnelles courantes.

— *Politique d'échange de lots batch swapping* : les palettes restantes d'un lot sont systématiquement échangées avec des palettes provenant du lot qui lui est destiné pour consolidation. Par conséquent, le nombre de palettes à consolider devient la somme des palettes restantes du lot précédent et des palettes du lot en cours de déchargement. Cette approche garantit qu'au maximum deux lots sont combinés dans un même conteneur, simplifiant ainsi la consolidation au sein d'une étape. Bien que cette hypothèse puisse parfois interrompre la phase de chargement en vrac, nous considérons que le respect de la contrainte de mélange de lots prime sur les retards potentiels que cette politique d'échange pourrait engendrer. La Figure 3.1 illustre l'effet de cette hypothèse. Elle présente une situation où des palettes jaunes — provenant du stockage, de l'étape précédente ou d'une autre ligne de chargement en cours — doivent être combinées avec des palettes bleues. La partie gauche de la figure montre une stratégie de consolidation par défaut, produisant un groupe mixte de palettes issues de deux lots distincts et nécessitant donc des opérations supplémentaires pour éviter un

mélange avec un troisième lot. En revanche, la partie droite illustre l'approche par échange de lots : les palettes jaunes restantes sont échangées avec des palettes bleues appartenant à un conteneur complet. Le nombre final de palettes à consolider reste inchangé, mais elles appartiennent toutes au même lot (bleu), ce qui élimine tout mélange supplémentaire et garantit que les consolidations suivantes ne concernent qu'un seul lot.

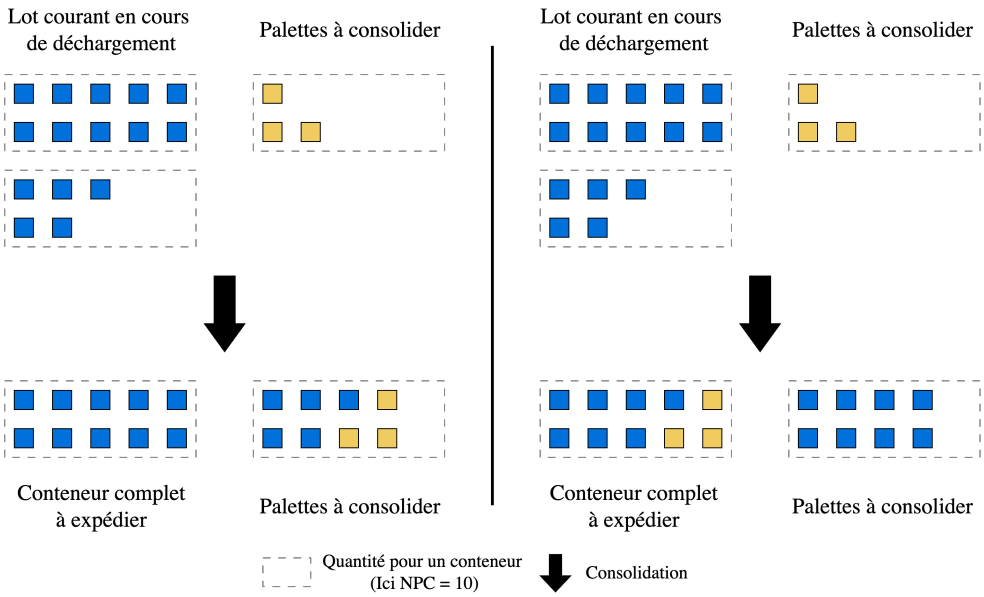

FIGURE 3.1 Comparaison de l'hypothèse d'échange de lots *batch swapping* . À gauche, nous illustrons une consolidation sans échange de lots, tandis qu'à droite, l'échange de lots est appliqué.

— *Pas d'optimisation intra-étape* : les décisions de consolidation ne sont pas optimisées au sein d'une même étape de déchargement. Le modèle suppose que les opérations les plus chronophages se produisent lors des transferts vers et depuis le stockage, et non entre les lignes de chargement.
— *Coût linéaire* : le temps nécessaire pour déplacer les palettes est supposé proportionnel, de manière linéaire, au nombre de palettes déplacées. Dans ce modèle, le nombre de palettes sert de base à notre indicateur de congestion.

Dans la suite, la stratégie courante FIFO est analysée, elle servira de point de comparaison pour évaluer la performance de l'approche proposée.

3.2 Référence opérationnelle : FIFO

La règle FIFO est couramment appliquée dans les opérations logistiques. Dans notre contexte, cela signifie que toute palette restante d'un produit est regroupée avec le lot suivant du même produit présent sur le train. La consolidation a généralement lieu soit au début, soit à la fin d'une étape de chargement/déchargement, car on veille en règle générale à ne pas interrompre la phase de déchargement en vrac une fois celle-ci entamée. Pour mettre en œuvre FIFO, un ensemble de règles statiques doit encadrer les décisions de consolidation. Ces règles sont appliquées pour un produit donné et à chaque étape :

- Si le produit n'apparaît pas à l'étape de chargement/déchargement suivante, consolider toutes les palettes provenant des lignes en cours et transférer les palettes restantes vers la zone de stockage.
- Si le produit apparaît à l'étape suivante :
  - Pour les lignes où le produit réapparaît, laisser les palettes dans la zone de chargement (la consolidation se fera au début de l'étape suivante).
  - Pour les lignes où le produit ne réapparaît pas, consolider ces palettes avec celles provenant d'autres lignes disponibles et les transférer directement vers les lignes où le produit réapparaîtra.

Nous notons que ces règles peuvent déclencher l'application de la politique d'échange de lots décrite précédemment, afin d'éviter toute violation de la contrainte de mélange de lots.

La Figure 3.2 présente deux exemples d'application des règles FIFO. Chaque rectangle représente un wagon correspondant à un ensemble de palettes restantes, tel qu'il apparaît à la position du chargeur. Les rectangles blancs indiquent les occurrences d'un même produit, les lettres servant à distinguer les lots individuels. Le schéma est structuré selon deux dimensions : le temps, indiqué par les étiquettes d'étapes, et l'espace, représenté ici par les trois lignes de chargement. Les points de contrôle temporels et l'évolution chronologique reflètent à la fois le temps nécessaire au repositionnement du train et celui requis pour l'arrivée des premières palettes.

Dans la partie supérieure de la Figure 3.2, la consolidation des lots A et B peut avoir lieu à l'étape s sur la ligne 1 ou la ligne 2, selon le nombre de palettes à déplacer et l'agencement de centre logistique. Après mise en stockage, les palettes restantes sont placées sur la ligne 1 pour être consolidées avec le lot C au début de l'étape $s + 2$. La partie inférieure de la figure illustre un autre scénario. Après la consolidation de A et B à l'étape s, les palettes restantes sont placées sur la ligne 3, car le lot C y apparaît à l'étape $s+1$. Le résultat de cette consolidation, effectuée sur la ligne 3, est ensuite transféré vers la ligne 2 pour être consolidé

avec le lot D à l'étape $s + 2$. On remarque que, dans ce second exemple, l'application de la politique FIFO ne génère aucun stockage de palettes.

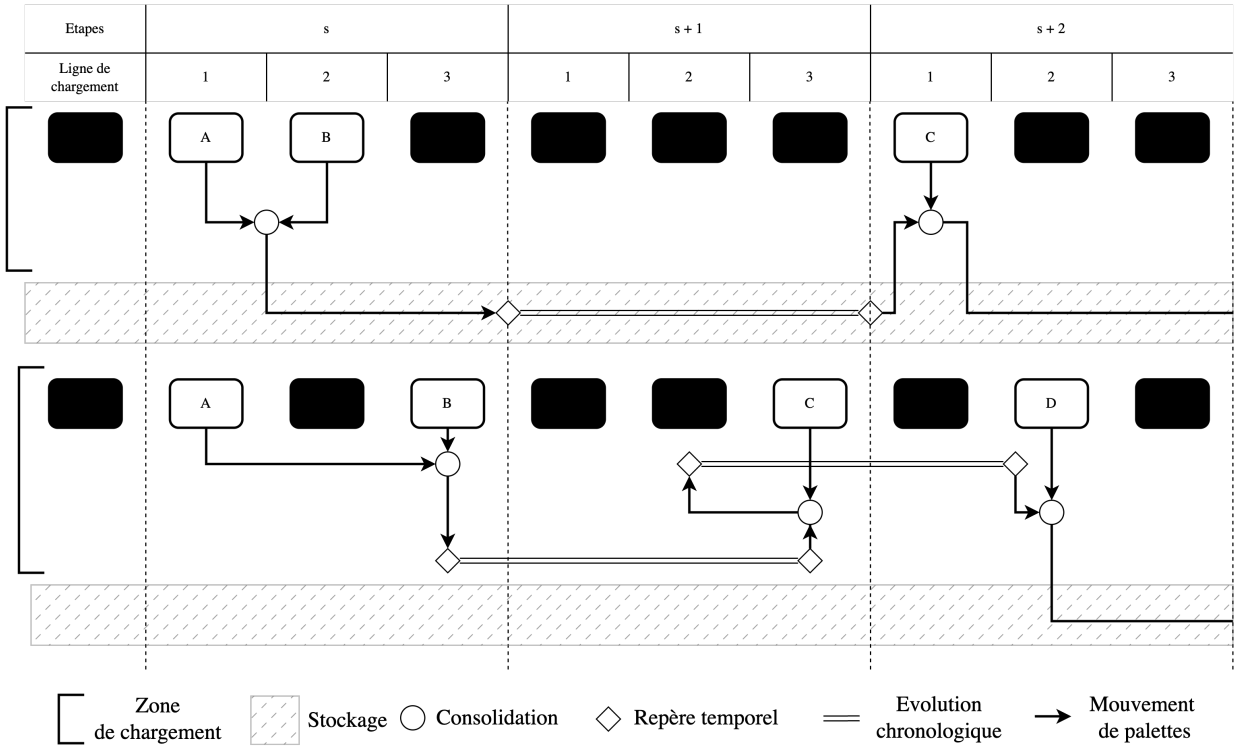

FIGURE 3.2 Exemples de séquences de consolidation selon la règle FIFO.
Haut : exemple avec mise en stockage intermédiaire
Bas : exemple avec apparition consécutive des lots

FIFO : Borne inférieure en stockage

Nous évaluons ici l'impact de la stratégie actuelle en termes de stockage. Les palettes sont mises en attente lorsque aucun autre lot du même produit n'apparaît à l'étape de déchargement suivante. En conséquence, les palettes restantes doivent être stockées jusqu'à ce qu'un lot ultérieur du même produit apparaisse dans le train.

Notre problème repose sur un système entrée/sortie dans lequel la sortie est conditionnée par la contrainte de mélange et par NPC. Par construction, la stratégie FIFO consolide les palettes dans les conteneurs dès que possible. Par conséquent, toute déviation par rapport au FIFO ne peut qu'augmenter — ou au mieux égaler — la demande de stockage du FIFO. En d'autres termes, FIFO fournit une borne inférieure pour le stockage. Empêcher la formation d'un conteneur entraîne nécessairement des volumes de stockage plus importants lors de l'exécution du plan de consolidation (voir Figure 3.3).

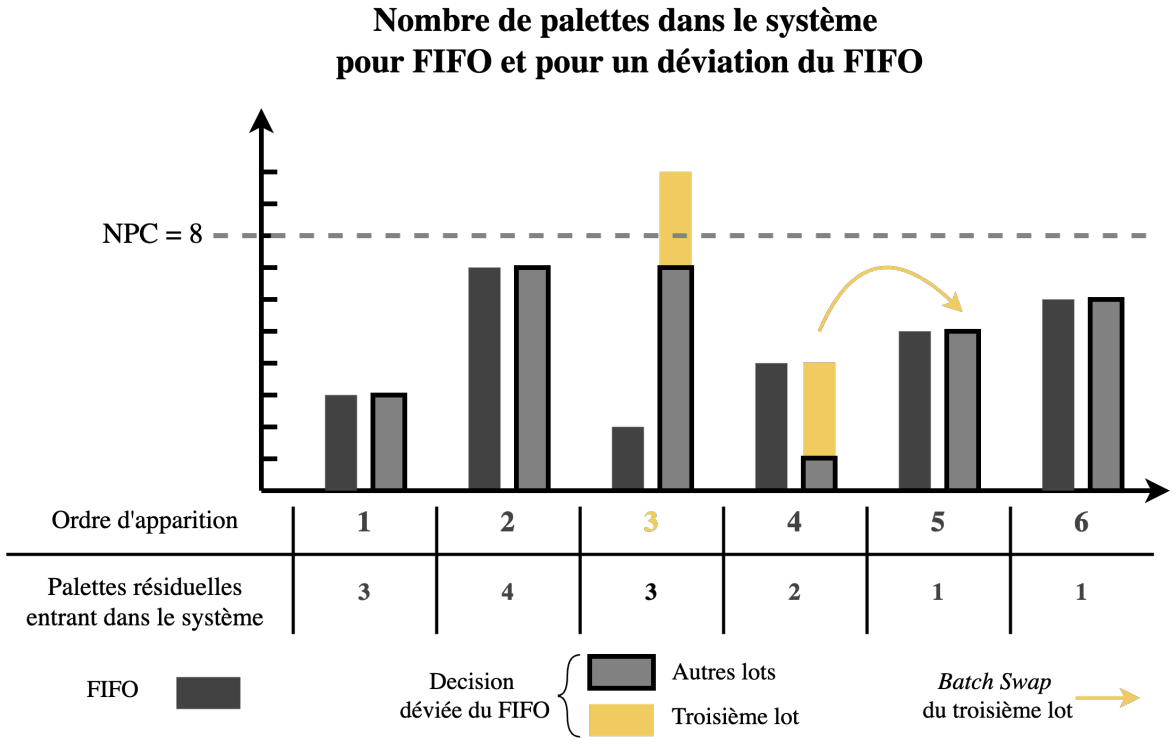

FIGURE 3.3 Illustration graphique de l'infériorité en stockage du FIFO

La figure 3.3 illustre un exemple qui représente les palettes résiduelles dans le système en fonction des ordres d'apparitions des lots d'un seul produit. Elle représente le nombre de palettes dans le cas du FIFO et dans le cas où l'on prend la décision que le troisième lot du produit va être ignoré dans la chaîne de consolidation du FIFO. On considère ici que les lots sont tous espacés d'au moins une étape temporelle à laquelle une optimisation du stockage intermédiaire ne serait pas nécessaire. La figure permet de schématiser ce qu'il se passe lorsque l'on décide de dévier, ne serait-ce qu'un peu, du FIFO. Dans la mesure où une décision de combinaison doit être prise sur les palettes du troisième lot, on considère ici qu'elles vont être consolidées avec le cinquième lot (non le quatrième car ce serait équivalent au FIFO). On connaît à l'avance le nombre de palettes résiduelles et ainsi le nombre de conteneurs pouvant être formés, ici 1, car :$(3+4+3+1+2+1)/NPC = 1$. Lorsque la quantité augmente d'une étape à l'autre, cela veut dire que les palettes résiduelles apportées par le lot courant n'ont pas suffi à former un conteneur, comme on peut le voir entre les apparitions du premier lot et du deuxième lot : $(3+4) \bmod NPC = 7$. Ici, dès que le nombre le permet, à l'apparition du troisième lot, il y a formation d'un conteneur et la quantité de palettes résiduelles dans le système baisse pour le FIFO, car : $(3+4+3) \bmod NPC = 2$.

