



HAL
open science

Optimisation de problèmes de production et de transport intégré

Sana Frifita

► **To cite this version:**

Sana Frifita. Optimisation de problèmes de production et de transport intégré. Gestion et management. Université de Technologie de Troyes, 2021. Français. NNT : 2021TROY0027 . tel-03810684

HAL Id: tel-03810684

<https://theses.hal.science/tel-03810684v1>

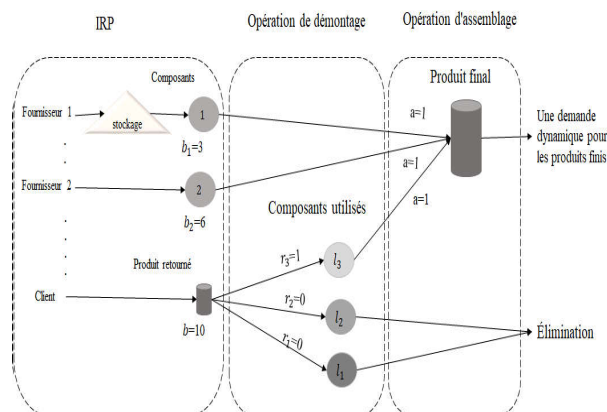
Submitted on 11 Oct 2022

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

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

Sana FRIFITA

Optimisation de problèmes de production et de transport intégré



Champ disciplinaire :
Sciences pour l'Ingénieur

THESE
pour l'obtention du grade de
DOCTEUR
de l'UNIVERSITE DE TECHNOLOGIE DE TROYES
en SCIENCES POUR L'INGENIEUR

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Sana FRIFITA

le 24 septembre 2021

Optimisation de problèmes de production et de transport intégré

JURY

M. Christian PRINS	PROFESSEUR DES UNIVERSITES	Président
M. Ayse AKBALIK	MAITRE DE CONFERENCES - HDR	Rapporteur
M. Oualid JOUINI	PROFESSEUR DES UNIVERSITES	Rapporteur
M. Alexandre DOLGUI	PROFESSEUR IMT ATLANTIQUE - HDR	Examineur
M. Hasan Murat AFSAR	MAITRE DE CONFERENCES	Directeur de thèse
M. Faicel HNAIEN	MAITRE DE CONFERENCES - HDR	Directeur de thèse

Personnalité invitée

Mme Alice YALAOUI	MAITRE DE CONFERENCES - HDR
-------------------	-----------------------------

Remerciements

Tout d'abord, je voudrais remercier le Fonds Européen de Développement Régional (FEDER) pour avoir rendu possible cette thèse au sein de l'Université de Technologie de Troyes (UTT).

Je suis reconnaissant à Messieurs Christian PRINS, Professeur à l'université de technologie de Troyes, et Alexandre DOLGUI, Professeur à l'IMT Atlantique, pour l'honneur qu'ils me font en ayant accepté d'être les examinateurs de cette thèse.

J'exprime toute ma gratitude à Monsieur Oualid JOUNI, Professeur à CentraleSupélec, et à Madame Ayse AKBALIK, Maître de conférences HDR à l'université de Lorraine pour avoir accepté d'être des rapporteurs pour ce manuscrit et pour leur participation au jury.

Je remercie également Madame Alice YALAOUI, Maître de conférences HDR à l'université de technologie de Troyes pour avoir accepté à sa participation au jury de soutenance.

J'adresse tout mon respect à mes directeurs de thèse, Hasan Murat AF-SAR et Faïcel Hnaïen, qui m'ont donné l'opportunité de réaliser cette thèse et m'ont encadré au cours de ces trois années.

Tous mes remerciements à ma famille, mon fiancé et les personnes avec qui j'ai partagé mes études et notamment ces années de thèse.

Résumé

Dans cette thèse, nous traitons essentiellement un problème intégré de tournées de véhicules avec la gestion de production. Ce problème combine les opérations d'assemblage et de désassemblage, avec la collecte des produits en fin de vie et des matières premières.

Premièrement, nous étudions le cas déterministe du problème multi-périodes dans sa version statique et dynamique. Nous proposons un modèle mathématique linéaire en nombres entiers mixtes incluant ces nouvelles contraintes. Une mat-heuristique itérative à deux phases basée sur la programmation mathématique et un algorithme de recherche à voisinage variable, est également développée pour résoudre les instances de plus grande taille.

Deuxièmement, nous étudions le cas stochastique où les quantités des composants désassemblés sont incertains en raison du statut inconnu des produits retournés. Nous proposons une approche basée sur une modélisation robuste définie par des scénarios discrets. Nous abordons différentes hypothèses, où différents niveaux de flexibilité sont donnés aux différentes décisions. Une mat-heuristique à trois étapes pour optimiser le pire cas est développée pour la résolution itérative de différents problèmes dans tous les scénarios.

Différents tests sont effectués afin de montrer l'efficacité et les limites de chaque approche dans chaque cas.

Mots clés : Planification, Production, Gestion des stocks, Transport, Logistique inverse, Programmation linéaire mixte, Heuristique.

Abstract

In this thesis, we study an integrated vehicle routing problem with production management. This problem combines assembly and disassembly operations with the collection of end-of-life products and raw materials.

First, we study the deterministic case of the multi-period problem in its static and dynamic version. We propose a mixed integer linear programming model including these new constraints. A two-phase mat-heuristic based on mathematical programming and the variable neighborhood search algorithm is also developed to solve the larger instances.

Second, we study the stochastic case where the yield of disassembled components are uncertain due to the unknown status of the returned products. We propose an approach based on robust modeling defined by discrete scenarios. We address different assumptions, where different levels of flexibility are given to different decisions. A three-stage mat-heuristic to optimize the worst case is developed to solve the different problems under all scenarios.

Different tests are performed to show the effectiveness and limitations of each approach in each case.

Keywords : Planning, Production, Transportation, Inventory control, Reverse logistics, Mixed linear programming, Heuristic.

Sommaire

Remerciements	i
Résumé	iii
Abstract	v
1 Introduction générale	1
1.1 Contexte	1
1.2 Problématique	6
1.3 Contributions et organisation de la thèse	8
2 État de l’art	11
2.1 Introduction	11
2.2 La logistique inverse	11
2.2.1 Définition de la logistique inverse	11
2.2.2 Problèmes liés à la logistique inverse	14
2.2.3 Désassemblage	16
2.2.4 Problèmes liés au désassemblage	17
2.2.5 Planification de désassemblage	18
2.3 Gestion de la production	21
2.3.1 Planification des besoins en composants	21
2.3.2 Dimensionnement de lots	24
2.4 Problèmes de production et de transport intégrés	30
2.4.1 Problèmes de production avec expédition directe	32
2.4.2 Problèmes de tournées de véhicules avec gestion des stocks	33
2.4.3 Problèmes de tournées de véhicules avec gestion de production	34
2.4.4 Problèmes de tournées de véhicules et d’assemblage	36
2.4.5 Problèmes de tournées de véhicules stochastiques	38
2.5 Conclusions	39
3 Problème d’assemblage, désassemblage et des tournées de véhicules : le cas déterministe	41
3.1 Introduction	41
3.2 Modélisation mathématique	43
3.2.1 Description du problème	43
3.2.2 Modèle mathématique	45

3.3	Approche Mat-heuristique	49
3.3.1	PLNE pour les problèmes de dimensionnement de lots	50
3.3.2	Algorithme de recherche à voisinage variable	51
3.3.3	Mise à jour des coûts approximatifs de transport	52
3.3.4	Mécanismes de diversification	53
3.4	Résultats expérimentaux	54
3.4.1	Paramétrages	54
3.4.2	Génération de données	55
3.4.3	Résultats	56
3.5	Conclusion	61
4	Problème d'assemblage, désassemblage et des tournées de véhicules : le cas stochastique	69
4.1	Introduction	69
4.2	Modélisation mathématique du problème	70
4.2.1	Définition du problème	71
4.2.2	Modèle mathématique robuste	71
4.2.3	Niveaux de flexibilité	75
4.3	Approche Mat-heuristique	76
4.4	Résultats expérimentaux	78
4.4.1	Impacts des niveaux de flexibilité	78
4.4.2	Interprétation des résultats	80
4.5	Conclusion	86
5	Conclusions et perspectives de recherches	93
5.1	Contributions	93
5.2	Perspectives de recherches	96
	Bibliographie	101

Table des figures

1.1	Fonctions principales de la chaîne logistique verte ¹	2
2.1	Problèmes liés à la logistique inverse	16
2.2	Un exemple d'une opération de désassemblage	17
2.3	Problèmes liés au désassemblage	18
2.4	Modèles de planification de la chaîne logistique	30
3.1	Exemple d'une instance 2D-ARP-R	43
3.2	Un exemple des chemins de flux pour un trajet de trois clients	45
3.3	Les étapes générales de notre mat-heuristique	49
4.1	Exemple d'une solution pour le 2D-ARP-R	70
4.2	Les étapes générales de la mat-heuristique	77

Liste des tableaux

2.1	Synthèse des articles sur le problème de planification de désassemblage	22
2.2	Synthèse des articles sur le problème de dimensionnement de lots	31
2.3	Articles sur les problèmes d'assemblage et de désassemblage	37
3.1	Notation	46
3.2	Le test de Friedman pour fixer la valeur $iter1_{max}$	54
3.3	Le test de Friedman pour fixer la valeur $iter2_{max}$	55
3.4	L'avantage du modèle intégré pour la version statique	57
3.5	L'avantage du modèle intégré pour la version dynamique	57
3.6	La meilleure solution du modèle décentralisé	57
3.7	La meilleure solution du modèle intégré	58
3.8	Solutions du modèle intégré avec et sans désassemblage	58
3.9	Résultats moyens pour 100 instances avec 14 nœuds	60
3.10	Résultats moyens pour 100 instances avec 50 nœuds	60
3.11	Résultats moyens pour 40 instances avec 100 nœuds	61
3.12	Résultats de 10 exécutions pour la mat-heuristique pour les instances à 14 nœuds	62
3.13	Résultats de 10 exécutions pour la mat-heuristique pour les instances à 50 nœuds	63
3.14	Résultats pour la version statique avec 14 nœuds	64
3.15	Résultats pour la version dynamique avec 14 nœuds	65
3.16	Résultats pour la version statique avec 50 nœuds	66
3.17	Résultats pour la version dynamique avec 50 nœuds	67
3.18	Résultats pour la version statique avec 100 nœuds	68
3.19	Résultats pour la version dynamique avec 100 nœuds	68
4.1	Résultats pour une instance avec 4 scénarios	78
4.2	Résultats pour les instances avec 14 nœuds sous 5 scénarios	79
4.3	Déviations pour les instances avec 14 nœuds sous 5 scénarios	80
4.4	Performances des méthodes pour des instances avec 14 et 50 nœuds avec 5 scénarios	82
4.5	Performances des méthodes pour des instances avec 14 et 50 nœuds avec 50 scénarios	84
4.6	Performances des méthodes pour des instances avec 14 nœuds avec 100 scénarios	85

4.7	Déviations pour les instances avec 14 et 50 nœuds	86
4.8	Résultats pour les instances avec 5 scénarios pour H_2	87
4.9	Résultats pour les instances avec 5 scénarios pour H_3	87
4.10	Résultats pour les instances avec 5 scénarios pour H_4	88
4.11	Résultats pour les instances avec 50 scénarios pour H_2	88
4.12	Résultats pour les instances avec 50 scénarios pour H_3	89
4.13	Résultats pour les instances avec 50 scénarios pour H_4	89
4.14	Résultats pour les instances avec 100 scénarios pour H_2	90
4.15	Résultats pour les instances avec 100 scénarios pour H_3	90
4.16	Résultats pour les instances avec 100 scénarios pour H_4	91
4.17	Différentes déviations pour tous les scénarios	91

Liste des Algorithmes

1	Algorithme de recherche à voisinage variable	51
2	Mise à jour de σ_{ut}	53

Introduction générale

1.1 Contexte

Les pays du monde entier essaient d'améliorer le niveau de vie de leurs citoyens en répondant à leurs besoins et assurant leur bien-être. Pour réussir cet objectif, les sociétés ont besoin de trouver des moyens efficaces pour réaliser des développements dans les différents domaines et secteurs. Dans ce sens, le développement durable est envisagé comme l'une des exigences les plus importantes des gouvernements depuis les années 90 du XXe siècle, il était considéré comme une condition préalable à une répartition équitable de développement et de richesse.

Le développement durable est une façon de vivre qui concerne tous les secteurs d'activité et qui permet de réaliser des objectifs de performance économique, des ambitions de protection de l'environnement et un développement social. [Munasinghe \(1993\)](#) a identifié trois enjeux du développement durable : Économique (maximiser les revenus avec un stock de capital constant ou croissant), écologique (maintenir la résilience des systèmes biologiques et physiques) et socioculturel (maintien de la stabilité des systèmes sociaux et culturels).

La protection de l'environnement est une problématique de développement durable par laquelle la gestion de la chaîne logistique est concernée sous les pressions réglementaires. Le développement durable, dans ses dimensions environnementale, économique et sociale, impacte les modes de gestion des flux physiques de la chaîne logistique et le management des relations avec les différents acteurs concernés (fournisseurs, prestataires, clients). En relation avec le développement durable, les chercheurs se sont intéressés aux secteurs de transports dans la logistique. En Europe, les objectifs de la politique des transports par rapport au développement durable passent par le report du mode routier vers des modes de transport durables, moins polluants, plus efficaces, à travers le développement de l'inter-modalité. [Salini \(1997\)](#) est parmi les premiers chercheurs qui a intégré les notions de développement durable dans le transport et logistique. Il a analysé le transport routier aux trois axes principaux du développement durable (économique, sociale et environnementale).

Les contraintes réglementaires et les soucis d'images des entreprises les motivent à accorder un grand intérêt à l'environnement. Les organisations logistiques se modifient par l'intégration des contraintes environnementales, amenant le développement de ce que on appelle la chaîne logistique verte (durable).

La chaîne logistique verte est une chaîne logistique visant à minimiser l'empreinte écologique d'un service ou d'un produit, pendant son cycle de vie. Une chaîne logistique verte doit également prendre en compte à la fois les flux amont et aval. La logistique amont verte consiste à améliorer les performances environnementales de production de fournisseurs. L'objectif est la réduction des gaspillages et de la pollution par la réutilisation de matériaux et du recyclage. Pour rendre plus verte la logistique aval, il faut intégrer une gestion des retours et des retraits des différents types des produits que ce soit des produits en fin de vie, ou détériorés pour les valoriser.

Selon [Hervani et al. \(2005\)](#), il y a des nombreux champs qui sont sollicités pour définir la gestion durable de la chaîne logistique : de l'achat vert aux chaînes intégrées comprenant le fournisseur, le producteur, le consommateur et la logistique inverse. Les principales fonctions de la chaîne logistique verte (Figure 1.1) concernent la dimension environnementale du développement durable, en l'occurrence de l'éco-conception, l'éco-sourcing, l'éco-manufacturing, l'éco-logistique et la logistique inverse.

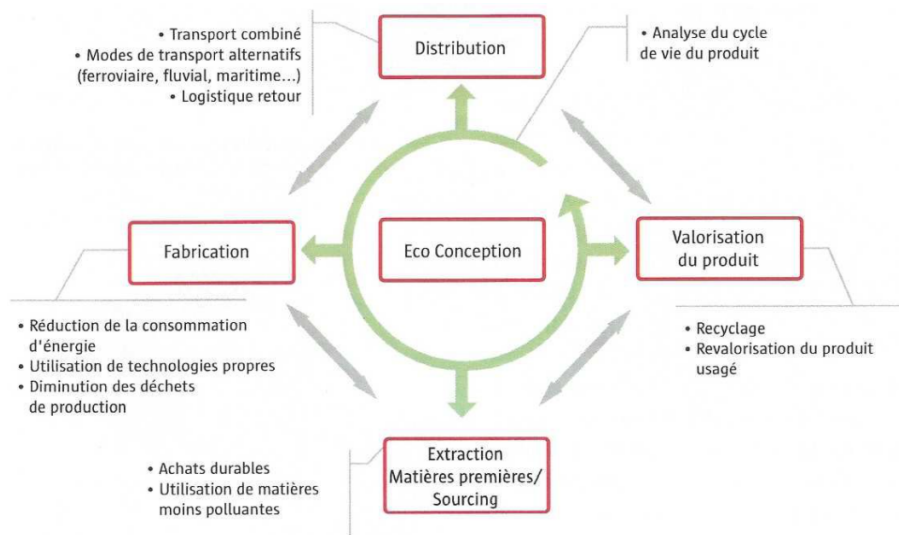


Figure 1.1 – Fonctions principales de la chaîne logistique verte ¹

La logistique inverse joue un rôle important dans la contribution à la gestion des déchets et des ressources. C'est le système qui peut prendre en charge

1. Source : Observatoire de la chaîne logistique (2008)

les déchets dans un cadre responsable, efficace et durable pour minimiser les coûts et respecter les exigences environnementales. Auparavant, les fabricants ou les détaillants estimaient qu'ils n'étaient pas responsables de la récupération de leurs produits après-vente. Lorsqu'un produit est endommagé, son propriétaire pourrait l'éliminer sans être responsable des conséquences. Ce qui augmente la consommation de matières vierges et les pertes de nutriments biologiques qui pourrait être utilisée pour générer de l'énergie.

La logistique inverse est l'un des principaux thèmes concernant la logistique durable ou verte. Elle est définie (Rogers & Tibben-Lembke (1998)) comme le processus de planification, de mise en œuvre et de contrôle d'une logistique efficace. Elle relie les informations du point de consommation au point d'origine. Elle comprend également le traitement des retours des marchandises endommagées, les stocks saisonniers, le réapprovisionnement, le sauvetage, et les stocks excédentaires. Les produits récupérés peuvent être des objets civils, biens de consommation, biens industriels, minéraux et composants chimiques, des articles de distribution et des pièces de rechange.

La logistique inverse est un concept important pour les entreprises qui organisent les systèmes de recyclage et les entreprises qui doivent payer pour l'élimination des déchets (Carter & Ellram (1998)). Elle est parfois considérée comme la colonne vertébrale du "développement durable" dans l'empreinte de ses aspects économiques et environnementaux sur les pratiques commerciales. Elle est considérée comme un instrument de développement durable et une force dynamique pour soutenir l'économie et l'environnement. La logistique inverse est un élément essentiel d'une chaîne logistique en boucle fermée dont les flux d'entrée sont généralement les matériaux des fournisseurs aux clients finaux ainsi que les flux inverses de produits (post-consommation) vers la fabrication ou distribution.

Les entreprises de vente au détail offrent principalement des services de logistique inverse pour gérer les retours de produits, les réparations et les garanties. Ces services sont souvent associés aux garanties de produits offertes par les fabricants d'équipements d'origine et vent inclure la revente et la redistribution des produits invendus. Une quantité importante de matériaux recyclés peut être collecter par ces entreprises pendant le processus de distribution. Autre type d'entreprises qui n'ont pas de magasins, pourraient donc externaliser des centres de retour à des prestataires de services logistiques tiers, ou fournir des canaux tels que les services postaux pour que les consommateurs puissent y retourner les produits usagés. Les commerçants sont des entreprises qui achètent et vendent des produits récupérés ou des matériaux recyclés collectés par les agences gouvernementales, les entreprises de vente au détail, les entreprises de gestion des déchets et les pilleurs. Les clients sont parmi les plus importants acteurs dans la boucle de la logistique

inverse. Ils forment les marchés de "réutilisation" : les consommateurs, les sociétés commerciales et les organisations à but non lucratif. Les organisations caritatives sont également des clients importants de produits récupérés. Il existe également des entreprises qui distribuent et vendent des pièces de rechange usagées ou des produits résinés. Sans une demande de ces acteurs, les systèmes de logistique inverse ne seront pas économiquement durables. La responsabilité d'un producteur peut améliorer de manière significative un système de logistique inverse. Des sociétés, telles que IBM, BMW, Mercedes-Benz, Dell, et bien d'autres, ont constamment amélioré leur logistique inverse pour atteindre la durabilité environnementale. Par exemple, BMW a mené des recherches sur le recyclage dans le cadre d'un projet pilote de démontage de véhicules à Landshut, en Allemagne, depuis 1990. BMW vise à réutiliser 80 % de tous les plastiques, et a réussi à d'incorporer des plastiques recyclés dans certains composants tels que le revêtement des compartiments et les pare-chocs recyclés. Elle vise aussi à la réutilisation de pièces de grande valeur provenant de voitures existantes dans le but de les remettre à neuf tels que les moteurs, les démarreurs et les alternateurs. Les pièces résinées sont vendues comme "pièces d'échange" pour 50 à 70 % des prix des nouveaux produits.

La valorisation désigne la transformation de déchets en énergie ou énergie issue de déchets. C'est une option lorsque les matériaux récupérés ne peuvent pas être réutilisés ou recyclés. Le processus de démontage ou désassemblage permet la valorisation des produits en fin de vie. [Gupta & Taleb \(1994\)](#) ont défini ce processus comme une méthode systématique pour la séparation du produit en modules, composants, sous-ensembles ou autres groupements. En général, il existe six grandes options de valorisation des produits ([Thierry et al. \(1995\)](#)).

- Réparation : c'est de remettre les produits usagés en état de marche. Ces produits usagés ne sont normalement pas démontés. Ils sont réparés et certaines pièces peuvent être remplacées. Les ordinateurs, les téléphones portables, les machines à laver sont des produits qui conviennent pour la réparation.
- Remise à neuf : c'est d'amener les produits usagés à une qualité déterminée. Il nécessite des processus de réparation similaires. Elle implique le démontage de produits usagés en modules, l'inspection des modules et ensuite la fixation et/ou le remplacement de certains modules. Les modules approuvés sont ensuite réassemblés en produits remis à neuf. Les produits tels que les caravanes peuvent être remis à neuf.
- Reconditionnement : c'est de mettre les produits usagés en conformité avec les normes de qualité qui sont aussi bons que celles des nouveaux produits. Les processus impliqués sont similaires à ceux de la remise à

neuf. Les produits usagés sont souvent complètement désassemblés en modules et en pièces. Il y aura une inspection approfondie des modules. Les pièces ou les modules usés sont remplacés. Les pièces réparables sont réparées et soumises à des tests approfondis. Les pièces approuvées et les modules sont réassemblés en produits reconditionnés. Les photocopieurs, les moteurs, les pièces de voiture, les machines peuvent être reconditionnés.

- Cannibalisation : c'est de récupérer un ensemble limité de pièces réutilisables à partir de produits usagés. Seule une petite partie de produits usagés est récupérée pour la réutilisation, la refabrication ou la réparation d'autres produits. Il implique le démontage sélectif des produits usagés et l'inspection potentielle des pièces réutilisables.
- Recyclage : c'est de réutiliser les matériaux des produits ou composants usagés. Les matériaux récupérés sont séparés en catégories. Les matériaux peuvent être utilisés pour fabriquer des produits originaux si leur qualité est élevée. Dans le cas contraire, les matériaux peuvent être utilisés pour produire d'autres produits. Exemples : bouteilles de lait, bouteilles de boissons non alcoolisées, etc.
- Upcyclage : La différence du recyclage est la nécessité d'une réutilisation créative. L'objectif est de produire quelque chose de nouveau et meilleur que l'ancien article. Le produit final est généralement unique en son genre et durable. Par exemple, les T-shirts usagés sont utilisés pour fabriquer des chiffons de nettoyage.

Il existe quelques différences majeures entre les options de valorisation ci-dessus. Les produits usagés sont démontés au niveau du produit pour la réparation, au niveau du module pour la remise à neuf, au niveau de la pièce pour la refabrication, et au niveau du matériau pour le recyclage. La qualité des produits recyclés ou remanufacturés est la plus élevée (à l'état neuf), et la qualité des produits réparés et remis à neuf sera moindre. En utilisant l'option de réparation, certaines pièces sont réparées ou remplacées par des pièces de rechange. En recourant à la remise à neuf, certains modules sont réparés ou remplacés et il est possible qu'une mise à niveau soit nécessaire. La remise à neuf permet de combiner des modules/pièces neufs et d'occasion pour en faire de nouveaux produits pouvant être mis à niveau. Le upcyclage est une nouvelle approche de la remise à neuf qui peut augmenter considérablement la valeur des articles non désirés. Le recyclage fonctionne au niveau des matériaux ; cela signifie que les pièces sont retraitées sous forme de matière première.

1.2 Problématique

En fonction du contexte décrit ci-dessus, les travaux de recherche dans cette thèse portent essentiellement sur la valorisation des produits retournés en désassemblant leurs composants et les réassembler avec des nouveaux composants pour obtenir des nouveaux produits. Les opérations d'assemblage et de désassemblage jouent un rôle très important dans le processus de production.

Dans une chaîne logistique classique, les différents processus de production, de stockage et de distribution sont planifiés de façon séquentielle. Chaque processus est généralement optimisé en fonction des décisions prises par les autres processus qui sont déjà fixés précédemment. Plusieurs études ont montré un potentiel important d'économies dans la chaîne logistique en combinant les décisions de production et de transport ([Chandra et al. \(1994\)](#)). Ces études mettent en évidence la nécessité d'optimiser les différents processus de la chaîne logistique en tant qu'entité unique pour réduire le coût total et rendre les différents services plus rentables. La prise en compte de toutes les décisions dans un cadre unique offre une vision globale de la planification du réseau logistique. Pour cela, nous allons traiter, dans cette thèse un problème intégré de production et de transport dans lequel deux opérations d'assemblage et désassemblage sont faites dans le même usine de production. Dans ce problème les périodes de la planification de la production et les périodes de la planification des tournées ne sont pas nécessairement identiques.

Pour analyser et comprendre les décisions de production des entreprises, il est important de connaître les différents types de coûts auxquels elles sont confrontées au cours de ce processus. Il existe différents types de coûts de production, tels que les coûts fixes et les coûts variables. Les coûts fixes sont des coûts qui ne changent pas avec la quantité de production, comme le coût de lancement. Les coûts variables sont des coûts qui varient en fonction de la quantité de production, ils sont généralement déclarés par unité, comme le coût unitaire de production, le coût unitaire de détention, etc.

L'opération de désassemblage classique consiste à démonter les modules et les pièces à partir des produits retournés pour satisfaire la demande externe de leurs pièces, composants ou sous-ensembles ([Brennan et al. \(1994\)](#)). Dans notre cas, les pièces désassemblées sont utilisées dans l'opération d'assemblage.

Selon [Duta \(2006\)](#), il existe des différences entre les deux opérations. La différence fondamentale est que les demandes se situent au niveau des composants élémentaires constituant le produit retourné. Par contre dans l'assemblage, les demandes se situent au niveau du produit final. Le désassemblage est un processus divergeant d'un produit vers plusieurs composants, contrairement à l'assemblage où plusieurs composants convergent vers un seul produit. Le pro-

blème de planification de désassemblage consiste à déterminer le moment et la quantité du désassemblage. Le problème de planification d'assemblage consiste à déterminer le moment et la quantité de chaque type de composants utilisés pour avoir un produit final. La planification du désassemblage comporte un certain nombre de particularités. Une première particularité du désassemblage est liée à la nomenclature des produits retournés. Contrairement à l'assemblage, lors du désassemblage, les nomenclatures des produits sont inversées puisque les demandes concernent les composants d'un ou plusieurs produits finis. Nous pouvons trouver des structures pour un seul ou plusieurs produits à deux ou plusieurs niveaux sans ou avec parties communes. Dans notre cas, nous traitons une structure de désassemblage à deux niveaux.

En pratique, il existe certains niveaux d'incertitude, notamment en ce qui concerne la disponibilité et la qualité des produits en fin de vie à désassembler. Cette incertitude est influencée par de nombreux facteurs tels que la mauvaise qualité des produits retournés ou les dommages subis lors de l'opération de désassemblage.

L'assemblage est une opération de production dans laquelle différents composants sont utilisés pour fabriquer un produit final. Ces composants sont généralement achetés auprès différents fournisseurs. Les quantités des composants disponibles chez les fournisseurs sont supposées être connues à l'avance. Si l'usine de production est chargée d'organiser le transport entrant des différents composants, les gains peuvent alors être réalisés par l'intégration de la planification de la production avec les tournées de véhicules. Ce problème est connu dans la littérature comme le problème de tournées de véhicules et d'assemblage. Ce problème considère une usine de fabrication qui fabrique un produit final pour répondre à des demandes dynamiques mais déterministes. L'usine achète les composants essentiels auprès de divers fournisseurs, chacun fournissant un ou plusieurs types de composants. L'usine gère un parc illimité de véhicules à capacité pour gérer l'expédition des composants des fournisseurs vers l'usine. L'objectif est de minimiser le coût total de production, de stockage et de transport sous réserve de certaines contraintes sur un horizon temporel fini et discret (Afsar & Hnaien (2020), Chitsaz et al. (2020)).

Dans cette thèse, nous nous intéressons à la planification des opérations de désassemblage pour les produits retournés par les clients. Les nouveaux produits sont constitués par des nouveaux composants et autres désassemblés. Notre objectif est de minimiser les coûts totaux de lancement (pour l'opération de désassemblage et d'assemblage), la production, les ventes perdues, les coûts de stockages pour les produits neufs et retournés et les composants ainsi que les coûts d'achat et de transport.

Dans le désassemblage il existe une forte incertitude sur l'état des produits retournés ce qu'il donne une incertitude sur les quantités de chaque type

de composant désassemblé (rendement de désassemblage). Pour cette raison, nous avons étudié dans la première partie le cas déterministe dans lequel les rendements sont supposés connus et dans la deuxième partie le cas stochastique dans lequel les rendements sont incertains.

1.3 Contributions et organisation de la thèse

La thèse est organisée sous forme de 5 chapitres. Nous commençons ce rapport par l'introduction générale.

Ensuite, le chapitre 2 présente une revue de la littérature. Tout d'abord, les études existant sur la logistique inverse sont présentées. Parmi les problèmes de planification étudiés dans la logistique inverse, nous nous intéressons aux problèmes de planification du désassemblage. Ensuite, les travaux existants sur les problèmes de la gestion de la production tels que le problème de la planification des besoins en composant et le problème de dimensionnement de lots sont adressés. Nous offrons une étude détaillée des méthodes de résolution utilisées dans chaque problème. Enfin, nous passons en revue les problèmes de planification de la production et de transport intégrés. Nous résumons les récentes recherches sur le problème de tournées de véhicules avec gestion de stock, le problème de production et tournées de véhicules, et les problèmes de tournées de véhicules et d'assemblage. Nous classons les problèmes selon deux catégories, déterministes ou stochastiques.

Dans le chapitre 3 nous traitons un nouveau problème d'assemblage, désassemblage et tournées de véhicules avec des produits en retour. Nous examinons ce problème multi-périodes dans sa version statique et dynamique. Une caractéristique importante est que une partie de données se change de manière dynamique d'une période à une autre. Nous proposons un modèle de programme linéaire en nombres entiers mixtes, ainsi qu'une mat-heuristique itérative à deux phases. L'approche proposée consiste à résoudre le problème de production avec un solveur CPLEX. Pour la partie transport, une méthode approchée de type recherche à voisinage variable est utilisé pour trouver les meilleurs itinéraires. Dans les deux versions, la résolution de problèmes de production et de transport est faite de façon itérative avec un échange des informations en utilisant des techniques de diversification. Nous illustrons la performance de mat-heuristique sur des nouvelles instances et nous comparons ses résultats avec du modèle de la PLNE. Les expérimentations soulignent les avantages du modèle intégré par rapport à l'approche hiérarchique, ainsi que les avantages de considérer des composants désassemblés dans le système de production. Les résultats montrent l'efficacité de mat-heuristique en terme de la qualité de solutions et le temps d'exécutions.

Le chapitre 4 étudie la version dynamique dans le cas stochastique du problème où les rendements des composants désassemblés sont incertains. Le problème est d'abord formulé sous la forme d'un modèle mathématique non linéaire en nombres entiers mixtes. Nous proposons une approche d'optimisation robuste basée sur une modélisation par scénario. Nous abordons différentes hypothèses, où différents niveaux de flexibilité sont donnés aux différentes décisions qui peuvent ou non être influencées par les scénarios. Les analyses indiquent que le fait de permettre une certaine flexibilité dans le processus d'achat est le plus rapide en terme de temps CPU et améliore le coût total. Ensuite, nous développons un mat-heuristique à trois étapes pour optimiser le pire cas. Le mat-heuristique est basé sur la programmation mathématique et l'algorithme de recherche à voisinage variable. D'après les résultats, l'approche mat-heuristique est rapide et fournit des solutions de bonne qualité. Nous observons que, les résultats de la mat-heuristique peuvent être améliorés par le modèle de la PLNE, en prenant ces résultats comme une solution initiale.