En revanche, dans le cas où l'on a pris la décision de ne pas combiner le troisième lot dans la chaîne de consolidation, le nombre de palettes dans le système va se cumuler et pourra dépasser NPC. Pour la décision déviée, la formation du conteneur se produira lors de l'apparition du quatrième lot, et ainsi égalera le nombre de palettes résiduelles du FIFO dans le système.

Compte tenu du fait qu'un train transporte plusieurs produits et que ceux-ci ne peuvent être mélangés, ils coexistent simultanément dans la zone de stockage. En appliquant un FIFO par défaut pour tous les produits, on peut atteindre le volume de stockage maximal nécessaire pour une instance donnée. Comme le nombre d'emplacements de stockage est fixé, ce nombre constituera notre référence de capacité de stockage pour une instance donnée. Dépasser cette référence impliquerait d'investir dans une capacité de stockage supplémentaire. En revanche, prendre des décisions alternatives tout en restant sous la limite FIFO est acceptable et permettrait une meilleure utilisation des zones tampons. Il arrive que l'on considère des espaces de stockage supérieurs à la limite FIFO pour une certaine instance, dans la mesure où une installation est dimensionnée pour répondre à une variété d'instances et non à la seule instance testée. Comprendre les arbitrages entre l'espace de stockage et la congestion peut éclairer le dimensionnement optimal des installations et la planification opérationnelle.

Exemples simplifiés

Puisque nous avons établi une borne inférieure pour le stockage, nous souhaitons désormais concentrer notre attention sur l'efficacité opérationnelle dans cette limite totale de stockage.

Pour illustrer l'utilité de notre approche, nous proposons deux exemples principaux. Chaque exemple correspond à un motif d'apparition qui, combiné à l'autre, peut définir la topologie d'apparition de tout produit dans le train. Notre analyse se concentre sur le type et le volume de mouvements de palettes, utilisés ici comme indicateurs de coût opérationnel et de congestion. Dans ces exemples, la politique d'échange de lots est appliquée et chaque schéma de consolidation doit récupérer des palettes en stockage lorsqu'un lot désigné apparaît pour consolidation.

La Figure 3.4 présente un cas où tous les lots nécessitent un stockage intermédiaire, car toutes les apparitions du produit sont séparées par au moins une étape. La partie supérieure illustre la stratégie FIFO standard, et la partie inférieure montre un choix alternatif avec deux variantes. Les lignes verticales pointillées indiquent les étapes de déchargement. Le choix alternatif présenté correspond à une consolidation de A avec C, et de B avec D. Comme il s'agit d'un exemple simplifié, tous les lots doivent finalement être consolidés à la fin. On observe que le nombre total de mouvements de palettes est identique entre la stratégie

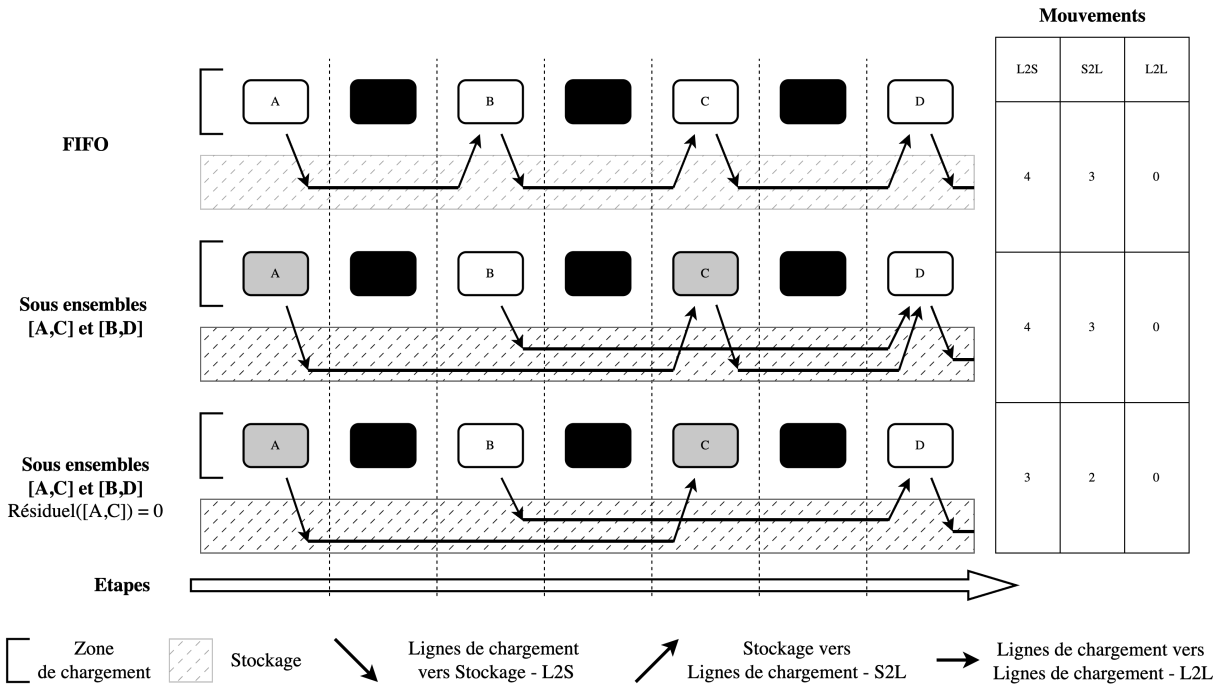

FIGURE 3.4 Exemple illustrant les mouvements lors de l'application de différentes stratégies de consolidation pour des apparitions non consécutives de lots

FIFO et l'alternative au centre. Cela indique que des décisions de consolidation alternatives n'augmentent pas nécessairement la quantité des mouvements.

En outre, le second scénario alternatif, en bas, montre que les palettes résiduelles peuvent être complètement éliminées grâce à une combinaison appropriée des lots — entraînant ainsi zéro palette restante. Cela réduit les allers-retours entre la zone de chargement et la zone de stockage, diminuant ainsi la congestion.

Globalement, il est important de noter que des décisions alternatives peuvent réduire les mouvements de palettes entre la zone de chargement et le stockage. Plus précisément, le nombre total de palettes déplacées dans un plan de consolidation alternatif peut être inférieur à celui du FIFO, dépendamment de la formation des conteneurs.

La Figure 3.5 illustre un cas où deux lots apparaissent consécutivement. Comme le suggère la partie centrale, il n'y a aucun avantage particulier à décider de stocker lorsque le même produit apparaît à l'étape suivante. Cela augmente le nombre de mouvements en remplaçant une opération L2L par deux opérations coûteuses, L2S et S2L. Cependant, ce type de décision peut encore être utile si le coût additionnel entraîne des économies sur d'autres portions du plan de consolidation.

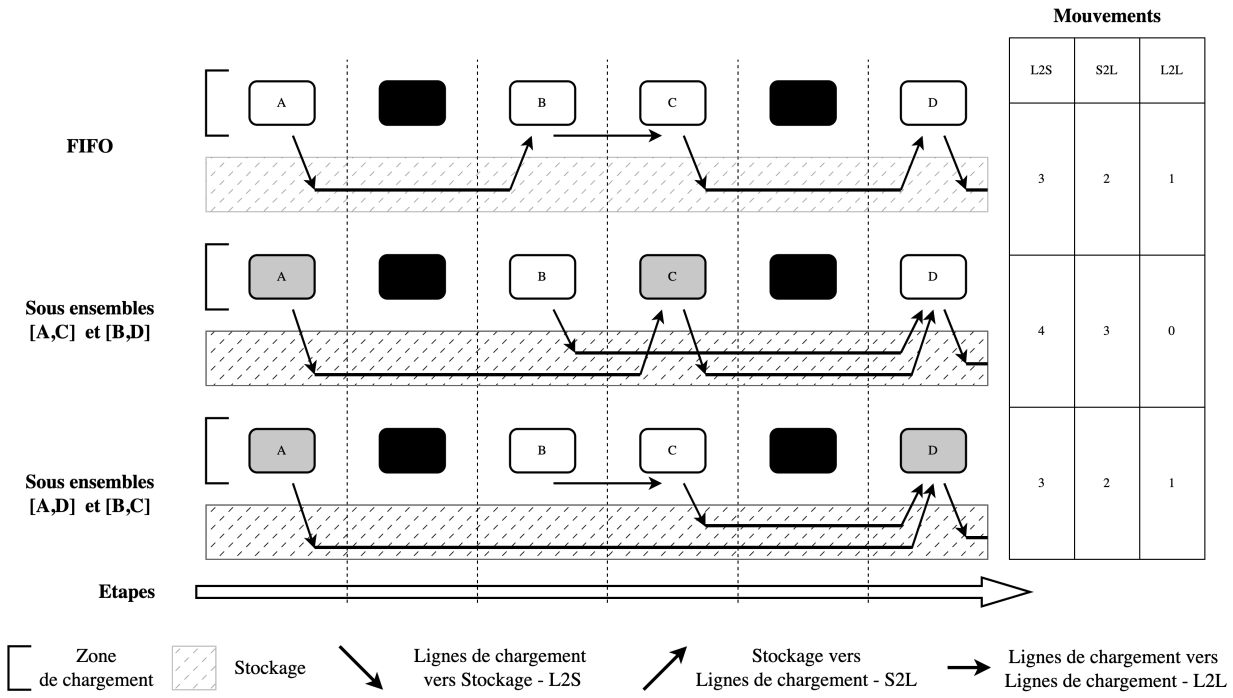

FIGURE 3.5 Exemple illustrant les mouvements lors de l'application de différentes stratégies de consolidation pour des apparitions consécutives de lots

Dans l'exemple du bas, le mouvement L2L est conservée, mais A n'est pas systématiquement consolidé avec B. Le nombre de mouvements est similaire, mais le volume total de palettes transitant peut être significativement différent. Par exemple, lorsque ni [A,B] ni [A,B,C] ne forment un conteneur, il n'est pas nécessaire d'apporter les palettes de A lors de l'apparition de B, puisqu'il faudrait alors déplacer deux fois les palettes restantes de A — une fois pour les amener vers B et une autre pour les renvoyer en stockage après C. Il est préférable de les mobiliser uniquement lorsque cela contribue à former un conteneur plus tard dans le processus de déchargement.

Dans les deux exemples simplifiés, nous constatons que le choix de règles de consolidation alternative à FIFO peuvent égaler, voire surpasser, une politique FIFO stricte en termes de mouvements totaux de palettes, grâce à des combinaisons de lots plus avantageuses et à l'évitement des trajets inutiles vers le stockage. Ainsi, l'efficacité opérationnelle découle de choix de consolidation flexibles qui réduisent les déplacements et allègent la congestion sur l'ensemble du processus de chargement.

Basé sur ces exemples, nous avons formulé un modèle mathématique permettant de traduire des décisions de la nature de ceux présenté.

3.3 Formulation

3.3.1 Modélisation

La formulation MILP considère en entrée :

- la *composition du train*, définie par ses produits et les lots associés ;
- la *longueur du train* T_L, correspondant au nombre de wagons ;
- le *nombre de lignes de déchargement* L_D disponibles dans l'installation ;
- le *nombre de palettes* NPC nécessaires pour former un conteneur complet ;
- la *la limite de stockage des palettes* ST imposé.

Chaque étape de chargement/déchargement $s \in \{1, \ldots, S_D\}$ implique le déchargement simultané de N_l wagons. Par conséquent, le nombre d'étapes temporelles S_D est donné par :

$$S_D = \begin{cases} \left\lfloor \dfrac{T_L}{N_l} \right\rfloor & \text{if } (T_L \bmod N_l) = 0 \\ \left\lfloor \dfrac{T_L}{N_l} \right\rfloor + 1 & \text{otherwise.} \end{cases} \tag{3.1}$$

Chaque wagon transporte un lot distinct d'un produit appartenant à $P = \{p_1, \ldots, p_n\}$. Par conséquent, les lots de produits sont indexés par $b \in B = \{1, \ldots, T_L\}$. Pour chaque produit $p_i \in P$, on note $B_i \subseteq B$ l'ensemble des indices des lots de p_i. L'étape temporelle t_b d'un lot b correspond au moment où il est programmé pour être déchargé du train et est donné par $t_b = \left\lfloor \frac{b}{N_l} \right\rfloor + 1$. Lors du traitement de ces lots sur la ligne de déchargement, les palettes sont formées. On note $w_b \in [0, NPC - 1]$ le nombre de palettes restantes créées à partir du lot b.

La formulation proposée pour notre problème de transbordement de marchandises, ci-après notée `TCP`, énumère dans Ω_i, pour $i = 1, \ldots, n$, tous les sous-ensembles d'occurrences de lots pour chaque produit $p_i \in P$, où chaque sous-ensemble représente une sélection possible de lots d'un même produit planifiés pour être consolidés ensemble. Pour chaque $q \in \Omega_i$, on note c_q son coût opérationnel, correspondant au nombre total de palettes déplacées vers et depuis le stockage lorsque les lots indexés dans q sont consolidés ensemble. Pour chaque plan de consolidation, le calcul du coût repose sur l'application d'une politique FIFO avec échange de lots, ce qui nécessite de récupérer des palettes en stockage, si elles sont disponibles, chaque fois qu'un nouveau lot de q doit être consolidé. Le stockage résultant de la consolidation des lots de q dépend du nombre de palettes restantes générées lors du processus de consolidation.

Définissons, pour chaque sous-ensemble $q \in \Omega_i$, ℓ_{qs} comme le nombre de palettes restantes du produit p_i en stockage à l'étape s, résultant de la consolidation des lots de q. En particulier, pour $q = q^1, \ldots, q^{|q|}$,

$$\ell_{qs} = \begin{cases} 0 & \text{si } s \leq t_{q^1} \text{ ou } (s = t_{q^i} \text{ et } t_{q^i} - t_{q^{i-1}} = 1 \text{ pour } i > 1) \\ \left(\sum_{j \in q, t_j \leq s} w_j \right) \bmod NPC & \text{sinon.} \end{cases} \tag{3.2}$$

Cette définition par cas distingue deux situations en fonction du calendrier d'arrivée des lots au sein de l'ensemble q. Le premier cas, où $\ell_{qs} = 0$, s'applique lorsque l'étape s a lieu avant la consolidation du premier lot de q (c'est-à-dire $s \leq t_{q^1}$), ou lorsque le lot courant q^i est immédiatement précédé du lot précédent q^{i-1} à des étapes consécutives ($t_{q^i} - t_{q^{i-1}} = 1$). Selon notre règle opérationnelle, les palettes issues de lots qui réapparaissent à des étapes consécutives restent dans la zone de chargement et sont consolidées au début de l'étape suivante, de sorte qu'aucune palette restante ne s'accumule en stockage. Dans le second cas, lorsqu'il existe un intervalle entre les lots ou lorsque plusieurs lots sont arrivés jusqu'à l'étape s, le nombre total de palettes $\sum_{j \in q, t_j \leq s} w_j$ peut ne pas être un multiple de la capacité de consolidation NPC. L'opération modulo calcule alors le nombre de palettes restantes qui ne peuvent pas être consolidées en unités complètes.