Enfin, nous terminons cette thèse par une conclusion générale et nous abordons les perspectives de la future recherche.

CHAPITRE 2

État de l'art

2.1 Introduction

Nous allons voir dans ce chapitre un échantillon des travaux qui sont en relation avec notre problème. Nous avons axé cette étude sur les travaux qui se rapportent au contexte de notre étude, la logistique inverse. Puis, nous verrons les travaux sur la gestion de la production qui correspondent à la base du problème qui nous intéresse. Les problèmes de production et de transport intégrés font l'objet de la troisième partie de ce chapitre.

2.2 La logistique inverse

2.2.1 Définition de la logistique inverse

Au cours des dernières années, la logistique inverse a suscité une attention croissante de la part des chercheurs et de l'industrie. Cette situation s'explique par des raisons environnementales aussi bien qu'économiques. Parmi les principales préoccupations environnementales, nous trouvons la saturation de zones de décharge, le réchauffement climatique et l'épuisement rapide des matières premières, etc. Pour remédier à ces problèmes, les gouvernements imposent de nouvelles réglementations environnementales plus strictes qui obligent les fabricants à récupérer leurs produits en fin de vie par l'intermédiaire d'un réseau de la logistique inverse. Les entreprises peuvent non seulement se conformer aux réglementations légales, mais aussi d'utiliser la valeur économique restante contenue dans les produits en fin de vie grâce aux différentes options de récupération des produits, à savoir la réutilisation, le recyclage et la refabrication.

Dans la littérature, la logistique inverse est définie de différentes manières, Nous trouvons la logistique inverse ou inversée ou la logistique verte. Depuis les années 90, plusieurs définitions sont utilisées pour définir la logistique inverse. En 1995, [Thierry et al. \(1995\)](#) considèrent la gestion de la récupération des produits et des matériels usés par des fabricants comme un processus de la logistique inverse. En 1999, [Rogers et al. \(1999\)](#) définissent la logistique inverse

comme étant un processus de planification du point de consommation jusqu'au point d'origine. Il s'agit de la gestion unidirectionnelle de flux de matières et d'information, dont le but est d'améliorer l'élimination des déchets. Plus tard, [Chouinard et al. \(2007\)](#) considère la logistique inverse comme un système qui permet la récupération des produits retournés à partir des consommateurs et les traiter dont l'objectif est d'assurer la gestion et la planification des activités de ramassage, d'évaluation, de désassemblage et de retransmission.

Selon [Bai & Sarkis \(2013\)](#) la logistique inverse est l'inverse de la logistique directe. La logistique inverse diffère de la logistique directe par plusieurs aspects ([Pochampally et al. \(2009\)](#)). Dans la logistique directe, les nouveaux produits fabriqués dans l'usine sont transportés vers de nombreux destinataires. Par contre, dans la logistique inverse, les produits retournés, collectés auprès de nombreux centres de collecte, sont transportés vers le producteur ou vers un centre de récupération des produits (réusinage, recyclage ou d'élimination). En d'autres termes, les flux de transport dans la chaîne logistique directe sont d'un à plusieurs (one-to-many) alors qu'ils sont de plusieurs à un (many-to-one) dans la logistique inverse.

Même si certaines études dans la littérature qui utilisent les termes "logistique inverse" et "chaîne logistique inverse" de manière interchangeable, il existe une légère différence entre les deux. La logistique inverse traite principalement le transport, la planification de la production et la gestion des stocks, tandis que la chaîne logistique inverse a une orientation plus large qui implique d'éléments supplémentaires tels que la coordination et la collaboration entre les partenaires de la chaîne ([Prahinski & Kocabasoglu \(2006\)](#)). Autrement dit, la logistique inverse est l'un des éléments d'une chaîne logistique inverse. La logistique inverse permet la valorisation des produits en fin de vie à travers un réseau intégrant les processus de récupération, de traitement, de désassemblage et de recyclage ou d'élimination. La robustesse et la durabilité d'un réseau de logistique inverse sont assurés par la cohérence entre les trois niveaux de décisions stratégiques, tactiques et opérationnelles au niveau de différentes activités.

Il existe des différents types de produits en retour dans la logistique inverse tel que les retours de clients, les retours de réparation/service, les retours de fin de vie, les retours de conteneurs réutilisables et les retours de produits loués. Différents travaux traitent différents types de produit en retours de différents niveaux de décisions :

- Produits électroniques et électriques : Dans ce contexte [Shih \(2001\)](#) traite le cas d'un réseau de récupération d'appareils électriques dont l'objectif est d'optimiser le coût total afin de déterminer le nombre des sites de collectes, de recyclages et d'allocation des flux entre ces sites. [Dat](#)

et al. (2012) traitent le problème du recyclage des produits électriques et électroniques en fin de vie à Taiwan. Les auteurs développent un modèle de programmation linéaire pour minimiser les coûts de transport, d'exploitation et d'élimination. Dans le même contexte, Shokouhyar & Aalirezaei (2017) proposent un modèle mathématique de réseau de logistique inverse en deux étapes basées sur des objectifs de développement durable dans lequel les objectifs économiques, environnementaux et sociaux sont considérés simultanément. Un algorithme génétique à objectifs multiples est développé pour déterminer les meilleurs emplacements des centres de collecte et des usines de recyclage. Le modèle proposé est examiné à travers un cas réel en Iran. L'étude menée par Bal & Satoglu (2018) permet d'étudier le processus de collecte des produits d'équipements électriques et électroniques dans les points de service, leur transport vers les centres de recyclage et la valorisation des déchets. Ils proposent un modèle mathématique multi-sites, multi-produits et multi-périodes, en tenant compte des conditions réelles. Ils proposent aussi une approche du triple bilan et ils utilisent la programmation des objectifs pour résoudre un problème des appareils électroménagers.

- Les déchets dangereux : Une étude est réalisée par Alumur & Kara (2007) sur le recyclage des déchets dangereux. Les auteurs proposent un modèle mathématique afin de déterminer les localisations des centres d'élimination et l'itinéraire du transport de déchets dangereux au centre de stockage. L'objectif est de minimiser le coût de transport, le coût de traitement, le coût d'élimination et le risque sur l'environnement provoqué par le transport de déchets dangereux. Dinler & Güngör (2017) proposent un modèle de programmation linéaire floue avec plusieurs objectifs pour la planification des processus de recyclage des produits qui contiennent des substances dangereuses et explosives. L'application du modèle proposé est réalisée dans une usine où divers produits devenus inopérants sont livrés à l'usine à des moments incertains à partir de divers entrepôts. Une simulation hybride de Monte Carlo est utilisée pour obtenir des solutions pareto optimales afin de résoudre le modèle. Ndanguza et al. (2020) s'intéressent à la modélisation et à l'analyse d'un système dynamique où l'intoxication de la population est due à une mauvaise gestion de ces déchets. Le modèle précise la relation entre la densité des individus dans une certaine zone, la production de déchets et la concentration de la substance toxique.
- Industrie automobile : Krikke et al. (1999) traitent un problème du retour des produits dans l'industrie d'automobile dont l'objectif est de déterminer la localisation des installations et de tournées de véhicules.

Ils proposent un modèle qui prend en compte les fournisseurs de produits, les centres de traitement et de recyclage afin de minimiser le coût total de transport et d'investissement. [Schultmann et al. \(2006\)](#) traitent le cas des véhicules hors d'usage en Allemagne. Ils présentent les différentes options de conception d'une chaîne logistique en boucle fermée, avec une concentration sur la manière dont les flux de matériaux inversés peuvent être traités en vue de les réintégrer dans leur véritable chaîne logistique. Les aspects de logistique inverse sont modélisés avec la planification de l'itinéraire des véhicules ainsi qu'un algorithme adapté au problème est proposé. [Özceylan et al. \(2017\)](#) étudient les constructeurs automobiles en Turquie qui sont responsables de la reprise gratuite des véhicules en fin de vie aux consommateurs pour réintégrer les pièces usagées dans l'industrie. Un modèle de programmation linéaire est développé pour gérer les flux de matériaux inversés en vue de les réintégrer dans les chaînes d'approvisionnement. Plusieurs scénarios sont examinés pour montrer les performances du modèle proposé et son applicabilité dans l'industrie automobile. [Kuşakcı et al. \(2019\)](#) développent un modèle linéaire en nombres entiers mixtes (PLNE) flous d'allocation et de localisation pour un réseau logistique inverse de véhicules hors d'usage en Turquie. En conséquence, cette étude utilise une nouvelle approche et suppose que l'approvisionnement en véhicules hors d'usage dans le réseau est incertain.

2.2.2 Problèmes liés à la logistique inverse

Les problèmes de la logistique inverse liés aux produits en fin de vie sont classés sous forme de trois grandes catégories : les problèmes stratégiques, les problèmes de planification et les problèmes de traitement (Figure 2.1).

Problèmes stratégiques :

- Conception du réseau : La conception du réseau sur la chaîne logistique est le processus de construction et de modélisation de cette chaîne pour déterminer le meilleur emplacement et la taille optimale des facilités à inclure dans le réseau avec les ressources et les emplacements disponibles ([Lieckens & Vandaele \(2012\)](#)). L'emplacement de toutes les facilités de la chaîne logistique telles que les sites de production, les entrepôts et les centres de distribution doit être optimisé pour accroître l'efficacité de l'ensemble du système.
- Sélection des produits usagés : Il existe de nombreuses entreprises qui collectent des produits usagés pour faire des bénéfices. Lors de la sélection des produits usagés, ces entreprises comparent les revenus provenant

du recyclage ou de la revente des composants des produits et les coûts de collecte et de retraitement des produits usagés (Pochampally & Gupta (2008)).

- Technologie de l'information : Une infrastructure informatique efficace est indispensable dans un système de logistique inverse, compte tenu de la nécessité d'une projection précise du temps et de la quantité de produits retournés. En outre, la coordination entre les différentes parties impliquées dans un système de logistique inverse est assurée par l'infrastructure informatique (Kumar & Chan (2011)).

Planification :

- Prévision : il s'agit d'une tâche statistique courante dans les entreprises, où elle contribue à éclairer les décisions concernant la programmation de la production, du transport et du personnel, et fournit un guide pour la planification stratégique à long terme. La prévision consiste à prévoir l'avenir aussi précisément que possible, compte tenu de toutes les informations disponibles, y compris les données historiques et la connaissance de tout événement futur qui pourrait avoir un impact sur les prévisions. Le niveau élevé d'incertitude quant au moment et à la quantité des retours rend nécessaire l'utilisation de diverses méthodes de prévision inédites pour prévoir les retours de produits (De Brito & Van Der Laan (2009)).
- Planification de la production : La planification de la production implique des décisions sur la quantité et le moment du démontage, la quantité et le moment de la refabrication, la quantité à produire et/ou la commande de nouveaux matériaux, et la manière de coordonner le démontage et le remontage (Ferrer et Whybark (2001) Ferrer & Whybark (2001)).
- Gestion des stocks : L'objectif de la logistique inverse est de réduire le plus possible les stocks de produits et de pièces retournés et de réduire le coût des stocks dans la logistique inverse en transformant les produits et pièces retournés en produits et pièces réutilisables, ce qui maximise la valeur des pièces et produits retournés (Inderfurth & van der Laan (2001)).

Traitement :

Les problèmes de traitement sont liés au traitement physique des produits en fin de vie. Le nettoyage, le recyclage, le désassemblage et le réassemblage des produits en fin de vie peuvent être évalués dans cette catégorie.

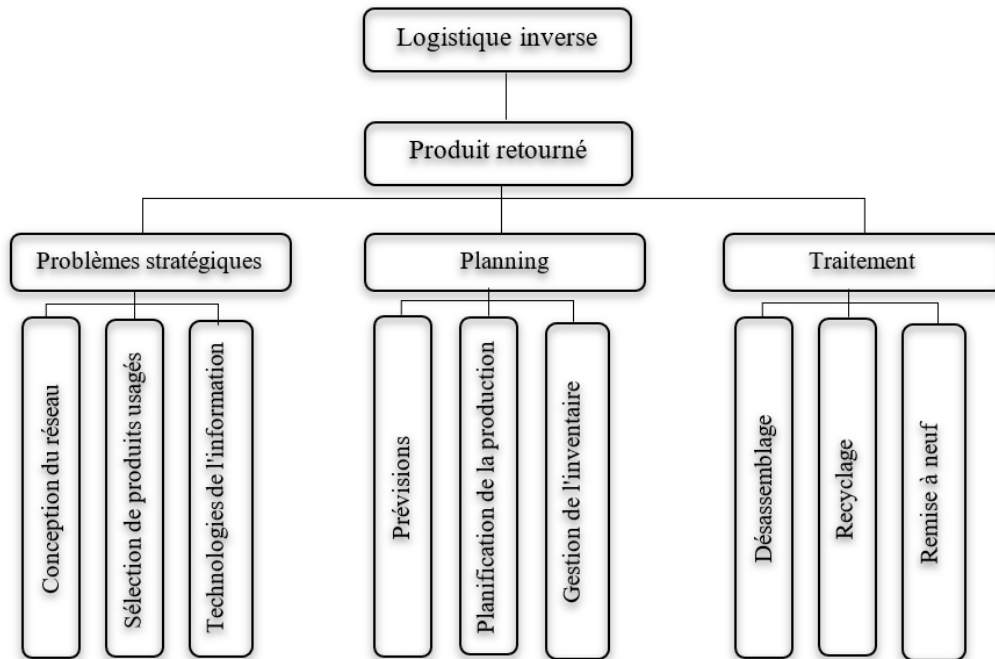


Figure 2.1 – Problèmes liés à la logistique inverse

2.2.3 Désassemblage

Selon [Brennan et al. \(1994\)](#), le désassemblage est défini comme étant un processus de récupération de composants souhaités constituant un produit. Une autre définition du désassemblage est annoncée par [Gungor & Gupta \(1997\)](#) le désassemblage est considéré comme étant un processus systématique qui permet la récupération sélective des composants et/ou des sous-ensembles réutilisables, recyclables ou polluants à partir d'un produit. Selon [Lambert & Gupta \(2005\)](#), le désassemblage est la séparation d'un ensemble en plusieurs sous-ensembles, ou bien la rupture d'une ou plusieurs liaisons entre différents composants. Dans la figure 2.2, nous présentons un exemple d'une opération de désassemblage pour un produit qui donne trois différents types de composants où r_i présente le rendement (la quantité obtenue de chaque type de composant). Un produit désassemblé peut donner un composant réutilisable avec un rendement qui est égal à 1. Les composants dont le rendement égale à 0 comme r_2 sont jetés.

Le désassemblage est un processus qui joue un rôle fondamental dans la valorisation des produits retournés. Il peut être considéré comme une technique de récupération des composants qui peuvent être réutilisés comme des produits neufs, recyclés ou encore réutilisés comme pièces de rechanges. Les

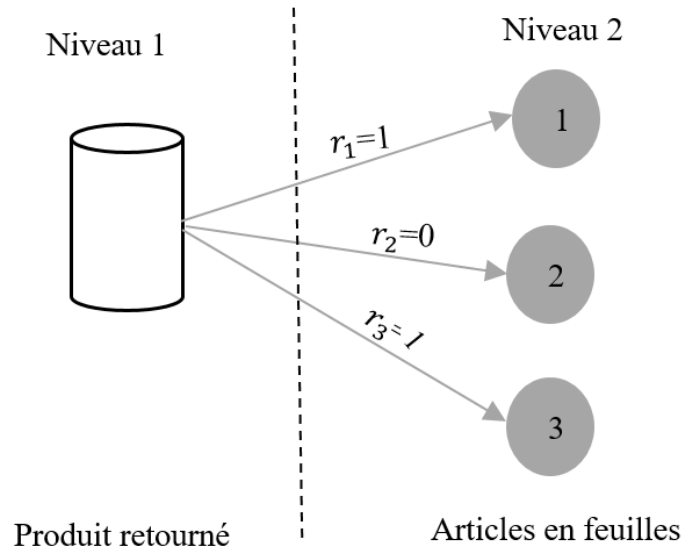


Figure 2.2 – Un exemple d’une opération de désassemblage

composants désassemblés sont ensuite vendus en vue de générer des revenus et de minimiser l’impact sur l’environnement. Dans les opérations de désassemblage, l’optimisation est nécessaire afin de rendre ce processus efficace et durable.

2.2.4 Problèmes liés au désassemblage

D’après la littérature, il existe différents problèmes de décisions dans le domaine du désassemblage que nous les présentons dans la figure 2.3. Le processus de désassemblage est une séquence d’opérations. L’ordre de ces opérations dans un processus spécifique est appelé séquence de désassemblage (Hui et al. (2008), Xu et al. (2020)). Le problème de séquence du désassemblage et le problème de planification de désassemblage (Lee et al. (2001)) sont parmi les problèmes les plus connus dans le processus de désassemblage. La séquence de désassemblage permet de trouver le meilleur ordre pour désassembler des composants. Tandis que la planification du désassemblage permet de savoir la quantité à désassembler et le moment. Le problème d’équilibrage des lignes de désassemblage (Liu et al. (2020), Li et al. (2020)) consiste à répartir les tâches de désassemblage aux postes de travail en équilibrant la ligne du désassemblage. Nous trouvons également le problème d’automatisation du processus de désassemblage (Veerakamolmal & Gupta (2002)), le problème de désassemblage sur commande (Kongar & Gupta (2006)) et le problème d’ergonomie (Youssef et al. (2011)).

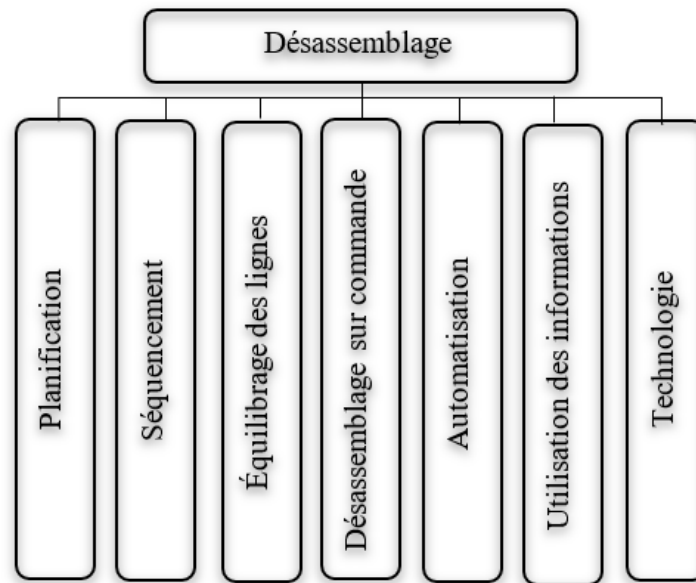


Figure 2.3 – Problèmes liés au désassemblage

2.2.5 Planification de désassemblage

Depuis les années 90, la planification des opérations de désassemblage a reçu un grand intérêt. Il s'agit d'organiser dans un horizon de planification périodique ou continu une succession d'actions pour désassembler une quantité de composants à partir d'un produit retourné tout en satisfaisant différentes demandes de ces composants. Les chercheurs n'ont pas envisagé, au début de la planification du désassemblage, l'optimisation des différents coûts associés aux opérations, ils cherchent seulement à déterminer la quantité de désassemblage des produits en fin de vie pour répondre aux exigences de leurs composants constitutifs. Au début des années 2000, les chercheurs commencent à intégrer d'autres coûts tels que le coût de lancement lié au produit en fin de vie. Ce dernier concerne la préparation de l'opération de désassemblage. Concernant le coût de stockage, il concerne les composants qui sont désassemblés et stockés pour répondre aux demandes futures.

Dans le problème de planification de désassemblage, il arrive que les producteurs aient souvent moins de contrôles sur l'état des produits retournés. Pour faire face à ce problème et afin de valoriser les retours des produits, les industriels essaient de développer des méthodes de résolution efficaces permettant de prendre en compte les incertitudes liées à la quantité et à la qualité des produits retournés d'une part, et à la demande des composants et des matières à recycler d'autre part. Notre analyse des travaux de la littérature nous a permis de diviser les travaux de recherche en deux catégories; le cas

déterministe et le cas stochastique qui sont résumés dans le tableau 2.1.

2.2.5.1 Le cas déterministe

Le premier modèle mathématique est proposé par [Gupta & Taleb \(1994\)](#). Le moment et la quantité des produits retournés à désassembler sont déterminés de manière à satisfaire certaines demandes de leurs pièces, sous-ensembles ou composants. Ils étudient le problème sans contrainte de capacité de désassemblage et sans parties communes entre les produits retournés afin d'optimiser une planification inversée des besoins en matériaux. Ils étendent ensuite la procédure pour inclure les pièces communes pour le désassemblage de plusieurs produits ([Taleb & Gupta \(1997\)](#)). [Lee & Xirouchakis \(2004\)](#) proposent une heuristique basée sur l'algorithme de Gupta et Taleb avec un objectif supplémentaire de minimiser les divers coûts liés aux opérations de désassemblage. [Lee et al. \(2004\)](#) proposent trois modèles de programmations entières pour le mono et multi-produits et avec ou sans partie commune pour le problème sans capacité. Ils expliquent qu'une approche optimale ou quasi-optimale des solutions peuvent être obtenues en utilisant le solveur CPLEX dans des temps de calcul pour les problèmes de petite taille.

En 2007, [Langella \(2007\)](#) formule le problème de planification de désassemblage comme un PLNE et il propose plusieurs heuristiques pour résoudre le problème avec des parties communes. Ensuite en 2008, [Kim et al. \(2009\)](#) développent un algorithme de branch-and-bound pour résoudre le problème de planification de désassemblage avec un seul item et sans parties communes. [Gao & Chen \(2008\)](#) développent un algorithme génétique pour résoudre le le problème de planification de désassemblage avec une structure à plusieurs niveaux et un seul type de produit. L'objectif est de minimiser la somme des coûts de lancement, de stockage et de désassemblage. [Prakash et al. \(2012\)](#) présentent une nouvelle forme d'algorithme de recuit simulé basé sur des contraintes pour résoudre le problème avec des plusieurs items et avec des parties communes. [Ullerich & Buscher \(2013\)](#) formulent le problème de planification de désassemblage sous la forme d'un modèle de programmation linéaire en nombres entiers et ils ont pris en compte la contrainte de capacité dans chaque période. Cependant, ce travail ignore le coût de lancement qui est un paramètre commun et indispensable dans la réalité industrielle. [Ji et al. \(2016\)](#) abordent le problème de planification de désassemblage avec des contraintes de capacité et avec des pièces communes et un coût de démarrage et ils résolvent le problème avec une heuristique de relaxation lagrangienne.

L'étude proposée par [Godichaud et al. \(2015\)](#) traite le problème de planification de désassemblage avec des ventes perdues et coût de pénalité de la surcharge de la capacité de désassemblage, ce problème est formulé comme un

modèle de programmation en nombres entiers mixtes et un algorithme génétique est proposée pour donner une solution à ce problème. Dans leur étude, [Hrouga et al. \(2016\)](#) combinent un algorithme génétique et des heuristiques de correction et d'optimisation pour résoudre le problème de planification de désassemblage avec des produits multiple, vente perdu et contraintes de capacité. [Tian & Zhang \(2019\)](#) abordent le même problème mais avec plusieurs items et périodes. Ils formulent le problème sous la forme d'un programme mixte qui est résolu par une méta-heuristique d'optimisation d'essaim de particules et la programmation dynamique.

[Slama et al. \(2020\)](#) traitent le problème de planification de désassemblage dynamiques avec contrainte de capacité avec des achats externes, des articles défectueux et avec retard et des délais de préparation. Le problème consiste à déterminer combien de produits en fin de vie doivent être désassemblés au cours de chaque période. Ils proposent un PLNE pour formuler le problème afin de maximiser le gain du processus de désassemblage, qui est obtenu par la différence entre les revenus réalisés par la revente des articles récupérés après le désassemblage et les coûts liés à l'exploitation des tâches de désassemblage. [Pour-Massahian-Tafti et al. \(2020\)](#) résolvent le problème de planification de désassemblage pour le type de produit unique ayant un stock excédentaire. Les décisions d'élimination sont prises pour éviter l'accumulation de stocks tout au long de l'horizon de planification. Trois nouvelles PLNE ainsi que deux heuristiques sont proposées pour résoudre le problème.

Les recherches existantes se concentrent principalement sur le problème de planification de désassemblage avec la prise en compte des décisions dans les autres activités de la chaîne logistique. Cependant, l'intégration avec le problème de tournées de véhicules est rarement envisagée. Le problème de tournées de véhicules avec désassemblage est initialement introduit par [Habibi et al. \(2017\)](#). Ils étudient le cas où la collecte de produits en fin de vie et leur démontage conduisent à la satisfaction des demandes des clients dans différents types de composants et de sous-ensembles.

2.2.5.2 Le cas stochastique

Les études sur le cas stochastique de problème de planification de désassemblage sont limitées, car elles sont plus complexes que les problèmes déterministes. Les articles existants montrent un intérêt pour l'incertitude de la demande et/ou du rendement. En effet, le rendement aléatoire ne résulte pas nécessairement d'articles défectueux, mais il peut également être dû à des pénuries chez les fournisseurs ([Moon et al. \(2012\)](#)).

Il existe deux approches pour représenter le rendement aléatoire dans un contexte de production. Le premier groupe de recherche représente le rende-

ment aléatoire avec un processus de Bernoulli (Teunter & Flapper (2011)), où une certaine fraction des marchandises est défectueuse, et le nombre d'articles défectueux dépend de la taille du lot. Le second groupe de recherche considère que tous les articles produits sont défectueux et les rendements aléatoires sont modélisés avec une distribution géométrique (Salameh & Jaber (2000)).

Inderfurth (2003) propose une heuristique de type un-à-un et un-à-plusieurs pour résoudre le problème de planification de désassemblage stochastique à période unique avec incertitude de rendement. Kongar & Gupta (2006) utilisent une programmation mathématique floue pour résoudre l'incertitude concernant le nombre de produits en fin de vie récupérés ; l'objectif est de maximiser le profit total et le nombre des composants recyclés. Barba-Gutiérrez & Adenso-Díaz (2009) étendent le modèle proposé par Barba-Gutiérrez et al. (2008) en intégrant l'incertitude de la demande. Pour résoudre ce problème, Ils proposent un algorithme de logique floue. Kim & Xirouchakis (2010) étudient le problème de planification de désassemblage stochastique avec une demande stochastique et de multiples types de produits avec une structure de produits à deux niveaux. Ils proposent un modèle de stockage stochastique et ils développent une méthode de relaxation lagrangienne pour le résoudre.

Ensuite, Wang & Huang (2013) examinent l'incertitude de la demande et proposent un modèle de programmation linéaire à deux niveaux utilisant une approche basée sur des scénarios. Inderfurth et al. (2015) présentent un modèle mathématique à deux racines et trois feuilles qui illustre l'effet de l'incertitude du rendement dans les modèles stochastiques proportionnels.

Liu & Zhang (2018) traitent le problème de planification de désassemblage stochastique à seul item et à périodes multiples avec contrainte de capacité et avec des rendements et des demandes aléatoires. Ce problème est formulé comme un programme non linéaire en nombres entiers mixtes et un algorithme basé sur une approximation externe est proposé pour le résoudre. Une approche de contrainte de hasard est introduite pour s'assurer que la probabilité de satisfaire la demande est supérieure à un niveau de service prédéterminé, puis approximée par une contrainte de cône de second ordre.

2.3 Gestion de la production

2.3.1 Planification des besoins en composants

À la fin du 20^{ème} siècle, le système de la planification des besoins en matériel (MRP) est devenu un outil importante de la gestion des flux de matières premières et des composants dans les usines (Mabert (2007)). La MRP est une méthode de gestion des stocks (Heizer & Render (2013)) qui

Tableau 2.1 – Synthèse des articles sur le problème de planification de désassemblage

Références	Multi-Niveaux	Multi-Produits	Avec capacité	Déterministe	Paramètres	Approche de résolution
Gupta & Taleb (1994)	*			*	N1	RMRRP
Taleb & Gupta (1997)	*	*		*	N1, d_c	Heuristique en deux phases
Lee et al. (2004)	*	*		*	S_c, P_c, H_c, d_c	PL
Lee & Xirouchakis (2004)	*			*	S_c, P_c, H_c, d_c	Heuristiques
Inderfurth & van der Laan (2001)	*	*		*	d_c, Z_c, W_c	Heuristiques
Kongar & Gupta (2006)	*	*		*	Max Profit, N2	Programmation à objectifs flous
Langella (2007)	*	*		*	S_c, H_c, d_c, Z_c, W_c	Heuristiques
Gao & Chen (2008)	*	*		*	S_c, H_c, d_c	algorithme génétique
Barba-Gutiérrez & Adenso-Díaz (2009)	*			*	N1	F-RMRP
Kim et al. (2009)	*			*	S_c, H_c, R_c	branch and bound
Kim & Xirouchakis (2010)	*	*		*	S_c, H_c, R_c	Heuristique lagrangienne
Prakash et al. (2012)	*	*		*	H_c, d_c	CBSA
Ullrich & Buscher (2013)	*	*		*	S_c, H_c, d_c, Z_c	Heuristiques
Ji et al. (2016)	*	*		*	S_c, H_c, L_c, O_c	Heuristique lagrangienne
Godichaud et al. (2015)	*	*		*	S_c, H_c, d_c, Z_c, W_c	PLNE, AG
Inderfurth (2009)	*	*		*	S_c, H_c, d_c, Z_c, W_c	Analyse empirique et mathématique
Inderfurth et al. (2015)	*	*		*	S_c, H_c, L_c	Analyse empirique et mathématique
Hrunga et al. (2016)	*	*		*	S_c, H_c, P_c	algorithme génétique et Fixer et optimiser
Lin & Zhang (2018)	*	*		*	S_c, H_c, P_c	Programmation non linéaire, Approximation extérieure
Tian & Zhang (2019)	*	*		*	S_c, P_c, H_c, W_c	OEP et programmation dynamique
Shama et al. (2020)	*	*		*	S_c, H_c, Z_c, O_c	PLNE
Pour-Massahian-Tafti et al. (2020)	*	*		*	S_c, H_c, d_c	3 PNEM, 2 Heuristiques

N1 : nombre de produits à démonter, N2 : nombre de produits à éliminer, S_c : coût de lancement, P_c : coût de démarrage, H_c : coût de la racine, d_c : coût de possession, d_e : coût de désassemblage, Z_c : coût de achat des pièces, W_c : coût d'élimination, R_c : coût de la pénalité pour demande insatisfaite, C_c : coût de démarrage, L_c : coût de la pénalité pour vente perdue, O_c : coût de surcharge, PNEM : programmation en nombres entiers mixtes, RMRRP : algorithme de planification inverse des besoins en matériel, PL : programmation linéaire, F-RMRP : RMRRP basé sur une logique floue, CBSA : algorithme de recuit simulé basé sur les contraintes, AG : algorithme génétique, OEP : méta-heuristique d'optimisation d'essaim de particules .

permet d'obtenir le coût de stockage minimum. Les systèmes MRP calculent la taille des lots à produire (ou à commander) pour chaque composant de chaque période. Ces calculs sont basés sur les nomenclatures, qui indiquent la hiérarchie des composants (c'est-à-dire la structure du produit et le nombre de composants requis pour produire chaque article final ou composant ultérieur). [Iasya & Handayati \(2015\)](#) montre que la mise en œuvre de la MRP aide l'entreprise à gérer le gaspillage des matériaux, d'avoir une production plus efficace et de mener une activité plus rentable. Les méthodes MRP produisent le calendrier de production et le besoin de matériel ce qui nécessite la technique de dimensionnement de lots. Déterminer la taille des lots dans les zones de production est une tâche essentielle dans la planification et le contrôle de la production ([Schmidt et al. \(2015\)](#)). La MRP utilise trois techniques de dimensionnement de lots ([Heizer & Render \(2013\)](#)), à savoir lot pour lot, quantité de commande économique, et quantité de commande périodique.

- Lot pour lot (LFL) : Le LFL est une technique qui permet de produire la quantité de matières premières nécessaire pour répondre aux plans qui sont faites. La technique LFL est conforme aux objectifs du MRP pour répondre aux besoins de demande dépendante.
- Quantité de commande économique (EOQ) : L'objectif de l'EOQ est de minimiser les coûts de commande et de stockage. Cette technique est très utilisée, et repose sur les hypothèses suivantes : a) la demande est connue, constante et indépendante, b) le délai de livraison est connu et constant, c) la réception des stocks est instantanée et complète, d) les remises sur quantité ne sont pas possibles, e) seuls les coûts variables sont mis en place et maintenus, f) les ruptures de stock peuvent être complètement évitées.
- Quantité de commande périodique (POQ) : Le POQ est une technique de réservation qui permet de passer des commandes dans des intervalles de temps déterminés entre les commandes. Le nombre d'articles commandés devant couvrir le nombre d'articles requis pendant l'horizon de planification.