Enfin, nous définissons $\sigma_q = t_{q^{|q|}} + 1$, comme le moment après la réalisation de q. Pour $s \geq \sigma_q$, ℓ_{qs} est constant, et nous considérons que les palettes restantes du produit p_i dues à q peuvent être consolidées avec les palettes restantes d'autres sélections $q' \in \Omega_i$, avec $q' \neq q$.

La Figure 3.6 illustre la dynamique du stockage des palettes restantes due à la consolidation de deux sous-ensembles différents de lots q_1 (en jaune) et q_2 (en bleu). Dans notre approche de modélisation, nous supposons que les opérations de consolidation ont lieu à la fin de la phase de chargement en vrac d'une étape, ce qui signifie que les palettes nécessitant un stockage sont reportées à l'étape suivante. Étant donné que la FIFO est appliquée à chaque sous-ensemble, l'occupation du stockage ne dépasse jamais NPC. Le traitement du lot q_1^1 dans q_1 (en jaune) entraîne le placement d'un certain nombre de palettes restantes en stockage à l'étape 2. Dans l'exemple, l'occupation du stockage augmente en raison de la consolidation des palettes restantes du lot q_1^1 avec celles du lot q_1^2. Il est important de noter que, dans le cadre de notre hypothèse d'échange de lots, toutes les palettes stockées à l'étape 4 proviennent du lot q_1^2. Après la consolidation du dernier lot q_1^3 dans q_1, c'est-à-dire à partir de $s = \sigma_{q_1} = 8$, le nombre de palettes restantes générées par la réalisation de q_1 reste constant. Le comportement de consolidation des lots dans q_2 suit un schéma similaire. Des palettes restantes sont produites lors du traitement de q_2^1 à l'étape 3. Cependant, la consolidation de ces palettes avec celles du lot q_2^2 à l'étape 5 ne génère pas de stockage supplémentaire, puisque le lot q_2^3 survient immédiatement après q_2^2 à des étapes consécutives. Après la consolidation du dernier lot q_2^4

dans q_2, c'est-à-dire à partir de $s = \sigma_{q_2} = 10$, le nombre de palettes restantes générées par la réalisation de q_2 reste constant. Dans notre formulation, après les étapes σ_{q_1} et σ_{q_2}, les palettes restantes générées par q_1 et q_2, respectivement, peuvent être combinées pour former des unités complètes de conteneur, à condition qu'elles appartiennent au même produit.

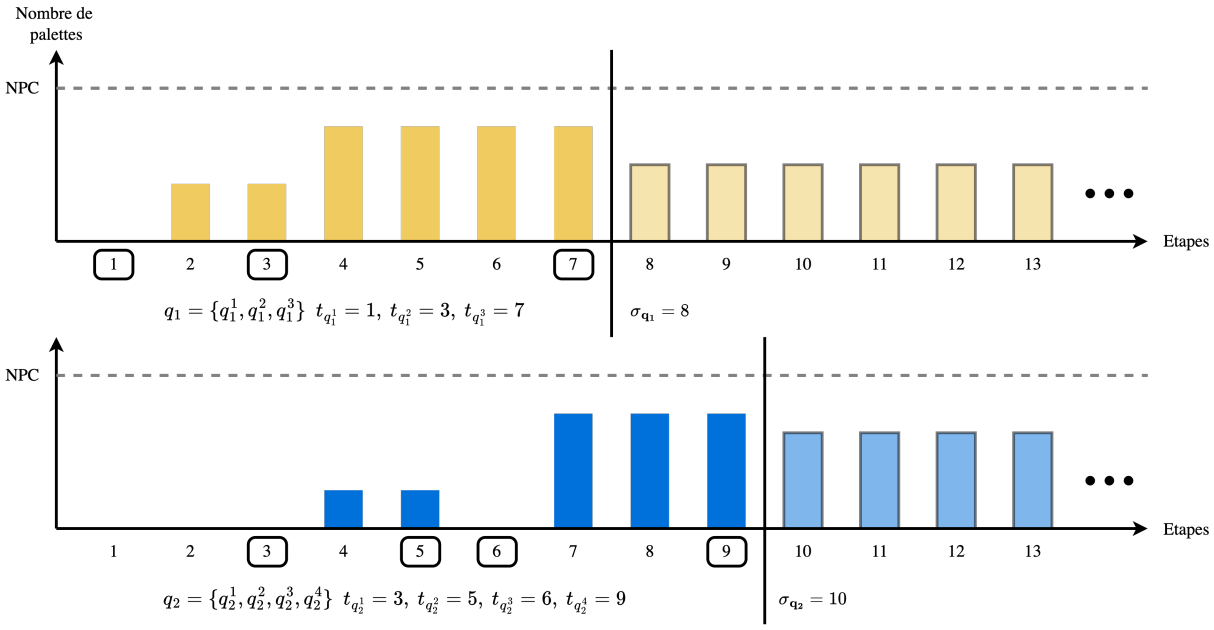

FIGURE 3.6 Représentation de ℓ_{qs} pour les sous-ensembles q_1 et q_2

La formulation mathématique de TCP est la suivante :

$$\min \quad \sum_{i=1}^{n} \sum_{q \in \Omega_i} c_q \boldsymbol{X_q} \tag{3.3}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{q \in \Omega_i} \ell_{qs} \boldsymbol{X_q} - \left(\sum_{i=1}^{n} \boldsymbol{Y_{is}} \right) NPC \leq ST, \qquad \forall s \in \{1, \ldots, S_D\} \tag{3.4}$$

$$\boldsymbol{Y_{is}} NPC \leq \sum_{\substack{q \in \Omega_i \\ s \geq \sigma_q}} \ell_{qs} \boldsymbol{X_q} < (\boldsymbol{Y_{is}} + 1) NPC, \quad \forall\, i \in \{1, \ldots, n\},\ \forall\, s \in \{1, \ldots, S_D\} \tag{3.5}$$

$$\sum_{\substack{q \in \Omega_i \\ b \in q}} \boldsymbol{X}_q = 1, \qquad \forall i \in \{1, \ldots, n\}, \forall b \in B_i \tag{3.6}$$

$$\boldsymbol{X_q} \in \{0, 1\}, \quad \forall i \in \{1, \ldots, n\},\ \forall\, q \in\ \Omega_i; \quad \boldsymbol{Y_{is}} \in \mathbb{N}, \quad \forall\, i \in \{1, \ldots, n\},\ \forall s \in \{1, \ldots, S_D\}, \tag{3.7}$$

où la variable $\boldsymbol{X_q} = 1$ si le sous-ensemble q est choisi pour la consolidation dans notre plan de transbordement, et $\boldsymbol{X_q} = 0$ sinon. $\boldsymbol{Y_{is}}$ est une variable entière représentant le nombre d'unités complètes de stockage (chacune d'une capacité NPC) utilisées pour stocker les palettes restantes du produit p_i à l'étape s.

La fonction objectif (3.3) vise à minimiser le coût opérationnel total induit par la sélection de sous-ensembles de lots pour la consolidation, tous produits confondus. Afin d'éviter tout biais, la formation de conteneurs n'est pas incluse dans l'objectif, car les variables $\boldsymbol{Y}$ ne tiennent compte que des conteneurs formés à partir des palettes restantes des sous-ensembles sélectionnés, et non de ceux implicitement générés lors du traitement des variables $\boldsymbol{X}$. De plus, comme le nombre total de conteneurs résultant du processus de consolidation est prédéterminé et fixe, les inclure dans l'objectif n'influencerait pas le résultat de l'optimisation.

Les contraintes (3.4) garantissent que le nombre total de palettes restantes stockées à l'étape s, ajusté des palettes consolidées en unités complètes de conteneurs, ne dépasse pas la capacité de stockage ST de l'installation.

Les contraintes (3.5) assurent la cohérence entre le nombre de palettes restantes et le nombre d'unités complètes de conteneurs ($\boldsymbol{Y_{is}}$) formées pour chaque produit p_i à l'étape s. Plus précisément, l'expression centrale de l'inégalité calcule le nombre total de palettes restantes provenant des sous-ensembles sélectionnés qui peuvent être combinées à l'étape s pour le produit p_i. Ce calcul inclut uniquement les sous-ensembles q pour lesquels $s \geq \sigma_q$. Ces contraintes réalisent ainsi le quotient de la division euclidienne par NPC, de sorte que $\boldsymbol{Y_{is}}$ reflète correctement le nombre d'unités complètes de conteneurs formées.

Les contraintes (3.6) garantissent que chaque lot b du produit p_i est inclus dans exactement un sous-ensemble $q \in \Omega_i$ sélectionné. Chaque variable de décision est, par nature, spécifique à un produit. Ainsi, le partitionnement défini en (3.6) pour chaque produit constitue un sous-problème indépendant. Une représentation visuelle de cette matrice de contraintes en blocs diagonaux est présentée à la Figure 3.7, où chaque variable couvre un sous-ensemble d'un produit, et où l'équation garantit que les sous-ensembles choisis sont mutuellement exclusifs.

La Figure 3.8 illustre une solution réalisable dans notre formulation `TCP`. Elle considère que les mêmes sous-ensembles de lots q_1 et q_2 présentés à la Figure 3.6 sont sélectionnés pour la consolidation, c'est-à-dire $\boldsymbol{X_{q^1} = 1}$ et $\boldsymbol{X_{q^2} = 1}$. On observe que, dans l'exemple fourni, le stockage dépasse le volume NPC nécessaire à la formation d'un conteneur, car les deux processus de consolidation se déroulent en parallèle avant $\sigma_{q_1} = 8$ pour q_1, et $\sigma_{q_2} = 10$ pour q_2. Les palettes restantes de q_1 (en jaune) aux étapes 8 et 9 ne peuvent pas être combinées avec celles de q_2 (en bleu), car q_2 conserve encore ses palettes en stockage pour les consolider avec le lot q_2^4 à l'étape 9. À l'étape 10, la somme des palettes restantes de q_1 et q_2 dépasse

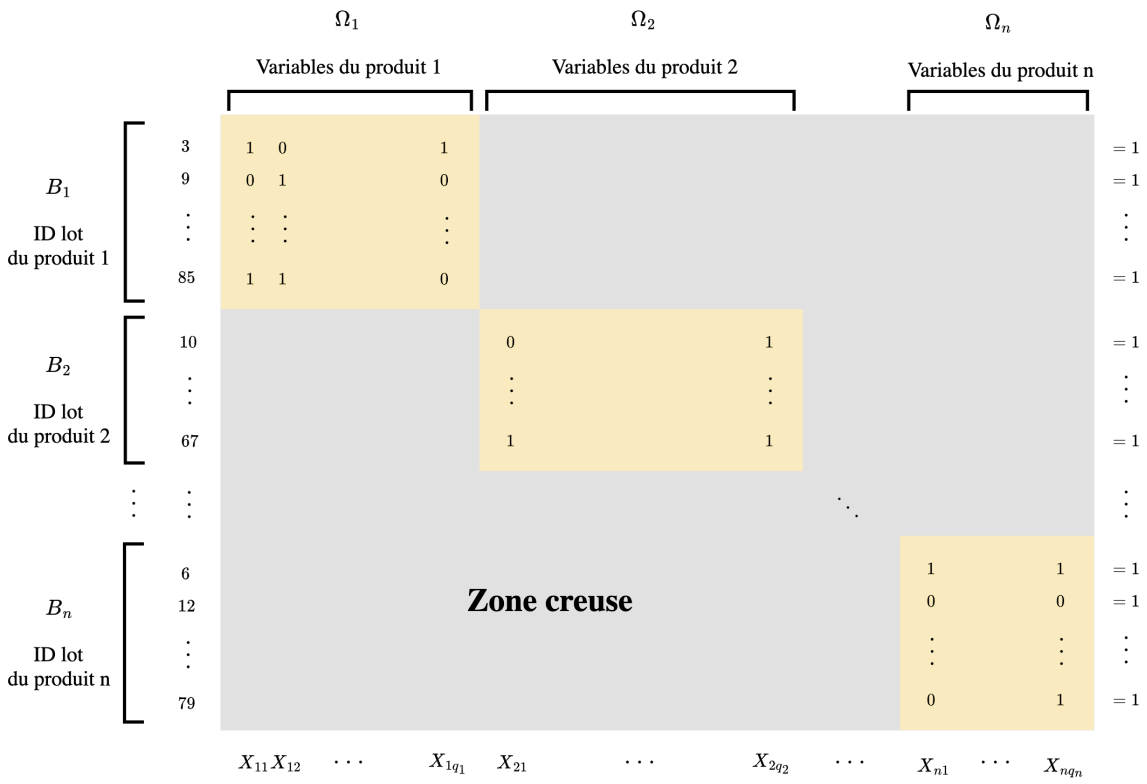

FIGURE 3.7 Représentation de la matrice des contraintes 3.6 de partitionnement

NPC (sans toutefois excéder $2 \cdot NPC$). Par conséquent, $Y_{is} = 1$ pour $s \geq 10$.

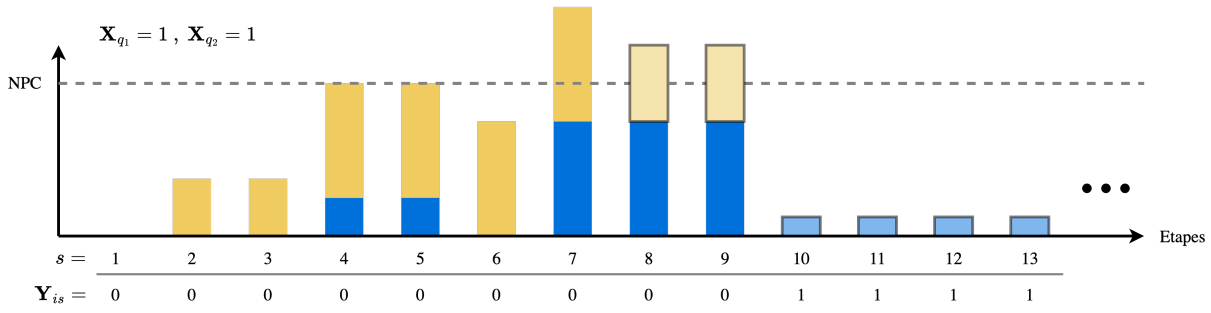

FIGURE 3.8 Exemple de représentation de solution dans la formulation `TCP`.