Différents modèles mathématiques sont étudiés dans un contexte déterministe ([Billington et al. \(1983\)](#), [Clark & Armentano \(1995\)](#)). En pratique, les systèmes de MRP sont soumis à diverses formes d'incertitude telles que la demande, les délais, le rendement de production, la capacité de production ([Dolgui & Prodhon \(2007\)](#), [Guide Jr & Srivastava \(2000\)](#)). Plusieurs chercheurs ([Kadipasaoglu & Sridharan \(1995\)](#), [Zhao et al. \(2001\)](#), [Bai et al. \(2002\)](#)) évaluent par simulation l'impact de l'incertitude de la demande sur les systèmes MRP pour les problèmes de production multi-échelons. Bien que des nombreuses études suggèrent l'utilisation des stocks de sécurité dans les systèmes

MRP avec une incertitude sur la demande (Benton (1991)). Par exemple, Inderfurth (2009) étudie un système MRP à un seul échelon avec une politique de stock critique, où les quantités de production sont calculées pour amener les niveaux de stock au-dessus de certains seuils critiques. Des autres recherches comme Thevenin et al. (2020), étudient le MRP dans des conditions d'incertitude de la demande dont un modèle à deux étapes et un modèle à plusieurs étapes sont proposés pour traiter le problème.

Pour les études sur les délais aléatoires nous trouvons l'article de Louly et al. (2008). Les auteurs mènent une recherche sur la planification optimale en MRP avec des délais incertains. Ce système est conçu pour les assemblages. La fonction objective de ce modèle était de minimiser la somme des différents coûts qui comprend le coût moyen de détention des composants et le coût moyen d'accumulation. Louly & Dolgui (2013) étudient le MRP sous incertitudes des délais. Ils considèrent l'approche MRP avec la politique POQ. L'objectif est de trouver le phasage temporel optimal de la MRP correspondant à chaque période de la politique POQ. Ils proposent un modèle mathématique et des algorithmes pour minimiser la somme des coûts de lancement et de possession tout en satisfaisant une contrainte sur le niveau de service.

2.3.2 Dimensionnement de lots

Dans cette section, les problèmes de dimensionnement de lots sont classés en plusieurs catégories de plus simples aux plus complexes.

2.3.2.1 Problème de dimensionnement de lots à un seul item

Le problème de dimensionnement de lots (problème de dimensionnement de lots) à un seul item est étudié pour la première fois par Wagner & Whitin (1958). Ils développent une méthode exacte basée sur la programmation dynamique pour résoudre ce problème.

La solution de ce problème consiste à planifier la production sur un nombre des périodes discrètes de l'horizon de planification afin de minimiser les coûts de production et de stockage. Le coût de production présente le coût unitaire pour chaque unité produite. Le coût de stockage présente le coût unitaire pour chaque unité stockée à la fin de chaque période.

Dans ce qui suit, nous allons commencer par présenter les différents modèles mathématiques pour le problème de dimensionnement de lots à un seul item, puis les approches de résolution exactes et approchées.

(a) Modélisation :

La formulation agrégée : C'est la formulation classique de problème de dimensionnement de lots proposé par Wagner & Whitin (1958). Le

nom de la formulation agrégée est donné car la production de l’item est définie par sa période de production.

La formulation basée sur le problème de localisation : Krarup & Bilde (1977) présentent la formulation du problème de dimensionnement de lots en un problème de localisation d’entrepôts (Facility Location Problem). Ils introduisent la période à laquelle ces productions sont consommées. Dans cette formulation, la production est définie par sa période de production et par sa période de consommation.

Formulation basée sur le problème du plus court chemin : Eppen & Martin (1987) proposent une autre reformulation du problème de dimensionnement de lots inspiré de problème dans un réseau de transport. Ils considèrent un graphe orienté avec $T + 1$ nœuds, $t \in [0; T + 1]$ et (t, t') est un arc qui existe pour tout $t < t'$. Un coût de lancement de la production à la période $t + 1$ est associé à l’arc (t, t') pour satisfaire la demande à $[t + 1, \dots, t']$. Le plus court chemin du 0 à T dans ce réseau est équivalent à la solution optimale du problème de dimensionnement de lots.

- (b) Les approches de résolution : Le problème de dimensionnement de lots à un seul item a reçu une attention particulière en raison de sa simplicité et facilité de résolution (en un temps polynomial) et de son importance comme étant un sous-problème de problème de dimensionnement de lots plus complexe. Brahim et al. (2017) proposent un état de l’art sur le problème de dimensionnement de lots à un seul item.

Différentes approches de résolution sont proposés pour résoudre le problème de dimensionnement de lots avec un seul item. Les méthodes exactes sont les méthodes les plus utilisées. Parmi ces méthodes nous trouvons la programmation dynamique pour résoudre le problème de dimensionnement de lots sans contrainte de capacité (Aksen et al. (2003)) ou avec contraintes de capacité (Florian & Klein (1971)). Ainsi, les méthodes des coupes (Wolsey (1998)). Plusieurs auteurs proposent des heuristiques afin de fournir de bonnes solutions réalisables au problème (Maes et al. (1991), Hardin et al. (2007)). Parmi les récents articles nous trouvons celui de Rapine et al. (2018). Les auteurs étudient un problème de dimensionnement de lots à un seul item avec une contrainte d’énergie limitant la consommation d’énergie à chaque période. Ils fournissent un PLNE pour le problème dans le cas général. Ils développent aussi un algorithme de temps polynomial. Carvalho & Nascimento (2018) étudient le problème de dimensionnement de lots avec contrainte de capacité et avec plusieurs usines. Toutes les usines produisent le même item avec une capacité limitée et dans certaines conditions tel que les configurations

de leurs machines peuvent être reportées aux périodes suivantes. L'horizon de planification est fini (12 périodes) où chaque usine a ses propres demandes. Cet article présente une nouvelle mat-heuristique nommée kernel search pour résoudre le problème. La méthode de résolution proposée est testée sur 120 instances de la littérature et 8 instances nouvellement générées. [Rodoplu et al. \(2020\)](#) abordent un nouveau problème de problème de dimensionnement de lots à un seul item pour les configurations de type flow-shop en intégrant le problème de sélection du contrat avec des sources d'énergie multiples. Ils introduisent un modèle mathématique qui définit un plan de production optimale et une option de contrat de fournisseur d'énergie. Une heuristique Fix-and-Relax est introduite pour résoudre le problème. Deux procédures de relaxation différentes sont appliquées et la performance de l'approche de la solution est testée sur des instances générées de façon aléatoire. [Suzanne et al. \(2020\)](#) proposent un modèle pour le problème de dimensionnement de lots à un seul item, qui intègre la gestion des résidus de production inévitables classés comme sous-produits. Ils développent un algorithme de programmation dynamique polynomiale.

2.3.2.2 Problème de dimensionnement de lots à plusieurs items

Le problème de dimensionnement de lots à plusieurs items est NP-Difficile donc les auteurs se sont principalement intéressés aux méthodes approchées. Parmi les méthodes les plus efficaces nous trouvons la méthode de la relaxation lagrangienne ([Thizy & Van Wassenhove \(1985\)](#)) aussi des approches hiérarchiques ([Mehra et al. \(1996\)](#)). Il y a des méthodes exactes qui sont développées pour résoudre ce problème tel que les méthodes de branch-and-cut (Miller et al 2000 [Miller et al. \(2000\)](#)) et branch-and-bound ([Haase & Kimms \(2000\)](#)).

[Hein et al. \(2018\)](#) traitent le problème de dimensionnement de lots avec plusieurs items et avec demande déterministe et dynamique dont l'objectif est de minimiser le coût d'inventaires et le coût de lancement. Ils développent une heuristique de Dixon–Silver et une méthode de programmation génétique. Pour tester la performance de leur approche, ils génèrent 720 instances comprenant 360 problèmes avec 12 produits et 12 périodes et 360 problèmes avec 24 produits et 24 périodes. [Zhou et al. \(2018\)](#) proposent un PLNE pour résoudre un problème de dimensionnement de lots dynamique avec plusieurs items (supérieures à 500 produits), prenant en compte des coûts de configuration de production variables dans un horizon de 9 périodes et comportant deux contraintes en termes de capacités dynamiques et de limites de ressources.

[Li \(2020\)](#) étudie le problème de dimensionnement de lots avec plusieurs items. Dans ce problème, il existe un ensemble de demandes sur un horizon

de planification de T périodes discrètes, et toutes les demandes doivent être satisfaites à temps. Le fait de transporter des stocks de temps en temps entraîne un coût de détention des stocks. Le but du problème est de trouver une solution réalisable qui minimise la somme des coûts de commande et de détention. Il résout le problème en donnant un algorithme d'approximation.

2.3.2.3 Problème de dimensionnement de lots avec reconditionnement

Le reconditionnement est l'ensemble des opérations de démontage et de récupération visant à réparer un produit afin qu'il soit équivalent au produit d'origine. [Guide Jr \(2000\)](#) présentent le schéma général des systèmes de reconditionnement, y compris le reconditionnement ou la réutilisation de composants provenant du démontage des produits retournés. [Pineyro & Viera \(2010\)](#) étudient un problème de dimensionnement de lots avec différents flux de demande d'articles neufs et reconditionnés, dans lesquels la demande de produits reconditionnés peut également être satisfaite par de nouveaux produits, mais pas l'inverse. Ils examinent le problème de la recherche des quantités reconditionnés d'un plan dans le cas multi-périodes. Ils fournissent un PLNE et l'ont résolu avec une recherche tabou.

En 2013, [Sahling \(2013\)](#) étudie le problème de dimensionnement de lots avec plusieurs items, avec la contrainte de capacité et avec reconditionnement. Pour résoudre ce problème, il propose un PLNE ainsi qu'une approche de génération de colonnes. En 2014, [Baki et al. \(2014\)](#) développent un PLNE pour résoudre le problème sur un horizon de 12 périodes ainsi qu'une heuristique basée sur la programmation dynamique. En 2015, [Sifaleras et al. \(2015\)](#) développent une méta-heuristique de type recherche à voisinage variable pour résoudre ce problème. Ils testent cette approche sur 6480 instances de la littérature (avec 12 périodes) et 108 instances nouvellement générées (avec 52 périodes).

D'autres travaux sont intéressés par l'étude d'un système hybride avec fabrication et reconditionnement, où un produit hybride est fabriqué avec des produits manufacturés et reconditionnés. Les produits manufacturés et reconditionnés sont produits sur des lignes distinctes, mais le produit hybride est utilisé pour répondre aux excédents fabriqués et intégrer les deux lignes distinctes. [Koken et al. \(2018\)](#) étudient le problème avec trois types de produits ; les produits manufacturés, les produits reconditionnés et les produits hybrides. Ils traitent à la fois le problème sans et avec la contrainte de capacité. L'objectif est d'identifier le nombre de produits manufacturés, reconditionnés et hybrides qui devraient être produits à chaque période de planification pour maximiser le profit total. Une heuristique basée sur un algorithme génétique

est proposée pour résoudre le problème.

Cunha et al. (2019) proposent une mat-heuristique pour résoudre le problème. Deux approches constructives sont proposées : une heuristique de génération de colonne suivie d'un mécanisme de réparation par programmation linéaire, et une procédure de recherche locale correctrice et optimisée est également appliquée pour améliorer la qualité des solutions obtenues par les techniques de construction.

Récemment, Devoto et al. (2020) proposent une formulation de la programmation mathématique et ils suggèrent plusieurs règles basées sur différentes décisions d'inspection et de reconditionnement. Ils montrent que le reconditionnement peut offrir des avantages économiques, même dans le cas d'une inspection indépendante et relativement coûteuse.

2.3.2.4 Les hypothèses sur la demande

Généralement, la demande est supposée déterministe et doit être satisfaite dans les délais. Dans de nombreux cas, la demande peut être reportée à des périodes ultérieures ou partiellement ou totalement perdues.

- (a) Accumulation de retards : Dans le problème de dimensionnement de lots, si l'accumulation de retards (Backlogging) est autorisée, un coût de rupture de stock est engagé pour chaque unité. L'accumulation de retard est considéré par Zangwill (1969). L'auteur analyse deux systèmes de production économiques de dimensionnement de lots dynamiques, le premier étant un modèle de produit unique avec retard (le niveau de stock d'une période peut devenir négatif) et le second un modèle multi-échelons. L'objectif est de minimiser les coûts totaux de production et d'inventaire. Il développe un algorithme de programmation dynamique pour le calcul du calendrier de production optimale. Un produit est fabriqué par lots puis placé en stock. Une fois le stock épuisé, un autre lot est créé, ce processus se répète autant de fois que nécessaire.
- (b) Vente perdu : Dans des cas particuliers, le détaillant ne peut pas satisfaire les demandes des clients dans une période donnée. Dans ce cas il doit décider de perdre la totalité ou une partie de demande à faible profit, d'où la possibilité de servir qu'une partie de la demande. Par exemple, Farhat et al. (2019) étudient le problème de dimensionnement de lots déterministe avec un seul item dans le cas où un détaillant signe un contrat de rachat avec un fournisseur qui lui fournit les quantités par lots. Ils considèrent un coût fixe par lot reconstitué en plus des coûts classiques de dimensionnement de lots et un coût de vente perdu pour chaque unité de demande non satisfaite.

2.3.2.5 Problème de dimensionnement de lots stochastique

Les articles sur le problème de dimensionnement de lots stochastique se concentrent principalement sur la demande comme source d'incertitude et peu d'entre eux considèrent l'aspect stochastique dans d'autres éléments tels que le coût, le rendement, le délai et la capacité.

Pour le problème des demandes stochastiques, la contrainte de niveau de service est l'une des méthodes utilisées par [Bookbinder & Tan \(1988\)](#). Les auteurs proposent une contrainte aléatoire pour modéliser une contrainte de niveau de service, et développent trois stratégies pour gérer le cadre résultant, qui sont l'incertitude statique, l'incertitude dynamique et l'incertitude statico-dynamique. Par la suite, [Tempelmeier & Herpers \(2011\)](#) utilisent la même méthode pour résoudre un problème de dimensionnement de lots dynamique à capacité multi-items sous des demandes aléatoires. [Rossi et al. \(2015\)](#) étudient le problème de dimensionnement de lots stochastique dans lequel la demande est supposée incertaine et non stationnaire. Ils proposent quatre modèles basés sur la stratégie d'incertitude statique-dynamique, c'est-à-dire la taille de lot stochastique, les contraintes de niveau de service, le schéma de coût de pénalité et les contraintes de niveau de service.

L'approche par scénarios est également utilisée pour traiter l'incertitude de la demande dans le cadre du problème de dimensionnement de lots stochastique. [Gutiérrez et al. \(2004\)](#) traitent le problème de dimensionnement de lots stochastique à un seul item où les coûts et la distribution de la demande dépendent du scénario sont pris en compte. Ils résolvent le problème en se basant sur une approche branch-and-bound. [Helber et al. \(2013\)](#) utilisent des scénarios de demande pour se rapprocher d'une version non linéaire du problème de dimensionnement de lots stochastique.

Les pannes de machines sont supposées stochastiques par [Nourelfath \(2011\)](#) pour le problème de dimensionnement de lots stochastique multi-périodes et multi-items. Le modèle proposé comprend un ensemble de contraintes pour assurer une probabilité minimale pour satisfaire le niveau de service à la clientèle, et il est résolu par une approche de solution en deux phases. [Witthayapraphakorn & Charnsethikul \(2019\)](#) étudient le problème de dimensionnement de lots avec un seul item et avec une demande incertaine impliquant un grand nombre de scénarios de simulations pour un nombre de périodes entre 5 et 20. Ils appliquent la décomposition de Benders avec programmation linéaire stochastique. [Gurkan & Tunc \(2021\)](#) considèrent le cas où les demandes périodiques sont des variables aléatoires variant dans le temps. L'objectif est de déterminer le plan de production à coût minimum attendu de manière à répondre aux demandes de période stochastiques sur l'horizon de planification. Ils proposent un PLNE ainsi qu'une heuristique fixe et optimise. [Hnaïen &](#)

Afsar (2017) proposent le problème de dimensionnement de lots stochastique robuste avec des délais incertains. Ils explorent les cas avec et sans contraintes de capacité et sans ventes perdu, par des scénarios discrets. Taş et al. (2019) formulent le problème de dimensionnement de lots avec des temps de réglage stochastiques et des heures supplémentaires comme un problème de programmation stochastique en deux étapes ainsi qu'une procédure d'approximation de moyenne d'échantillon. Dans le tableau 2.2 nous présentons un résumé sur le problème de dimensionnement de lots.

2.4 Problèmes de production et de transport intégrés

Les grandes catégories de problèmes de production et de transport intégrés sont présentées dans la figure 2.4. Le premier problème est la version intégrée de la planification de dimensionnement de lots et de la distribution avec des expéditions directes (Federgruen & Tzur (1999)). Ce problème détermine la taille des lots de la production sur un horizon de planification discret afin de minimiser la production, la configuration, les stocks et les coûts d'expédition directe. Ce problème peut également être considéré comme un problème de dimensionnement de lots à deux niveaux ou comme un problème d'entrepôt unique à plusieurs détaillants. Le deuxième problème est le problème de production et de tournées de véhicules intégré (Bard & Nananukul (2009)). Ce problème est une généralisation de problème de tournées de véhicules avec gestion des stocks (Andersson et al. (2010)), obtenu en considérant les décisions de production (c'est-à-dire les décisions relatives à la quantité de production et à la configuration). Ce dernier se réduit au problème de dimensionnement de lots intégré avec expédition directe en abandonnant l'aspect tournées de véhicules, et à un problème de tournées de véhicules avec gestion des stocks en négligeant l'aspect production.

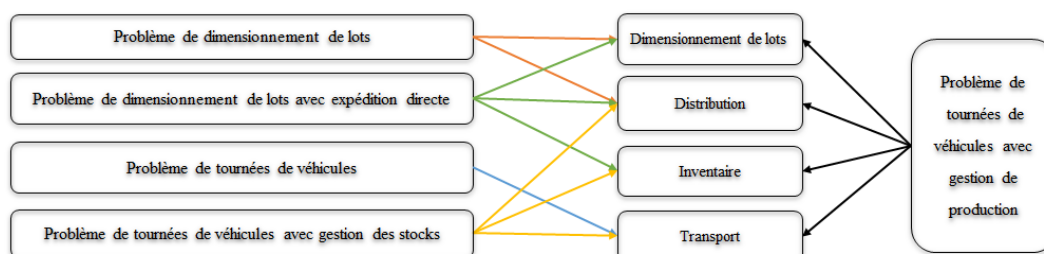


Figure 2.4 – Modèles de planification de la chaîne logistique

Tableau 2.2 – Synthèse des articles sur le problème de dimensionnement de lots

Références	Seul item	Avec la contrainte de capacité	Reconditionnement	Stochastique	Dynamique	Méthodes de résolution
Zangwill (1969)	*				*	Programmation dynamique
Pineyro & Viera (2010)		*				recherche tabou programmation en nombres entiers mixtes
Sahling (2013)		*				PLNE + Génération de colonnes
Baki et al. (2014)		*				PLNE, Programmation dynamique
Sifaleras et al. (2015)		*				Recherche à voisinage variable
Jiao et al. (2017a)	*			*		Heuristique
Hein & Almeder (2016)		*			*	Programmation génétique
Cunha et al. (2019)		*			*	Génération de colonnes
Carvalho & Nascimento (2018)	*					PLNE
Rapine et al. (2018)		*				Algorithme approximatif
Fiorotto et al. (2018)		*				PLNE
Koken et al. (2018)	*				*	Algorithme génétique
Rapine et al. (2018)	*				*	Heuristique, Programmation en nombres entiers mixtes
Zhou et al. (2018)		*				PLNE
Witthayapraphakorn & Charnsethikul (2019)	*			*		Décomposition de Benders
Farhat et al. (2019)		*				Heuristique
Cunha et al. (2019)	*			*		Mat-heuristique, Génération de colonne, recherche locale
Taş et al. (2019)	*			*		Procédure d'approximation de moyenne d'échantillon
Rodoplu et al. (2020)		*				PLNE, Fix-and-Relax
Suzanne et al. (2020)	*					Programmation dynamique
Li et al. (2020)		*				Algorithme d'approximation
Gurkan & Tunc (2021)	*			*		PLNE, Fix-and-Relax

2.4.1 Problèmes de production avec expédition directe

Dans le problème de la planification intégrée de la distribution par expédition directe et le dimensionnement de lots, la production peut être réalisée à chaque période et les produits sont directement transportés de l'usine de fabrication aux clients. Les produits peuvent être stockés à l'usine ou chez les clients, ce qui entraîne des coûts de stockage pour répondre à la demande. Les coûts de production, de lancement, de stockage et d'expédition sont minimisés sur l'horizon de planification. La planification intégrée de la production et de la distribution par expédition directe est étudiée par certains chercheurs où le coût de distribution est considéré comme un coût fixe ou une fonction de coût mixte. L'objectif est de minimiser les coûts de production, de préparation, de stockage et d'expédition directe sur l'horizon de planification. L'expédition directe est une politique de transport, elle consiste à expédier un produit directement du vendeur au client. La situation de coût fixe est réalisée lorsque les expéditions sont livrées par camion complet, tandis que le coût est variable lorsque les expéditions sont livrées par camion incomplet. Une fonction combinée intègre à la fois les aspects fixes et variables dans la structure des coûts.

Federgruen & Tzur (1999) étudient un cas particulier de problème de taille des lots et d'expédition directe. Ils considèrent le problème du multi-détaillant à un entrepôt unique avec une production et des véhicules sans contrainte de capacité. Ils examinent le cas des articles multiples et ils développent une heuristique de répartition du temps pour résoudre le problème. Rizk et al. (2006) décomposent le problème intégré en sous-problèmes liés à la taille des lots et à l'indépendance temporelle et ont mis au point une technique de relaxation lagrangienne pour obtenir des bornes inférieures.

Deux études (Li et al. (2004); Jiao et al. (2017b)) proposent des algorithmes de programmation dynamique pour résoudre le problème avec les structures de coûts de livraison par camion complet et incomplet. L'algorithme de programmation dynamique est également utilisé par Chand et al. (2007) et Jiao et al. (2017a) pour résoudre le problème de l'accumulation de retard et de la livraison fractionnée avec des restrictions de fenêtre de temps. Par la suite, Solyali & Süral (2012) développent une nouvelle formulation basée sur le modèle du transport combiné et du chemin le plus court et ils l'ont résolu en utilisant des approches heuristiques. Melo & Wolsey (2012) étudient le problème de production et de transport avec capacité à deux niveaux et ils proposent des heuristiques hybrides.

Récemment, Lmariouh et al. (2019) considèrent une étude de cas qui implique la production et la distribution d'eau en bouteille pour une usine à partir de laquelle toutes les demandes sont livrées en expéditions directes et

généralement par camion complet aux clients. Ils proposent un PLNE pour résoudre le problème de tournées de véhicules et de production multi-véhicules et multi-produits.

2.4.2 Problèmes de tournées de véhicules avec gestion des stocks

Le problème de tournées de véhicules avec gestion des stocks est une généralisation du problème de tournées de véhicules. Il combine la gestion des stocks chez le client et l'acheminement des véhicules sans tenir compte de la planification de la production à l'usine centrale. Il est largement étudié au cours des dernières années dans le secteur de transport et logistique. En outre, ce problème se pose dans les contextes où un système de stockage est géré par le fournisseur. Dans ce système, un fournisseur surveille les niveaux de stockage de ses clients et prend la décision de réapprovisionner les produits à ses clients. Ce système transfère les tâches de stockage et de commande des clients au fournisseur afin de réduire la charge de travail, de diminuer les coûts et d'obtenir une meilleure efficacité grâce à la collaboration au sein de la chaîne logistique.

Bell et al. (1983) sont les premiers auteurs qui introduisent le problème de tournées de véhicules avec gestion des stocks dans une étude sur la livraison du gaz, ils l'ont résolu avec une méthode de relaxation lagrangienne. La même étude est menée pour le problème avec une seule période (Berman & Wang (2006)) et multi-périodes (Mjirda et al. (2014)), à la fois avec plusieurs fournisseurs et une seule usine d'assemblage dans laquelle chaque fournisseur fournit différent type de pièce. Certaines études abordent le problèmes de tournées de véhicules avec gestion des stocks en juste à temps pour coordonner l'entrée de matériaux avec le taux de production (Satoglu & Sahin (2013)). Une étude de cas est envisagée dans ce contexte par Montagné et al. (2019). Ils étudient le problème de la collecte des déchets d'huile végétale régulièrement. Les déchets collectés sont traités et utilisés comme matière première pour de nouveaux produits. Comme le problèmes de tournées de véhicules avec gestion des stocks est un problème combinatoire complexe, plusieurs méta-heuristiques, par exemple, la recherche tabou (Rusdiansyah & Tsao (2005)), l'algorithme génétique (Jahangir et al. (2019)), la procédure de recherche adaptative randomisée gourmande (Savelsbergh & Song (2007)), l'heuristique hybride avec une recherche tabou combinée avec le PLNE (Archetti et al. (2011)), et la recherche adaptative à voisine large (Coelho et al. (2012)), la méta-heuristique d'optimisation d'essaim de particules (Widyadana & Irohara (2019)) sont proposées.

Comme mentionné dans Andersson et al. (2010), quelques algorithmes

exacts sont proposés pour résoudre les problèmes de tournées de véhicules avec gestion des stocks en raison de sa complexité. Archetti et al. (2007) proposent un algorithme de branch-and-cut pour résoudre les problèmes de tournées de véhicules avec gestion des stocks avec un seul véhicule avec une contrainte de capacité. Savelsbergh & Song (2008) envisagent le problème de tournées de véhicules avec gestion des stocks avec des mouvements continus et développent un algorithme de branch-and-cut pour résoudre le problème lorsque le réseau est constitué de plusieurs usines et de plusieurs clients. Solyali & Süral (2012) renforcent la formulation de problèmes de tournées de véhicules avec gestion des stocks avec la politique d'ordre à niveau de Archetti et al. (2007) en utilisant une reformulation du réseau de chemin le plus court.

Babagolzadeh et al. (2020) abordent les problèmes de tournées de véhicules avec gestion des stocks stochastique avec les coûts énergétiques et d'émission. Ils considèrent que la quantité livrée à chaque détaillant, la voie de distribution des produits et le coût de vente perdu dépendent des scénarios. Ils proposent une programmation stochastique à deux étapes et un mat-heuristique basé sur la recherche locale itérative et un PLNE pour résoudre le problème avec moins de 65 scénarios à partir d'une étude de cas réel. Gruler et al. (2020) proposent une approche hybride qui combine la méta-heuristique recherche à voisinage variable avec une simulation pour résoudre les problèmes de tournées de véhicules avec gestion des stocks stochastique avec une seule période et des exigences de rupture de stock.

2.4.3 Problèmes de tournées de véhicules avec gestion de production

Nous présentons ici le problème de production et tournées de véhicules qui combine les décisions relatives au dimensionnement de lots, la gestion des stocks et les tournées de véhicules. Le réseau de la chaîne logistique se compose d'une usine de production et de plusieurs détaillants, que nous pouvons considérer comme des clients de l'usine. L'usine et les détaillants disposent tous de leurs propres zones de stockage pour conserver les produits finis. La demande de chaque détaillant doit être satisfaite à chaque période de l'horizon de planification. À chaque période, l'usine doit décider de fabriquer ou non le produit et déterminer la quantité de la production correspondante. Si la production a lieu, ce processus engendre un coût de lancement fixe ainsi que des coûts de production unitaires. En outre, la taille des lots ne peut pas dépasser la capacité de production. Au cours de chaque période, les livraisons sont effectuées de l'usine aux détaillants par des véhicules équipés et des coûts d'acheminement sont encourus. Si les produits sont stockés à l'usine ou chez les détaillants, des coûts unitaires de stockage sont également encourus. L'op-

timisation de la distribution quotidienne est modélisée comme un problème du voyageur de commerce (Absi et al. (2018)) ou un problème de tournées de véhicules (Neves-Moreira et al. (2019)).

Chandra (1993) introduit le problème de tournées de véhicules avec gestion de production. Il étudie le problème de dimensionnement de lots avec un nombre illimité de véhicules avec capacité. Plus tard, le même auteur (Chandra et al. (1994)) définit le problème de tournées de véhicules avec gestion de production comme multi-produits où les demandes pour chaque produit sont connues à chaque période. Pour résoudre ce problème, l'auteur suggère que les opérations de fabrication utilisent le modèle de traitement par lots avec une capacité de ressources limitée. Il montre qu'il est possible de réaliser des économies de 3 à 20 % en résolvant le problème de tournées de véhicules avec gestion de production par rapport à la résolution séquentielle des problèmes séparés. Les différentes variantes de problème de tournées de véhicules avec gestion de production sont étudiées de manière approfondie en se concentrant sur le développement d'algorithmes efficaces.

On trouve ce problème avec une seule usine produisant un seul type de produit (Boudia & Prins (2009); Absi et al. (2015a)), ou bien à plusieurs types de produits (Brahimi & Aouam (2016)) aussi le problème avec plusieurs usines et à plusieurs produits (Zhang et al. (2017)). Quelques études sur ce problème envisagent une capacité de production et de stockage illimitée (Absi et al. (2018)). Plusieurs études abordent des importants aspects tels que les ventes perdues (Qiu et al. (2017)), les commandes en attente (Brahimi & Aouam (2016)), l'incertitude de la demande (Agra et al. (2018)), le tournées de véhicules riche (Miranda et al. (2018)), la logistique inverse (Qiu et al. (2018)) et les émissions de gaz à effet de serre (Qiu et al. (2017)).

Différents algorithmes pour résoudre le problème de tournées de véhicules avec gestion de production sont proposés, tels que les mat-heuristiques (Neves-Moreira et al. (2019)), et les méta-heuristiques comme la procédure de recherche adaptative randomisée gourmande (Boudia et al. (2007)), l'algorithme mémétique (Boudia & Prins (2009)), une recherche tabou (Armentano et al. (2011)), et une recherche adaptative à voisinage large (Bard & Nananukul (2009)). Absi et al. (2015a) proposent une méthode itérative en deux phases. Solyalı & Süral (2017) proposent une heuristique multi-phase pour le problème de tournées de véhicules avec gestion de production où les itinéraires sont calculés à l'avance et, dans un deuxième temps, les itinéraires prédéterminés sont utilisés pour simplifier les modèles de programmation en nombres entiers mixtes.

En raison de la complexité du problème, les approches exactes sont assez rares dans le contexte du problème de tournées de véhicules avec gestion de production. Solyalı & Süral (2009) développent une approche de relaxation

lagrangienne pour obtenir des limites inférieures basées sur la formulation de flux de produits multiples. [Bard & Nananukul \(2010\)](#) introduisent un cadre de branch-and-price.

[Adulyasak et al. \(2015\)](#) étudient le problème de tournées de véhicules avec gestion de production stochastique sous l'incertitude de la demande. Ils considèrent que les quantités de production, la quantité livrée au client, la quantité de demande non satisfaite chez le client dépendent des scénarios. Ils proposent une formulation en deux étapes, une approche de décomposition de benders et une méthode d'approximation de la moyenne des échantillons pour résoudre le problème avec 100, 500 ou 1000 scénarios. [Shuang et al. \(2019\)](#) considèrent que le nombre d'articles usés disponibles pour le ramassage chez le client et le nombre d'articles usés expédiés du client à l'installation de reconditionnement dépendent des scénarios. Ils proposent un problème de programmation stochastique à deux niveaux et génèrent trois scénarios.

2.4.4 Problèmes de tournées de véhicules et d'assemblage

Le problème de tournées de véhicules et d'assemblage réside dans la prise en compte particulière de la collecte de tous les composants nécessaires auprès de divers fournisseurs pour accomplir le processus d'assemblage. Ainsi, un composant manquant implique l'impossibilité de l'opération d'assemblage ([Kuhn & Liske \(2011\)](#)). Le problème de tournées de véhicules et d'assemblage vise à minimiser la somme des coûts tels que la production, les stocks et l'approvisionnement, sous réserve de plusieurs types de contraintes de capacité à un horizon temporel fini et discret ([Kuhn & Liske \(2014\)](#)). Certaines études portent sur la production en juste à temps, où le stockage du produit final ou des matières premières n'est pas autorisé dans l'usine ([Hein & Almeder \(2016\)](#)). Différentes versions de modèles mathématiques sont proposées. [Díaz-Madroño et al. \(2017\)](#) proposent un PLNE qui prend en compte différents modes d'expédition ; par camion complet et incomplet. [Chitsaz et al. \(2019\)](#) formulent le problème de tournées de véhicules et d'assemblage comme un PLNE et l'ont résolu par une mat-heuristique. [Chitsaz et al. \(2020\)](#) formulent ce problème avec un PLNE modèle résolu par un algorithme de branch-and-cut. Récemment, [Afsar & Hnaien \(2020\)](#) proposent une nouvelle formulation du problème dans laquelle ils étudient l'aspect dynamique de la demande.

Dans le tableau 2.3 nous présentons un résumé sur les travaux les plus proches à la problématique étudiée dans cette thèse.

Tableau 2.3 – Articles sur les problèmes d'assemblage et de désassemblage

Références	Assemblage		Désassemblage		Transport		Méthodes de résolutions
	multi-item	Capa	multi-item	Capa	Stocha	LTL	
Kim & Xirouchakis (2010)			*	*	*		Heuristique lagrangienne
Prakash et al. (2012)				*			Algorithme de recuit simulé basé sur les contraintes
Ullrich & Buscher (2013)				*			Heuristiques
Godichaud et al. (2015)				*			PLNE, algorithme génétique
Inderfurth et al. (2015)					*		Analyse empirique, mathématique
Hein & Almeder (2016)		*				*	PLNE
Hrouga et al. (2016)				*			Algorithme génétique, Fixe-and-optimize
Ji et al. (2016)			*	*			Heuristique lagrangienne
Díaz-Madroño et al. (2017)	*			*			PLNE
Liu & Zhang (2018)				*	*		Programmation non linéaire, Approximation externe
Tian & Zhang (2019)			*	*			Algorithme d'optimisation d'essai de particules
Habibi et al. (2019)			*	*	*	*	PLNE, mat-heuristique
Chitsaz et al. (2019)				*	*	*	PLNE, mat-heuristique
Chitsaz et al. (2020)		*			*	*	branch-and-cut
Slama et al. (2020)				*			PLNE
Afsar & Hnaien (2020)						*	PLNE, décomposition de Benders
Pour-Massahian-Tafti et al. (2020)				*			3 PLNE, 2 Heuristiques

2.4.5 Problèmes de tournées de véhicules stochastiques

Le problème de tournées de véhicules, est une généralisation du problème du voyageur de commerce. Ce dernier est formulé en 1959 par [Dantzig & Ramser \(1959\)](#) sous le nom de « Truck Dispatching Problem », il fait déjà l'objet d'études intensives pour le modéliser et le résoudre. Un problème de tournées de véhicules a pour objectif de construire un ensemble des tournées de coût minimal permettant à chaque commerçant (véhicule) de desservir un ensemble de clients tout en satisfaisant un certain nombre de contraintes. Dans ce problème, la flotte de véhicules est homogène et le nombre de clients (villes, magasins, entrepôts, machines de fabrication dans un atelier, etc.) est connue a priori. Le coût minimal peut être la distance totale parcourue, le temps de parcours, le nombre de véhicules utilisés, etc. De nombreuses variantes de problème de tournées de véhicules sont définies par extension de ce problème de base, en rajoutant différentes hypothèses, contraintes et/ou objectifs. Dans cette sous-section, nous nous concentrons sur le problème de tournées de véhicules avec l'incertitude de la demande ([Tillman \(1969\)](#)). Plusieurs modèles mathématiques et algorithmes différents sont proposés et explorés pour traiter le problème de tournées de véhicules avec des demandes stochastiques ([Gendreau et al. \(2016\)](#)). [Sungur et al. \(2008\)](#) introduisent une approche d'optimisation robuste pour résoudre les problèmes de tournées de véhicules. L'objectif est de minimiser les coûts de transport tout en satisfaisant toutes les demandes incertaines. [Lei et al. \(2011\)](#) formulent un modèle de recours stochastique à deux étapes et proposent une heuristique de recherche adaptative à voisinage large. [Lee et al. \(2012\)](#) étudient le problème de tournées de véhicules avec des demandes stochastiques avec deux types d'ensembles d'incertitude avec des paramètres ajustables pour les réalisations possibles du temps de déplacement et de la demande. Ils proposent une décomposition de type Dantzig-Wolfe pour résoudre le problème. [Marinaki & Marinakis \(2016\)](#) développent un algorithme basé sur une méta-heuristique d'optimisation d'essai de particules hybride avec la recherche à voisinage variable pour résoudre le problème de tournées de véhicules avec des demandes stochastiques.

Différents aspects du problème de tournées de véhicules avec des demandes stochastiques sont étudiées, comme les demandes stochastiques simple, avec des fenêtres de temps ou avec ramassage et livraison.

[Li et al. \(2010\)](#) étudient le problème avec des fenêtres de temps avec des temps de déplacement et de service stochastiques et introduisent un modèle à contraintes aléatoires et un modèle de recours stochastique. [Corso & Wallace \(2015\)](#) traite l'incertitude de la demande dans le problème de ramassage et de livraison en utilisant l'algorithme génétique. [Zhang et al. \(2016\)](#) étudient le problème avec des fenêtres de temps impliquant des demandes stochastiques

et ils proposent trois modèles probabilistes pour traiter la probabilité de livraison à temps tout en minimisant le coût total attendu. [Goodson et al. \(2017\)](#) résolvent le cas des véhicules multiples dans le cadre de l'approche de réoptimisation avec une méthode de programmation dynamique approximative. [Hu et al. \(2018\)](#) étudient le problème avec des fenêtres de temps sous l'angle de l'incertitude de la demande et du temps de trajet. Ils proposent un modèle d'optimisation robuste pour traiter les cas de petite taille et un algorithme à deux étapes basé sur une heuristique de recherche adaptative à voisinage variable modifiée pour traiter les cas de grande taille. [Florio et al. \(2020\)](#) étudient le cas de problème de tournées de véhicules avec des demandes stochastiques avec un seul véhicule et avec un repeuplement optimal comme processus de décision Markov. Un modèle d'attente et de vision est proposé et est utilisé dans une heuristique parallèle pour résoudre des instances de la littérature et des autres plus importantes avec jusqu'à 150 nœuds et avec une demande distribuée avec la loi de Poisson.

2.5 Conclusions

Dans ce chapitre nous avons passé en revue les différents problèmes en relation directe ou indirecte avec notre problématique. Nous nous concentrons sur les problèmes de planification qui intègrent les activités de production et de transport dans le flux direct et inverse de la chaîne logistique. Nous constatons que les problèmes de tournées de véhicules avec gestion de production et ses variantes attirent plus en plus l'attention des chercheurs au cours des dernières années. Cependant, le problèmes de tournées de véhicules et d'assemblage l'un des plus importants problèmes de la chaîne logistique a reçu moins d'attention. Nous constatons aussi que les opérations d'assemblage et de désassemblage dans le processus de production sont largement étudiés mais de façon séparée. Toutefois, à notre connaissance, le problème de tournées de véhicules et d'assemblage avec désassemblage des produits retournés n'est pas encore abordé. Motivée par les observations mentionnées ci-dessus, cette thèse est consacrée à l'étude de ce nouveau problème de planification qui regroupe les opérations d'assemblage et de désassemblage avec des tournées de véhicules dans les cas déterministe et stochastique.

Problème d'assemblage, désassemblage et des tournées de véhicules : le cas déterministe

3.1 Introduction

Pour rester compétitives, les entreprises de divers secteurs cherchent des moyens pour réduire les coûts dans divers domaines où la production reste la préoccupation centrale. Cependant, le transport doit être pris en compte en même temps que la production afin de minimiser le coût logistique total. C'est pour cela l'intégration des décisions de transport et de production est devenue de plus en plus importante depuis les années 70. Les chercheurs ont prouvé que la chaîne logistique était plus efficace et plus rentable en coordonnant ses étapes grâce au partage d'informations. En outre, si toutes les étapes de la chaîne logistique optimisent indépendamment leurs coûts, le coût total augmente en raison d'un manque de coordination.

L'intégration des différentes décisions liées à la production et le transport comme deux activités principales de la chaîne logistique attire l'attention de l'industrie et de chercheurs dans différents secteurs. Le problème de dimensionnement de lots, un des fameux problèmes du processus de production, combiné au problème de tournées de véhicules, représente un défi pour de nombreux chercheurs. [Hein & Almeder \(2016\)](#) et [Ruokokoski et al. \(2010\)](#) montrent les avantages des approches intégrées pour le problème de tournées de véhicules avec gestion de production par rapport aux approches séquentielles classiques. La planification séquentielle suppose que les décisions relatives aux tournées de véhicules et au production sont optimisées de manière indépendante. Ce problème se déploie en deux sous-problèmes : (i) le dimensionnement de lots et (ii) les tournées de véhicules. Les décisions conjointes ou intégrées consistent à résoudre un seul problème d'optimisation qui intègre les objectifs et les contraintes des deux problèmes dans un cadre d'optimisation unique.

Dans certains pays, le traitement des produits usagés est considéré par des nombreuses usines sous les pressions législatives afin de réduire les émissions

de pollution et la consommation de ressources naturelles. Ces produits usagés peuvent être réutilisés, recyclés ou désassemblés en composants. Les composants désassemblés peuvent être réutilisés comme pièces de rechange. Ces derniers peuvent être utilisés, avec des composants neufs, pour assurer la fabrication des nouveaux produits [Guide Jr \(2000\)](#). L'intégration des opérations d'assemblage et de désassemblage dans un système de production hybride est étudiée par [Mete et al. \(2018\)](#).

Dans ce travail, nous nous intéressons à la résolution d'un problème de production qui regroupe les opérations de désassemblage et d'assemblage pour la première fois avec des tournées de véhicules où des produits usés sont retournés par les clients. Nous considérons un centre de production où un ensemble des composants neufs et désassemblés sont convertis en produits finis. Nous traitons la version statique et dynamique du problème. La différence entre les deux versions se focalise dans les variations de valeurs des paramètres avec le temps. La problématique abordée dans ce chapitre se retrouve dans des entreprises spécialisées dans la fabrication de véhicules et de produits électroniques. Ces entreprises, telles que BMW, Dell et Samsung, produisent des téléphones cellulaires, des ordinateurs portables et des appareils électroménagers avec des composants neufs et usagés.

La figure 3.1 présente un exemple de système de production avec des opérations de désassemblage et d'assemblage. Dans ce qui suit, nous considérons une structure de produit simple où il faut une unité ($a = 1$) de chaque type de composant, neuf ou désassemblé à partir d'un produit retourné, pour fabriquer un produit final. Chaque composant, qu'il soit neuf ou désassemblé, a une taille b_i , et chaque produit a une taille ($b = 10$). La taille est prise en compte dans les capacités de stockage et des véhicules lors de la planification. Un produit désassemblé peut donner un composant réutilisable de type l_3 avec un rendement r_3 qui est égal à 1. Les composants dont le rendement est égal à 0 sont jetés.

Les contributions dans ce chapitre sont doubles :

1. Nous traitons un nouveau problème intégrant la planification de production et de problème de tournées de véhicules avec retours des produits usagés. Selon nos connaissances, le problème de production, qui combine les opérations d'assemblage et de désassemblage et le problème de tournées de véhicules, n'est pas étudié auparavant. Nous étudions les versions statiques et dynamiques de ce problème multi-période et nous fournissons un modèle de programmation linéaire en nombres entiers mixtes.
2. Nous développons une mat-heuristique efficace basée sur la programmation mathématique et un algorithme de recherche à voisinage variable

pour résoudre les versions statiques et dynamiques du problème. Nous évaluons expérimentalement ses performances, ainsi que les avantages de considérer les composants usés dans le processus de production. Nous prouvons aussi l'apport du modèle intégré par rapport à l'approche hiérarchique.

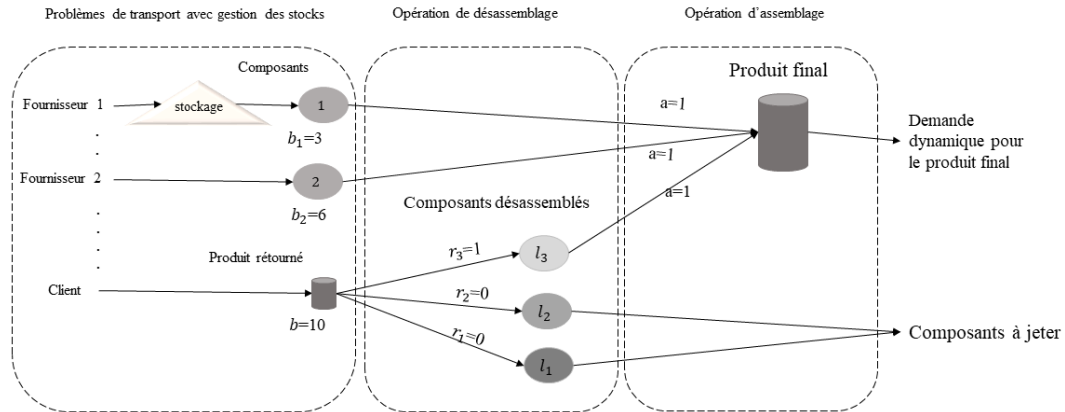


Figure 3.1 – Exemple d'une instance 2D-ARP-R

3.2 Modélisation mathématique

3.2.1 Description du problème

Le problème d'assemblage, désassemblage et des tournées de véhicules dynamique (2D-ARP-R) considéré dans ce chapitre est une combinaison de deux problèmes : le problème du dimensionnement de lots et le problème de tournées de véhicules. Ce problème considère un système d'assemblage des composants avec contrainte de capacité pour fabriquer un seul type de produit final pour satisfaire la demande extérieure, d_t , sans rupture de stock. Nous supposons qu'une unité de chaque composant est nécessaire pour fabriquer une unité de produit final. L'usine fournit un espace de stockage partagé avec une capacité L_0 pour les composants et les produits.

L'emplacement des fournisseurs et des clients ainsi que l'usine de production est modélisé comme des nœuds sur un graphe symétrique orienté $G = (N, A)$, avec l'ensemble de nœuds $N^+ = N \cup \{0\}$, où 0 représente l'usine, et l'ensemble d'arcs $A = \{(u, v) : u, v \in N^+, u \neq v\}$. Chaque arc (u, v) a un coût de transport $c_{uv} > 0$.

Il existe deux types de nœuds ; le premier est celui des fournisseurs qui approvisionnent les composants. Le second type est celui des clients, qui alimentent le produit en fin de vie à désassembler. Nous prenons en compte une

usine disposant d'une flotte de véhicules homogènes illimitée avec une capacité fixe Q , ramassant les composants et les produits retournés des fournisseurs et des clients représentés par l'ensemble $N = \{1, \dots, n\}$.

Un coût d'achat unitaire est associé pour les composants et les produits retournés à chaque période. Tous les produits retournés ont une structure à deux niveaux, ce qui signifie que les produits retournés et les composants désassemblés sont considérés comme des éléments de base et des éléments de feuilles, respectivement. L'horizon de planification comprend T périodes ($t \in \{1, \dots, |T|\}$) où la demande est connue mais dynamique dans le temps. Tous les coûts sont aussi dynamique sauf le transport. Une fonction bijective ψ définit le fournisseur d'un composant ($v = \psi(j)$).

Dans notre travail nous proposons un PLNE basée sur la formulation de bi-commodités pour éliminer les sous tours. [Finke \(1984\)](#) introduit cette formulation pour résoudre le problème du voyageur de commerce. Cette formule considère deux types de marchandises A et B. Une unité de la première marchandise (A), doit être livrée à chaque nœud et une unité de la seconde marchandise (B), doit être collectée à chaque nœud. Le voyageur commence sa tournée à partir du dépôt avec m unités de la marchandise A et 0 unité de la marchandise B . À chaque nœud de la tournée, il laisse une unité de A et ramasse une unité de B . À la fin de la tournée, le voyageur revient au dépôt avec 0 unité de produit A et m unités de B .

Plus tard, [Baldacci et al. \(2004\)](#) proposent un modèle de programmation en nombres entiers basée sur la formule de bi-commodités pour résoudre le problème de tournées de véhicules. La même formulation est utilisée par [Afsar & Hnaïen \(2020\)](#). Pour éliminer les sous tours, nous trouvons aussi la formule MTZ de [Miller et al. \(1960\)](#).

Le principe de la formule de bi-commodités est expliqué dans la figure 3.2. Dans cette figure un dépôt doit servir trois nœuds avec un véhicule dont la capacité est égale à 15. Les q_u présentent les demandes au niveaux des noeuds u .

Le $Flux_{(0;3)} = 14$ indique que le véhicule quitte le dépôt avec un chargement égale à la demande totale des trois clients. le $Flux_{(n+1;2)} = 15$ indique que le véhicule arrive vide au dépôt.

Dans notre modèle, la variable continue F_{uvt} présente la capacité occupée du véhicule sur l'arc $(u, v) \in A$ pour la période t avant la visite du nœud v .

Par contre, la variable continue F_{vut} présente la capacité restante du véhicule sur l'arc $(v, u) \in A$ pour la période t après la visite du nœud v . Les nœuds 0 et $n+1$ correspondent au dépôt.

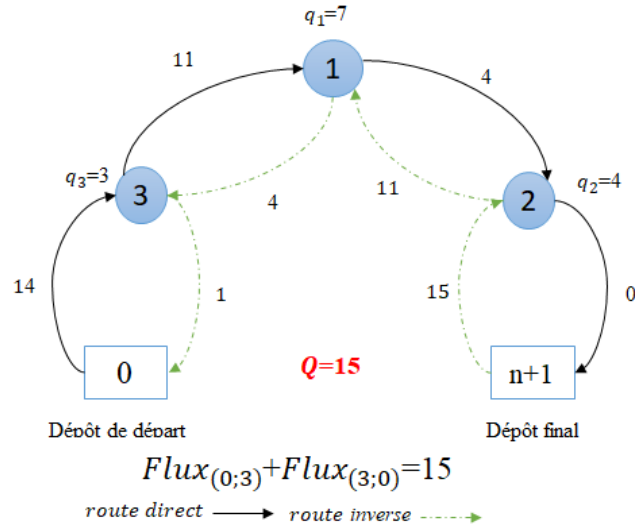


Figure 3.2 – Un exemple des chemins de flux pour un trajet de trois clients

3.2.2 Modèle mathématique

Les notations utilisées dans le modèle sont données dans le tableau 3.1.

Les termes de la fonction objectif sont expliqués comme suit :

- Coût de production de la période t

$$\pi_t = f_t p_t.$$

- Coût de lancement de la période t

$$\varphi_t = f'_t (y_t + y'_t).$$

- Coût d'achat pour les composants et le produit retourné de la période t

$$\rho_t = \sum_{i \in K} a_{it} q k_{\psi(i)t} + \sum_{u \in N} ar_t q_{ut}$$

- Coût de stocks à l'usine, sur la période t

$$\theta_t = \sum_{i \in K} h k_{it} I k_{it} + h r_t I r_t + h_t I_t.$$

- Coût de transport de la période t

$$\tau_t = \sum_{(u,v) \in A} c_{uv} x_{uvt}$$

Tableau 3.1 – Notation

Ensemble	
N^+	Ensemble de nœuds, $N^+ = \{0, \dots, n\}$, où 0 représente l'usine
N	Ensemble des fournisseurs et des clients, $N = N^+ \setminus \{0\}$
K	Ensemble de composants
T	Horizon de planification $t \in \{1, \dots, T \}$
A	Ensemble d'arcs, $A = \{(u, v) : u, v \in N^+, u \neq v\}$
Variables de décision	
p_t	Quantité de production pour la période t à l'usine
y_t'	Égal à 1 s'il y a une opération de désassemblage à l'usine pendant la période t , 0 sinon
y_t	Égal à 1 si l'usine produit pendant la période t , 0 sinon
Ir_t	Niveau de stock de produits retournés à l'usine à la fin de la période t
I_t	Niveau de stock d'articles à l'usine en fin de période t
Ik_{it}	Niveau de stock de composants à l'usine en fin de période t
q_{ut}	Quantité d'expédition du produit retourné du nœud $u \in N$ à l'usine pour la période t
qk_{ut}	Quantité de composants expédiés du nœud $u \in N$ à l'usine pour la période t
dis_t	Quantité de produits retournés pour le désassemblage à l'usine pendant la période t
x_{uvt}	Nombre de fois qu'un véhicule traverse l'arc $(u, v) \in A$ pour la période t
z_{ut}	Egal à 1 si le nœud $u \in N$ est visité dans la période t , 0 sinon
Paramètres	
d_t	Demande de produits finis à l'usine pendant la période t
f_t	Coûts de production unitaire pour les nouveaux articles pour la période t
f_t'	Coûts fixes de lancement pour la période t
a_{it}	Coût d'achat pour les composants pour la période t
ar_t	Coût d'achat pour les produits retournés pour la période t
h_t	Coût unitaire de possession des nouveaux articles à l'usine pour la période t
hr_t	Coût unitaire de possession du produit retourné à l'usine pendant la période t
hk_{it}	Coût unitaire de possession des composants à l'usine pour la période t
c_{uv}	Coût du transport entre les nœuds u et v , $(u, v) \in A$
σ_{ut}	Coût approximatif de la visite au nœud $u \in N$ pour la période t
C	Capacité de production
L_0	Capacité de stockage globale de l'usine
Q	Capacité de véhicules
S_{ut}	Quantité disponible du produit retourné au nœud $u \in N$ dans la période t
Sk_{ut}	Quantité disponible du composant au nœud $u \in N$ pour la période t
b_i	Taille unitaire du composant i
b	Taille unitaire du produit
r_i	Rendement de composant i par rapport à l'élément racine

Le problème 2D-ARP-R est donc modélisé :

$$\min \sum_{t \in T} (\pi_t + \varphi_t + \rho_t + \theta_t + \tau_t) \quad (3.1)$$

s.t

$$I_{t-1} + p_t = d_t + I_t \quad \forall t \in T \quad (3.2)$$

$$Ik_{i,t-1} + qk_{\psi(i)t} + r_i dis_t = p_t + Ik_{it} \quad \forall t \in T, \forall i \in K \quad (3.3)$$

$$Ir_{t-1} + \sum_{u \in N} q_{ut} = dis_t + Ir_t \quad \forall t \in T \quad (3.4)$$

$$p_t \leq Cy_t \quad \forall t \in T \quad (3.5)$$

$$dis_t \leq Cy'_t \quad \forall t \in T \quad (3.6)$$

$$dis_t + p_t \leq C \quad \forall t \in T \quad (3.7)$$

$$q_{ut} \leq S_{ut} \quad \forall t \in T, \forall u \in N \quad (3.8)$$

$$qk_{ut} \leq Sk_{ut} \quad \forall t \in T, \forall u \in N \quad (3.9)$$

$$\sum_{i \in K} b_i Ik_{it} + bI_t + bIr_t \leq L_0 \quad \forall t \in T \quad (3.10)$$

$$b_{\psi^{-1}(u)} qk_{ut} + bq_{ut} \leq Qz_{ut} \quad \forall t \in T, \forall u \in N \quad (3.11)$$

$$\sum_{u \in N^+ | u \neq v} x_{uvt} = \sum_{u \in N^+ | u \neq v} x_{vut} \quad \forall t \in T, \forall v \in N^+ \quad (3.12)$$

$$\sum_{u \in N^+} x_{uvt} = z_{ut} \quad \forall t \in T, \forall v \in N \quad (3.13)$$

$$F_{uvt} + F_{vut} = Qx_{uvt} \quad \forall t \in T, \forall u \in N^+, \forall v \in N^+ | u \neq v \quad (3.14)$$

$$\sum_{v \in N^+} F_{uvt} - \sum_{v \in N^+} F_{vut} = 2(bq_{ut} + b_{\psi^{-1}(u)} qk_{ut}) \quad \forall t \in T, \forall u \in N \quad (3.15)$$

$$\sum_{u \in N} F_{0ut} = \sum_{u \in N} b_{\psi^{-1}(u)} qk_{ut} + \sum_{u \in N} bq_{ut} \quad \forall t \in T \quad (3.16)$$

$$\sum_{u \in N} Qx_{0ut} - \left(\sum_{u \in N} b_{\psi^{-1}(u)} qk_{ut} + \sum_{u \in N} bq_{ut} \right) = \sum_{u \in N} F_{u0t} \quad \forall t \in T \quad (3.17)$$

$$\sum_{v \in N^+} F_{n+1vt} = Q \sum_{v \in N^+} x_{0vt} \quad \forall t \in T \quad (3.18)$$

$$p_t, dis_t \geq 0, z_{ut}, y_t, y'_t \in \{0, 1\} \quad \forall t \in T, \forall u \in N \quad (3.19)$$

$$Ik_{it}, Ir_{jt}, I_t \geq 0 \quad \forall t \in T, \forall i \in K \quad (3.20)$$

$$qk_{ut}, q_{ut} \geq 0 \quad \forall t \in T, \forall u \in N \quad (3.21)$$

$$F_{uvt} \geq 0 \quad \forall t \in T, \forall u \in N^+, \forall v \in N^+ \quad (3.22)$$

$$x_{uvt} \in \{0, 1\} \quad \forall t \in T, \forall u \in N^+, \forall v \in N^+ \mid u \neq v \quad (3.23)$$

La fonction objectif (3.1) minimise le coût total qui inclut les coûts de production, de lancement (à la fois les opérations de désassemblage et d'assemblage), d'achat, de stockage et de transport. Les coûts de stockage comprennent les composants et les produits retournés ainsi que les produits finis à l'usine. Les contraintes de (3.2) à (3.11) concernent le problème de dimensionnement de lots avec retours tandis que les contraintes de (3.12) et (3.18) définissent le problème du transport. Les contraintes (3.2) assurent le flux des stocks de produits finis fabriqués. Les contraintes (3.3) et (3.4) assurent l'équilibre des flux de composants (neufs ou démontés) et de produits retournés, respectivement. Les contraintes (3.5) et (3.6) fixent les variables de décision de réglage pour l'opération d'assemblage et de désassemblage, respectivement. Les contraintes (3.7) imposent une limite maximale aux opérations d'assemblage et de désassemblage.

Les contraintes (3.8) et (3.9) garantissent que chaque nœud visité peut fournir la quantité demandée de composants ou de produits retournés. Les contraintes (3.10) limitent la capacité de stockage du produit final, des composants et des produits retournés à l'usine. Les contraintes (3.11) imposent la limite de la quantité totale de composants et de produits retournés expédiés de chaque nœud à chaque période par la capacité du véhicule.

Les contraintes entre (3.12) et (3.18) présentent les contraintes d'élimination des sous tours. Ces contraintes garantissent que la solution ne contient aucun cycle.

Les contraintes (3.12) stipulent que le nombre d'arcs entrants est égal au nombre d'arcs sortants à chaque nœud. Les contraintes (3.13) impliquent que si un nœud u est visité, il n'y a qu'un seul arc (uv) qui est parcouru par un véhicule. Les contraintes (3.14) indiquent que la somme de la charge du véhicule et de la capacité restante est égale à la capacité totale du véhicule sur l'arc traversé. La variation totale de la charge du véhicule et de la capacité restante est égale à deux fois le volume de la quantité achetée à partir d'un nœud visité (contraintes (3.15)). Le chargement de la flotte des véhicules quittant le dépôt est égal au total des quantités achetées (contraintes (3.16)). Les contraintes (3.17) stipulent que la capacité restante de la flotte des véhicules entrant dans le dépôt est égal à la capacité totale de la flotte des véhicules moins le volume total des composants achetés. Les contraintes (3.18) indiquent que la capacité totale restante de la flotte des véhicules entrant dans le dépôt $(n + 1)$ est égale à la capacité totale des véhicules utilisés à partir de dépôt 0. Les contraintes (3.19)-(3.23) sont des contraintes qui définissent le type des variables des décisions.

3.3 Approche Mat-heuristique

Nous développons une mat-heuristique inspirée de la méthode de [Absi et al. \(2015b\)](#), qui est utilisée pour résoudre le problème de tournées de véhicules avec gestion de production. L'heuristique est composée de deux phases qui sont itérées jusqu'à ce qu'un critère d'arrêt soit satisfait. Dans la première phase, un problème de dimensionnement de lots est résolu avec des coûts de distribution approximatifs. Sur la base de ce plan de production, un plan de distribution est calculé, qui fournit la première solution. Cette solution est utilisée pour mettre à jour les coûts de distribution approximatifs et la procédure est itérée jusqu'à un nombre fini d'itérations.

L'heuristique itérative en deux phases est utilisée pour résoudre plusieurs problèmes de la littérature tels que le problème de tournées de véhicules et de désassemblage ([Habibi et al. \(2017\)](#)), et le problème de tournées de véhicules et d'assemblage ([Chitsaz et al. \(2019\)](#)).

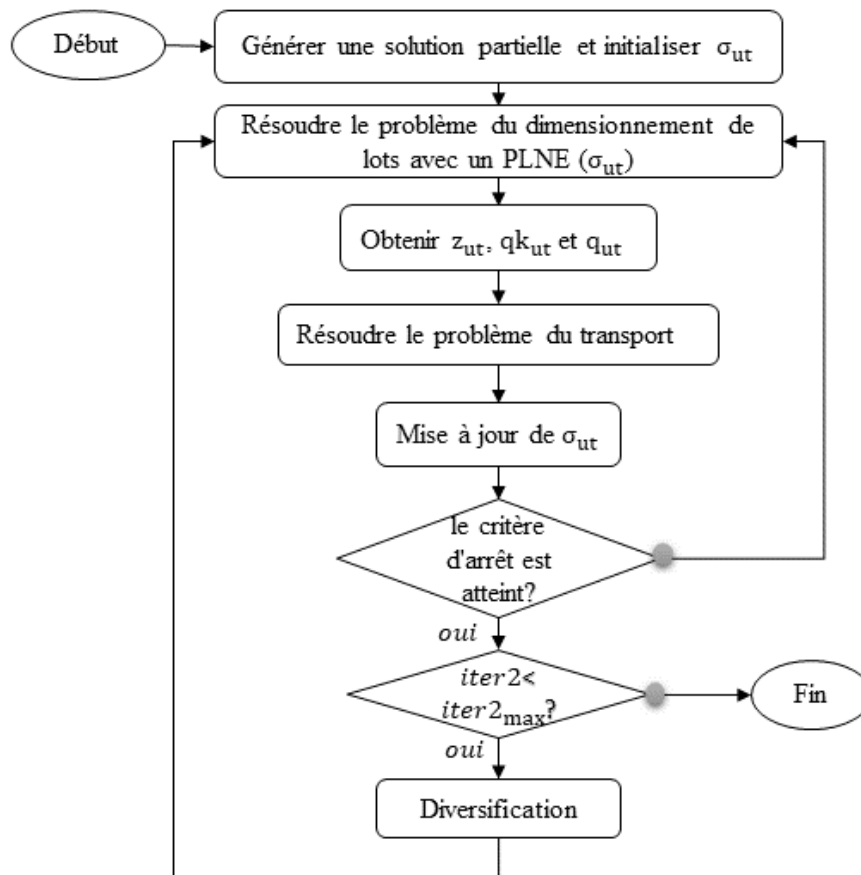


Figure 3.3 – Les étapes générales de notre mat-heuristique

La mat-heuristique proposée dans ce travail est basée sur le modèle de programmation linéaire en nombres entiers mixtes et l'algorithme de recherche à voisinage variable (Figure 3.3). Avant de commencer la résolution avec notre méthode, nous générons une solution partielle. Ensuite, nous initialisons la valeur de σ_{ut} en fonction de cette solution. σ_{ut} c'est le coût approximatif de visite (qui est mis à jour tout au long de l'algorithme) pour la collecte à partir du nœud u à la période t

La méthode proposée dans ce chapitre comporte deux étapes principales ; La première étape consiste à résoudre les problèmes de dimensionnement de lots et de dimensionnement de lots inverse. Il s'agit de traiter les décisions de $dis_t, p_t, I_t, Ir_t, Ik_t, q_{ut}, qk_{ut}$, et z_{ut} . En utilisant la valeur de z_{ut} , nous obtenons l'ensemble des nœuds visités dans chaque période. De plus, les valeurs q_{ut} et qk_{ut} révèlent la quantité expédiée qui forme chaque nœud. Pour les décisions de tournée dans cette phase, nous utilisons σ_{ut} . La deuxième étape, consiste à résoudre le problème de tournées de véhicules en appliquant l'algorithme de recherche à voisinage variable. Une fois que l'itinéraire de chaque période est construit, la valeur de σ_{ut} est mise à jour et la fonction objectif est recalculée.