3.3.2 Remarques

Récursivité Dans notre problème, la récursivité introduit une complexité importante : chaque wagon génère des palettes restantes, et chaque combinaison de lots produit sa propre

quantité — variable — de palettes résiduelles. Comme dans le cas de base, ces palettes restantes apparaissent à des moments précis du déchargement, ont un nombre fixe de palettes et nécessitent une consolidation. Pour simplifier le processus et éviter une complexité opérationnelle excessive, nous adoptons une politique FIFO pour consolider les palettes restantes des sous-ensembles q pour lesquels $\boldsymbol{X_q = 1}$. Selon cette politique, les conteneurs sont formés dès qu'un nombre suffisant de palettes est atteint. La variable $\boldsymbol{Y_{is}}$ capture le nombre de conteneurs formés à partir des palettes restantes des sous-ensembles, à partir du début du processus de déchargement.

Notre approche traite la récursivité à travers deux niveaux de prise de décision. Le premier niveau concerne les opérations sur $\boldsymbol{X_q}$ avant σ_q, où l'on prend en compte la formation de conteneurs et le stockage des palettes issues des lots au sein de chaque sous-ensemble q sélectionné. Le second niveau, représenté par $\boldsymbol{Y}$, correspond à des décisions qui dépendent de la partition spécifique choisie et interviennent après σ_q. Bien que ces opérations entraînent des coûts, ceux-ci ne sont pas directement inclus dans le modèle, mais sont pris en compte séparément en pratique.

Solution FIFO de référence La solution FIFO de référence peut être représentée de plusieurs manières dans notre formulation. Un exemple trivial consiste à sélectionner uniquement des sous-ensembles composés d'un seul lot (i.e. $|q_j| = 1$ et $\cup_j q_j = B_i$) pour chaque produit. Une autre représentation peut être obtenue en limitant les sous-ensembles à des lots consécutifs.

Ces différentes représentations de la solution FIFO peuvent entraîner des coûts opérationnels différents, car le coût de congestion associé à la variable $\boldsymbol{Y_{is}}$ n'est pas explicitement inclus dans le modèle `TCP`. Le seul sous-ensemble $q \in \Omega_i$ dont le plan de consolidation est équivalent à la solution FIFO de référence et présente le même coût est $q = B_i$. Comme un tel plan est réalisable dans notre formulation, toute solution alternative qui s'écarte de la représentation FIFO « lot complet » pour l'ensemble des produits peut être interprétée comme un plan de planification alternatif — à condition qu'il fasse l'objet d'une évaluation approfondie a posteriori par l'utilisateur.

Le modèle adopte une vision globale, dans laquelle un coût légèrement plus élevé pour un produit peut être justifié s'il permet de réaliser des économies encore plus importantes pour un autre, ce qui peut se traduire par une réduction nette de la congestion opérationnelle totale.

Contraintes de mélange Notre formulation distingue deux niveaux principaux de mélange : le mélange de produits et le mélange de lots. Pour le mélange de produits, nous nous assurons que les lots provenant de produits différents ne sont jamais consolidés. Plus précisément, la formulation partitionne les combinaisons de lots en ensembles Ω_i, pour $i = 1, \ldots, n$, de telle sorte que chaque ensemble ne contienne que des sous-ensembles de lots issus d'un seul produit. De plus, les variables $\boldsymbol{Y_{is}}$ permettent la combinaison de palettes exclusivement issues du produit p_i, l'indexation rendant cette association explicite.

Le mélange de lots, quant à lui, est traité de manière implicite. Il est capturé par l'hypothèse selon laquelle un échange de lots est effectué chaque fois que des palettes provenant de deux lots différents du même produit sont regroupées, ce qui garantit qu'au maximum deux lots du même produit sont présents dans un conteneur.

Il est important de noter que les distinctions entre lots ne sont pas modélisées explicitement. Les opérations spécifiques d'échange de lots doivent être déterminées a posteriori. Le modèle `TCP` présenté se concentre sur le calcul des quantités de palettes et sur l'ajustement des valeurs en fonction de la complétion des variables $\boldsymbol{Y}$. La planification, l'ordonnancement et l'exécution détaillée des opérations d'échange de lots ne relèvent pas du périmètre de cette approche. Notre objectif principal est de déterminer quelles unités de palettes restantes doivent être combinées et d'évaluer comment ces décisions influencent l'occupation du stockage et les mouvements de palettes, plutôt que de décomposer davantage le temps ou de prescrire la séquence exacte des échanges de lots.

3.3.3 Taille du modèle

Présentons ici le défi en termes de nombre de variables et de nombre de contraintes. Le nombre de contraintes algébriques est entièrement déterminé par $|P|$, T_L et S_D. Le nombre total de contraintes algébriques est donné par :

$$\#\ contraintes = S_D + 2S_D|P| + T_L = S_D(1 + 2|P|) + T_L$$

où :

- S_D correspond aux contraintes de l'équation (3.4),
- $2S_D|P|$ regroupe les contraintes de formation des conteneurs de l'équation (3.5),
- T_L représente le partitionnement des sous ensembles dans l'équation (3.6).

Par exemple, avec $T_L = 100$ lots, $|P| = 10$ produits, et $L_D = 2$ portes, on obtient $S_D = 50$ et $\#\ contraintes\ = 1150$, ce qui correspond à un nombre de contraintes modéré.

Concentrons-nous maintenant sur les contraintes d'intégrité et de positivité (3.7) des va-

riables. Les variables Y_{is} sont facilement dénombrables à partir des paramètres précédents, il y a $|P|S_D$ variables Y_{is}.

En revanche, le nombre de variables X_q dépend du nombre de lots par produit. Plus précisément, pour chaque produit p_i, le nombre de sous-ensembles possibles est $2^{m_i} - 1$, où m_i désigne le nombre d'apparitions du produit p_i. Par exemple, si un produit apparaît 15 fois, cela génère plus de 32 000 variables rien que pour ce produit, ce qui devient ingérable.

Ainsi, le nombre total de variables est fortement influencé par la quantité et la distribution des occurrences de produits dans une instance. Il est donc essentiel de mettre en œuvre des stratégies de résolution adaptatives afin de contrôler et de limiter en conséquence l'ensemble des variables.

3.4 Énumération de sous-ensembles

L'énumération de l'ensemble des sous-ensembles $q \in \Omega_i$ pour chaque produit $p_i \in P$ peut entraîner une explosion combinatoire du nombre de variables de décision, rendant la résolution du `TCP` difficile. Pour éviter cette énumération exhaustive, nous adoptons une phase de construction qui génère uniquement des sous-ensembles q jugés prometteurs selon un ensemble de critères de sélection, seules les variables ainsi obtenues sont transmises au MILP. Cette approche réduit significativement la complexité de calcul, mais peut aussi écarter certains sous-ensembles optimaux, ce qui peut conduire à des solutions sous-optimales.

Un sous-ensemble $q \in \Omega_i$ peut être vu comme un chemin dans le graphe orienté $G_i = (N_i, E_i)$, où chaque nœud $n_a \in N_i$ représente un lot du produit p_i, et chaque arc $(n_a, n_b) \in E_i$ relie un lot n_a à un lot n_b ultérieur déchargé du train. Nous définissons un nœud terminal spécifique $n_{term} \in N_i$, représentant un lot artificiel final dans tout chemin valide. Comme G_i est un graphe orienté acyclique (Directed Acyclic Graph (DAG)), ses nœuds peuvent être ordonnés topologiquement, reflétant l'ordre de traitement des lots.

Nous utilisons une procédure de construction itérative pour bâtir progressivement un chemin dans G_i, en ajoutant un nœud-lot à la fois. Soit $\overline{q}$ le chemin partiel construit à une itération donnée, et $\mathcal{Q} \subseteq N_i$ l'ensemble des nœuds candidats éligibles pour étendre $\overline{q}$.

L'ensemble candidat $\mathcal{Q}$ est défini dynamiquement comme suit :

- Initialement, lorsque $\overline{q}$ est vide, tous les nœuds sont admissibles, donc $\mathcal{Q} = N_i$.
- Si $\overline{q}$ n'est pas vide et que son dernier nœud est n_a, alors $\mathcal{Q}$ est restreint aux successeurs de n_a, soit $\mathcal{Q} = \{n_b \in N_i \mid (n_a, n_b) \in E_i\}$.

À chaque itération, un nœud de $\mathcal{Q}$ est sélectionné et ajouté à $\overline{q}$, et le processus se poursuit jusqu'à ce que le chemin atteigne le nœud terminal n_{term}. Chaque étape de sélection est

guidée par un ensemble de critères $\mathcal{C}$, tels que

$$\sum_{j \in \mathcal{Q}} prob_j^{\mathcal{C}_k} = 1 \quad \forall\, \mathcal{C}_k \in \mathcal{C}$$

où $prob_j^{\mathcal{C}_k} \in [0,1]$ correspond à la probabilité d'ajouter n_j à $\overline{q}$ selon le critère $\mathcal{C}_k \in \mathcal{C}$.

Ces probabilités servent à orienter une construction gloutonne randomisée, favorisant les lots plus attractifs — par exemple, ceux apparaissant rapidement après le lot courant et ne générant pas une occupation excessive du stockage. Nous résumons ci-dessous nos critères de sélection $\mathcal{C}$:

— Coût ($\mathcal{C}_1$) : Nous calculons le coût incrémental d'ajout d'un nœud lot n_j à $\overline{q}$.

$$u_1^j = c_{\overline{q} \cup \{n_j\}} - c_{\overline{q}} \quad , \quad \text{avec} \;\; prob_j^{\mathcal{C}_1} = \frac{\frac{1}{u_1^j}}{\sum_{j \in \mathcal{Q}} \frac{1}{u_1^j}}$$

— Nombre d'étapes ($\mathcal{C}_2$) : Nous considérons le nombre d'étapes entre le dernier nœud lot n_a ajouté à $\overline{q}$ et le nœud lot candidat n_j. Ce critère est utile pour déterminer combien de temps les palettes entre les lots a et j resteront en stockage.

$$u_2^j = (t_j - t_a)^2 + 1 \quad , \quad \text{avec} \;\; prob_j^{\mathcal{C}_2} = \frac{\frac{1}{u_2^j}}{\sum_{j \in \mathcal{Q}} \frac{1}{u_2^j}}$$

— Étapes consécutives ($\mathcal{C}_3$) : Ce critère indique si le nœud lot candidat n_j se situe dans la même étape que le lot n_a ou dans l'étape immédiatement suivante. Comme mentionné précédemment, consolider de tels lots évite des transferts inutiles vers et depuis le stockage.

$$u_3^j = \begin{cases} 1 & \text{si} \;\; t_a \leq t_j \leq t_a + 1 \\ 0 & \text{sinon.} \end{cases} \quad , \; \text{avec} \;\; prob_j^{\mathcal{C}_3} = \begin{cases} \dfrac{1}{|\mathcal{Q}|} & \text{si} \;\; \forall j \in \mathcal{Q} \, , u_3^j = 0 \\ \dfrac{u_3^j}{\sum_{j \in \mathcal{Q}} u_3^j} & \text{sinon.} \end{cases}$$

— Formation de conteneur ($\mathcal{C}_4$) : Ce critère indique si la combinaison des restes de palettes de $\overline{q}$ et du lot n_j aboutit à la consolidation d'un conteneur complet. Étant donné que les restes varient de 0 à $NPC - 1$, la combinaison peut produire au maximum un

conteneur complet.

$$u_4^j = \begin{cases} 1 & \text{si les palettes restantes de } \overline{q} \text{ combinées avec } n_j \text{ complètent un conteneur} \\ 0 & \text{sinon.} \end{cases} \quad , \text{ avec } \quad prob_j^{\mathcal{C}_4} = \begin{cases} \dfrac{1}{|\mathcal{Q}|} & \text{si } \forall j \in \mathcal{Q}\ , u_4^j = 0 \\ \dfrac{u_4^j}{\sum_{j \in \mathcal{Q}} u_1^j} & \text{sinon.} \end{cases}$$

Pour décider de l'arrêt de la construction du chemin, c'est-à-dire en ajoutant le nœud terminal n_{term} à $\overline{q}$, nous considérons un ensemble $\mathcal{Z}$ de métriques décrites ci-dessous. Chaque métrique de $\mathcal{Z}$ est normalisée pour appartenir à l'intervalle $[0, 1]$.

— Fin du train ($\mathcal{Z}_1$) : Ce critère indique à quelle distance se trouve le wagon du lot courant par rapport à la fin du train. Il correspond au rapport entre l'étape du dernier lot n_a dans $\overline{q}$ et le nombre total d'étapes.

$$\mathcal{Z}_1 = \frac{t_a}{S_D}$$

— Nombre de palettes stockées ($\mathcal{Z}_2$) : Ce critère reflète la proportion de palettes restantes après la consolidation de $\overline{q}$. Une valeur proche de 1 indique que $\overline{q}$ génère peu de palettes restantes, tandis qu'une valeur proche de 0 indique que la consolidation de $\overline{q}$ est proche de former un conteneur complet. Dans ce dernier cas, l'ajout de lots à $\overline{q}$ est souhaitable afin de réduire le volume des mouvements de palettes vers et depuis le stockage.

$$\mathcal{Z}_2 = 1 - \frac{\left(\sum_{n_i \in \overline{q}} w_i\right) \bmod NPC}{NPC}$$

— Couverture des lots ($\mathcal{Z}_3$) : Ce critère mesure la proportion de lots inclus dans $\overline{q}$ par rapport au nombre total de lots $|B_i|$.

$$\mathcal{Z}_3 = \frac{|\overline{q}|}{|B_i|}$$

Des valeurs proches de 1 signifient que presque tous les lots de p_i sont consolidés dans un seul sous-ensemble, tandis que des valeurs intermédiaires laissent plus de flexibilité pour d'autres schémas de consolidation.

— Conteneurs formés ($\mathcal{Z}_4$) : Rapport entre le nombre de conteneurs pouvant être assemblés à partir des palettes restantes des lots de $\overline{q}$ et le nombre total de conteneurs pouvant être formés à partir des palettes restantes de tous les lots de ce produit.

$$\mathcal{Z}_4 = \frac{\left\lfloor \left(\sum_{n_i \in \overline{q}} w_i \right) / NPC \right\rfloor}{\left\lfloor \left(\sum_{b \in B_i} w_b \right) / NPC \right\rfloor}$$

Comme notre objectif secondaire est de compléter des conteneurs à partir des palettes restantes, une valeur élevée de ce critère est souhaitable. À l'inverse, un sous-ensemble de palettes qui sont combinées sans jamais former un conteneur ne fait qu'accroître la congestion.