La procédure itérative s'arrête lorsqu'un critère d'arrêt est atteint (après un nombre d'itérations sans amélioration). Un mécanisme de diversification est effectué et l'ensemble du schéma est répété jusqu'à ce qu'un nombre maximal d'itérations ($iter2_{max}$) soit atteint.

Dans les sous-sections suivantes, nous décrivons les différentes étapes du schéma mat-heuristique proposée.

3.3.1 PLNE pour les problèmes de dimensionnement de lots

Dans la phase de production, nous proposons un PLNE qui minimise la fonction objectif (3.24). L'objectif est de minimiser les coûts d'assemblage, de désassemblage et de stockage ainsi que les coûts (τ_t^0) liés à la visite de l'ensemble des clients ou des fournisseurs. Il conserve les mêmes contraintes que dans le modèle de base.

$$\min \sum_{t \in T} (\pi_t + \varphi_t + \rho_t + \theta_t + \tau_t^0) \quad (3.24)$$

avec $\tau_t^0 = \sum_{u \in A} \sigma_{ut} z_{ut}$.

Nous approximations le coût de transport de la fonction objectif en utilisant les variables de visite du nœud z_{ut} . Nous notons que σ_{ut} est un coût de visite approximatif utilisé dans l'étape de production. C'est le lien crucial entre les problèmes de production et de transport. En résolvant le plan de production avec le coût de visite approximatif, nous obtenons rapidement une première

solution réalisable. La valeur initiale de σ_{ut} est obtenue à partir de la solution partielle de l'itération initiale : D'abord, nous résolvons le problème de dimensionnement de lots sans transport, et nous obtenons le coût total de production. Deuxièmement, nous résolvons le problème de tournées de véhicules avec la recherche à voisinage variable et nous obtenons le coût total de transport. La somme du coût de production et de transport est une première borne supérieure du problème. Nous résolvons à plusieurs reprises la phase de production avec une approximation actualisée du coût de transport jusqu'à ce que le critère d'arrêt soit rempli.

3.3.2 Algorithme de recherche à voisinage variable

Le modèle fixe pour chaque période les décisions de visite du nœud z_{ut} et les quantités de collecte q_{ut} et qr_{ut} . Nous devons donc résoudre un problème de tournées de véhicules pour chaque période. Pour résoudre ce sous-problème, nous utilisons l'algorithme de recherche à voisinage variable (algorithme 1). Celui-ci est proposé pour la première fois par Mladenović & Hansen (1997) et il prend en compte un ensemble de voisinages k_{max} et exécute alternativement une procédure de recherche locale et une procédure de perturbation pour échapper à l'optimal local.

Algorithme 1 Algorithme de recherche à voisinage variable

```

1:  $sol$  = solution initiale à partir de l'heuristique constructive
2:  $sol_{Meilleure} = sol$ 
3: répéter
4:    $k = 1$ 
5:   tant que  $k < k_{max}$  faire
6:      $sol_1 = Perturbation(sol, k)$ 
7:      $sol_2 = Recherche\ locale(sol_1)$ 
8:     si  $f(sol_2) < f(sol)$  alors
9:        $sol_{Meilleure} = sol_2$ 
10:       $k = 1$ 
11:    sinon
12:       $k++$ 
13: jusqu'à  $iter > iter_{max}$ 

```

L'algorithme de recherche à voisinage variable commence par une solution initiale obtenue en appliquant une heuristique de construction aléatoire. À chaque itération, une solution aléatoire est générée à partir d'un voisinage actuel, puis une procédure de recherche locale est appliquée pour améliorer la solution. Si la nouvelle solution est meilleure que la meilleure solution actuelle

$sol_{Meilleure}$, la procédure est répétée en réinitialisant le voisinage, sinon en passant au voisinage suivant .

Cette approche est relativement facile à mettre en œuvre, ne contient pas un grand nombre de paramètres nécessitant des processus de réglage longs, et offre un excellent compromis en termes de qualité des solutions ainsi qu’en termes de temps de calcul (Frifita & Masmoudi (2020)). C’est pourquoi nous sommes motivés pour utiliser cet algorithme et nous avons l’intention de le combiner avec une programmation mathématique.

Quatre structures de voisinage sont utilisées à la fois dans les procédures de recherche locale et de perturbation (Frifita et al. (2020)) :

- Voisinage N_1 : Shift-Intra ; un nœud est déplacé de sa position actuelle à une autre position sur le même itinéraire.
- Voisinage N_2 : Swap-intra ; deux nœuds d’un même itinéraire sont échangés.
- Voisinage N_3 : Shift-Inter ; un nœud est supprimé d’un itinéraire à un autre.
- Voisinage N_4 : Swap-inter ; deux nœuds de deux itinéraires différents sont échangés.

3.3.3 Mise à jour des coûts approximatifs de transport

La mise à jour des coûts de visite approximatifs (Algorithme 2) consiste à modifier σ_{ut} selon la meilleur solution. Nous testons deux techniques pour mettre à jour σ_{ut} pour l’itération suivante.

1. Dans la première technique, nous supposons que u est un nœud visité dans un itinéraire r de période t obtenu en résolvant le problème de tournées de véhicules. Soit p le prédécesseur de u avec le coût c_{up} et s est son successeur avec le coût c_{us} . M_t est l’ensemble des nœuds $u \in N$ pour lesquels $z_{ut} = 1$. Si $u \in M_t$ alors $\sigma_{ut} = c_{up} + c_{us} - c_{ps}$, ce qui signifie que si un nœud u est éliminé de son itinéraire actuel, un itinéraire acceptable peut être obtenu en connectant les nœuds prédécesseur et successeur. Autrement, $\sigma_{ut} = \Delta_{ut}$ où Δ_{ut} serait le coût d’insertion le moins cher de u dans les itinéraires de véhicules disponibles dans la période t . Ainsi, en insérant le nœud u dans une certaine période t , un itinéraire acceptable peut être obtenu par la meilleure insertion parmi tous les itinéraires de cette période. Dans certaines périodes, aucun achat n’est nécessaire, donc aucun itinéraire n’est requis. Avec cette technique, nous imposons des visites aux fournisseurs dans ces périodes, ce qui peut changer les périodes de production. Nous pouvons donc dire que cette technique permet d’équilibrer le nombre de visites entre les périodes.

Cette procédure de mise à jour des coûts est également utilisée par [Absi et al. \(2015b\)](#).

2. La deuxième technique divise le coût total de transport de la période desservie entre les nœuds visités. Soit $cost_t$ le coût total de transport dans la période t , Nous définissons $\sigma_{ut} = cost_t(c_{0u} / \sum_{v \in M_t} c_{0v})$ si $u \in M_t$ sinon $\sigma_{ut} = (cost_t + \Delta_{ut})(c_{0u} / \sum_{v \in M_t \cup \{u\}} c_{0v})$.

Algorithme 2 Mise à jour de σ_{ut}

```

1: pour tous  $t \in T$  faire
2:   pour tous  $u \in N$  faire
3:     si  $iter1 < iert1_{max}$  alors
4:       si  $u \in M_t$  alors
5:          $\sigma_{ut} = c_{up} + c_{us} - c_{ps}$ 
6:       sinon
7:          $\sigma_{ut} = \Delta_{ut}$ 
8:       sinon
9:         si  $u \in M_t$  alors
10:           $\sigma_{ut} = cost_t(c_{0u} / \sum_{v \in M_t} c_{0v})$ 
11:        sinon
12:           $\sigma_{ut} = (cost_t + \Delta_{ut})(c_{0u} / \sum_{v \in M_t \cup \{u\}} c_{0v})$ 

```

Nous alternons périodiquement entre deux techniques de mise à jour. En conséquence, la plupart du temps, nous obtenons de meilleurs résultats que si nous utilisons ces deux techniques. Afin de trouver le coût réel, nous recalculons le coût global qui est la somme des coûts de production obtenus dans la première phase, plus le coût de transport trouvé dans la deuxième phase. Enfin, la meilleure solution est actualisée en fonction de cette nouvelle valeur de la fonction objectif.

3.3.4 Mécanismes de diversification

Pour éviter une convergence rapide vers la meilleure solution, deux mécanismes de diversification sont utilisés ([Absi et al. \(2015b\)](#)) : une procédure de démarrage multiple et un mécanisme de diversification de mise à jour. Ces deux mécanismes de diversification réinitialisent les coûts de visite σ_{ut} et relancent la procédure itérative. À chaque redémarrage d'itération, nous avons choisi au hasard l'un des deux mécanismes.

1. La procédure à plusieurs démarrages : réinitialise les coûts de visite approximatifs σ_{ut} par multiplication avec une valeur aléatoire $\varepsilon \in [0.5...1.5]$ ($\sigma_{ut} = (c_{0u} + c_{u0}) * \varepsilon$).

2. Le mécanisme de diversification des mises à jour modifie σ_{ut} selon la meilleure solution. Il consiste à multiplier σ_{ut} par le nombre de nœuds desservis M_t plus un (pour éviter la multiplication par zéro) $\sigma_{ut} = |M_t| \sigma_{ut} + 1$.

3.4 Résultats expérimentaux

La mat-heuristique est codée en Java. Le solveur Cplex 12.8 est utilisé pour résoudre les modèles mathématiques et obtenir des solutions optimales. Tous les tests sont effectués sur Intel (R) Core (TM) i7 2.9GHz avec 16GO de RAM.

3.4.1 Paramétrages

Les valeurs des paramètres du nombre d'itérations $iter1_{max}$ et $iter2_{max}$ sont fixées avec le test de Friedman en utilisant le logiciel Minitab.18 (voir tableaux 3.2 et 3.3).

Le test de Friedman ([Baddeley \(1966\)](#)) est un test statistique non paramétrique. Il classe d'abord les valeurs de chaque ligne de bas en haut. Chaque ligne est classée séparément. Il fait ensuite la somme des classements de chaque groupe. Si les sommes sont très différentes, la p-valeur sera faible.

Il existe deux hypothèses ;

- Hypothèse nulle H_0 : indique que les différents paramètres ont le même effet sur les résultats.
- Hypothèse alternative H_1 : les effets du traitement ne sont pas tous égaux à zéro) sont examinés.

Tableau 3.2 – Le test de Friedman pour fixer la valeur $iter1_{max}$

la valeur de $iter1_{max}$	Nombre d'instances	Somme des rangs
5	5	11.5
10	5	13
15	5	13
20	5	13
Globale	20	
Degrees of freedom (<i>d.f</i>)	3	
chi-square (corrected for ties)	16	
A.symp.Sig. (<i>p-value</i>)	0.003	

Tableau 3.3 – Le test de Friedman pour fixer la valeur $iter2_{max}$

la valeur de $iter2_{max}$	Nombre d'instances	Somme des rangs
5	5	16.5
10	5	14.5
15	5	10
20	5	9
Globale	20	
Degrees of freedom ($d.f$)	3	
chi-square (corrected for ties)	16.99	
A.symp.Sig. (p -value)	0.002	

Les résultats de l'analyse du test de Friedman rejettent H_0 avec un p -value = 0.003 pour le premier test et p -value = 0.002 pour le second.

La p -valeur des données étant inférieure au niveau de signification de 0.05, nous devons donc rejeter l'hypothèse nulle et conclure qu'au moins une des quatre valeurs du nombre d'itérations (pour les $iter1_{max}$ et $iter2_{max}$) a un effet significatif.

En outre, la valeur inférieure de la somme des rangs pour $iter1_{max}$ est égale à 11.5 et pour $iter2_{max}$ est égale à 9. Ces résultats indiquent que la valeur du nombre d'itérations est égale à 5 pour $iter1_{max}$ et à 20 pour $iter2_{max}$.

Le temps de résolution d'une instance par CPLEX est fixé à une heure (3600 s). Nous imposons une limite de temps de 60 seconds à la résolution du modèle PLNE lors de la mat-heuristique. La valeur maximale de l'itération VNS $iter_{max}$ est de 100. Il faut noter que la première itération de la mat-heuristique est équivalente au problème décentralisé (3.4.3.1).

3.4.2 Génération de données

Pour tester l'efficacité de la méthode proposée, nous générons un ensemble d'instances basées sur ceux de [Chitsaz et al. \(2019\)](#) qui est adapté des instances de problème de tournées de véhicules et de la gestion de production de [Archetti et al. \(2011\)](#). Les instances sont regroupées en trois catégories différentes en fonction du nombre de nœuds : Le premier groupe contient 14 nœuds (fournisseur ou client), le deuxième groupe contient 50 nœuds, et le troisième groupe contient 100 nœuds. Nous générons 200 instances pour le premier et le deuxième groupe, et 20 instances pour le troisième groupe. Nous définissons un horizon de 12 périodes. Les nouvelles instances conservent la capacité de production initiale et les distances entre les nœuds. Dans nos instances,

les produits de retour, les nouveaux produits et les composants partagent la capacité de stockage de l’usine .

- La demande dynamique par période est générée de manière aléatoire par une distribution uniforme entre $0,5 * D$ et $1,5 * D$ où D est la demande par période dans les instances de [Chitsaz et al. \(2019\)](#).
- Les coûts de lancement, d’achat, de stockage et de production sont générés de manière aléatoire par une distribution uniforme $\pm 30\%$ à partir des valeurs obtenues à partir des instances de [Chitsaz et al. \(2019\)](#).

Les instances utilisées dans les versions statique et dynamique restent inchangées, sauf pour les demandes et les coûts relatifs à chaque période qui modifient la version dynamique.

3.4.3 Résultats

Pour vérifier l’efficacité de l’algorithme proposé en termes de qualité de la solution, des expériences numériques sont mises en œuvre.

3.4.3.1 Contribution des décisions intégrées

Dans cette sous-section, nous comparons les résultats des décisions intégrées avec l’approche hiérarchique, nous pouvons dire aussi séquentielle. Ces résultats sont obtenus par l’exécution de la première itération de mat-heuristique. Cette première étape de mat-heuristique est considérée comme une approche séquentielle qui peut être utilisée entre les départements de production et de transport lorsque les données ne sont pas disponibles au niveau central.

Les tableaux 3.4 et 3.5 présentent les valeurs minimale (Min), maximale (Max), moyenne (Avg) et médiane (Med) de l’avantage du modèle intégré ($Bénéfice(\%) = \frac{sol.decent.model - sol.integ.model}{sol.integ.model}$). Pour la version statique, il peut atteindre un bénéfice maximal égal à 29.08% pour les petites instances, et 117.38 % pour les instances à 50 nœuds. Le bénéfice médian est égal à 7.24 % pour les petites instances et 13.35% pour les instances de taille moyenne. Dans la version dynamique, le bénéfice des décisions intégrées atteint 17.08 % pour les petites instances et 43.82 % pour les instances de taille moyenne. Le bénéfice médian est égal à 6.52% pour les petites instances et à 11.12% pour les instances de taille moyenne. Ces résultats nous permettent de confirmer que les décisions conjointes en matière de dimensionnement de lots et de transport présentent un grand avantage sur le coût total.

Nous présentons deux cas où les décisions de production et transport sont prises dans une manière consécutive (problèmes décentralisés) ; et le problème

Tableau 3.4 – L’avantage du modèle intégré pour la version statique

Bénéfice (%) pour 14 nœuds				Bénéfice (%) pour 50 nœuds			
Min	Max	Avg	Med	Min	Max	Avg	Med
0.90	29.08	9.57	7.24	0.28	117.38	25.79	13.35

Tableau 3.5 – L’avantage du modèle intégré pour la version dynamique

Bénéfice (%) pour 14 nœuds				Bénéfice (%) pour 50 nœuds			
Min	Max	Avg	Med	Min	Max	Avg	Med
0.21	17.08	6.77	6.52	3.26	43.82	14.53	11.12

de production et le problème de tournées de véhicules sont définis simultanément (problème intégré). Dans le tableau 3.5, l’exemple où le bénéfice maximal est égal à 17.08% est sélectionné et détaillé ci-dessous.

Le tableau 3.6 présente la solution pour les problèmes décentralisés avec 14 nœuds. Dans le problème de la production, nous avons une opération de production aux deuxième, sixième et neuvième périodes où les quantités d’achat de composants et de produits retournés sont effectuées aux mêmes périodes. Ainsi, nous avons le coût total du dimensionnement de lots (qui comprend les coûts de production, d’achat et de stockage) égal à $LC = 19673.78$, le coût total du transport égal à $TC = 5122.23$ et le coût total est égal à $LC + TC = 24796.01$.

Tableau 3.6 – La meilleure solution du modèle décentralisé

Périodes	Quantité de production	Coûts de production	Routes
2	48	336	0-4-13-10-5-6-7-8-14-1-12-3-9-0
6	41	746	0-11-2-13-10-5-6-7-8-14-4-1-12-3-9-0
9	43	1133	0-11-2-13-10-5-6-7-8-14-4-1-12-3-9-0

Le tableau 3.7 présente la solution optimale du même exemple, obtenue par le modèle intégré. Dans cette solution, le nombre des collectes est ramené à deux périodes. Nous pouvons observer que le transport a un impact sur la période de production (deux périodes au lieu de trois). En conséquence, les coûts de production augmentent au même temps, le coût de transport diminue. Nous avons donc un coût total plus faible de 21179.12. Le bénéfice de la coordination est de $(24796.01 - 21179.12)/21179.12 = 17.08\%$ par rapport au coût total. Dans l’approche décentralisée, le problème de production et le problème de transport sont résolus de manière consécutive. Par conséquent, nous obtenons la solution en un temps de résolution plus court que l’approche

Tableau 3.7 – La meilleure solution du modèle intégré

Périodes	Quantité de production	Coûts de production	Routes
2	56	392	0-4-13-10-5-6-7-8-14-1-12-3-9-0
7	76	1608	0-7-8-14-11-5-6-0

intégrée. Dans ce dernier cas, nous résolvons un seul problème d’optimisation qui incorpore les objectifs et les contraintes des deux problèmes dans un seul cadre d’optimisation. Les résultats montrent les avantages significatifs de la coordination, mais avec plus de temps de calcul.

3.4.3.2 Contribution des produits en retours

Le tableau 3.8 présente des résultats de PLNE pour 10 instances avec 14 nœuds avec et sans opération de désassemblage. La colonne *Obj* indique la valeur de la fonction objectif. La colonne *Cpu(s)* indique le temps de calcul en secondes. La colonne *Ecart(%)* indique l’écart d’optimalité donné par CPLEX. La colonne *Dev(%)* présente la déviation entre les meilleures solutions fournies par le PLNE avec et sans désassemblage.

Tableau 3.8 – Solutions du modèle intégré avec et sans désassemblage

Instance	Sans désassemblage			Avec désassemblage			Dev (%)
	Obj	Cpu(s)	Ecart(%)	Obj	Cpu(s)	Ecart(%)	
1	36849.98	17	0.0	36390.99	30	0.0	1.3
2	43679.00	11	0.0	43282.00	41	0.0	0.9
3	38538.93	226	0.0	37258.29	2465	0.0	3.4
4	41540.38	13	0.0	41252.38	70	0.0	0.7
5	27535.66	91	0.0	27038.66	3600	0.9	1.8
6	35418.92	154	0.0	34942.25	433	0.0	1.4
7	47110.55	5	0.0	46450.66	393	0.0	1.4
8	40193.42	19	0.0	40019.42	393	0.0	0.4
9	45953.27	1761	0.0	45162.27	617	0.0	1.8
10	45716.16	3600	0.7	45446.16	3600	1.2	0.6
Avg							1.4

D’après ce tableau, nous pouvons confirmer l’avantage de considérer les composants désassemblés dans le processus de production (Déviation moyenne = 1.4%). Toutefois, le traitement des produits en retours réduit les impacts négatifs sur l’environnement ainsi que le coût logistique total.

3.4.3.3 Analyse de la performance

Dans un premier temps, nous testons la stabilité de la méthode proposée. Ensuite, le comportement de calcul de notre algorithme sur différents types d’instances de test est rapporté.

Les tableaux de 3.12 à 3.19 donnent des résultats numériques détaillés pour chaque groupe d’instances (instances avec 14, 50 et 100 nœuds). Dans

ces tableaux, nous faisons une comparaison entre les résultats de PLNE et ceux de la mat-heuristique pour les versions statique et dynamique du problème. La colonne *Obj* indique la valeur de la fonction objectif. La colonne *BI* montre la limite inférieure donnée par CPLEX en 3600 secondes. La colonne *Cpu(s)* indique le temps de calcul en secondes. La colonne *Ecart1(%)* indique l'écart d'optimalité donné par CPLEX en 3600 secondes de temps de calcul. La colonne *Ecart(%)* présente l'écart d'optimalité de solution mat-heuristique par rapport à la borne inférieure ($Ecart(\%) = \frac{Meilleure.sol.mat - BI.PLNE}{BI.PLNE}$). La colonne *Dev(%)* présente la déviation entre la meilleure solution fournie par la mat-heuristique et la meilleure solution fournie par le PLNE ($Dev(\%) = \frac{Meilleure.sol.PLNE - Meilleure.sol.mat}{Meilleure.sol.mat}$).

Dans les tableaux 3.12 et 3.13, les résultats de 10 exécutions pour 10 instances avec 14 et 50 nœuds sont illustrés, respectivement. Les mesures suivantes sont prises en compte ; *Pire* : la pire valeur de la fonction objectif parmi les 10 exécutions, *Meilleure* : la meilleure valeur de la fonction objectif parmi les 10 exécutions, *Stb* : la valeur de stabilité qui est égale à $\frac{Pire - Meilleure}{Meilleure}$. En général, le taux de stabilité est égal à 0.00% pour les instances de petite taille (tableau 3.12) et le taux maximal de stabilité est égal à 1.10% pour les instances de taille moyenne (tableau 3.13). Les résultats montrent que la méthode proposée est stable.

Les tableaux 3.9, 3.14 et 3.15 montrent l'expérimentation sur un ensemble d'instances de petite taille. Pour la version statique, les résultats montrent que le PLNE peut prouver l'optimalité pour 17 instances sur 50 avec un temps moyen de résolution CPU égal à 2550.35 secondes. L'écart d'optimalité est égal à 0.73% en moyen. La mat-heuristique trouve 16 solutions optimales sur les 17 instances, ce qui améliore la solution de PLNE avec une déviation moyen de 0.06%.

Par rapport à la borne inférieure, l'écart d'optimalité pour la solution de la mat-heuristique est égal à 0.69% avec un temps de résolution CPU égal à 405.34 secondes en moyen. Pour la version dynamique, les résultats montrent que le PLNE trouve les solutions optimales pour 23/50 instances avec un temps moyen de résolution CPU égal à 2313.84 secondes. L'écart d'optimalité moyen est égal à 0.80%. La mat-heuristique trouve 17/23 solutions optimales, il améliore la solution de PLNE avec une déviation moyenne égal à 0.14%. Par rapport à la borne inférieure, l'écart d'optimalité moyen pour les solutions de la math-heuristique est égal à 0.67% avec un temps de résolution CPU égal à 351.34 secondes en moyen.

Tableau 3.9 – Résultats moyens pour 100 instances avec 14 nœuds

Statique					Dynamique				
PLNE		mat-heuristique			PLNE		mat-heuristique		
Cpu (s)	Ecart1 (%)	Cpu (s)	Dev (%)	Ecart (%)	Cpu (s)	Ecart1 (%)	Cpu (s)	Dev (%)	Ecart (%)
2550.35	0.73	405.34	0.06	0.69	2313.84	0.80	351.57	0.14	0.67

Les tableaux 3.10, 3.16 et 3.17 montrent les résultats de mat-heuristique et de PLNE sur un ensemble d'instances de taille moyenne (50 nœuds). Il faut noter que CPLEX est arrêté à une limite de temps égale à 3600 secondes. Pour les versions statique et dynamique, les résultats montrent que le PLNE ne peut trouver aucune solution optimale dans les 3600 secondes. La mat-heuristique améliore les solutions 37/50 et 45/50 trouvées par le PLNE avec une déviation moyenne égale à 2.86% et 1.67% pour la version statique et dynamique, respectivement. Par rapport à la borne inférieure, l'écart d'optimalité moyen pour les solutions de mat-heuristique est égal à 5.14% avec un temps moyen de résolution CPU égal à 976.77 secondes. Pour la version dynamique, l'écart d'optimalité moyen des solutions de la mat-heuristique par rapport à la borne inférieure est égal à 3.44% avec un temps de calcul moyen égal à 627.55 secondes.

Tableau 3.10 – Résultats moyens pour 100 instances avec 50 nœuds

Statique					Dynamique				
PLNE		mat-heuristique			PLNE		mat-heuristique		
Cpu (s)	Ecart1 (%)	Cpu (s)	Dev (%)	Ecart(%)	Cpu (s)	Ecart1 (%)	Cpu (s)	Dev (%)	Ecart(%)
3600	7.21	976.77	2.86	5.14	3600	4.79	627.55	1.67	3.44

Les tableaux 3.11, 3.18 et 3.19 présentent les résultats de l'analyse de la mat-heuristique et de PLNE sur un ensemble d'instances de grande taille. Pour les versions statique et dynamique, les résultats montrent que le PLNE ne peut pas trouver des solutions optimales dans un délai de 3600 secondes. Dans certains cas, 7/20 (version statique) et 4/20 (version dynamique), le CPLEX ne peut trouver aucune solution faisable. La mat-heuristique améliore toutes les solutions trouvées par le PLNE avec une déviation moyenne égal à 10.13% pour la version statique et 26.05% pour la version dynamique. Par rapport à la borne inférieure, l'écart d'optimalité moyen de la math-heuristique est égal à 4.01% et 6.74% pour les versions statique et dynamique respectivement.

Ces résultats nous permettent de confirmer l'efficacité de mat-heuristique, qui surpasse les résultats trouvés par CPLEX en termes de qualité de la solution et de temps de résolution CPU.

Tableau 3.11 – Résultats moyens pour 40 instances avec 100 nœuds

Statique					Dynamique				
PLNE		mat-heuristique			PLNE		mat-heuristique		
Cpu (s)	Ecart1 (%)	Cpu (s)	Dev (%)	Ecart(%)	Cpu (s)	Ecart1 (%)	Cpu (s)	Dev (%)	Ecart(%)
3600	18.92	1020.87	10.13	4.01	3600	19.91	957.67	26.05	6.74

3.5 Conclusion

Dans ce chapitre, nous avons étudié les versions statique et dynamique d'un problème de tournées de véhicules avec la gestion de la production. Ce problème traite pour la première fois les opérations d'assemblage et désassemblage avec des tournées de véhicules conjointement. Dans un premier temps, nous avons formulé ce problème comme un modèle de programmation linéaire en nombres entiers mixtes avec une formulation de bi-commodités. Les résultats montrent que le solver ne peut résoudre que des instances de petite et moyenne taille. Par conséquent, pour résoudre les grandes instances, une mat-heuristique efficace est développée.

Nous avons comparé les performances de l'approche intégrée avec celles de l'approche décentralisée. D'après les résultats, les décisions intégrées ont réduit considérablement les coûts totaux.

Les résultats de comparaisons entre les cas d'un système de production avec et sans considération des composants désassemblés ont montré une minimisation de coût total.

En outre, nous avons résolu les versions statiques et dynamiques du problème où l'approche mat-heuristique fournit des solutions de bonne qualité dans un temps de résolution CPU raisonnable.

Dans certains cas, la qualité de pièces désassemblées ne peut pas être connue à l'avance. Pour cette raison, nous allons prendre en compte l'incertitude des rendements dans le chapitre suivant.

Tableau 3.12 – Résultats de 10 exécutions pour la mat-heuristique pour les instances à 14 nœuds

	Instance 1		Instance 2		Instance 3		Instance 4		Instance 5		Instance 6		Instance 7		Instance 8		Instance 9		Instance 10	
	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)	Obj	Cpu (s)
1	36849.99	68.00	43449.94	47.92	38106.42	75.17	41540.38	56.03	27341.42	63.58	34959.57	52.86	46981.55	36.07	45871.42	71.44	45376.89	77.21	44265.49	28.90
2	36849.99	95.61	43449.94	47.60	38106.42	85.59	41646.82	24.10	27341.42	76.94	34959.57	24.72	46981.55	33.08	45871.42	52.78	45474.17	66.53	44265.49	18.42
3	36849.99	45.26	43449.94	70.05	38106.42	56.53	41540.38	25.10	27341.42	92.52	34959.57	48.75	46981.55	60.45	45871.42	23.69	45376.89	73.66	44265.49	21.33
4	36849.99	31.22	43449.94	14.45	38106.42	86.24	41540.38	30.49	27341.42	83.57	49.97	46981.55	71.79	45871.42	98.81	45376.89	49.84	44265.49	82.87	
5	36849.99	67.16	43449.94	67.07	38106.42	84.50	41646.82	95.34	27341.42	66.74	34959.57	35.04	46981.55	81.38	45871.42	41.21	45376.89	37.93	44265.49	21.98
6	36849.99	34.39	43449.94	89.62	38106.42	67.05	41540.38	50.04	27341.42	66.21	34959.57	32.77	46981.55	52.88	45871.42	84.21	45376.89	58.47	44265.49	84.03
7	36849.99	66.28	43449.94	38.37	38106.42	83.16	41540.38	44.47	27376.19	85.47	34959.57	25.19	46981.55	80.86	45871.42	23.96	45376.89	30.84	44265.49	54.23
8	36849.99	44.61	43449.94	50.35	38106.42	85.70	41540.38	33.54	27375.78	77.92	34959.57	32.57	46981.55	73.60	45871.42	23.47	45376.89	43.52	44265.49	24.56
9	36849.99	22.20	43449.94	79.12	38106.42	89.74	41646.82	61.11	27341.42	74.36	46981.55	45.64	46981.55	29.67	45871.42	50.80	45376.89	23.65	44265.49	86.85
10	36849.99	24.56	43449.94	72.20	38106.42	82.87	41540.38	43.90	27341.42	68.78	34959.57	42.38	46981.55	76.23	45871.42	39.68	45376.89	24.36	44265.49	19.44
Pire	36849.99	95.61	43449.94	89.62	38106.42	89.74	41646.82	95.34	27376.19	92.52	34959.57	52.86	46981.55	81.38	45871.42	98.81	45474.17	77.21	44265.49	86.85
Meilleure	36849.99	22.20	43449.94	14.45	38106.42	56.53	41540.38	24.10	27341.42	63.58	34959.57	24.72	46981.55	29.67	45871.42	23.47	45376.89	23.65	44265.49	18.42
Std	0.00		0.00		0.00		0.26		0.13		0.00		0.00		0.00		0.21		0.00	