Enfin, la probabilité de sélectionner le nœud terminal $n_{term} \in N_i$ et de l'ajouter à $\overline{q}$ est donnée par

$$\pi_{term} = \min \left(\sum_{r=1}^{4} \zeta_r \mathcal{Z}_r, , 1 \right), \tag{3.8}$$

où $\zeta_r \in [0, 1]$ représente un paramètre de pondération associé au critère $\mathcal{Z}_r$. Ainsi, nous définissons la probabilité d'ajouter à $\overline{q}$ un nœud $n_j \neq n_{term}$ comme suit :

$$\pi_j = (1 - \pi_{term}) \sum_{\mathcal{C}_k \in \mathcal{C}} \phi_k prob_j^{\mathcal{C}_k}, \tag{3.9}$$

où $\phi_k \in [0, 1]$ représente le poids associé au critère $\mathcal{C}_k \in \mathcal{C}$, avec la contrainte $\sum k\phi_k = 1$. Nous remarquons que, par construction,

$$\sum_{j \in \mathcal{Q}} \pi_j + \pi_{term} = 1$$

puisque :

$$\sum_{j \in \mathcal{Q}} \pi_j + \pi_{term} = (1 - \pi_{term}) \sum_{j \in \mathcal{Q}} \sum_{\mathcal{C}_k \in \mathcal{C}} \phi_k prob_j^{\mathcal{C}_k} + \pi_{term} = (1 - \pi_{term}) \sum_{\mathcal{C}_k \in \mathcal{C}} \phi_k \sum_{j \in \mathcal{Q}} prob_j^{\mathcal{C}_k} + \pi_{term} = 1.$$

3.4.1 Réglage des hyper-paramètres

Étant donné un processus paramétré qui génère des sous-ensembles $q \in \Omega$, notre objectif est de déterminer l'ensemble de paramètres optimal permettant de produire des sous-ensembles de haute qualité. À cette fin, nous définissons une fonction objectif destinée à évaluer la qualité d'un ensemble de sous-ensembles $q \in \Omega$.

Le processus étant stochastique, pour évaluer un ensemble de paramètres $\Lambda := \{\phi_k, \mathcal{C}_k \in \mathcal{C}\} \cup \{\zeta_r, \mathcal{Z}_r \in \mathcal{Z}\}$, nous générons Γ (de taille suffisamment grande) sous-ensembles afin

de disperser l'aléatoire. Notons ces sous-ensembles générés par la suite $\mathcal{E}(\Lambda) = (q^\gamma(\Lambda))_{\gamma=1}^{\Gamma}$ où chaque $q^\gamma \in \Omega$, tel que $\Omega = \bigcup \Omega_i$. Comme le processus de construction peut produire plusieurs fois les mêmes plans de consolidation, on note $\mathcal{E}^*(\Lambda) = \{q^\gamma(\Lambda) : \gamma = 1, \ldots, \Gamma\}$. Par conséquent, $|\mathcal{E}^*(\Lambda)| \leq \Gamma$.

Afin de déterminer les valeurs optimales des paramètres Λ, nous considérons la fonction objective suivante :

$$\min_{\Lambda} \quad \alpha_1 \times f_1(\Lambda) + \alpha_2 \times f_2(\Lambda) - \alpha_3 \times f_3(\Lambda) - \alpha_4 \times f_4(\Lambda) \tag{3.10}$$

Le premier terme de (3.10) correspond à la moyenne, sur l'ensemble des éléments de $\mathcal{E}^*(\Lambda)$ de l'occupation du stockage par lot, notée $f_1(\Lambda)$, qui s'écrit :

$$f_1(\Lambda) = \frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} \frac{\sum_{s=1}^{S_D} \ell_{q^\gamma s}}{|q^\gamma(\Lambda)|}.$$

L'élément $\sum_s \ell_{q^\gamma s}$ représente la somme des niveaux de stockage associés à l'ensemble des étapes liées au sous-ensemble q^γ. Il permet ainsi de mesurer l'impact global de ce sous-ensemble sur l'utilisation de l'espace de stockage. Cette valeur est ensuite rapportée au nombre de lots traités par le sous-ensemble, de sorte qu'un sous-ensemble couvrant un plus grand nombre de lots soit considéré comme plus légitime à mobiliser davantage de capacité de stockage. Enfin, l'ensemble est moyenné sur tous les sous-ensembles de Γ, ce qui donne une mesure représentative de l'occupation du stockage par lot et en moyenne sur tous les sous-ensembles construits.

Le deuxième terme, noté $f_2(\Lambda)$ dans (3.10), prend en compte le coût de déplacement des palettes pour les lots lors de l'évaluation de la phase de construction. Il correspond à une normalisation min–max de la moyenne du coût de mouvement par lot, et s'écrit :

$$f_2(\Lambda) = \frac{\frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} \bar{c}_\gamma(\Lambda) - \min_\gamma \bar{c}_\gamma(\Lambda)}{\max_\gamma \bar{c}_\gamma(\Lambda) - \min_\gamma \bar{c}_\gamma(\Lambda)},$$

où

$$\bar{c}_\gamma(\Lambda) = \frac{c_{q^\gamma(\Lambda)}}{|q^\gamma(\Lambda)|}$$

désigne le coût moyen de mouvement de palettes par lot associé au sous-ensemble $q^\gamma(\Lambda)$.

Comme pour le terme relatif à l'occupation du stockage, le coût total des mouvements est rapporté au nombre de lots traités. Ainsi, un sous-ensemble couvrant plus de lots peut légi-

timement entraîner un coût total plus élevé, puisque celui-ci est réparti proportionnellement sur davantage d'unités.

Le troisième terme $f_3(\Lambda)$ indique la variabilité du processus de construction, étant définie par $f_3(\Lambda) = \dfrac{|\mathcal{E}^*(\Lambda)|}{\Gamma}$. Ce membre de l'objectif mesure le rapport entre le nombre de sous-ensembles uniques générés et le nombre total de sous-ensembles construits.

Enfin, le quatrième terme $f_4(\Lambda)$ inclut la formation de conteneur dans la fonction. Notons κ_γ le nombre de conteneurs formés par $q^\gamma(\Lambda)$ et $\widehat{\kappa_\gamma}$ le nombre de conteneurs pouvant être formés au maximum pour le produit de q^γ. On va ainsi définir $f_4(\Lambda) = \left(\sum_\gamma \dfrac{\kappa_\gamma}{\widehat{\kappa_\gamma}}\right) \Big/ \Gamma$ correspondant à la moyenne du ratio du nombre de conteneurs formés sur le nombre de conteneurs pouvant être formés.

Les coefficients ont été choisis de manière empirique, en se basant sur les intervalles de valeurs observés lors d'expériences préliminaires, et sont donnés par $\alpha_1 = 5,\ \alpha_2 = 3,\ \alpha_3 = 5,\ \alpha_4 = 10$.

L'ajustement des paramètres de Λ est réalisé sur l'ensemble des produits d'une instance donnée. Comme le processus de construction dépend de chaque produit, et qu'une instance comprend plusieurs produits, l'optimisation se fait en construisant des quantités de variables équilibrées pour chacun d'entre eux. L'optimisation de la fonction (3.10) est réalisée à l'aide d'`Optuna` [36]. L'espace de recherche est défini en considérant pour chaque paramètre de Λ des valeurs comprises entre 0,01 et 1,00, avec un incrément de 0,01 (soit 100 possibilités). Compte tenu l'existence des huit paramètres, i.e. $|\Lambda| = 8$, l'espace de recherche est déjà vaste, et il a été jugé qu'une granularité plus fine ne serait pas nécessaire, d'autant que le processus conserve une composante aléatoire.

Un paramètre clé dans l'optimisation avec `Optuna` est le nombre d'essais (*trials*), qui correspond au nombre de configurations différentes de Λ explorées par l'algorithme. Il a été constaté qu'il était plus efficace de privilégier un plus grand nombre d'essais plutôt que d'évaluer un nombre restreint de configurations de manière plus précise. En pratique, cette stratégie permet une meilleure exploration de l'espace des poids et conduit à des résultats plus robustes. Les valeurs retenues pour l'optimisation sont $\Gamma = 3000$ et $trials = 20$.

3.5 Logique de résolution finale

Nous considérons le nombre maximal de variables β comme une entrée de notre logique de résolution. Dans un premier temps, ce budget est réparti équitablement entre les n produits de l'instance, de sorte que β/n variables sont alloués pour chaque produit p_i. Si, pour un produit p_i, ce nombre excède le total des sous-ensembles possibles $(2^{m_i} - 1)$, une énumération

exhaustive est réalisée et l'excédent de capacité de génération est redistribué aux autres produits. Cette redistribution est effectuée proportionnellement à la fréquence d'apparition des produits, de manière à privilégier ceux qui apparaissent le plus souvent. Dans le cas où l'énumération exhaustive n'est pas réalisable, la méthode de construction heuristique présentée en 3.4 est utilisée.

Il est néanmoins nécessaire de garantir que la contrainte de partitionnement soit respectée. Ainsi, pour chaque produit $p_i \in P$, l'ensemble des singletons $Q_i^{(1)} = \{\{b\} : b \in B_i\}$ et la représentation FIFO notée $q = B_i$ sont joints à l'ensemble des variables générées du modèle TCP.

Le processus complet de résolution est représenté en Figure 3.9. La décision de recourir à une énumération exhaustive ou à une approche heuristique est prise individuellement pour chaque produit. La phase d'ajustement des paramètres de la méthode de construction est indépendante de la phase de construction et aucun sous-ensemble n'est transféré de l'une à l'autre.

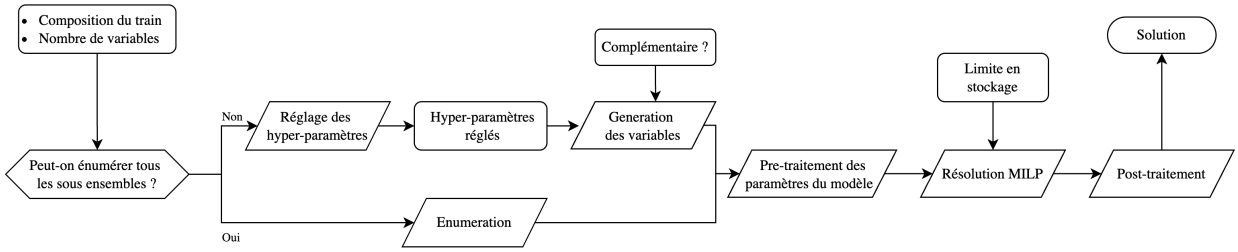

FIGURE 3.9 Processus de résolution de bout en bout

Lors de l'étape de génération des variables, nous nous réservons la possibilité de créer également la variable *complémentaire* d'une variable générée par le processus de construction. Pour $q \in \Omega_i$, sa variable complémentaire $\bar{q} \in \Omega_i$ est définie par

$$\bar{q} = B_i \setminus q,$$

c'est-à-dire l'ensemble des lots de p_i qui ne sont pas sélectionnés par q. L'ajout des variables complémentaires facilite l'optimisation du partitionnement effectué par `TCP`, tout en accélérant le temps de construction par produit.

Enfin, étant donné que le coût associé aux variables Y n'est pas calculé dans le modèle, une étape de post-traitement est nécessaire pour arriver à la solution et au coût de congestion finals.

CHAPITRE 4 RÉSULTATS EXPÉRIMENTAUX

L'implémentation a été mise en œuvre en utilisant Python pour les pré- et post-traitement des données, et Julia pour l'exécution et l'optimisation du modèle à l'aide d'IBM CPLEX. L'ajustement des hyperparamètres a été réalisé avec la bibliothèque Optuna [36]. Toutes les expériences ont été effectuées sur un MacBook Pro équipé d'une puce M3 avec une fréquence maximale de 4,05 GHz et 16 Go de RAM.

La base de comparaison est la règle de consolidation FIFO. L'accent est principalement mis sur les différences d'utilisation du stockage et, surtout, sur la réduction des coûts de mouvement de palettes. Étant donné que FIFO est une règle simple, l'exécution de sa logique en termes de mouvement et de stockage s'effectue en un temps fixe proche de 2 millisecondes. Pour la suite, l'analyse des temps d'exécution se limitera aux approches proposées, afin de déterminer quelle configuration de résolution est la plus adaptée aux contextes opérationnels.

4.1 Scénarios testés

Nous avons testé notre modèle par rapport à la règle de consolidation FIFO dans une série de scénarios synthétiques, décrits ci-dessous :

- Scénario 1 : Uniformité dans les produits et les palettes résiduels. Étant donné un nombre de produits et une longueur de train, pour chaque wagon, nous choisissons uniformément un produit et une quantité de palettes résiduels dans l'intervalle [0, $NPC - 1$].
- Scénario 2 : Uniformité du nombre d'apparitions de chaque produit et de leurs palettes résiduels. Étant donné un nombre d'apparitions M et un nombre de produits n, nous construisons une instance contenant exactement $M \times n$ apparitions. Chaque produit apparaît ainsi M fois, ce qui conduit à une instance de taille $M \times n$. Cette instance est utile pour contrôler précisément le nombre de variables du modèle `TCP`, en particulier pour pouvoir toutes les énumérer en cas d'un faible nombre d'apparitions.
- Scénario 3 : Uniformité dans les produits et distribution normale pour leurs palettes résiduels. Étant donné un nombre de produits et une longueur de train, nous sélectionnons d'abord un produit de manière uniforme. Pour les palettes résiduelles, nous procédons à un échantillonnage à partir d'une distribution gaussienne discrète, centrée sur une moyenne aléatoire propre à chaque produit. L'écart-type est fixé à deux palettes.

— Scénario 4 :Étant donné un nombre de produits et une longueur de train, nous sélectionnons d'abord un produit de manière uniforme, puis nous tirons aléatoirement dans l'intervalle [1, 5] correspondant au nombre de wagons consécutifs qui auront ce produit. Les palettes résiduelles sont uniformes pour chaque wagon. Cela forme une instance avec des blocs d'un même produit. Le but de cette instance est d'analyser une structure de composition fréquemment observée en pratique, où les produits forment des groupes de wagons le long du train.
— Scénario 5 : Résidus bimodaux. Le produit est choisi uniformément, mais les palettes résiduelles proviennent d'une distribution bimodale, plus précisément un mélange équilibré de gaussiennes. Les moyennes sont volontairement placées pour des valeurs faibles et élevées dans l'intervalle $[0, NPC-1]$. Cette instance permet d'analyser le potentiel de notre approche à combiner des quantités de palettes se complétant. En effet, il est souhaitable d'associer de grandes valeurs à de petits résidus afin de former des conteneurs pleins et de minimiser le nombre de palettes résiduelles envoyées et récupérées du stockage.
— Scénario 6 : Produit séquentiel. À partir d'un nombre de produits et d'une longueur de train, les occurrences des produits suivent un ordre séquentiel, seules les palettes résiduelles étant choisies de façon uniforme aléatoire. Ce type d'instance permet d'analyser la manière dont les méthodes se compare en présence d'un schéma d'apparition cyclique.

Cinq instances distinctes ont été générées pour chaque scénario. Pour chacune d'elles, la longueur du train, le nombre de produits et le nombre de lignes de chargement ont été variés afin de constituer un échantillon représentatif et diversifié, tant en termes de difficulté de résolution (nombre d'occurrences) qu'en termes de capacité de combinaison (lignes de chargement). Toutes les instances ont par ailleurs un $NPC = 16$, valeur recommandée par notre partenaire industriel. Néanmoins, la méthode développée reste flexible et peut s'adapter à d'autres valeurs, l'enjeu combinatoire sur les palettes demeurant le même.

Il convient de préciser que certaines configurations particulières n'ont pas été retenues dans notre batterie de tests. C'est notamment le cas des trains mono-produit ou encore des situations où chaque produit apparaît à chaque étape comme un scénario 6 lorsque le nombre de produits est inférieur ou égal au nombre de lignes de chargement. Dans ces situations, les palettes vont toujours être gardées près des zones de chargement pour la prochaine étapes et ne nécessitent donc pas un stockage plus longue durée. Dans ces cas, l'optimisation des mouvements entre la zone de stockage et la zone de chargement ne présente pas d'intérêt, puisque le FIFO constitue déjà une solution optimale. Ces instances relèvent davantage d'une problématique d'optimisation inter-étapes, qui dépasse le cadre de l'étude. Elles n'ont donc

pas été incluses dans l'ensemble des instances testées.

Les ensembles de paramètres des instances générées sont présentés en annexe A.

4.2 Analyse des résultats sur une instance type

Dans cette section, nous examinons une instance particulière afin de mieux illustrer le fonctionnement de l'optimisation. À cet effet, une instance du Scénario 1 a été retenue, et ses caractéristiques sont présentées dans le tableau 4.1.

Tableau 4.1 Caractéristiques de l'instance choisie pour illustration

Paramètre	NPC	T_L	L_D	S_D	n
Valeur	16	100	3	35	10

Produit	1	2	3	4	5	6	7	8	9	10
Nombre d'apparitions	3	11	15	7	10	9	13	11	12	9
Nombre de sous-ensembles	7	2047	32767	127	1023	511	8191	2047	4095	511

Le processus de résolution de notre approche est évalué sur deux hyperparamètres principaux : (i) le nombre maximal de variables énumérées pour le modèle `TCP` ; et (ii) l'énumération des variables complémentaires à celles générées afin potentiellement améliorer la couverture obtenue par le modèle `TCP`.

4.2.1 Évaluation du nombre maximal de variables énumérées

L'instance type a été résolue à quatre reprises, en utilisant des graines aléatoires différentes, pour un éventail de tailles maximales de variables compris entre 2500 et 12500. Chaque exécution a été réalisée indépendamment, avec un ajustement individuel des hyperparamètres. La figure 4.1 illustre, pour chacune des expérimentations, la répartition du temps d'exécution total par étape de résolution en considérant différents limites β de variables générées pour le modèle `TCP`.

Le premier point à remarquer est qu'il est naturel que l'augmentation du nombre de variables accroisse le temps de résolution. En effet, un plus grand nombre de variables, en particulier lorsqu'elles sont entières, entraîne un accroissement du branchement dans l'arbre d'exploration du solveur, et donc un temps de calcul plus élevé. Il est important de noter que, pour l'ensemble des résolutions lancées, le réglage des hyperparamètres reste globalement fixe et dépend principalement du nombre de sous-ensembles construits Γ ainsi que du nombre d'es-

sais (*trials* d'Optuna). Dans le cas des paramètres qui ont été fixés, ce réglage s'effectue en $5,68 \pm 0,16$ s. En tenant compte de cet aspect, on comprend que les composantes les plus variables du temps de résolution total sont la construction des sous-ensembles et la résolution du `TCP`. En particulier, on peut voir que le temps nécessaire pour construire les variables (en rose) a une variance importante.

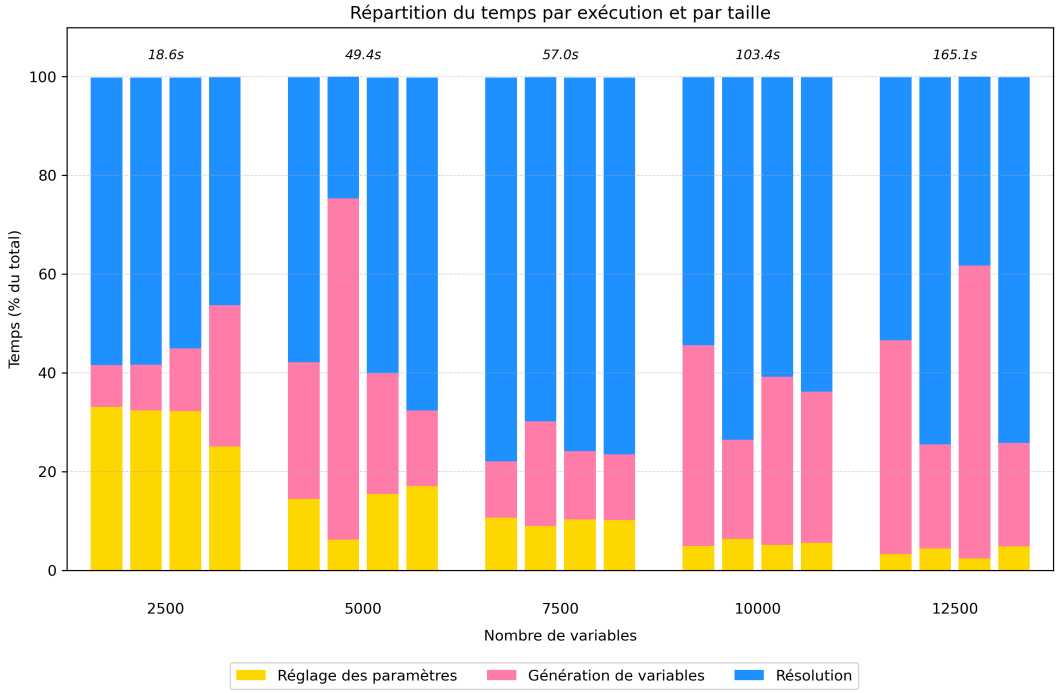

FIGURE 4.1 Décomposition du temps pour la résolution de l'instance 4.1 avec construction standard

4.2.2 Évaluation de l'énumération des variables complémentaires

Dans cette section, l'analyse porte sur les expérimentations de résolution de l'instance 4.1 par le modèle `TCP`, avec l'ajout de variables complémentaires à celles générées par le processus de construction présenté à la section 3.4. Par la suite, nous désignons par standard l'approche sans ajout de variables complémentaires, et par complémentaire celle intégrant l'addition de ces variables.

La figure 4.2 montre les réductions moyennes (en pourcentage) du nombre de mouvements en fonction du nombre de variables par rapport à la solution fournie par la stratégie FIFO. Sur l'ensemble des expérimentations lancé, la méthode standard a eu une réduction des coûts de mouvements moyenne de **23.17%** et la méthode avec complémentaire a eu une réduction de **24.17%**. Toutes les configurations ont permis une réduction de coûts d'au moins 20 %

par rapport à FIFO, ce qui confirme la robustesse de l'approche. De plus, étant donné la proximité des performances entre des résolutions avec beaucoup et peu de variables, il ne semble pas pertinent d'utiliser un très grand nombre de variables, puisque le gain obtenu apparaît marginal pour un temps de résolution augmentant de façon importante. Finalement, les gains présentés à la figure 4.2 révèlent la supériorité de l'approche complémentaire. On observe ici une légère corrélation positive : la réduction croît de $23,31\%$ à $24,21\%$ de moyenne, avec un saut à $25,95\%$ pour 10 000 variables. Il convient toutefois de souligner que l'ampleur de cette corrélation demeure limitée.

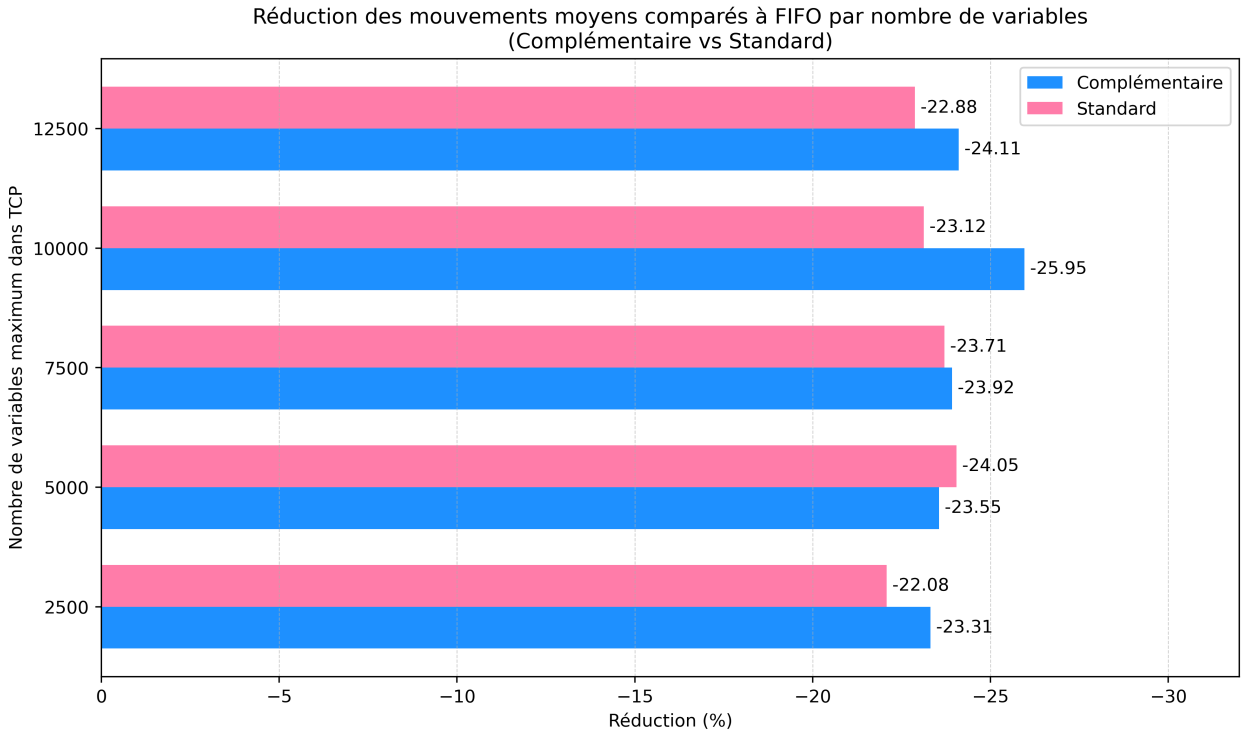

FIGURE 4.2 Résultats d'optimisation comparé à FIFO pour l'instance type en fonction du nombre de variables et de la complémentarité

La décomposition des temps de calcul pour les résolutions avec variables complémentaires est présentée en figure 4.3. Comme pour les résolutions standard, il est attendu que le temps total de résolution augmente avec le nombre de variables. De manière cohérente, on constate que plus le nombre de variables croît, plus la résolution de `TCP` occupe une part importante. Le temps de réglage des hyperparamètres demeure fixe et similaire, puisque l'ajout des variables complémentaires intervient uniquement durant la phase de génération (partie en rose). On remarque également que la variance de cette phase est globalement moins élevée et plus stable lorsque le nombre de variables augmente. Cependant, des épisodes de génération plus

longs peuvent toujours survenir, notamment lorsque le nombre de variables est élevé. C'est le cas, par exemple, pour l'expérimentation à 10 000 variables sur la figure 4.3. Dans ce contexte, le processus de construction peut éprouver des difficultés à proposer de nouveaux sous-ensembles distincts de ceux déjà générés, ce qui nécessite davantage de temps pour que l'aléatoire intégré au procédé produise de nouveaux sous ensembles.

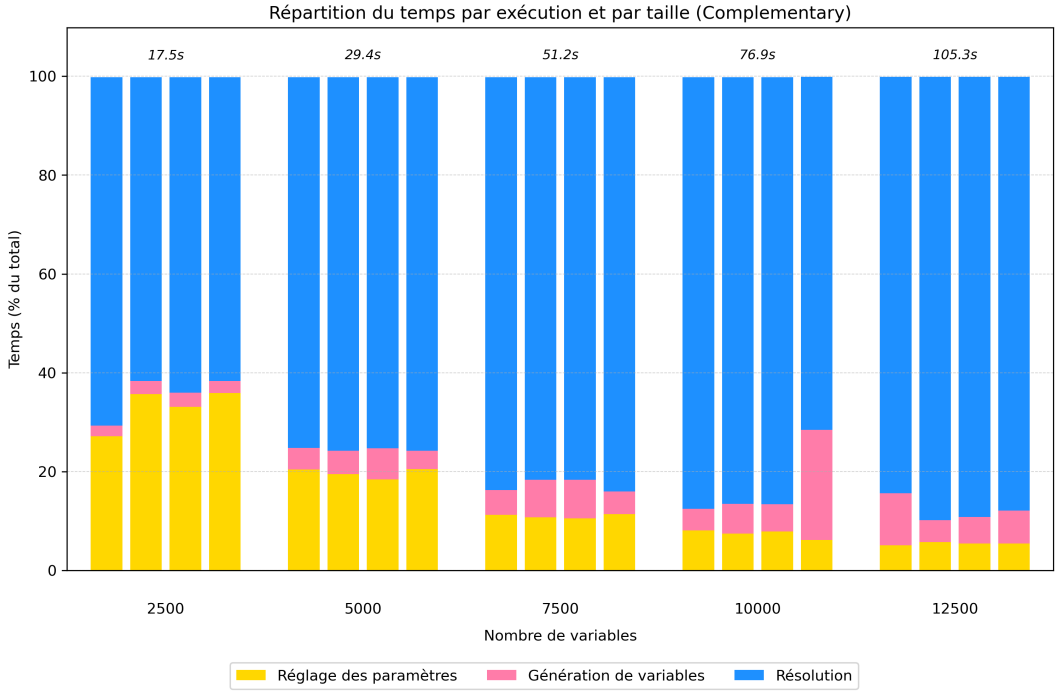

FIGURE 4.3 Décomposition du temps pour la résolution de l'instance type avec construction complémentaire

La figure 4.4 présente le temps total et le temps de résolution du modèle `TCP` en fonction du nombre de variables pour les deux approches. On observe que l'option complémentaire conduit systématiquement à des temps totales plus faibles et surtout plus prévisibles que le standard. La courbe du temps total avec complémentaire suit étroitement celle du temps de résolution du modèle MILP, ce qui traduit la stabilité de la phase de génération lorsque cette option est activée. À l'inverse, la génération standard se révèle beaucoup plus sensible en temps à cette étape. On remarque également que les courbes du temps de résolution du modèle `TCP` avec et sans complémentaire se recouvrent largement. Cela confirme que le facteur déterminant reste le contrôle du nombre de variables, qui impacte directement le temps de résolution du modèle. Ce résultat suggère que le temps de résolution est davantage lié à la quantité de variables qu'à leur capacité à se combiner efficacement pour former des partitions pertinentes dans chaque sous-problème.