Tableau 3.14 – Résultats pour la version statique avec 14 nœuds

Instances	PLNE				mat-heuristique			
	Obj	Cpu(s)	Ecart1(%)	BI	Obj	Cpu(s)	Dev(%)	Ecart(%)
1	29576.20	10.30	0.00	29576.20	29576.20	122.41	0.00	0.00
2	47775.18	393.02	0.00	47775.18	47775.18	66.14	0.00	0.00
3	45539.56	46.31	0.00	45539.56	45539.56	278.77	0.00	0.00
4	26927.77	731.85	0.00	26927.77	27009.02	267.77	-0.30	0.30
5	18063.54	52.05	0.00	18063.54	18063.54	933.02	0.00	0.00
6	23381.17	45.27	0.00	23381.17	23381.54	784.94	0.00	0.00
7	70807.36	5.35	0.00	70807.36	70807.36	390.96	0.00	0.00
8	21589.42	3600.40	2.40	21071.27	21347.80	257.40	1.13	1.31
9	39291.57	3602.40	0.30	39173.70	39271.85	621.27	0.05	0.25
10	40643.35	3602.00	1.30	40114.99	40369.46	706.42	0.68	0.63
11	20201.01	3601.99	1.50	19898.00	20019.10	132.35	0.91	0.61
12	29289.87	3606.91	2.90	28440.47	29100.51	205.93	0.65	2.32
13	24930.45	3606.76	0.80	24731.01	25003.06	140.63	-0.29	1.10
14	24504.42	3602.48	0.60	24357.39	24491.59	63.42	0.05	0.55
15	30798.66	3602.81	0.30	30706.27	30991.72	387.87	-0.62	0.93
16	40605.57	1605.71	0.00	40605.57	40605.57	338.63	0.00	0.00
17	17709.96	194.94	0.00	17709.96	17709.96	103.92	0.00	0.00
18	23505.50	1110.32	0.00	23505.50	23505.50	295.77	0.00	0.00
19	20321.36	605.53	0.00	20321.36	20321.36	58.31	0.00	0.00
20	22790.59	3602.24	0.90	22585.47	22780.05	887.44	0.05	0.86
21	30666.17	3605.58	1.10	30328.84	30632.39	142.40	0.11	1.00
22	15140.98	190.97	0.00	15140.98	15140.98	422.29	0.00	0.00
23	27319.98	3603.24	0.40	27210.70	27303.95	535.28	0.06	0.34
24	28343.12	3602.97	1.40	27946.32	28153.30	311.93	0.67	0.74
25	18892.49	3604.49	1.40	18627.99	18665.73	174.09	1.21	0.20
26	28005.10	3603.36	0.90	27753.06	28033.49	384.49	-0.10	1.01
27	22330.63	3603.56	0.90	22129.65	22142.25	178.77	0.85	0.06
28	12011.48	3603.12	1.70	11807.29	12011.48	337.61	0.00	1.73
29	5412.00	8.26	0.00	5412.00	5412.00	5.12	0.00	0.00
30	40737.66	3604.78	1.30	40208.07	40400.48	108.71	0.83	0.48
31	30744.17	3602.07	0.90	30467.47	31595.80	782.01	-2.70	3.70
32	20853.11	3602.17	0.30	20790.55	20799.95	558.19	0.26	0.05
33	30335.52	3606.03	0.90	30062.50	30335.52	641.90	0.00	0.91
34	32005.24	3601.03	2.00	31365.14	32026.92	435.40	-0.07	2.11
35	17308.12	889.18	0.00	17308.12	17346.10	272.66	-0.22	0.22
36	27232.34	3602.74	0.60	27068.94	27168.94	682.04	0.23	0.37
37	26708.01	3603.68	1.00	26440.93	26782.73	158.16	-0.28	1.29
38	30981.10	3601.44	1.40	30547.37	31022.24	858.04	-0.13	1.55
39	30981.10	2435.49	1.40	30547.37	30658.77	207.79	1.05	0.36
40	11310.48	3601.28	0.40	11265.24	11285.79	419.84	0.22	0.18
41	31768.34	3604.39	0.20	31704.80	31741.74	282.40	0.08	0.12
42	21198.47	3603.65	1.70	20838.10	21385.66	589.60	-0.88	2.63
43	19468.05	3606.47	1.10	19253.90	19483.97	742.28	-0.08	1.19
44	32171.72	3601.68	2.20	31463.94	32253.05	463.59	-0.25	2.51
45	24052.02	3601.48	1.70	23643.14	24082.97	160.92	-0.13	1.86
46	10440.91	357.61	0.00	10440.91	10440.91	1033.45	0.00	0.00
47	27187.08	3601.67	0.10	27159.89	27323.36	821.86	-0.50	0.60
48	15985.17	3601.08	0.60	15889.26	15928.59	334.80	0.36	0.25
49	30198.82	1965.40	0.00	30198.82	30198.82	632.01	0.00	0.00
50	32693.98	1570.06	0.00	32693.98	32693.98	545.93	0.00	0.00
Avg		2550.35	0.73			405.34	0.06	0.69

Tableau 3.15 – Résultats pour la version dynamique avec 14 nœuds

Instances	PLNE				mat-heuristique			
	Obj	Cpu(s)	Ecart1(%)	BI	Obj	Cpu(s)	Dev(%)	Ecart(%)
1	36849.99	21.70	0.00	36849.99	36849.99	190.59	0.00	0.00
2	43644.00	16.17	0.00	43644.00	43964.10	264.24	-0.73	0.73
3	38225.56	67.19	0.00	38225.56	38451.04	522.21	-0.59	0.59
4	41540.38	35.23	0.00	41540.38	41540.38	1034.80	0.00	0.00
5	27330.46	993.45	0.00	27330.46	27341.26	156.31	-0.04	0.04
6	35418.92	236.64	0.00	35418.92	35418.92	950.40	0.00	0.00
7	46981.55	5.68	0.00	46981.55	46981.55	685.33	0.00	0.00
8	40088.42	43.42	0.00	40088.42	40088.42	395.49	0.00	0.00
9	45880.22	3603.70	0.30	45742.58	45862.85	272.39	0.04	0.26
10	45752.09	3602.08	1.10	45248.82	45727.89	308.05	0.05	1.06
11	25480.07	3602.60	1.60	25072.39	25248.12	124.37	0.92	0.70
12	29269.80	3601.79	1.10	28947.84	29090.47	44.29	0.62	0.49
13	24823.82	3601.80	1.30	24501.11	24697.28	87.43	0.51	0.80
14	32189.28	3605.46	1.30	31770.82	31819.54	36.20	1.16	0.15
15	28114.29	3602.03	2.90	27298.97	27892.84	132.72	0.79	2.18
16	33839.10	571.87	0.00	33839.10	33839.10	195.68	0.00	0.00
17	25228.47	437.92	0.00	25228.47	25228.47	148.06	0.00	0.00
18	25187.42	1909.22	0.00	25187.42	25187.42	897.46	0.00	0.00
19	29939.65	3602.74	0.50	29789.95	29939.65	109.79	0.00	0.50
20	21179.12	2704.23	0.00	21179.12	21179.12	593.21	0.00	0.00
21	27661.56	1647.79	0.00	27661.56	27661.56	547.46	0.00	0.00
22	28143.09	43.12	0.00	28143.09	28143.09	396.33	0.00	0.00
23	27811.20	762.01	0.00	27811.20	27811.20	119.97	0.00	0.00
24	28550.61	368.31	0.00	28550.61	28550.61	100.11	0.00	0.00
25	26941.34	3602.52	1.10	26644.98	26952.35	154.77	-0.04	1.15
26	23553.86	3602.86	0.60	23412.54	23481.35	72.30	0.31	0.29
27	29247.03	2449.45	0.00	29247.03	29326.65	36.32	-0.27	0.27
28	23830.86	3605.10	2.70	23187.43	23396.68	92.19	1.86	0.90
29	25719.17	3603.05	1.30	25384.82	25458.72	693.18	1.02	0.29
30	30010.49	3601.07	3.20	29050.15	29754.45	67.78	0.86	2.42
31	18708.28	88.57	0.00	18708.28	18801.56	112.95	-0.50	0.50
32	21233.21	3602.61	0.40	21148.28	21343.76	138.31	-0.52	0.92
33	23680.09	3604.22	1.40	23348.57	23617.85	213.86	0.26	1.15
34	23175.75	3601.70	1.40	22851.29	22876.68	489.42	1.31	0.11
35	17493.47	943.64	0.00	17493.47	17493.47	661.62	0.00	0.00
36	24906.40	3024.26	0.00	24906.40	25006.56	266.70	-0.40	0.40
37	19277.14	1289.20	0.00	19277.14	19277.14	304.07	0.00	0.00
38	22682.79	3602.06	0.30	22614.74	22690.53	122.24	-0.03	0.34
39	24119.43	3602.72	1.70	23709.40	24101.47	556.88	0.07	1.65
40	23529.71	3604.72	2.80	22708.88	23377.40	920.04	0.65	2.21
41	23978.56	3600.90	1.30	23666.84	24004.15	112.35	-0.11	1.43
42	21263.46	3603.77	1.80	20880.71	21287.71	156.88	-0.11	1.95
43	20035.08	3602.24	2.00	19634.38	20064.80	148.22	-0.15	2.19
44	27973.63	3602.72	2.50	27274.29	28062.26	639.46	-0.32	2.89
45	23187.56	3603.13	2.00	22723.81	23038.59	714.48	0.65	1.39
46	22618.30	3604.97	1.40	22301.64	22716.76	606.41	-0.43	1.86
47	23968.49	519.16	0.00	23968.49	23968.49	418.27	0.00	0.00
48	17292.17	3601.58	1.30	17067.37	17286.39	734.58	0.03	1.28
49	26068.43	3603.85	0.50	25938.08	26056.60	116.93	0.05	0.46
50	22968.42	235.63	0.00	22968.42	22968.42	715.48	0.00	0.00
Avg		2313.84	0.80			351.57	0.14	0.67

Tableau 3.16 – Résultats pour la version statique avec 50 nœuds

Instances	PLNE			mat-heuristique			
	Obj	Ecart1(%)	BI	Obj	Cpu(s)	Dev(%)	Ecart(%)
1	70625.22	9.90	63633.32	69677.81	727.68	1.36	9.50
2	55244.31	27.80	39886.39	43800.98	1254.54	26.13	9.81
3	72303.70	31.10	49817.25	56613.88	958.24	27.71	13.64
4	72026.67	11.40	63815.63	72025.17	966.59	0.00	12.86
5	135605.68	7.30	125706.46	130655.66	921.80	3.79	3.94
6	73093.16	3.60	70461.81	73354.70	1399.60	-0.36	4.11
7	96229.33	5.70	90744.26	96141.42	929.96	0.09	5.95
8	107590.58	4.40	102856.60	107952.97	826.27	-0.34	4.95
9	77751.19	8.00	71531.10	77883.42	1384.18	-0.17	8.88
10	112447.58	9.20	102102.40	110765.93	1312.12	1.52	8.49
11	92797.30	10.60	82960.79	88984.05	986.05	4.29	7.26
12	114276.03	3.80	109933.54	110750.24	791.53	3.18	0.74
13	57683.89	14.20	49492.77	54251.90	1043.28	6.33	9.62
14	62558.47	6.80	58304.50	59414.00	1120.61	5.29	1.90
15	58430.76	6.30	54749.63	56649.64	1021.04	3.14	3.47
16	134501.76	1.60	132349.73	133651.14	1063.66	0.64	0.98
17	86450.52	3.70	83251.85	87215.36	303.38	-0.88	4.76
18	65455.12	9.30	59367.80	61513.93	959.08	6.41	3.61
19	123019.45	2.90	119451.89	120591.80	207.44	2.01	0.95
20	89076.21	5.90	83820.71	86516.19	1108.81	2.96	3.22
21	62598.90	5.60	59093.36	62943.79	1048.80	-0.55	6.52
22	82513.91	4.20	79048.32	80332.74	750.11	2.72	1.62
23	79879.58	3.70	76924.04	78515.05	1015.80	1.74	2.07
24	116806.17	5.00	110965.86	117356.80	1094.47	-0.47	5.76
25	59769.62	3.10	57916.76	58518.87	1017.90	2.14	1.04
26	78355.16	7.80	72243.46	77897.49	917.65	0.59	7.83
27	104242.09	3.30	100802.10	101749.34	460.22	2.45	0.94
28	119164.84	5.20	112968.27	118337.06	772.91	0.70	4.75
29	107775.98	8.40	98722.80	108416.31	819.33	-0.59	9.82
30	152232.76	11.90	134117.06	146713.27	1312.55	3.76	9.39
31	37190.94	2.30	36335.55	37511.34	1194.31	-0.85	3.24
32	88288.49	4.30	84492.09	88788.28	1179.99	-0.56	5.08
33	66200.04	4.80	63022.44	66726.59	938.67	-0.79	5.88
34	83134.12	6.50	77730.40	80539.39	954.95	3.22	3.61
35	66299.06	8.70	60531.04	63145.16	938.95	4.99	4.32
36	40186.11	5.20	38096.43	40708.06	980.58	-1.28	6.86
37	48789.87	4.20	46740.69	47219.27	1036.52	3.33	1.02
38	31417.59	4.20	30098.05	31050.72	976.95	1.18	3.17
39	91160.52	12.90	79400.81	86707.00	1049.20	5.14	9.20
40	58850.32	5.40	55672.40	57609.68	1113.23	2.15	3.48
41	68070.40	9.90	61331.43	63562.27	1070.83	7.09	3.64
42	79750.04	3.10	77277.79	78105.46	904.54	2.11	1.07
43	71532.84	8.00	65810.21	70928.54	1097.82	0.85	7.78
44	97065.60	7.60	89688.61	96766.22	985.57	0.31	7.89
45	32085.19	5.70	30256.33	31549.73	800.66	1.70	4.27
46	80587.55	4.80	76719.34	79596.54	946.98	1.25	3.75
47	85596.08	4.40	81829.86	85780.36	991.87	-0.21	4.83
48	64281.26	3.70	61902.85	63769.41	1077.42	0.80	3.02
49	52497.72	6.50	49085.37	51759.31	1080.57	1.43	5.45
50	105513.50	6.40	98760.63	99845.42	1023.26	5.68	1.10
Avg		7.21			976.77	2.86	5.14

Tableau 3.17 – Résultats pour la version dynamique avec 50 nœuds

Instances	PLNE			mat-heuristique			
	Obj	Ecart1(%)	BI	Obj	Cpu(s)	Dev(%)	Ecart(%)
1	73467.44	8.85	66965.57	70113.85	780.58	4.78	4.70
2	62820.22	9.40	56915.12	60023.80	769.66	4.66	5.46
3	58017.86	10.40	51984.01	56630.70	803.73	2.45	8.94
4	70832.20	6.70	66086.44	70189.93	573.13	0.92	6.21
5	64734.31	10.30	58066.68	60688.94	794.68	6.67	4.52
6	93597.36	3.70	90134.26	90356.19	629.16	3.59	0.25
7	66498.49	4.00	63838.55	65669.16	644.00	1.26	2.87
8	98346.16	5.20	93232.16	96110.39	663.09	2.33	3.09
9	105775.30	5.70	99746.11	100236.41	682.10	5.53	0.49
10	76697.65	2.20	75010.30	75558.44	613.73	1.51	0.73
11	76917.04	9.10	69917.59	75861.62	760.40	1.39	8.50
12	86827.29	1.60	85438.06	86197.92	504.40	0.73	0.89
13	68197.22	2.50	66492.29	67321.79	691.97	1.30	1.25
14	79622.65	4.50	76039.63	79697.57	632.88	-0.09	4.81
15	77020.75	6.00	72399.50	76292.39	593.57	0.95	5.38
16	119644.29	2.00	117251.41	118401.24	828.14	1.05	0.98
17	96897.28	4.10	92924.49	97670.42	640.12	-0.79	5.11
18	84527.78	4.20	80977.61	83769.69	648.14	0.90	3.45
19	90002.05	3.20	87121.98	88799.85	588.82	1.35	1.93
20	92887.11	3.70	89450.29	90674.62	590.70	2.44	1.37
21	95103.36	5.40	89967.78	95054.16	763.93	0.05	5.65
22	88982.81	2.70	86580.28	88527.06	372.51	0.51	2.25
23	79571.44	2.80	77343.44	79811.78	602.09	-0.30	3.19
24	100050.99	2.80	97249.56	100007.07	437.82	0.04	2.84
25	97466.35	6.10	91520.90	93124.33	688.65	4.66	1.75
26	88664.26	5.00	84231.05	88694.68	710.12	-0.03	5.30
27	79738.40	2.90	77425.98	79582.11	369.75	0.20	2.78
28	94209.00	3.90	90534.85	93032.35	708.43	1.26	2.76
29	109855.67	8.10	100957.36	108315.69	652.30	1.42	7.29
30	88721.65	4.00	85172.79	87618.87	560.29	1.26	2.87
31	68234.07	2.90	66255.28	68311.92	768.34	-0.11	3.10
32	71740.14	2.30	70090.12	70210.71	599.76	2.18	0.17
33	41764.81	3.60	40261.28	41133.77	725.21	1.53	2.17
34	61550.29	3.70	59272.93	61340.68	556.73	0.34	3.49
35	52821.95	6.20	49546.99	50353.89	798.18	4.90	1.63
36	59836.27	2.60	58280.53	59618.19	672.08	0.37	2.30
37	47649.08	2.10	46648.45	46926.71	598.84	1.54	0.60
38	52150.74	4.10	50012.56	50355.40	728.82	3.57	0.69
39	60393.03	4.20	57856.52	59850.16	739.81	0.91	3.45
40	51685.85	5.10	49049.87	50425.57	248.25	2.50	2.80
41	89731.24	17.90	73669.35	88731.63	348.25	1.13	20.45
42	59223.94	3.30	57269.55	58413.19	853.36	1.39	2.00
43	56443.74	4.80	53734.44	54336.92	752.36	3.88	1.12
44	77498.90	6.00	72848.97	77143.31	453.69	0.46	5.89
45	46717.61	8.70	42653.18	46123.30	852.36	1.29	8.14
46	65603.78	2.60	63898.08	64136.87	751.36	2.29	0.37
47	62057.01	2.50	60505.58	61236.60	632.25	1.34	1.21
48	53033.05	3.20	51335.99	52356.34	456.36	1.29	1.99
49	54429.70	9.50	49258.88	50126.94	789.46	8.58	1.76
50	79510.41	4.70	75773.43	78236.41	635.28	1.63	3.25
Avg		4.79			627.55	1.67	3.44

Tableau 3.18 – Résultats pour la version statique avec 100 nœuds

Instances	PLNE			mat-heuristique			
	Obj	Ecart1(%)	BI	Obj	Cpu(s)	Dev(%)	Ecart(%)
1	137186.37	17.30	113453.13	119767.81	1073.75	14.54	5.57
2	-	-	-	90083.35	1097.18	-	-
3	52916.91	22.30	41116.44	43539.89	927.85	21.54	5.89
4	-	-	-	83794.78	1065.64	-	-
5	67139.75	7.40	62171.41	65661.01	970.94	2.25	-
6	48957.00	16.60	40830.14	42626.69	897.09	14.85	4.40
7	62965.85	14.80	53646.91	58696.60	1227.04	7.27	-
8	-	-	-	83700.16	1106.28	-	-
9	49925.22	17.20	41338.09	44975.91	878.26	11.00	8.80
10	-	-	-	38091.87	3642.94	-	-
11	-	-	-	182272.86	710.22	-	-
12	-	-	-	171640.26	763.48	-	-
13	-	-	-	200135.47	799.65	-	-
14	158671.62	17.60	130745.42	139304.35	688.74	13.90	6.55
15	151445.02	31.40	103891.28	116442.18	787.82	30.06	12.08
16	153777.67	19.80	123329.69	131327.58	705.04	17.09	6.48
17	247484.65	31.10	170516.92	189645.52	737.84	30.50	11.22
18	202739.24	12.80	176788.62	189566.16	744.52	6.95	7.23
19	261016.82	21.30	205420.24	217013.76	819.17	20.28	5.64
20	95602.39	16.30	80019.20	85082.01	773.89	12.36	6.33
Avg		18.92			1020.87	10.13	4.01

(-) : Aucune solution réalisable n'est disponible

Tableau 3.19 – Résultats pour la version dynamique avec 100 nœuds

Instances	PLNE			mat-heuristique			
	Obj	Ecart1(%)	BI	Obj	Cpu(s)	Dev(%)	Ecart(%)
1	84728.18	3.40	81847.42	82521.15	1186.00	2.67	0.82
2	94354.73	9.90	85013.61	89951.56	1384.78	4.90	5.81
3	68751.46	9.60	62151.32	63159.95	1328.13	8.85	1.62
4	-	-	-	107307.86	1078.29	-	-
5	-	-	-	91160.41	1323.10	-	-
6	76417.78	7.00	71068.53	75858.55	1277.66	0.74	6.74
7	230943.45	67.80	74363.79	88335.96	1338.23	161.44	18.79
8	-	-	-	85603.10	1079.87	-	-
9	85541.52	13.40	74078.95	83927.43	1276.74	1.92	13.29
10	-	-	-	80166.85	1316.87	-	-
11	187005.193	19.80	149978.16	160164.44	550.67	16.76	6.79
12	135363.699	14.30	116006.69	128077.27	617.36	5.69	10.41
13	192886.67	10.70	172247.80	182054.21	515.12	5.95	5.69
14	177863.082	12.10	156341.65	168076.93	658.54	5.82	7.51
15	217759.426	15.30	184442.23	200240.37	803.80	8.75	8.57
16	160886.026	12.40	140936.16	147198.68	768.72	9.30	4.44
17	481952.921	71.00	139766.35	186309.87	661.69	158.68	33.30
18	150501.711	8.70	137408.06	142967.77	774.93	5.27	4.05
19	156262.539	6.70	145792.95	150178.53	535.70	4.05	3.01
20	294650.832	56.30	128762.41	133749.58	677.30	120.30	3.87
Avg		19.91			957.67	26.05	6.74

(-) : Aucune solution réalisable n'est disponible

Problème d'assemblage, désassemblage et des tournées de véhicules : le cas stochastique

4.1 Introduction

Dans ce chapitre, nous abordons le problème d'assemblage, désassemblage et des tournées de véhicules avec retours (2D-ARP-R), dans lequel les rendements (r_i) sont incertains. Le 2D-ARP-R qui est déjà traité dans le chapitre précédent, concerne la synchronisation de différentes activités telles que la production, la gestion des stocks et les tournées de véhicules. Le processus de production permet la fabrication de mono-produit. Ces produits sont assemblés à l'aide d'un ensemble de composants neufs et désassemblés, qui sont obtenus auprès des fournisseurs ou désassemblés à partir de produits usagés. Les produits usagés, à chaque période, sont situés en quantité limitée dans des entrepôts. En raison de leur état d'usage, certains de ces composants ne sont pas récupérables et doivent être éliminés lors du désassemblage. Pour refléter les variations de la qualité des composants utilisés, le rendement du processus de désassemblage est supposé incertain. Cette incertitude est influencée par de nombreux facteurs tels que la mauvaise qualité des produits retournés ou les dommages subis lors de l'opération de désassemblage.

Nous utilisons la méthodologie d'optimisation robuste minmax introduite par [Ben-Tal & Nemirovski \(1998\)](#) pour formuler le 2D-ARP-R. Différents scénarios sont générés pour traiter l'incertitude des rendements. Dans certains cas, les décideurs peuvent être plus préoccupés par le pire scénario que par le moyen. Pour cette raison, nous étudions le pire des cas pour traiter les incertitudes. L'objectif est de minimiser le coût total du pire scénario, qui comprend les coûts de production des opérations d'assemblage et de désassemblage, les ventes perdues, les stocks et les coûts de tournées de véhicules. À notre connaissance, aucune recherche existante n'a abordé ce problème.

La prise en compte de l'incertitude dans les décisions intégrées peut affecter le processus de production et d'approvisionnement. Pour cette raison, différents niveaux de flexibilité sont accordés aux différentes décisions liées à

ces deux activités de la chaîne logistique qui peuvent ou non être influencées par les scénarios.

Toutefois, le système pourrait ne pas être en mesure de satisfaire la demande en temps voulu en raison de l'incertitude sur les quantités des composants. Dans cette situation, la demande correspondante est perdue, ce qui entraîne un coût de pénalité élevé.

La figure 4.1 présente un exemple de cas déterministe du problème étudié. Dans cet exemple, les r_i sont supposés être connus, mais dans le cas stochastique, il peut prendre une valeur binaire de 0 ou 1.

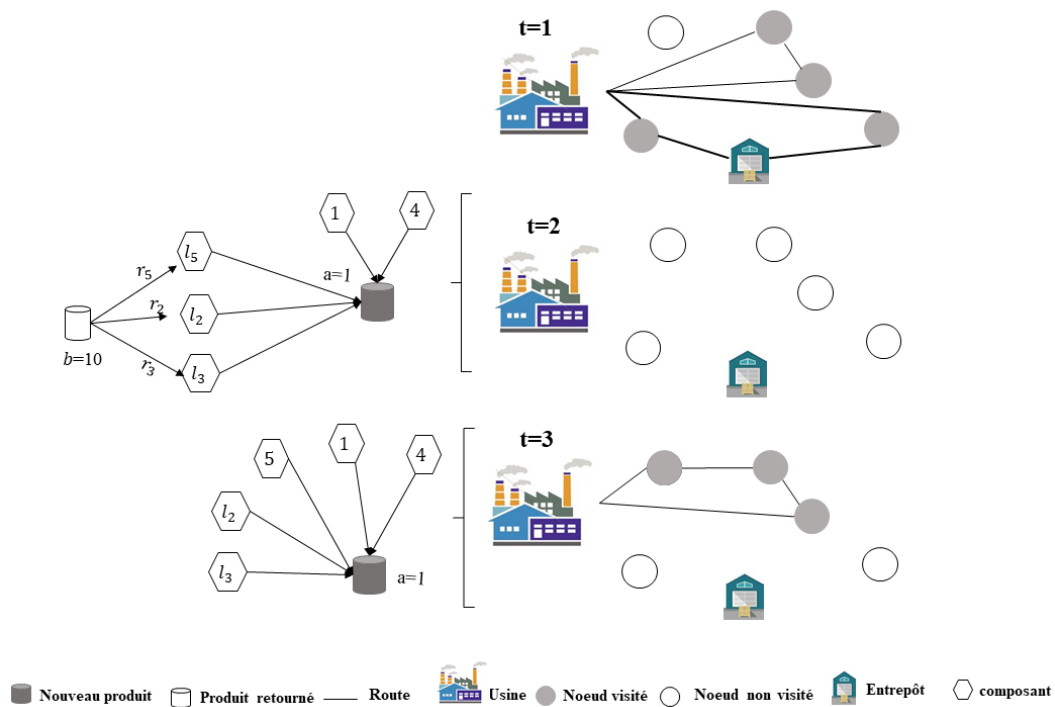


Figure 4.1 – Exemple d'une solution pour le 2D-ARP-R

Les contributions de la recherche dans ce chapitre sont doubles :

1. Nous étudions le 2D-ARP-R robuste dans lequel les rendements sont incertains. Nous utilisons un nombre de scénarios discret.
2. Nous analysons les impacts de différents niveaux de flexibilité de décisions sur le coût total.

4.2 Modélisation mathématique du problème

Dans cette section, nous présentons la définition du problème, le modèle mathématique robuste ainsi que les niveaux de flexibilité.

4.2.1 Définition du problème

Nous considérons un seul site d'assemblage et de désassemblage ayant une capacité de stockage commune C_0 pour les composants, et pour les produits neufs et retournés. L'opération de production est réalisée pour satisfaire la demande extérieure, d_t , en assemblant un ensemble de composants neufs et usés en respectant la capacité de production C . Les ventes perdues sont autorisées dans le modèle avec un coût de pénalité (pr). Nous supposons qu'une unité de chaque composant est nécessaire pour fabriquer une unité du nouveau produit. L'emplacement des fournisseurs et des entrepôts ainsi que l'usine de production sont modélisés comme des nœuds sur un graphe symétrique orienté $G = (N, A)$, avec l'ensemble de nœuds $N^+ = N \cup \{0\}$, où 0 représente l'usine, et l'ensemble d'arcs $A = \{(u, v) : u, v \in N^+, u \neq v\}$. Chaque arc (u, v) a un coût de transport $c_{uv} > 0$. L'ensemble des nœuds regroupe les fournisseurs et les entrepôts. Chaque fournisseur dispose d'un seul type de composant. Il se trouve un stock d'un seul type de produits à désassembler dans les entrepôts. Ces produits sont soit en fin de vie, soit en fin d'utilisation par le consommateur. Une fonction bijective ψ définit le fournisseur d'un composant ($v = \psi(j)$).

En général, la qualité de chaque type de pièces démontées ne peut pas être connue à l'avance. Nous avons donc généré un ensemble de scénarios avec des rendements aléatoires. Les scénarios sont équiprobables et les rendements suivent la loi de distribution de Bernoulli. Tous les produits retournés ont une structure à deux niveaux, ce qui signifie que les produits retournés et les composants désassemblés sont considérés comme des éléments de racine et de feuille, respectivement. Chaque produit retourné a une taille unitaire de b identique à celle du nouveau produit. Cette taille influence la capacité des véhicules sur les routes et la capacité de stockage dans l'usine. Nous considérons un coût de production unitaire f_t et un coût de lancement f'_t au niveau de l'usine. Nous supposons qu'une flotte illimitée de véhicules homogènes d'une capacité de Q est stationnée à l'usine, pour ramasser les composants et les produits retournés des fournisseurs et des entrepôts représentés par l'ensemble $N = \{1, \dots, n\}$. L'horizon de planification comprend T des périodes ($t \in \{1, \dots, |T|\}$) où la demande est connue mais dynamique dans le temps, comme tous les coûts sauf le transport.

4.2.2 Modèle mathématique robuste

La liste complète des notations utilisées tout au long de ce chapitre est donnée dans ce qui suit :

- Les ensembles :

T : Horizon de planification $t \in \{1, \dots, |T|\}$.
 N^+ : Ensemble de nœuds, $N^+ = \{0, \dots, n\}$, où 0 représente l'usine.
 N : Ensemble de fournisseurs et des entrepôts spécifiques qui fournissent des composants et des produits retournés, $N = N^+ \setminus \{0\}$.
 A : Ensemble d'arcs, $A = \{(u, v) : u, v \in N^+, u \neq v\}$.
 K : Ensemble de composants.
 S : Ensemble de scénarios.

— Variables de décisions :

p_t : Quantité de production pour la période t à l'usine.
 y'_t : Égal à 1 s'il y a une opération de désassemblage à l'usine pendant la période t , 0 sinon.
 y_t : Égal à 1 si l'usine fabrique des produits pendant la période t , 0 sinon.
 Ir_t : Niveau de stock de produits retournés à l'usine à la fin de la période t .
 I_t : Niveau de stock des nouveaux produits à l'usine en fin de période t .
 Ic_{it}^s : Niveau de stock du composant i à l'usine en fin de période t dans le scénario s .
 x_{uvt} : Nombre de fois qu'un véhicule traverse l'arc $(u, v) \in A$ pour la période t .
 F_{uvt} : La capacité occupée du véhicule sur l'arc $(u, v) \in A$ pour la période t avant la visite de nœud v .
 F_{vut} : La capacité restante du véhicule sur l'arc $(v, u) \in A$ pour la période t après la visite de nœud v .
 z_{ut} : Égal à 1 si le nœud $u \in N$ est visité dans la période t , 0 sinon.
 q_{ut} : Quantité d'expédition du produit retourné du nœud $u \in N$ à l'usine pour la période t .
 qc_{ut} : Quantité de composants expédiés du nœud $u \in N$ à l'usine pour la période t .
 dis_t : Quantité de désassemblage du produit retourné à l'usine pendant la période t .
 l_t : Quantité de ventes perdues à la période t .

— Paramètres :

d_t : Demande de produits finis à l'usine pendant la période t .
 f_t : Coût de production unitaire de produit final pour la période t .
 f'_t : Coût de lancement pour le produit final pour la période t .
 a_{it} : Coût d'achat unitaire du composant i pour la période t .
 ar_t : Coût d'achat unitaire du produit retourné pour la période t .
 h_t : Coût unitaire de possession des nouveaux articles et à l'usine pour la période t .
 hr_t : Coût unitaire de possession des nouveaux produits à l'usine pour

la période t .

hc_{it} : Coût unitaire de possession du composant i à l'usine pour la période t .

c_{uv} : Coût du transport entre les nœuds u et v , $(u, v) \in A$.

pr : Coût unitaire des ventes perdues.

σ_{ut}^s : Coût approximatif de la visite au nœud $u \in N$ pour la période t dans le scénario s .

C : Capacité de production.

C_0 : Capacité de stockage globale de l'usine.

Q : Capacité des véhicules.

S_{ut} : Quantité disponible du produit retourné au nœud $u \in N$ dans la période t .

Sc_{ut} : Quantité disponible du composant au nœud $u \in N$ pour la période t .

r_i^s : Taux de rendement de l'élément feuille i par rapport à l'élément racine dans le scénario s .

b_i : Taille unitaire de composant i .

b : Taille unitaire du produit.