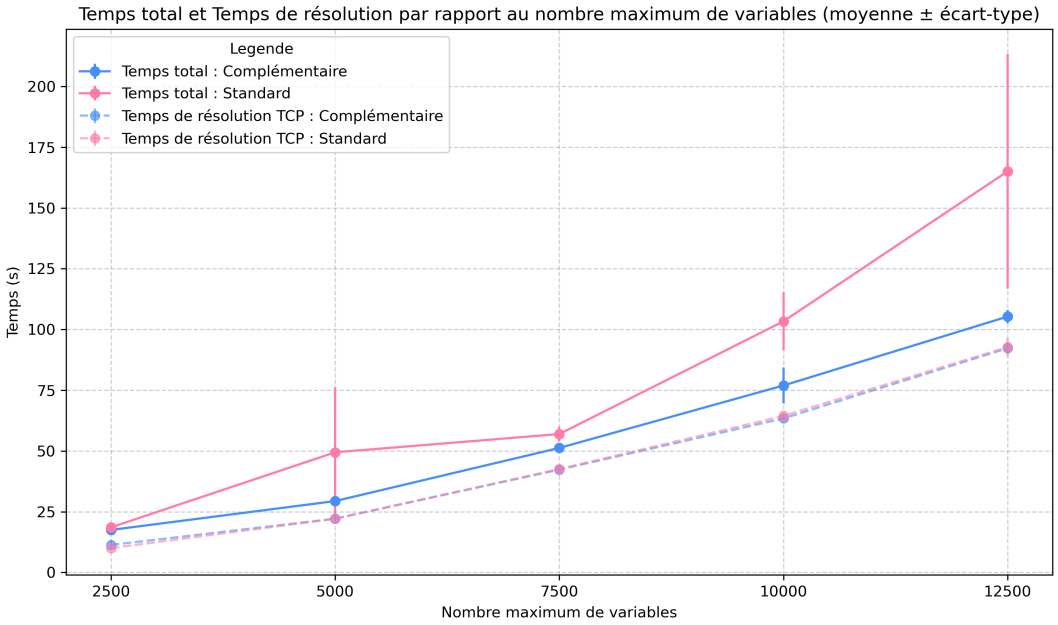

FIGURE 4.4 Temps total et de résolution du modèle `TCP` en fonction du nombre de variables (standard et complémentaire)

La majorité du temps total est consacrée à la phase de résolution du modèle `TCP`. Cela souligne que l'enjeu principal n'est pas tant la quantité de variables que leur qualité, c'est-à-dire leur capacité à conduire à des réductions de coûts significatives. En effet, il n'est pas évident qu'un grand nombre de variables entraînent systématiquement de meilleures réductions de coûts. Cette tendance semble légèrement moins marquée dans le cas de la génération complémentaire, mais, dans l'ensemble, la résolution avec un nombre élevé de variables n'apparaît pas rentable en termes de temps d'exécution total. À l'inverse, résoudre plusieurs fois avec un nombre modéré de variables peut s'avérer plus avantageux. L'aléatoire introduit dans le processus permet d'explorer des solutions plus variées, et l'augmentation du nombre d'essais accroît les chances d'obtenir une réduction significative. Dans le même temps, une stratégie fondée sur un très grand nombre de variables fournirait un volume moindre de solutions explorées dans un temps équivalent.

Il convient toutefois de préciser que ces observations dépendent de la nature des instances considérées. Certaines instances, notamment celles caractérisées par un nombre important d'apparitions de produits, peuvent nécessiter davantage de variables dans le cadre de notre approche, afin de garantir au modèle un espace de choix suffisant pour atteindre des solutions de qualité.

4.2.3 Illustration d'une solution pour l'instance type

La solution retenue a été obtenue avec 5000 variables et l'option complémentaire activée. La limite de stockage ST a été fixée à la valeur maximale observée pour FIFO (ici 104). Après moins de 37 secondes de résolution totale, la solution atteint un coût de **725** (vs. 1049 pour FIFO) mouvements de palettes, soit une réduction de plus de **30%** par rapport à FIFO.

La figure 4.5 présente les profils cumulés des coûts de mouvements (en haut) et de stockage (en bas) par étapes du déchargement. Puisque FIFO constitue une borne inférieure en termes de stockage, toute autre solution ne peut être que supérieure ou égale. Comme illustré sur la partie inférieure de la figure, les barres jaunes correspondent à la quantité de palettes dépassant le niveau FIFO à une étape donnée. Les niveaux de stockage sont largement similaires au début, jusqu'à l'étape 10, car c'est une phase d'accumulation de palettes ne permettant pas beaucoup de combinaisons alternatives nécessitant plus de stockage que le FIFO. Après cette phase d'approvisionnement, entre les étapes 10 et 23, les décisions de combinaisons alternatives sont permises tant qu'elles ne surpassent pas la limite. Ces décisions restent limitées dans la mesure où à l'étape 23, le maximum du stockage, le nombre de palettes étant entrées dans le système pour FIFO doit être le même pour la solution de l'optimisation. Ce qui veut dire, qu'à l'étape 23, la solution doit aligner son nombre conteneurs formés avec celui du FIFO. Ensuite, l'approche optimisée exploite plus efficacement la capacité de stockage disponible, sans dépasser la limite fixée, afin de consolider les palettes et de réduire leurs mouvements superflus. Ainsi, les solutions optimisées par notre approche parviennent à tirer un meilleur parti de l'espace de stockage disponible.

En tenant compte de l'importance de la nature de l'instance dans la performance des approches, nous poursuivons l'étude sur différents scénarios, à la lumière des résultats mis en évidence jusqu'à présent.

4.3 Analyse comparative sur les scénarios

Afin d'étudier le comportement de la méthode sur des instances de nature différente, nous l'avons appliquée à chacune des instances issues des scénarios présentés à la section 4.1. Chaque instance a été résolue avec 5000 variables, en incluant l'option complémentaire. Cinq graines aléatoires fixes ont été utilisées pour la construction des sous-ensembles lors du réglage des hyperparamètres ainsi que pour la phase de génération des variables. Les résultats, exprimés en termes de réduction des coûts par scénario par rapport à la stratégie FIFO, sont présentés à la figure 4.6.

On peut remarquer que, dans l'ensemble des scénarios, une réduction des coûts de congestion

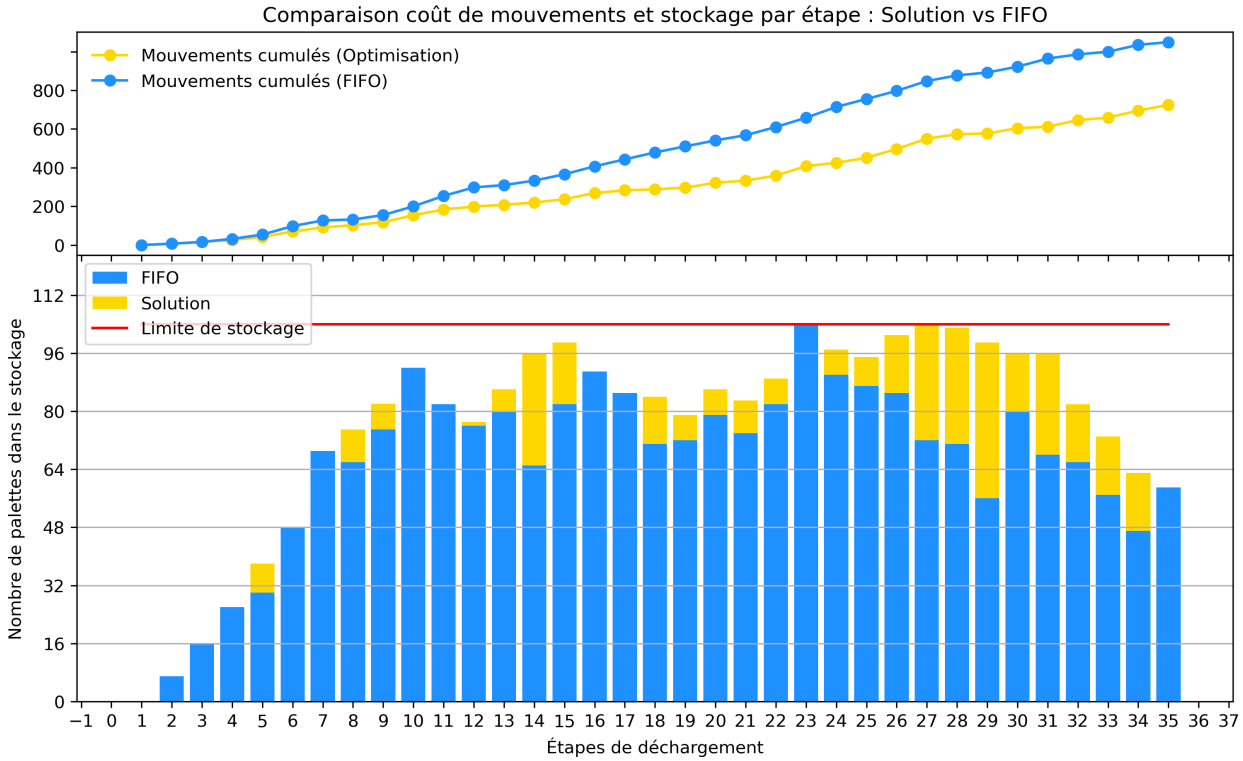

FIGURE 4.5 Représentation du stockage et des coûts de mouvement par étapes pour l'instance 4.1

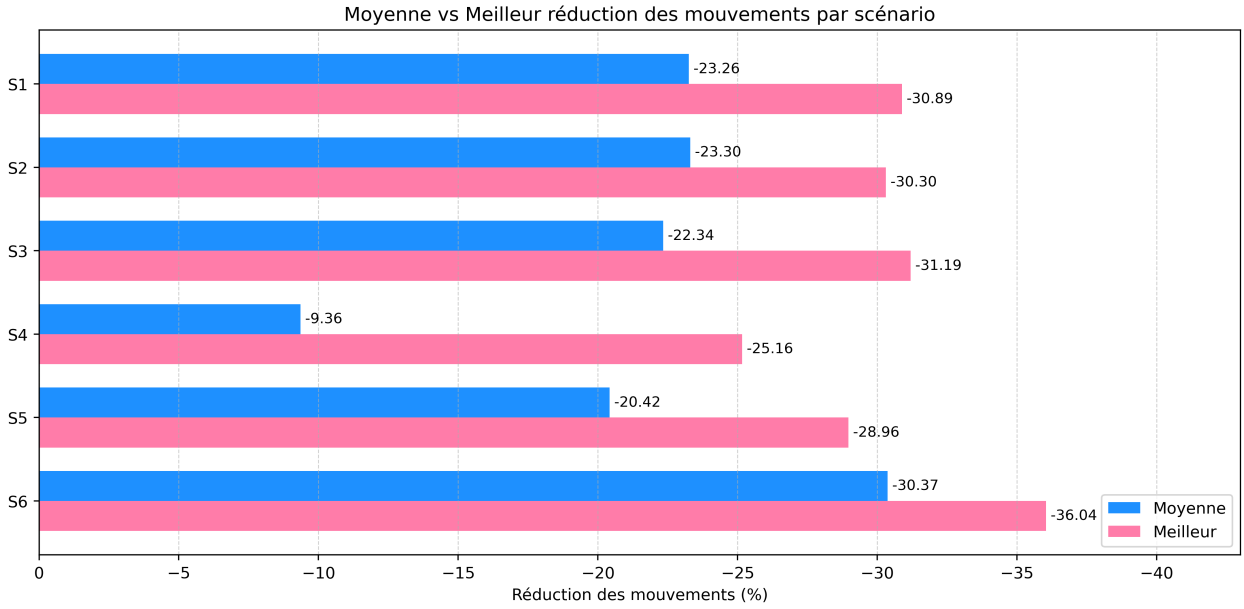

FIGURE 4.6 Réductions de coût obtenues par notre méthode pour chacun des scénarios testés

est observée.

On peut identifier trois catégories de résultats dans cette figure. Les scénarios 1, 2, 3 et 5 présentent des dynamiques similaires, avec des réductions moyennes par rapport à FIFO comprises entre 20% et 24% et des valeurs maximales proches de 30%.

Les scénarios 1 et 2 sont strictement de même nature. La seule différence réside dans le fait que le scénario 2 a bénéficié d'une énumération exhaustive de tous les sous-ensembles des apparitions de ses produits. Les résultats obtenus sont très proches. Pour le scénario 1, certaines instances comportent effectivement un grand nombre de variables théoriques. Il est donc encourageant de constater que l'approche peut atteindre des niveaux de gain comparables, avec ou sans énumération exhaustive.

Les scénarios 3 et 5 présentent la même structure d'apparition des produits que le scénario 1, c'est-à-dire uniforme. Ce qui les différencie, en réalité, réside dans la distribution du nombre de palettes résiduelles. Le scénario 5, en revanche, présente des résultats légèrement inférieurs à ceux des scénarios 1, 2 et 3, ce qui pourrait indiquer une sensibilité à la manière de combiner les palettes. En effet, le scénario 5 a une forte probabilité de former un conteneur avec une paire de lots, car la somme des espérances de chacun des modes est égale à NPC. Cela entraîne des combinaisons par paires et des palettes résiduelles faibles après combinaison, réduisant ainsi les marges pour des combinaisons impliquant un nombre arbitraire de lots.

Les résultats pour le scénario 4 présentent un grand écart entre la moyenne et le meilleur résultat obtenu, ce qui peut indiquer une forte dépendance à l'instance générée plutôt qu'aux caractéristiques du scénario de support.

On peut également voir que la méthode est plus sensible à certains scénarios comme le scénario 4. Cela peut être dû à la quantité importante de produit apparaissant de manière consécutive, rendant plus difficile une énumération des sous-ensembles de qualité, impactant ainsi le résultat. La méthode considère chaque lot comme une unité combinatoire, alors que dans ce cas, il n'y a pas trop d'intérêt (comme vue dans l'exemple 3.5) de séparer les combinaisons d'un bloc.

Le scénario 6 correspond à des produits apparaissant dans un ordre fixe et séquentiel. Parmi les instances choisies de ce scénario, deux présentent un nombre d'apparitions de produits suffisamment faible pour que la méthode puisse énumérer tous les sous-ensembles. Ces deux instances ont respectivement généré des réductions de 26,3% et 28,1%. En revanche, les trois autres instances, plus complexes, ont permis d'obtenir des gains plus importants. Lorsque l'instance présente une combinatoire limitée, les réductions sont restreintes. En revanche, lorsque de nombreuses combinaisons sont possibles, le gain potentiel augmente. Ce scénario

peut également se rapporter de manière plus générale à un cas où les produits apparaissent selon un schéma fixe, sans être nécessairement séquentiel.

Il est important de noter qu'à l'issue du processus de résolution, le coût réel de la solution fournie par le solveur est calculé et peut dépasser celui de FIFO, car les coûts d'opération des Y ne sont pas pris en compte. En moyenne, les coûts hors modèle représentent $40{,}1\% \pm 4{,}8\%$, soit une part significative du coût total. De la même manière, au sein du MILP, plusieurs solutions minimales peuvent conduire à des coûts totaux différents après post-traitement.

Comme le montrent les écarts entre la moyenne et la meilleure réduction des coûts, le potentiel de gain peut dépendre fortement de l'instance. Dans les scénarios définis, deux degrés de liberté existent : la distribution des produits et la distribution des palettes résiduels pour ces produits. Pour obtenir des réductions de mouvements importantes, il est nécessaire de disposer d'un nombre suffisant de combinaisons possibles afin d'aboutir à des solutions à coût plus faible. De plus, le potentiel de réduction est significatif si la solution FIFO implique un nombre important de mouvements d'aller-retour au stockage pouvant être améliorés. Une solution avec un schéma d'apparition fixe et des palettes résiduelles fixes pour chaque produit favorise une règle de combinaison fixe, comme FIFO. Néanmoins, la combinatoire induite par le nombre d'apparitions peut également élargir de manière substantielle les possibilités, renforçant le caractère aléatoire de l'approche et permettant ainsi d'engendrer des réductions plus importantes.

CHAPITRE 5 CONCLUSION

Les travaux présentés dans ce mémoire ont permis d'atteindre les objectifs fixés en étroite collaboration avec l'industriel partenaire. Après une analyse approfondie de la stratégie actuelle FIFO et de ses limites, nous avons élaboré une modélisation intégrant de manière réaliste les contraintes physiques de stockage, les mouvements de palettes et les contraintes temporelles des opérations. Sur cette base, une méthode de résolution combinant formulation exacte et génération heuristique des variables a été développée, de manière à rester applicable à des instances représentatives du contexte industriel.

Les résultats obtenus sont particulièrement encourageants. Sur un large éventail d'instances, incluant des configurations variées de tailles et de compositions, la méthode proposée permet de réduire jusqu'à 30 % le nombre de mouvements par rapport à la stratégie FIFO de référence. Ces gains, obtenus de manière robuste sur l'ensemble des cas étudiés, confirment la pertinence de l'approche et son potentiel d'amélioration significatif dans un contexte opérationnel réel.

Un aspect essentiel de cette contribution réside également dans les temps de résolution obtenus. Les solutions sont calculées dans des délais cohérents avec les attentes de l'industrie, garantissant une intégration possible dans un environnement de prise de décision rapide. Cette performance résulte de l'équilibre recherché entre précision de la modélisation et maîtrise de la complexité computationnelle, assurant que les méthodes développées restent utilisables en pratique, même sur des instances de grande taille.

Enfin, ce travail met en évidence l'intérêt des collaborations entre le milieu académique et le secteur privé. Ce type de partenariat favorise la mise en commun des expertises : la rigueur scientifique et la capacité d'expérimentation de la recherche viennent enrichir la connaissance fine des enjeux et contraintes terrain apportée par l'industriel. Cette synergie permet non seulement de concevoir des solutions innovantes et applicables, mais aussi de renforcer la compétitivité et la performance opérationnelle des entreprises tout en contribuant aux avancées méthodologiques dans le domaine de l'optimisation.

5.1 Limitations de la solution proposée

Plusieurs limites peuvent être soulignées. Le modèle proposé ne prend pas en compte l'ensemble des composantes de coût associées aux opérations, ce qui peut influencer la pertinence économique des solutions obtenues. Dans certains cas spécifiques, certaines instances peuvent

même aboutir à des coûts supérieurs à ceux de la stratégie FIFO de référence. Par ailleurs, bien que la méthode de génération de variables permette d'atteindre des solutions de qualité, le temps nécessaire à cette étape peut s'avérer important, en particulier pour des instances de grande taille, ce qui pourrait limiter son utilisation dans un contexte opérationnel où l'optimisation doit être relancée sur le tas.

L'approche se révèle particulièrement efficace pour des instances présentant une variabilité distribuée, qui ouvre l'accès à davantage de combinaisons. En revanche, elle est moins adaptée aux instances caractérisées par des structures rigides ou des schémas, comme celle en bloc. Elle montre également une forte sensibilité au nombre d'apparitions des produits.

L'optimisation proposée ne prend pas en compte les dynamiques intra-étape, seuls les mouvements vers et depuis la zone de stockage sont considérés. Les transferts entre lignes de chargement, bien que potentiellement optimisables, dépendent fortement de la configuration spécifique du centre, ce que nous avons volontairement écarté afin de préserver l'aspect haut niveau de la modélisation.

Enfin, la méthodologie ne conserve aucune mémoire entre deux résolutions. Lorsque plusieurs exécutions sont lancées, seule la meilleure solution parmi elles est retenue, sans exploitation d'informations inter-résolutions.

5.2 Améliorations futures

Plusieurs pistes de recherche peuvent être envisagées pour renforcer l'approche proposée. Tout d'abord, il serait pertinent d'intégrer les coûts actuellement non modélisés, soit par l'ajout de pénalités adaptées, soit par une modélisation plus fine de ces composantes. Par ailleurs, l'optimisation pourrait être étendue aux dynamiques intra-étape, afin de prendre en compte les possibilités d'amélioration au sein d'une même étape.

Une autre direction consisterait à affiner le réglage des hyperparamètres, par exemple en les adaptant spécifiquement à chaque produit, ou encore en développant des mécanismes de mémoire entre résolutions successives. De plus, la reconnaissance de motifs récurrents dans les instances permettrait de mieux ajuster l'approche aux structures spécifiques rencontrées.

Enfin, des processus récursifs conservant les meilleures variables identifiées pour chaque produit pourraient être explorés afin de renforcer la qualité des solutions. L'élaboration d'un modèle de simulation représentant plus fidèlement la dimension temporelle des opérations constituerait également un apport majeur, permettant d'évaluer l'impact réel des solutions en conditions opérationnelles.

RÉFÉRENCES

[1] Y. E. Vasiliev *et al.*, "Railroad cement warehouse digital system," *IOP Conference Series : Materials Science and Engineering*, vol. 1159, n°. 1, p. 012035, juin 2021, publisher : IOP Publishing. [En ligne]. Disponible : https://dx.doi.org/10.1088/1757-899X/1159/1/012035

[2] T. H. Cormen *et al.*, *Introduction to algorithms*, third edition éd. MIT Press.

[3] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, n°. 2, p. 146–160, 1972. [En ligne]. Disponible : https://doi.org/10.1137/0201010

[4] F. Rothlauf, *Design of Modern Heuristics : Principles and Application*, ser. Natural Computing Series. Springer Berlin Heidelberg. [En ligne]. Disponible : https://link.springer.com/10.1007/978-3-540-72962-4

[5] M. G. Resende et C. C. Ribeiro, *Optimization by GRASP : Greedy Randomized Adaptive Search Procedures*. Springer New York. [En ligne]. Disponible : https://link.springer.com/10.1007/978-1-4939-6530-4

[6] T. G. Crainic et K. H. Kim, "Chapter 8 Intermodal Transportation," dans *Handbooks in Operations Research and Management Science*, ser. Transportation, C. Barnhart et G. Laporte, édit. Elsevier, janv. 2007, vol. 14, p. 467–537. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0927050706140086

[7] B. Jennings, "An investigation of transload : The use of non-containerized multimodal bulk shipments within the u.s. freight carrier industry." [En ligne]. Disponible : https://trace.tennessee.edu/utk_graddiss/10380

[8] D. M. Thomson, "Transloads : Freight movement efficiencies in the next decade."

[9] I. Rasul, "EVALUATION OF POTENTIAL TRANSLOAD FACILITY LOCATIONS IN THE UPPER PENINSULA (UP) OF MICHIGAN." [En ligne]. Disponible : https://digitalcommons.mtu.edu/etds/805

[10] A. Braham *et al.*, "Locating transload facilities to ease highway congestion and safeguard the environment," number : TRC1608. [En ligne]. Disponible : https://trid.trb.org/View/1687825

[11] M. I. Asborno et S. Hernandez, "Using data from a state travel demand model to develop a multi-criteria framework for transload facility location planning," vol. 2672, n°. 9, p. 12–23. [En ligne]. Disponible : https://journals.sagepub.com/doi/10.1177/0361198118772699

[12] J. Warner *et al.*, "Use of rail in oil and gas regions of texas to reduce truck impacts," dans *2019 Joint Rail Conference*. American Society of Mechanical Engineers, p. V001T05A002. [En ligne]. Disponible : https://asmedigitalcollection.asme.org/JRC/proceedings/JRC2019/58523/Snowbird,%20Utah,%20USA/954485

[13] M. C. Burke, "An analysis of economic impacts and spatial influence of transload facilities," ISBN : 9798494443069 Publication Title : ProQuest Dissertations and Theses 28747827. [En ligne]. Disponible : https://www.proquest.com/dissertations-theses/analysis-economic-impacts-spatial-influence/docview/2595126116/se-2?accountid=40695

[14] S. Smith *et al.*, "Development of a cost estimation framework for potential transload facilities," vol. 2672, n°. 9, p. 24–34, publisher : SAGE Publications Inc. [En ligne]. Disponible : https://doi.org/10.1177/0361198118774690

[15] M. Patel, "Conceptual design of dry bulk terminals." [En ligne]. Disponible : http://hdl.handle.net/10019.1/109794

[16] R. Cigolini et T. Rossi, "Sizing off-shore transshipment systems in dry-bulk transportation," vol. 21, n°. 5, p. 508–522. [En ligne]. Disponible : https://www.tandfonline.com/doi/full/10.1080/09537280903514587

[17] C. T. Dick et L. E. Brown, "Design of bulk railway terminals for the shale oil and gas industry," dans *Shale Energy Engineering 2014*. American Society of Civil Engineers, p. 704–714. [En ligne]. Disponible : http://ascelibrary.org/doi/10.1061/9780784413654.071

[18] T. Newdome et C. Gibson, "Modeling of blending/transloading facilities for use in optimal fuel scheduling," vol. PAS-104, n°. 11, p. 3050–3055. [En ligne]. Disponible : http://ieeexplore.ieee.org/document/4112983/

[19] L. Cheng *et al.*, "Direct transshipment optimization in hub bulk transshipment terminals." [En ligne]. Disponible : https://papers.ssrn.com/abstract=4960223

[20] I. F. Vis et R. De Koster, "Transshipment of containers at a container terminal : An overview," vol. 147, n°. 1, p. 1–16. [En ligne]. Disponible : https://linkinghub.elsevier.com/retrieve/pii/S037722170200293X

[21] A. Baykasoğlu et K. Subulan, "A multi-objective sustainable load planning model for intermodal transportation networks with a real-life application," vol. 95, p. 207–247. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1366554516302368

[22] P. Corry et E. Kozan, "An assignment model for dynamic load planning of intermodal trains," vol. 33, n°. 1, p. 1–17. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0305054804001170

[23] ——, “Optimised loading patterns for intermodal trains,” vol. 30, n°. 4, p. 721–750. [En ligne]. Disponible : https://doi.org/10.1007/s00291-007-0112-5

[24] T. Benna et M. Gronalt, “Generic simulation for rail-road container terminals,” dans *2008 Winter Simulation Conference*. IEEE, p. 2656–2660. [En ligne]. Disponible : https://ieeexplore.ieee.org/document/4736381/

[25] Y. Xie et D.-P. Song, “Optimal planning for container prestaging, discharging, and loading processes at seaport rail terminals with uncertainty,” vol. 119, p. 88–109. [En ligne]. Disponible : https://linkinghub.elsevier.com/retrieve/pii/S1366554518302527

[26] P. Buijs, I. F. Vis et H. J. Carlo, “Synchronization in cross-docking networks : A research classification and framework,” vol. 239, n°. 3, p. 593–608. [En ligne]. Disponible : https://linkinghub.elsevier.com/retrieve/pii/S0377221714002264

[27] A. A. Ardakani et J. Fei, “A systematic literature review on uncertainties in cross-docking operations,” vol. 2, n°. 1, p. 2–22, publisher : Emerald Publishing Limited. [En ligne]. Disponible : https://www.emerald.com/insight/content/doi/10.1108/mscra-04-2019-0011/full/html

[28] W. Yu et P. J. Egbelu, “Scheduling of inbound and outbound trucks in cross docking systems with temporary storage,” vol. 184, n°. 1, p. 377–396. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0377221706011052

[29] Arabani. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage - ScienceDirect. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S095741741000758X#bib35

[30] G. Luo et J. S. Noble, “An integrated model for crossdock operations including staging,” vol. 50, n°. 9, p. 2451–2464. [En ligne]. Disponible : http://www.tandfonline.com/doi/abs/10.1080/00207543.2011.581007

[31] N. Zaerpour, Y. Yu et R. B. de Koster, “Storing fresh produce for fast retrieval in an automated compact cross-dock system,” vol. 24, n°. 8, p. 1266–1284, _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/poms.12321. [En ligne]. Disponible : https://onlinelibrary.wiley.com/doi/abs/10.1111/poms.12321

[32] S. Sumit, “Staging approaches to reduce overall cost in a crossdock environment.” [En ligne]. Disponible : http://hdl.handle.net/10355/4246

[33] E. S. Suh, “Cross-docking assessment and optimization using multi-agent co-simulation : a case study,” vol. 27, n°. 1, p. 115–133. [En ligne]. Disponible : https://doi.org/10.1007/s10696-014-9201-3

[34] A. Bortfeldt et G. Wäscher, “Constraints in container loading – a state-of-the-art review,” vol. 229, n°. 1, p. 1–20. [En ligne]. Disponible : https://linkinghub.elsevier.com/

retrieve/pii/S037722171200937X

[35] O. X. d. Nascimento, T. Alves de Queiroz et L. Junqueira, “Practical constraints in the container loading problem : Comprehensive formulations and exact algorithm,” vol. 128, p. 105186. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0305054820303038

[36] T. Akiba *et al.*, “Optuna : A next-generation hyperparameter optimization framework,” dans *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, p. 2623–2631.

ANNEXE A DÉTAIL DES SCÉNARIOS

Tableau A.1 Paramètres d'instances pour le scénario 1 (Uniforme)

Paramètre	T_L	L_D	n	NPC	Graine
Valeur	100	3	10	16	42
	150	5	16	16	26
	100	2	13	16	18
	100	2	8	16	19
	200	4	20	16	55

Tableau A.2 Paramètres d'instances pour scénario 2 (Exact)

Apparition par produit	n	L_D	NPC	Graine
10	5	1	16	26
7	10	3	16	34
9	6	2	16	11
8	20	6	16	18
6	50	2	16	55

Tableau A.3 Paramètres d'instances pour le scénario 3 (Gaussien)

Paramètre	T_L	L_D	n	NPC	Graine
Valeur	120	4	12	16	18
	200	6	25	16	24
	120	4	12	16	23
	60	1	7	16	22
	130	2	14	16	33

Tableau A.4 Paramètres d'instances pour le scénario 4 (Bloc)

Paramètre	T_L	L_D	n	NPC	Graine
Valeur	120	6	15	16	35
	200	6	25	16	24
	120	4	12	16	23
	60	1	7	16	22
	110	2	13	16	126

Tableau A.5 Paramètres d'instances pour le scénario 5 (Bimodal)

Paramètre	T_L	L_D	n	NPC	Graine
Valeur	120	4	12	16	36
	200	6	25	16	24
	120	4	12	16	23
	60	1	7	16	22
	120	3	14	16	33

Tableau A.6 Paramètres d'instances pour le scénario 6 (Séquentiel)

Paramètre	T_L	L_D	n	NPC	Graine
Valeur	60	2	15	16	16
	105	3	15	16	19
	200	6	15	16	26
	50	2	3	16	25
	80	1	5	16	44