Le problème robuste minmax modifie la fonction objectif du problème déterministe en tant que fonction objectif minmax après avoir introduit un ensemble de scénarios de rendements incertains. Nous proposons un modèle mathématique basée sur la formulation de bi-commodités (Afsar & Hnaien (2020)). Les nœuds 0 et $n+1$ correspondent au nœud de dépôt. Ce modèle présente une fonction objectif minmax comme suit :

$$\min \left(\sum_{t \in T} (f'_t(y_t + y'_t) + f_t p_t + pr l_t + hr_t I r_t + h_t I_t + \sum_{u \in N} ar_t q_{ut} + \sum_{i \in K} a_{it} q c_{\psi(i)t} + \sum_{(u,v) \in A} c_{uv} x_{uvt} + \max_{s \in S} (\sum_{i \in K} hc_{it} I c_{it}^s)) \right) \quad (4.1)$$

s.t

$$I_{t-1} + p_t + l_t = d_t + I_t \quad \forall t \in T \quad (4.2)$$

$$l_t \leq d_t \quad \forall t \in T \quad (4.3)$$

$$I c_{i,t-1}^s + q c_{\psi(i)t} + r_i^s dis_t = p_t + I c_{it}^s \quad \forall t \in T, \forall i \in K, \forall s \in S \quad (4.4)$$

$$I r_{t-1} + \sum_{u \in N} q_{ut} = dis_t + I r_t \quad \forall t \in T \quad (4.5)$$

$$p_t \leq C y_t \quad \forall t \in T \quad (4.6)$$

$$dis_t \leq C y'_t \quad \forall t \in T \quad (4.7)$$

$$q_{ut} \leq S_{ut} \quad \forall t \in T, \forall u \in N \quad (4.8)$$

$$q_{cut} \leq S_{cut} \quad \forall t \in T, \forall u \in N \quad (4.9)$$

$$\sum_{i \in K} b_i I_{c_{it}^s} + b(I_t + I_{r_t}) \leq C_0 \quad \forall t \in T, \forall s \in S \quad (4.10)$$

$$b_{\psi^{-1}(u)} q_{cut} + b q_{ut} \leq Q z_{ut} \quad \forall t \in T, \forall u \in N \quad (4.11)$$

$$\sum_{u \in N^+ | u \neq v} x_{uvt} = \sum_{u \in N^+ | u \neq v} x_{vut} \quad \forall t \in T, \forall v \in N^+ \quad (4.12)$$

$$\sum_{u \in N^+} x_{uvt} = z_{ut} \quad \forall t \in T, \forall v \in N \quad (4.13)$$

$$F_{uvt} + F_{vut} = Q x_{uvt} \quad \forall t \in T, \forall u \in N^+, \forall v \in N^+ | u \neq v \quad (4.14)$$

$$\sum_{v \in N^+} F_{uvt} - \sum_{v \in N^+} F_{vut} = 2(b q_{ut} + b_{\psi^{-1}(u)} q_{kut}) \quad \forall t \in T, \forall u \in N \quad (4.15)$$

$$\sum_{u \in N} F_{0ut} = \sum_{u \in N} b_{\psi^{-1}(u)} q_{cut} + \sum_{u \in N} b q_{ut} \quad \forall t \in T \quad (4.16)$$

$$\sum_{u \in N} Q x_{0ut} - (\sum_{u \in N} b_{\psi^{-1}(u)} q_{cut} + \sum_{u \in N} b q_{ut}) = \sum_{u \in N} F_{u0t} \quad \forall t \in T \quad (4.17)$$

$$\sum_{v \in N^+} F_{n+1vt} = Q \sum_{v \in N^+} x_{0vt} \quad \forall t \in T \quad (4.18)$$

$$p_t, dis_t, l_t \geq 0, y'_t, y_t \in \{0, 1\} \quad \forall t \in T, \forall u \in N \quad (4.19)$$

$$I_{c_{it}^s}, I_{r_t}, I_t \geq 0 \quad \forall t \in T, \forall i \in K, \forall s \in S \quad (4.20)$$

$$q_{cut}, q_{ut} \geq 0 \quad \forall t \in T, \forall u \in N \quad (4.21)$$

$$F_{uvt} \geq 0 \quad \forall t \in T, \forall u \in N^+, \forall v \in N^+ \quad (4.22)$$

$$z_{ut}, x_{uvt} \in \{0, 1\} \quad \forall t \in T, \forall u \in N^+, \forall v \in N^+ \quad (4.23)$$

La fonction objectif (4.1) minimise le coût de lancement (à la fois pour les opérations de désassemblage et d'assemblage), la production, les ventes perdues, les coûts de stockage des produits neufs et retournés, les coûts d'achat et de transport ainsi que les coûts de stockage maximum des composants pendant tous les scénarios. Les contraintes de (4.2) à (4.11) s'appliquent au problème de dimensionnement de lots avec retours, tandis que les contraintes de (4.12) à (4.18) définissent le problème de transport.

Les contraintes (4.2) assurent le flux des stocks de produits finis fabriqués. Le volume des ventes perdues est inférieur à la demande (contrainte (4.3)). Les contraintes (4.4) et (4.5) assurent l'équilibre des flux de composants (neufs ou démontés) et de produits retournés, respectivement. Les contraintes (4.6) et

(4.7) fixent les variables de décision de réglage pour l'opération d'assemblage et de désassemblage, respectivement.

Les contraintes (4.8) et (4.9) garantissent que chaque nœud visité peut fournir la quantité demandée de composants ou de produits retournés. Les contraintes (4.10) limitent la capacité de stockage du produit final, des composants et des produits retournés à l'usine. Les contraintes (4.11) imposent la limite de la quantité totale de composants et de produits retournés expédiés de chaque nœud à chaque période par la capacité du véhicule.

Les contraintes (4.12) stipulent que le nombre d'arcs entrants est égal au nombre d'arcs sortants à chaque nœud. Les contraintes (4.13) impliquent que si un nœud u est visité, il n'y a qu'un seul arc (uv) qui est parcouru par un véhicule. Les contraintes (4.14) indiquent que la somme de la charge du véhicule et de la capacité restante est égale à la capacité totale du véhicule sur l'arc traversé. La variation totale de la charge du véhicule et de la capacité restante est égale à deux fois le volume de la quantité achetée à partir d'un nœud visité (contraintes (4.15)).

Le chargement de la flotte des véhicules quittant le dépôt est égal au total des quantités achetées (contraintes (4.16)). Les contraintes (4.17) stipulent que la capacité restante de la flotte des véhicules entrant dans le dépôt est égale à la capacité totale de la flotte des véhicules moins le volume total des composants achetés. Les contraintes (4.18) indiquent que la capacité totale restante de la flotte des véhicules entrant dans le dépôt $(n + 1)$ est égale à la capacité totale des véhicules utilisés à partir de dépôt 0. Les contraintes (4.19)-(4.23) sont des contraintes qui définissent le type des variables de décisions.

Le modèle proposé est non linéaire en raison de l'opérateur min-max imbriqué. Nous utilisons ϕ pour indiquer la valeur maximale du coût de stockage des composants à l'usine. Le modèle mathématique équivalent au modèle min-max devient le modèle de programmation linéaire suivant :

$$\min \sum_{t \in T} (f'_t(y_t + y'_t) + f_t p_t + p r l_t + h r_t I r_t + h_t I_t + \sum_{u \in N} a r_t q_{ut} + \sum_{i \in K} a_{it} q c_{\psi(i)t} + \sum_{(u,v) \in A} c_{uv} x_{uvt}) + \phi \quad (4.24)$$

$$\text{s.t (4.2)—(4.23)}$$

$$\sum_{t \in T} \sum_{i \in K} h c_{it} I c_{it}^s \leq \phi \quad \forall s \in S \quad (4.25)$$

4.2.3 Niveaux de flexibilité

Nous analysons cinq hypothèses suivantes :

- H_0 (zéro flexibilité) : aucune variable de décision (sauf Ic_{it}^s) ne dépend des scénarios. Le modèle linéaire à nombres entiers mixtes précédent présente ce niveau de flexibilité.
- H_1 : les décisions relatives à la production et aux ventes perdues dépendent de scénarios (p_t^s, l_t^s, dis_t^s). Cette hypothèse permet d'ajouter de la flexibilité à la production, ce qui signifie que les quantités à assembler et à désassembler dépendent des scénarios.
- H_2 : les décisions relatives aux quantités achetées de produits retournés et de composants dépendent des scénarios (q_{ut}^s, qc_{ut}^s).
- H_3 : les décisions d'achat qui incluent les quantités achetées et les décisions de transport dépendent des scénarios ($q_{ut}^s, qc_{ut}^s, x_{uvt}^s$). Cette hypothèse élargit la flexibilité de H_2 en incluant les décisions de transport.
- H_4 : toutes les variables de décisions dépendent des scénarios ($p_t^s, l_t^s, dis_t^s, q_{ut}^s, qc_{ut}^s, x_{uvt}^s$). Dans cette hypothèse, la production et l'approvisionnement dépendent des scénarios.

4.3 Approche Mat-heuristique

Dans cette section, nous adaptons la mat-heuristique proposé dans le chapitre précédent pour résoudre le problème proposé dans le cas stochastique.

La mat-heuristique proposée est résumée dans la Figure (4.2), elle contient deux étapes principales ; La première étape consiste à résoudre le problème d'assemblage et de désassemblage. Elle traite les décisions de $dis_t^s, p_t, Ic_t^s, q_{ut}^s, qc_{ut}^s, l_t$, et z_{ut}^s . En utilisant la valeur de z_{ut}^s , nous obtenons l'ensemble des nœuds visités dans chaque période et chaque scénario. De même, les valeurs de q_{ut}^s et qc_{ut}^s révèlent la quantité expédiée qui forme chaque nœud. La deuxième étape, consiste à résoudre le problème de transport par la méthode de recherche à voisinage variable.

Nous générons d'abord une solution partielle de coût total (la somme du coût de production et du coût de transport). Puis nous initialisons la valeur de σ_{ut}^s en fonction de cette solution. Le paramètre σ_{ut}^s (mis à jour tout au long de l'algorithme) est utilisé comme un coût de visite approximatif de la collecte depuis le nœud u à la période t sous le scénario s . De manière consécutive, le problème de tournées de véhicules est résolu et le meilleur coût est obtenu. Une fois que la route de toutes les périodes est construite, nous mettons en œuvre un processus de mémorisation inspiré de la recherche tabou (étape C). A chaque itération, nous gardons en mémoire la meilleure solution de transport trouvée à l'étape B. Ensuite, la valeur de σ_{ut}^s est mise à jour en fonction de cette solution et la fonction objectif est recalculée. Cette étape est

sui vie d'une procédure de mise à jour pour le pire cas. Avant de résoudre le problème de dimensionnement de lots, nous vérifions si cette solution existe déjà dans la mémoire. Si c'est le cas, nous passons à l'étape E du mécanisme de diversification. Sinon, nous passons à l'étape A. Cette technique nous permet d'échapper aux optimums locaux pour accélérer la résolution. La procédure itérative s'arrête lorsqu'un critère d'arrêt est atteint (après 10 itérations sans amélioration). Les étapes sont répétées jusqu'à ce qu'un nombre maximal d'itérations ($iter2_{max}$) soit atteint.

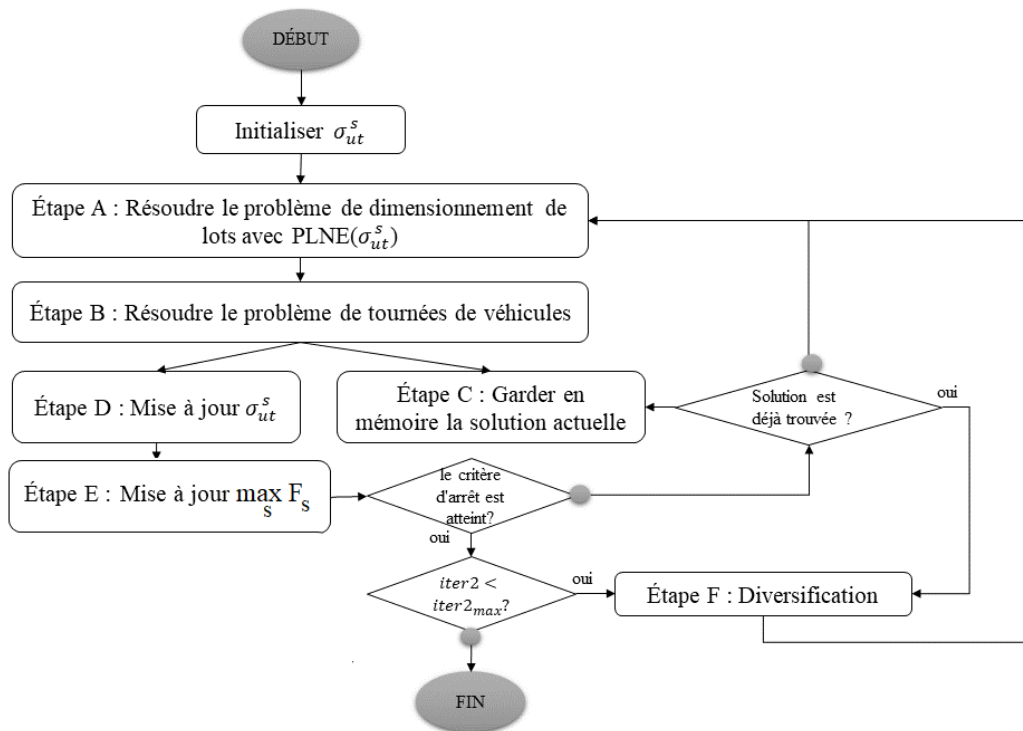


Figure 4.2 – Les étapes générales de la mat-heuristique

Dans certains cas, la limite de temps pour CPLEX dans l'étape A donne des solutions avec une valeur d'écart d'optimalité élevée. Cela nous amène à définir une limite de temps variable qui augmente dès qu'il n'y a pas d'amélioration des coûts. Ensuite, elle revient à sa valeur initiale.

Toutes les étapes de l'algorithme sont détaillées dans le chapitre précédent sauf l'étape E. Dans cette étape, nous recalculons le coût maximal pour chaque itération i de l'algorithme. Nous commençons par recalculer pour chaque scénario s le coût total obtenu par le PLNE ($f1_i^s$) plus le coût de transport obtenu par la méthode de recherche à voisinage variable ($f2_i^s$). Ainsi, $f_i^s = f1_i^s + f2_i^s$. Cela permet de changer la valeur du meilleur coût maximal d'une itération à

l’autre. Pour chaque scénario $s \in \{1, \dots, |S|\}$, nous comparons deux itérations consécutives i et $i+1$ en gardant la valeur minimale $F_s = \min(f_i^s, f_{i+1}^s)$. Nous récupérons la valeur maximale $\max_{s=1..S} F_s$ sur tous les scénarios S . Pour chaque itération i , nous mettons à jour le coût maximal correspondant à la valeur minimale donnée par $bestF = \min(\max_{s=1..S} F_s)$. S’il y a un changement, nous mettons à jour la meilleure solution correspondante. Dans l’exemple suivant (Tableau 4.1), nous présentons la valeur de la fonction objectif recalculée par notre algorithme pour un nombre de scénarios égal à 4 et un nombre d’itérations égal à 8. En conséquence, $bestF = 15885$ est le coût maximal parmi tous les scénarios. Les chiffres en gras dans le tableau montrent les changements dans le $\max_{s=1..S} F_s$ entre les itérations.

Tableau 4.1 – Résultats pour une instance avec 4 scénarios

	s=1	s=2	s=3	s=4
i=1	17188	15902	17053	16327
i=2	16570	15851	16465	15885
i=3	16639	15933	15775	15921
i=4	16639	15851	15738	15977
i=5	16534	15851	16501	15921
i=6	15839	15968	16093	15885
i=7	17188	15887	17053	16363
i=8	15839	15881	15738	15885
<i>bestF</i>				15885

4.4 Résultats expérimentaux

La mat-heuristique est codé en Java. Le solveur Cplex 12.8 est utilisé pour résoudre les modèles mathématiques. Tous les tests sont effectués sur Intel (R) Core (TM) i7 2,9 GHz avec 16 GO de RAM.

Nous résolvons les mêmes instances expliquées dans le chapitre précédent. De plus, pour chaque instance, nous générons un nombre de scénarios égal à 5, 50 et 100 qui représentent les rendements aléatoires. Dans chaque cas, nous vérifions la non-redondance des scénarios.

4.4.1 Impacts des niveaux de flexibilité

Afin d’identifier l’impact des différentes hypothèses expliquées dans la sous-section (4.2.3), une série de tests est effectuée.

Le tableau 4.2 présente les résultats entre les solutions obtenues avec CPLEX sous l'hypothèse H_0 et les autres hypothèses en terme de temps d'exécutions $CPU(s)$ et l'écart d'optimalité ($écart(\%)$).

Le tableau 4.3 présente les déviations $Dev0(\%)$ entre les solutions obtenues avec CPLEX sous l'hypothèse H_0 et les autres hypothèses H_x ($Dev0(\%) = \frac{sol.H_0 - sol.H_x}{sol.H_x}$) aussi la moyenne (Avg) et la médiane (Med) de ces déviations.

Tableau 4.2 – Résultats pour les instances avec 14 nœuds sous 5 scénarios

Inst	H_0		H_1		H_2		H_3		H_4	
	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)
1	4	0.0	5	0.0	77	0.0	3600	0.2	3600	0.2
2	18	0.0	28	0.0	36	0.0	3600	0.2	2830	0.0
3	69	0.0	69	0.0	7	0.0	75	0.0	52	0.0
4	12	0.0	24	0.0	76	0.0	1088	0.0	546	0.0
5	90	0.0	85	0.0	136	0.0	1217	0.0	426	0.0
6	12	0.0	23	0.0	13	0.0	2017	0.0	1676	0.0
7	22	0.0	48	0.0	115	0.0	3600	0.5	3600	0.3
8	32	0.0	55	0.0	35	0.0	3600	0.1	864	0.0
9	27	0.0	32	0.0	204	0.0	3600	0.8	3600	0.8
10	55	0.0	127	0.0	191	0.0	1455	0.0	1594	0.0
Med	25	0.0	40	0.0	77	0.0	2809	0.1	1635	0.0
Avg	34	0.0	49	0.0	89	0.0	2385	0.2	1879	0.1

Nous commençons notre expérimentation en comparant les résultats dans cinq scénarios pour 10 instances avec 14 nœuds. Selon les résultats présentés dans le tableau 4.2, Nous observons que la résolution du problème avec H_0 est plus rapide qu'avec les autres hypothèses, car il y a moins de décisions qui sont impactées par les scénarios. Par conséquent, avec H_0 également pour H_1 et H_2 , le problème est résolu de manière optimale. La comparaison des résultats de H_0 avec ceux de H_1 montre que l'ajout de la flexibilité à la production améliore légèrement le coût total ($Avg_Dev0 = 0.10\%$), où la quantité produite avec H_1 reste inchangée même si elle dépend des scénarios. Cependant, la quantité à désassembler a changé. La résolution du problème avec H_1 permet de trouver la majorité des mêmes solutions optimales fondées par H_0 (9 instances parmi les 10) mais avec un temps de résolution CPU plus élevé. L'ajout de flexibilité aux quantités achetées avec H_2 améliore les résultats de H_0 avec $Avg_Dev0 = 5.70\%$. La comparaison des résultats de H_0 et H_3 montre une amélioration du coût total qui peut atteindre 22.80%. Avec les deux hypothèses H_3 et H_4 , nous retrouvons majoritairement les mêmes solutions, mais avec des temps de résolution CPU différents. En effet, la résolution avec H_4 est plus rapide dans la plupart des cas par rapport à H_3 .

D'après les résultats de déviation présentés dans le tableau 4.3, H_1 améliore légèrement H_0 . Par contre, H_2 , H_3 et H_4 améliorent significativement H_0 . Ces améliorations sont pratiquement équivalentes. En revanche, le fait d'avoir plus de flexibilité dans les décisions améliore le coût total dans chaque hypothèse, mais la difficulté de résolution du problème augmente et nécessite plus de temps. Par conséquent, lors de la résolution du problème avec H_3 et H_4 , la majorité des instances ne sont pas résolues de manière optimale.

Tableau 4.3 – Déviations pour les instances avec 14 nœuds sous 5 scénarios

Instance	Dev0(H_0/H_1)	Dev0(H_0/H_2)	Dev0(H_0/H_3)	Dev0(H_0/H_4)
1	0.0	2.4	2.5	2.5
2	0.0	2.0	2.9	3.1
3	0.0	21.7	22.8	22.8
4	0.0	5.4	6.8	6.9
5	0.0	7.5	7.6	7.6
6	0.0	6.6	7.0	7.0
7	0.0	4.1	4.1	4.3
8	0.0	5.7	7.7	7.8
9	0.8	0.0	0.4	0.8
10	0.0	1.4	1.5	1.5
Med	0.00	4.75	5.45	5.60
Avg	0.10	5.70	6.30	6.40

Suite à cette observation, nous pouvons écarter H_0 et H_1 et résoudre plus d’instances de tailles différentes avec les 3 dernières hypothèses.

4.4.2 Interprétation des résultats

Dans cette section, les résultats numériques obtenus à partir de PLNE ainsi que de la mat-heuristique sont expliqués. Notons qu’un total de 150 instances avec 14 et 50 composants sous 5, 50 et 100 scénarios sont résolues pour comparer les résultats entre H_2 , H_3 et H_4 .

Nous résumons dans les tableaux 4.4, 4.5 4.6 et 4.7, les résultats pour les H_2 , H_3 et H_4 pour chaque groupe d’instances (instances avec 14 et 50 nœuds). Les tableaux de 4.8 à 4.17 en annexe donnent les résultats numériques détaillés.

Dans nos expériences préliminaires, nous constatons que le solveur ne peut pas trouver des solutions optimales pour les instances à 50 nœuds sous 50 scénarios après 3600 secondes. Ainsi, nous donnons au solveur la meilleure solution obtenue par l’approche mat-heuristique comme solution de démarrage. Cette technique est choisie pour évaluer la qualité de la solution mat-heuristique et l’améliorer si possible. L’utilisation de cette technique permet également de réduire le temps de résolution CPU.

Dans ces tableaux, la colonne *PLNE* donne la meilleure solution obtenue à l’aide du solveur Cplex. La colonne *Mat* présente les meilleures solutions de la mat-heuristique. La colonne *PLNE&WS* présente les solutions du PLNE avec une solution initiale de la mat-heuristique. Les colonnes *CPU(s)* présentent le temps de calcul en secondes. La colonne *écart(%)* présente l’écart d’optimalité donné par CLPEX en 3600 secondes de temps de calcul. La colonne *Dev(%)* présente la déviation entre la meilleure solution connue de H_2 (Bks_{H_2}) et la meilleure solution connue de H_3 (Bks_{H_3}) ($Dev(\%) = \frac{Bks_{H_2} - Bks_{H_3}}{Bks_{H_3}}$). La colonne *Dev1(%)* présente la déviation entre la meilleure solution connue de H_2 (Bks_{H_2}) et la meilleure solution connue de H_4 (Bks_{H_4}) ($Dev1(\%) =$

$\frac{Bks_{H_2} - Bks_{H_4}}{Bks_{H_4}}$). Les temps de calcul de *PLNE&WS* sont calculés comme la somme du temps de calcul pour trouver la solution initiale et du temps de calcul de la résolution avec *PLNE&WS*. Pour tous les tableaux, les valeurs minimale (*Min*), maximale (*Max*), moyenne (*Avg*) et médiane (*Med*) sont présentées. Dans tous les tableaux, les valeurs négatives des déviations indiquent une amélioration apportée par H_2 .

Le tableau 4.4 présente le résumé du temps de résolution CPU et l'écart d'optimalité de 20 instances avec 14 nœuds et 10 instances avec 50 nœuds sous 5 scénarios. Pour les instances à 14 nœuds, Cplex peut prouver l'optimalité de toutes les instances avec H_2 . Cependant, il prouve l'optimalité pour 11 des 20 instances dans le temps imparti avec un *Max_écart* = 2.5% et un *Avg_écart* = 0.4% pour H_3 . Pour H_4 , il prouve l'optimalité pour 13 des 20 instances dans le délai imparti avec un *Max_écart* = 1.7% et *Avg_écart* = 0.3%. Comme le montrent les tableaux 4.8, 4.9 et 4.10, le *PLNE&WS* est capable de trouver 20 solutions optimales pour toutes les hypothèses. La mat-heuristique converge vers les mêmes résultats pour presque toutes les instances pour les trois hypothèses. Pour toutes les hypothèses, la mat-heuristique et le *PLNE&WS* sont plus efficaces que PLNE. La mat-heuristique peut trouver une bonne solution très rapidement avec un *Avg_CPU* = 20 secondes pour H_2 , *Avg_CPU* = 64 secondes pour H_3 et *Avg_CPU* = 59 secondes pour H_4 .

Les résultats pour les instances avec 50 nœuds indiquent que la difficulté de résolution augmente avec la taille des nœuds, pour H_2 le *Avg_écart* = 40.7%, H_3 le *Avg_écart* = 70.1% et H_4 le *Avg_écart* = 56.4%. D'après ces exemples, le *Avg_écart* est plus serré pour H_2 que pour H_3 et H_4 .

La solution obtenue pour H_2 par *PLNE&WS* conduit à une amélioration de l'écart d'optimalité (par rapport au PLNE) de 0.7% en moyenne, et de 1.5% comme valeur maximale de l'écart d'optimalité. Pour toutes les instances, *PLNE&WS* trouve les meilleures solutions par rapport aux autres méthodes, mais avec un temps de résolution CPU plus important, qui est égal à la somme du temps de résolution CPU de la mat-heuristique et de PLNE.

Pour poursuivre l'évaluation des performances des méthodes proposées, nous avons exécuté la deuxième série de tests sur des instances de plus grande taille sous 50 scénarios. Pour H_2 , les écarts moyens d'optimalité pour le PLNE et le *PLNE&WS* obtenus pour 20 instances avec 14 nœuds sous 50 scénarios sont de 7.4% et 0.0%, respectivement. Comme nous pouvons le voir dans le tableau 4.11, le PLNE n'est pas en mesure de trouver une solution faisable pour les 10 instances avec 50 nœuds sous 50 scénarios. En revanche, le *PLNE&WS* peut résoudre les 20 instances avec *Avg_écart* = 3.6%. Les résultats montrent que la mat-heuristique peut trouver de bonnes solutions, qui sont améliorées par le *PLNE&WS*.

Tableau 4.4 – Performances des méthodes pour des instances avec 14 et 50 noeuds avec 5 scénarios

	H_2						H_3						H_4					
	PLNE		Mat	PLNE&WS		Mat	PLNE		PLNE&WS		Mat	PLNE		PLNE&WS		Mat	PLNE	
	CPU(s)	Ecart(%)		CPU(s)	Ecart(%)		CPU(s)	Ecart(%)	CPU(s)	Ecart(%)		CPU(s)	Ecart(%)	CPU(s)	Ecart(%)		CPU(s)	Ecart(%)
14	Max	2084	0.00	52	212	0.00	3600	2.50	136	921	0.00	3600	1.70	141	962	0.00		
	Min	7	0.00	5	14	0.00	75	0.00	14	82	0.00	52	0.00	11	136	0.00		
	Med	70	0.00	13	81	0.00	2822	0.00	59	406	0.00	2035	0.00	44	374	0.00		
	Ave	185	0.00	20	96	0.00	2502	0.40	64	436	0.00	2143	0.30	59	463	0.00		
50	Max	3600	85.30	951	4551	1.50	3600	99.80	421	4021	2.50	3600	99.20	1036	4636	5.70		
	Min	3600	6.00	40	3640	0.20	3600	17.50	65	3665	0.30	3600	29.00	62	3662	0.80		
	Med	3600	20.90	90	3691	0.80	3600	98.80	226	3826	1.30	3600	41.20	277	3878	2.30		
	Ave	3600	40.70	199	3799	0.70	3600	70.10	230	3830	1.30	3600	56.40	487	4087	2.90		

Pour H_3 , le *PLNE&WS* améliore les solutions trouvées par le PLNE avec un $Avg_écart = 0.1\%$ pour les instances avec 14 noeuds. Les résultats montrent que le PLNE a la plus mauvaise performance ; il est loin de la solution optimale avec un $Avg_écart = 77.1\%$ pour les instances avec 14 noeuds. De plus, il ne peut trouver aucune solution réalisable pour les instances à 50 noeuds, où le *PLNE&WS* prouve son efficacité avec un $Avg_écart = 3.9\%$.

Pour H_4 , le *PLNE&WS* améliore les solutions trouvées par le PLNE avec un $Avg_écart = 0.5\%$ pour les instances à 14 noeuds et avec un $Avg_écart = 4\%$ pour les instances à 50 noeuds. Les résultats montrent que le PLNE est loin de la solution optimale avec un $Avg_écart = 73.4\%$ pour les instances avec 14 noeuds. De plus, il ne peut trouver aucune solution réalisable pour les instances à 50 noeuds.

Le tableau 4.6 résume les résultats pour 20 instances avec 14 noeuds sous 100 scénarios. Pour ces 20 cas, les performances de *PLNE&WS* proposé sont supérieures à celles des autres méthodes (voir tableaux 4.14, 4.15 et 4.16).

Pour H_2 , l'écart d'optimalité moyen trouvé par PLNE ($Avg_écart = 13.5\%$) est plus élevé que celui de *PLNE&WS* ($Avg_écart = 1\%$). Pour H_3 , l'écart d'optimalité moyen trouvé par PLNE ($Avg_écart = 55.9\%$) est plus élevé que celui de *PLNE&WS* ($Avg_écart = 3.1\%$). Pour H_4 , l'amélioration de *PLNE&WS* est significative, elle réduit l'écart d'optimalité de PLNE de ($Avg_écart = 77\%$) à ($Avg_écart = 3.1\%$).

La difficulté de résolution du problème augmente avec 100 scénarios pour la PLNE avec H_3 et H_4 . Ainsi, dans 7 et 4 cas sur 20, la PLNE ne peut pas trouver de solution réalisable pour H_3 et H_4 , respectivement.

Le tableau 4.7 présente un résumé sur les déviations entre H_2 et H_3 aussi entre H_2 et H_4 pour les différents scénarios. Selon ces résultats, nous remarquons que pour 14 noeuds avec 5 scénarios, H_3 améliore H_2 au maximum de 3.20% et en moyenne 1.1%. L'hypothèse H_4 améliore au maximum H_2 de 3.20% et en moyenne 1.2%. Concernant les résultats avec 50 scénarios, nous observons que H_3 améliore au maximum H_2 de 16.20% et en moyenne 3.1%, et H_4 améliore au maximum H_2 de 17% et en moyenne 3.7%. Pour les résultats avec 100 scénarios, H_3 et H_4 améliore H_2 en moyenne de 1.4% et de 4.8%, respectivement. Et de même avec 50 noeuds et différents scénarios, comme le montre le tableau 4.17.

À partir de ces résultats, nous pouvons conclure que H_3 et H_4 améliorent dans la plupart des cas H_2 , en raison de la flexibilité des différentes décisions. En revanche, H_2 est plus rapide et facile à mettre en œuvre en industrie. Ainsi, les plannings de production et de tournées ne sont pas impactés par les changements de scénarios, sauf les décisions d'achat.

Tableau 4.5 – Performances des méthodes pour des instances avec 14 et 50 nœuds avec 50 scénarios

	H_2						H_3						H_4						
	PLNE		Mat		PLNE&WS		PLNE		Mat		PLNE&WS		PLNE		Mat		PLNE&WS		
	CPU(s)	Ecart(%)	CPU(s)	CPU(s)	Ecart(%)	CPU(s)	CPU(s)	Ecart(%)	CPU(s)	CPU(s)	Ecart(%)	CPU(s)	CPU(s)	Ecart(%)	CPU(s)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)
14	Max	3600	18.1	948	4548	0.4	3600	99.5	985	4585	0.8	3600	99.1	549	4149	2.5			
	Min	3600	0.8	29	3629	0.0	3600	12.6	83	3683	0.0	3600	1.6	124	3724	0.0			
	Med	3600	6.7	161	3761	0.0	3600	93.6	315	3915	0.0	3600	95.1	217	3817	0.3			
	Avg	3600	7.4	298	3899	0.0	3600	77.1	459	4059	0.1	3600	73.4	253	3853	0.5			
50	Max	-	-	975	4575	9.8	-	-	2095	5695	6.2	-	-	3509	7109	7.8			
	Min	-	-	342	3942	0.2	-	-	821	4421	0.7	-	-	421	4021	0.7			
	Med	-	-	502	4102	2.3	-	-	1003	4603	3.6	-	-	1455	5055	3.6			
	Avg	-	-	629	4229	3.6	-	-	1181	4781	3.9	-	-	1616	5216	4.0			

(-) : Aucune solution réalisable n'est disponible

Tableau 4.6 – Performances des méthodes pour des instances avec 14 nœuds avec 100 scénarios

	H_2						H_3						H_4					
	PLNE		PLNE&WS		PLNE		PLNE&WS		PLNE		PLNE&WS		PLNE		PLNE&WS			
	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)	CPU(s)	Ecart(%)		
Max	3600	45.9	975	4575	3600	99.9	1369	4969	3600	99.7	2543	6143	3600	99.7	6143	6.5		
Min	3600	0.2	125	3725	3600	37.0	145	3745	3600	66.7	169	3769	3600	66.7	3769	1.1		
Med	3600	9.8	346	3946	3600	94.8	698	4298	3600	99.0	567	4167	3600	99.0	4167	3.7		
Avg	3600	13.5	478	4078	3600	55.9	705	4305	3600	77.0	695	4300	3600	77.0	4300	3.1		

Tableau 4.7 – Déviations pour les instances avec 14 et 50 nœuds

		5		50		100	
		Dev (%)	Dev1 (%)	Dev (%)	Dev1 (%)	Dev (%)	Dev1 (%)
14	Max	3.20	3.20	16.2	17.0	8.8	15.0
	Min	0.00	0.00	0.0	0.6	-3.4	-0.9
	Med	1.20	1.20	1.4	3.0	1.4	7.0
	Avg	1.10	1.20	3.1	3.7	1.4	4.8
50	Max	10.90	12.00	11.6	14.2	-	-
	Min	0.00	1.10	0.2	2.3	-	-
	Med	2.60	3.00	1.8	4.6	-	-
	Avg	3.10	3.70	3.2	5.5	-	-

(-) : Aucune solution réalisable n'est disponible

4.5 Conclusion

Dans ce chapitre, nous avons étudié un problème dynamique d'assemblage de désassemblage avec tournées de véhicules avec retours sous des rendements incertains. Ce problème prend en compte la nature stochastique des rendements pour aider les gestionnaires à optimiser leur réseau de chaîne logistique en fonction des scénarios de rendement disponibles.

Dans un premier temps, nous avons formulé le 2D-ARP-R stochastique comme un minmax modèle de programmation linéaire en nombres entiers. Ensuite, nous avons développé une math-heuristique efficace basée sur un algorithme de recherche à voisinage variable et un *PLNE*. Dans la première phase de l'algorithme de résolution, un problème de dimensionnement de lot est résolu avec des coûts de transport approximatifs. La deuxième phase traite le problème de transport. La troisième phase présente une procédure pour recalculer le coût maximal. Le modèle et la mat-heuristique sont testés sur un ensemble d'instances issues de la littérature et modifiés en conséquence.

Nous avons commencé notre expérimentation en ajoutant de la flexibilité aux décisions considérées dans notre problème. Nous avons constaté que l'ajout de flexibilité dans l'approvisionnement, qui comprend les quantités achetées et les décisions de transport, permet un abaissement permanent de coût total. Cependant, le fait d'autoriser la flexibilité uniquement pour les décisions sur les quantités achetées est le plus pratique pour l'industrie.

Nous avons comparé les performances du modèle *PLNE* avec la mat-heuristique. D'après les résultats, l'approche mat-heuristique est plus rapide que les autres méthodes et fournit des solutions de bonne qualité. Toutefois, le *PLNE&WS* est plus efficace en termes de qualité de solution.

Les recherches futures peuvent étendre le modèle proposé de plusieurs façons. D'une part, les produits de types multiples peuvent être considérés. Dans une autre direction, examiner d'autres paramètres du modèle qui peuvent être stochastiques, tels que les demandes aléatoires des clients. Les études futures peuvent aborder ce point en développant par exemple une programmation

stochastique à deux étapes aussi des méthodes exactes et en comparant les résultats avec ceux produits par le PLNE et la mat-heuristique.

Tableau 4.8 – Résultats pour les instances avec 5 scénarios pour H_2

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	15890.09	77.20	0.0	15890.09	8	15890.09	82	0.0
	2	19923.87	36.21	0.0	19923.87	52	19923.87	74	0.0
	3	17218.15	6.70	0.0	17218.15	9	17218.15	14	0.0
	4	18590.95	75.66	0.0	18590.95	10	18590.95	67	0.0
	5	14348.89	135.83	0.0	14348.89	9	14348.89	161	0.0
	6	16419.54	13.03	0.0	16419.54	13	16419.54	24	0.0
	7	25596.61	114.74	0.0	25596.61	45	25596.61	169	0.0
	8	16198.59	35.16	0.0	16198.59	31	16198.59	59	0.0
	9	24556.68	204.29	0.0	24556.68	17	24556.68	173	0.0
	10	19835.72	190.51	0.0	19835.72	13	19835.72	80	0.0
	11	9940.03	63.35	0.0	9940.03	6	9940.03	92	0.0
	12	14530.81	178.39	0.0	14530.81	28	14530.81	212	0.0
	13	13574.99	2084.08	0.0	13574.99	31	13574.99	116	0.0
	14	14396.17	106.45	0.0	14396.17	9	14396.17	99	0.0
	15	16391.04	204.48	0.0	16391.04	40	16391.04	115	0.0
	16	15903.75	23.30	0.0	15903.75	18	15903.75	170	0.0
	17	10690.69	32.26	0.0	10690.69	8	10690.69	35	0.0
	18	11808.19	45.86	0.0	11808.19	42	11808.19	78	0.0
	19	15222.69	20.90	0.0	15222.69	5	15222.69	25	0.0
	20	10600.58	50.06	0.0	10600.58	8	10600.58	81	0.0
50	1	67697.96	3600	53.7	31984.11	304	31681.17	3904	1.1
	2	38262.00	3600	20.3	31137.17	178	30960.20	3778	1.5
	3	45523.74	3600	16.9	38098.20	71	38070.75	3671	0.6
	4	42752.64	3600	21.4	33714.79	98	33661.21	3698	0.2
	5	251852.95	3600	82.6	43990.34	951	43852.06	4551	0.2
	6	42021.93	3600	6.0	40047.79	40	39859.68	3640	0.9
	7	155986.38	3600	82.9	27029.76	57	26904.26	3657	0.9
	8	51307.91	3600	20.0	41446.58	137	41217.81	3737	0.4
	9	58893.32	3600	18.2	49094.89	83	48682.31	3683	1.0
	10	329672.55	3600	85.3	48682.54	72	48537.80	3672	0.4

Tableau 4.9 – Résultats pour les instances avec 5 scénarios pour H_3

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	15886.72	3600	0.2	15885.04	21	15851.07	832	0.0
	2	19740.79	3600	0.2	19740.79	18	19700.85	136	0.0
	3	16987.57	75	0.0	16987.57	21	16987.57	87	0.0
	4	18321.36	1088	0.0	18321.36	17	18321.36	82	0.0
	5	14330.78	1217	0.0	14330.78	14	14330.42	402	0.0
	6	16359.54	2017	0.0	16359.54	136	16357.93	204	0.0
	7	25601.59	3600	0.5	25555.55	115	25475.37	520	0.0
	8	15852.70	3600	0.1	15837.55	58	15837.55	246	0.0
	9	24457.25	3600	0.8	24318.38	26	24268.61	755	0.0
	10	19825.86	1455	0.0	19827.72	38	19825.86	118	0.0
	11	9914.63	827	0.0	9914.63	78	9914.63	632	0.0
	12	14564.27	3600	1.6	14356.38	135	14334.7	547	0.0
	13	13599.92	3600	2.5	13330.36	96	13259.72	200	0.0
	14	14298.02	3600	0.4	14295.95	61	14240.95	410	0.0
	15	16387.27	3600	1.7	16187.27	54	16108.93	109	0.0
	16	15416.78	1256	0.0	15416.78	55	15416.78	800	0.0
	17	10690.69	2038	0.0	10690.69	88	10690.69	892	0.0
	18	11599.33	3491	0.0	11600.28	113	11599.33	921	0.0
	19	15221.18	2030	0.0	15222.69	77	15221.18	363	0.0
	20	10345.74	2154	0.0	10345.74	67	10345.74	467	0.0
50	1	9473842.14	3600	99.7	31922.27	330	30880.27	3930	0.6
	2	42969.51	3600	29.3	31120.25	236	30946.70	3836	1.8
	3	45535.31	3600	17.5	37719.66	65	37700.95	3665	0.3
	4	42919.54	3600	30.0	31213.96	236	30360.96	3836	1.1
	5	19658169.33	3600	99.8	42643.16	421	41423.63	4021	1.9
	6	4747777.25	3600	99.2	39918.52	86	38812.32	3686	0.6
	7	3960572.04	3600	99.3	26880.29	136	26880.29	3736	1.4
	8	54826.89	3600	28.4	40044.14	216	40044.14	3816	1.9
	9	7595305.48	3600	99.4	48203.29	154	47723.29	3754	2.5
	10	2802442.90	3600	98.3	47429.04	416	47126.83	4016	0.9

Tableau 4.10 – Résultats pour les instances avec 5 scénarios pour H_4

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	15886.72	3600	0.2	15851.07	50	15851.07	353	0.0
	2	19700.85	2830	0.0	19700.85	37	19700.85	338	0.0
	3	16987.57	52	0.0	16987.57	11	16987.57	136	0.0
	4	18301.36	546	0.0	18301.12	20	18301.12	267	0.0
	5	14330.78	426	0.0	14330.78	27	14330.78	621	0.0
	6	16357.93	1676	0.0	16357.93	16	16357.93	441	0.0
	7	25555.55	3600	0.3	25475.37	31	25475.37	951	0.0
	8	15837.55	864	0.0	15837.55	16	15837.55	565	0.0
	9	24357.25	3600	0.8	24157.25	35	24157.25	380	0.0
	10	19825.86	1594	0.0	19825.86	68	19825.86	831	0.0
	11	9914.63	1442	0.0	9914.63	54	9914.63	599	0.0
	12	14520.12	3600	1.3	14334.7	141	14334.7	357	0.0
	13	13372.58	3600	0.9	13259.72	104	13259.72	400	0.0
	14	14240.95	2393	0.0	14240.95	16	14240.95	305	0.0
	15	16387.27	3600	1.7	16108.93	136	16108.93	336	0.0
	16	15416.78	702	0.0	15416.78	19	15416.78	238	0.0
	17	10690.69	3099	0.0	10690.69	118	10690.69	619	0.0
	18	11599.33	1131	0.0	11599.33	68	11599.33	368	0.0
	19	15221.18	912	0.0	15221.18	105	15221.18	186	0.0
	20	10420.95	3600	0.7	10345.74	104	10345.74	962	0.0
50	1	1160964.09	3600	97.4	31176.45	104	31016.06	3704	1.2
	2	-	3600	-	31050.07	1014	30623.07	4614	5.0
	3	62567.11	3600	41.2	37126.56	102	37072.01	3702	0.8
	4	43878.50	3600	32.7	30192.58	242	30041.98	3842	1.7
	5	3021857.05	3600	98.7	42184.22	943	41936.50	4543	4.2
	6	4673638.09	3600	99.2	39146.62	313	38440.62	3913	5.7
	7	36197.70	3600	29.0	26419.41	62	26067.38	3662	1.4
	8	4504898.81	3600	99.1	40011.24	1036	39965.59	4636	2.0
	9	68861.18	3600	33.3	48137.93	945	48062.04	4545	4.4
	10	68927.42	3600	33.2	47376.46	109	47211.54	3709	2.5

(-) : Aucune solution réalisable n'est disponible

Tableau 4.11 – Résultats pour les instances avec 50 scénarios pour H_2

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	19981.20	3600	16.0	16810.11	87	16789.55	3687	0.0
	2	20160.15	3600	1.1	19958.04	117	19947.95	3717	0.0
	3	19815.76	3600	2.0	19422.62	100	19419.75	3700	0.0
	4	20922.23	3600	5.7	19732.99	278	19732.39	3878	0.0
	5	16651.64	3600	2.9	16206.04	101	16162.89	3701	0.0
	6	16837.82	3600	1.1	17016.33	176	16648.23	3776	0.0
	7	28135.71	3600	7.8	25988.82	274	25944.12	3874	0.0
	8	17725.01	3600	7.9	16486.59	41	16331.32	3641	0.0
	9	28173.56	3600	12.2	24883.25	475	24724.45	4075	0.0
	10	25540.05	3600	18.1	20925.04	29	20919.64	3629	0.0
	11	13768.58	3600	13.7	11892.68	452	11883.57	4052	0.0
	12	17347.99	3600	13.5	15043.23	667	15038.14	4267	0.2
	13	15580.06	3600	4.9	14849.77	758	14823.62	4358	0.0
	14	15642.96	3600	7.8	14503.18	205	14427.61	3805	0.0
	15	20218.13	3600	12.7	17738.99	84	17669.94	3684	0.1
	16	16112.66	3600	0.8	15976.35	948	15976.63	4548	0.0
	17	12666.45	3600	5.1	12019.96	96	12014.62	3696	0.0
	18	13277.56	3600	3.7	12786.89	842	12786.98	4442	0.0
	19	17437.40	3600	7.6	16216.15	94	16176.73	3694	0.4
	20	11205.24	3600	3.6	10907.54	146	10800.43	3746	0.0
50	1	-	-	-	32976.54	975	32851.54	4575	4.4
	2	-	-	-	32816.51	912	32757.05	4512	1.7
	3	-	-	-	41237.59	451	41062.72	4051	1.3
	4	-	-	-	34856.67	543	34712.58	4143	2.4
	5	-	-	-	45168.79	342	44946.98	3942	1.5
	6	-	-	-	40836.47	943	40775.47	4543	2.1
	7	-	-	-	49716.71	862	47624.37	4462	6.0
	8	-	-	-	51156.71	461	50325.16	4061	9.8
	9	-	-	-	51856.84	346	51629.49	3946	6.3
	10	-	-	-	50852.76	456	50803.25	4056	0.2

(-) : Aucune solution réalisable n'est disponible

Tableau 4.12 – Résultats pour les instances avec 50 scénarios pour H_3

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	70865.29	3600	77.4	16048.64	245	16048.64	3845	0.0
	2	28452.41	3600	29.9	19977.91	342	19937.21	3942	0.0
	3	22095.58	3600	12.6	19320.40	83	19320.40	3683	0.0
	4	28304.94	3600	31.3	19444.64	145	19444.64	3745	0.0
	5	66314.95	3600	77.0	15254.37	256	15254.37	3856	0.0
	6	1783766.85	3600	99.1	16646.44	256	16646.44	3856	0.0
	7	1497790.91	3600	98.3	26002.91	259	25942.87	3859	0.8
	8	32921.71	3600	51.1	16114.29	259	16114.29	3859	0.0
	9	1702665.52	3600	98.6	24572.13	123	24495.43	3723	0.1
	10	1387092.73	3600	98.5	20803.57	562	20716.75	4162	0.5
	11	191392.61	3600	94.7	10224.36	287	10224.36	3887	0.0
	12	953120.43	3600	98.5	14667.05	216	14667.05	3816	0.0
	13	887125.25	3600	98.4	14367.02	356	14367.02	3956	0.0
	14	28139.80	3600	49.2	14304.21	945	14304.21	4545	0.0
	15	1687268.28	3600	99.0	16285.45	961	16285.45	4561	0.0
	16	211019.89	3600	92.5	15884.45	923	15884.45	4523	0.0
	17	137249.10	3600	91.3	11995.51	945	11995.51	4545	0.0
	18	22741.81	3600	47.6	11921.23	985	11921.23	4585	0.0
	19	864089.85	3600	98.2	15622.96	425	15491.76	4025	0.1
	20	2318123.09	3600	99.5	10518.24	612	10518.24	4212	0.0
50	1	-	-	-	32546.02	1845	32201.53	5445	3.4
	2	-	-	-	32469.31	1256	32469.31	4856	5.7
	3	-	-	-	40436.28	845	40436.28	4445	2.1
	4	-	-	-	31107.30	975	31107.30	4575	3.1
	5	-	-	-	42523.38	1024	42523.38	4624	0.7
	6	-	-	-	40654.88	983	40654.88	4583	3.7
	7	-	-	-	46001.11	2095	45601.11	5695	6.2
	8	-	-	-	50019.67	945	48319.67	4545	6.1
	9	-	-	-	51521.22	1023	51521.22	4623	6.2
	10	-	-	-	50774.46	821	50094.46	4421	1.5

(-) : Aucune solution réalisable n'est disponible

Tableau 4.13 – Résultats pour les instances avec 50 scénarios pour H_4

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	1523511.73	3600	99.0	16036.79	213	15902.38	3813	0.1
	2	25825.30	3600	23.5	19876.21	218	19804.15	3818	0.2
	3	19630.64	3600	1.6	19319.23	546	19306.79	4146	0.0
	4	28217.27	3600	32.7	19364.57	256	19064.73	3856	0.4
	5	23991.19	3600	38.0	15250.46	216	15249.42	3816	2.5
	6	1484584.59	3600	98.9	16645.44	549	16550.98	4149	0.1
	7	2214403.04	3600	98.8	25808.50	146	25670.38	3746	0.4
	8	151479.71	3600	89.4	16104.29	246	16101.44	3846	0.1
	9	669054.46	3600	96.4	24568.04	145	24411.28	3745	0.1
	10	1453002.07	3600	98.6	20794.58	124	20704.68	3724	1.2
	11	180677.73	3600	94.4	10154.78	543	10154.78	4143	0.4
	12	604089.37	3600	97.6	14667.54	246	14604.29	3846	0.4
	13	1592994.91	3600	99.1	14353.71	126	14345.52	3726	1.5
	14	19963.72	3600	28.3	14304.21	156	14304.21	3756	0.0
	15	377252.44	3600	95.7	16285.45	245	16285.45	3845	0.0
	16	194161.11	3600	91.9	15808.36	346	15802.24	3946	0.4
	17	784427.03	3600	98.5	11624.74	246	11528.35	3846	0.9
	18	17710.48	3600	33.0	11887.37	124	11881.55	3724	0.2
	19	659633.94	3600	97.7	15606.79	156	15469.38	3756	0.7
	20	23244.64	3600	54.9	10518.24	213	10490.35	3813	0.1
50	1	-	-	-	32109.70	2136	32109.70	5736	3.7
	2	-	-	-	31270.85	1346	31270.85	4946	2.2
	3	-	-	-	39203.20	421	38203.84	4021	3.4
	4	-	-	-	31093.32	1563	30403.15	5163	1.4
	5	-	-	-	42452.49	1096	42449.02	4696	0.7
	6	-	-	-	39453.15	945	39453.15	4545	3.1
	7	-	-	-	45960.13	2456	44660.13	6056	4.4
	8	-	-	-	49168.5974	3509	48168.5974	7109	7.10
	9	-	-	-	50179.28	2043	50179.28	5643	7.80
	10	-	-	-	49463.05	645	49463.05	4245	6.6

(-) : Aucune solution réalisable n'est disponible

Tableau 4.14 – Résultats pour les instances avec 100 scénarios pour H_2

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	32963.61	3600	45.9	17923.44	346	17913.44	3946	0.5
	2	21195.66	3600	1.7	20908.39	245	20908.39	3845	0.3
	3	29653.60	3600	34.4	19475.97	265	19475.97	3865	0.1
	4	22168.39	3600	10.3	19977.04	258	19977.04	3858	0.5
	5	26218.75	3600	23.9	19961.08	125	19961.08	3725	0.0
	6	28726.26	3600	35.1	18812.61	345	18812.61	3945	0.9
	7	27076.02	3600	0.2	27935.16	148	27935.16	3748	3.3
	8	20133.75	3600	11.5	18099.85	945	18099.85	4545	1.5
	9	26269.65	3600	1.4	25962.97	965	25962.97	4565	0.2
	10	22305.28	3600	1.1	22368.54	245	22368.54	3845	1.4
	11	13871.17	3600	13.8	12051.59	695	12051.59	4295	0.8
	12	17373.00	3600	3.6	16939.00	412	16839.00	4012	0.3
	13	16836.11	3600	8.6	15599.21	695	15599.21	4295	1.3
	14	21172.63	3600	22.4	16648.25	258	16619.25	3858	1.1
	15	20872.63	3600	12.7	18396.97	145	18396.97	3745	0.9
	16	17377.05	3600	4.3	17033.36	945	17033.36	4545	2.4
	17	14181.41	3600	9.2	13096.15	236	13096.15	3836	1.7
	18	14131.76	3600	4.4	13607.31	645	13607.31	4245	0.7
	19	17548.13	3600	6.4	16569.94	664	16569.94	4264	0.9
	20	17548.13	3600	19.2	14356.90	975	14256.90	4575	0.5

Tableau 4.15 – Résultats pour les instances avec 100 scénarios pour H_3

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	61688.27	3600	72.5	17007.13	156	17007.13	3756	0.3
	2	68987.08	3600	69.9	21065.52	365	21065.52	3965	1.3
	3	30815.67	3600	37.0	19426.09	456	19426.09	4056	0.0
	4	67085.86	3600	71.0	19509.90	145	19509.90	3745	0.3
	5	139632.76	3600	85.9	20333.18	698	20333.18	4298	3.3
	6	18761854.73	3600	99.9	18556.09	385	18556.09	3985	0.2
	7	-	3600	-	27645.21	269	27645.21	3869	4.2
	8	288375.28	3600	94.1	17405.98	1245	17405.98	4845	2.8
	9	-	3600	-	26241.73	1096	26241.73	4696	2.8
	10	-	3600	-	22157.82	698	22057.82	4298	2.8
	11	199062.40	3600	94.8	11079.00	546	11079.00	4146	5.8
	12	-	3600	-	16541.64	875	16541.64	4475	5.6
	13	1195781.71	3600	98.8	15886.47	985	15186.47	4585	3.4
	14	715131.12	3600	97.7	16120.98	746	16120.98	4346	0.1
	15	-	3600	-	18064.08	698	18064.08	4298	4.8
	16	1488487.83	3600	98.9	17414.25	783	17414.25	4383	4.7
	17	1553425.20	3600	99.2	13551.01	1069	13551.01	4669	6.3
	18	1169954.50	3600	99.0	12902.32	1369	12902.32	4969	6.0
	19	-	3600	-	16383.89	542	16383.89	4142	2.4
	20	-	3600	-	14733.95	975	14733.95	4575	3.9

(-) : Aucune solution réalisable n'est disponible

Tableau 4.16 – Résultats pour les instances avec 100 scénarios pour H_4

N	Inst	PLNE			Mat		PLNE&WS		
		Obj	CPU(s)	Ecart(%)	Obj	CPU(s)	Obj	CPU(s)	Ecart(%)
14	1	1473808.51	3600	98.9	16456.11	546	16456.11	4146	3.3
	2	1107279.01	3600	98.2	20374.21	256	20374.21	3856	2.8
	3	-	3600	-	20334.74	257	-	3857	-
	4	3721999.09	3600	99.5	20790.32	356	20090.32	4056	5.1
	5	2228707.39	3600	99.2	18578.96	254	18578.96	3854	4.1
	6	6392087.54	3600	99.7	17199.65	256	17199.65	3856	1.4
	7	4204361.40	3600	99.4	28021.55	259	28021.55	3859	5.9
	8	1268604.69	3600	98.7	17680.55	785	17680.55	4385	4.5
	9	6329571.80	3600	99.6	25243.16	236	25243.16	3836	1.4
	10	5155764.18	3600	99.6	22007.08	945	22007.08	4545	4.5
	11	194259.04	3600	94.7	10479.87	967	10479.87	4567	1.1
	12	4149682.17	3600	99.6	15639.39	587	15639.39	4187	1.4
	13	-	3600	-	15386.09	326	-	3926	-
	14	657702.72	3600	97.7	15528.55	169	15528.55	3769	3.7
	15	1657291.26	3600	99.0	17023.82	754	17023.82	4354	2.4
	16	-	3600	-	17056.05	961	-	4561	-
	17	36658.87	3600	66.7	13015.87	2543	13015.87	6143	6.2
	18	502154.35	3600	97.6	12360.13	1026	12360.13	4626	3.9
	19	-	3600	-	16723.98	1087	16723.98	4687	6.5
	20	160107.62	3600	92.5	12495.52	1324	12405.52	4924	3.1

(-) : Aucune solution réalisable n'est disponible

Tableau 4.17 – Différentes déviations pour tous les scénarios

N	Inst	5		50		100	
		Dev (%)	Dev1 (%)	Dev (%)	Dev1 (%)	Dev (%)	Dev1 (%)
14	1	0.25	0.25	4.62	5.58	5.06	8.86
	2	1.13	1.13	0.05	0.73	-0.75	2.62
	3	1.36	1.36	0.51	0.59	0.26	-
	4	1.47	1.58	1.48	3.50	2.39	-0.56
	5	0.13	0.13	5.96	5.99	-1.83	7.44
	6	0.38	0.38	0.01	0.59	1.38	9.38
	7	0.48	0.48	0.00	1.07	1.05	-0.31
	8	2.28	2.28	1.35	1.43	3.99	2.37
	9	1.19	1.65	0.93	1.28	-1.06	2.85
	10	0.05	0.05	0.98	1.04	1.41	1.64
	11	0.26	0.26	16.23	17.02	8.78	15.00
	12	1.37	1.37	2.53	2.97	1.80	7.67
	13	2.38	2.38	3.18	3.33	2.72	-
	14	1.09	1.09	0.86	0.86	3.09	7.02
	15	1.75	1.75	8.50	8.50	1.84	8.07
	16	3.16	3.16	0.58	1.10	-2.19	-
	17	0.00	0.00	0.16	4.22	-3.36	0.62
	18	1.80	1.80	7.26	7.62	5.46	10.09
	19	0.01	0.01	4.42	4.57	1.14	-0.92
	20	2.46	2.46	2.68	2.96	-3.24	14.92
50	1	2.59	2.14	2.02	2.31	-	-
	2	0.04	1.10	0.89	4.75	-	-
	3	0.98	2.69	1.55	7.48	-	-
	4	10.87	12.05	11.59	14.17	-	-
	5	5.86	4.57	5.70	5.88	-	-
	6	2.70	3.69	0.30	3.35	-	-
	7	0.09	3.21	4.44	6.64	-	-
	8	2.93	3.13	4.15	4.48	-	-
	9	2.01	1.29	0.21	2.89	-	-
	10	2.99	2.81	1.41	2.71	-	-

(-) : Aucune solution réalisable n'est disponible

Conclusions et perspectives de recherches

Dans ce manuscrit, nous avons étudié un problème de transport et de la gestion de production dans un contexte sensibilisé aux enjeux environnementaux. Ce problème de planification opérationnelle optimise conjointement les décisions de production, de stockage, d'approvisionnement et de tournées de véhicules. A notre connaissance, les décisions relatives à la production considèrent simultanément pour la première fois les opérations d'assemblage et de désassemblage. En effet, en pratique, des nombreuses usines produisent des articles avec des composants neufs et usagés désassemblés à partir de produits retournés. Généralement, dans ce type de système de production complexe, les deux opérations sont traitées indépendamment. Dans notre étude, différents types de composants neufs ou désassemblés sont nécessaires afin d'assembler un produit final. Les décisions relatives à l'approvisionnement et les tournées de véhicules concernent l'achat des produits en fin de vie et des matières premières. Le problème étudié dans cette thèse généralise le problème de planification de tournées de véhicules et d'assemblage. Le plan établi au début de l'horizon de planification doit répondre à des objectifs de satisfaction des demandes de clients, de minimisation de différents coûts agrégés et éventuellement de prendre en compte le cas incertain des quantités de composants désassemblés. Ce travail permet la valorisation des produits en fin de vie en les collectant et réutilisant leurs composants.

5.1 Contributions

Dans cette thèse, diverses contributions sont apportées. D'abord, nous avons étudié une nouvelle problématique qui intègre les opérations d'assemblage et de désassemblage avec des tournées de véhicules en considérant des produits en retours. Dans le cas déterministe, nous avons adapté un modèle de programmation mathématique ainsi qu'une mat-heuristique, capable de résoudre les différentes versions statiques et dynamiques du problème intégré. Nous avons abordé ce problème avec une incertitude sur la quantité de composants désassemblé et adapté en conséquence la méthode de résolution.

Nous avons fourni des analyses pour cinq hypothèses, où différents niveaux de flexibilité sont donnés aux décisions, ainsi que les résultats d'expériences de calcul étendues.

Planification intégrée du transport et de la gestion de production avec des produits en retours

Motivés par des nombreuses applications pratiques et les raisons environnemental et économique mentionnées au chapitre 1, nous avons étudié le problème intégré du transport et de la gestion de production avec des produits en retour. Ce problème optimise conjointement la planification intégrée du transport qui concerne l'activité d'approvisionnement, de la production et des stocks dans un horizon de planification fini dans le cas statique et dynamique. Par ailleurs, le problème de production considère les décisions intégrées d'assemblage et désassemblage. Dans le chapitre 3, nous avons introduit ce problème et nous avons proposé une formulation de programmation mathématique pour modéliser ce problème intégré. En premier lieu, nous avons prouvé l'avantage de considérer les produits en retour dans le processus de production. En second lieu, nous avons prouvé l'avantage de l'approche intégrée en comparaison avec l'approche hiérarchique. Ce problème est traité dans un article qui est déjà accepté dans le journal *European Journal of Industrial Engineering* .

Désassemblage avec rendements stochastiques

La plupart des travaux actuels qui sont présentés dans le chapitre 2, sur le problème de planification de désassemblage et ses variantes omettent l'aspect incertain de l'état des composants désassemblés en raison de sa complexité, dont il s'agit d'une abstraction du problème réel. Dans le chapitre 4, nous avons étudié le cas stochastique du problème intégré où les quantités des composants désassemblés sont incertains. Le problème est modélisé avec un modèle mathématique robuste basé sur l'approche par scénarios. Nous avons abordé différentes hypothèses, où des niveaux de flexibilité sont donnés aux différentes décisions qui peuvent ou non être impactés par les scénarios. Les analyses indiquent que le fait de permettre une certaine flexibilité dans le processus d'achat améliore le coût total et le plus simple à le mettre en oeuvre. Ce problème est traité dans un article qui est déjà soumis au journal *Expert Systems With Applications*.

Méthodes de résolutions et expérimentations

L'originalité des modèles présentés dans cette thèse exige le développement des algorithmes capables de les résoudre d'une manière efficace. Dans le chapitre 3, nous avons développé une mat-heuristique pour résoudre les instances de grande taille. L'approche proposée consiste à résoudre le problème de production avec un solveur CPLEX. Pour la partie transport, une méthode approchée de type recherche à voisinage variable est utilisée pour trouver les meilleurs itinéraires.

L'idée principale est de décomposer le problème en sous-problèmes et de résoudre ces sous-problèmes de manière itérative jusqu'à ce qu'un optimum local soit trouvé. Nous avons proposé également des mécanismes de diversification efficaces. La recherche se poursuit jusqu'à ce qu'une condition d'arrêt, par exemple une limite de temps, soit atteinte. Pour avoir une méthode de résolution efficace, le problème de production doit inclure une approximation de coût appropriée de problème de transport. La pertinence de l'approximation dépend du niveau des décisions à prendre dans le sous-problème actuel. Par exemple, lorsque, dans un sous-problème qui s'intéresse aux décisions relatives à la taille de production, le coût approximatif de la visite de chaque nœud (fournisseur ou client) est nécessaire pour intégrer des informations plus détaillées sur le coût de tournées de véhicules dans la fonction objectif. Un autre élément essentiel de notre méthode est la communication efficace entre les différents sous-problèmes. La communication concerne ici les informations à transmettre d'un problème à l'autre et comment la considérer. Nous avons proposé plusieurs mécanismes de mise à jour pour approximer le coût du transport.

Dans le chapitre 4, nous avons présenté comment adapter cette méthode pour résoudre le cas stochastique. Une étape est ajoutée à notre mat-heuristique pour optimiser le pire cas dans tous les scénarios. Nous avons généré des instances de petite, moyenne et grande taille pour ce problème intégré pour le cas déterministe (chapitre 3), et pour le cas stochastique (chapitre 4).

Nous avons présenté les résultats des expériences de calcul sur les nouvelles instances que nous avons générées. Les expérimentations dans le cas déterministe montrent les avantages du modèle intégré par rapport à l'approche hiérarchique, ainsi que l'efficacité de mat-heuristique en terme de la qualité de solutions et les temps d'exécution qui est plus rapide pour les instances de grande taille par rapport au solveur commercial CPLEX. D'après les résultats de cas stochastique, l'approche mat-heuristique est plus rapide que les autres méthodes et fournit des solutions de bonne qualité, mais la PLNE avec la solution initiale de la mat-heuristique est plus efficace.

5.2 Perspectives de recherches

Cette thèse ouvre plusieurs portes pour des futures recherches. Les possibilités peuvent être classées en deux grandes directions, la modélisation et la méthodologie de résolution.

Directions des futures recherches sur la modélisation

La première direction de recherche possible consiste à intégrer les décisions de transport en amont et en aval pour une usine de fabrication centrale. Il s'agit de combiner les trois activités d'approvisionnement, de production et de distribution ensemble. Plusieurs stratégies de collecte et de livraison peuvent être envisagées. Par exemple les routes mixtes ou séparées. Dans le cas de routes mixtes, les fournisseurs et les clients peuvent être visités sur la même route. Dans ce cas, les composants et les produits finis peuvent être transportés dans le même camion. Dans le cas de routes séparées, les fournisseurs et les clients doivent être visités sur des itinéraires séparés.

D'autres possibilités pour les futures directions de recherche sur le plan de la modélisation pour le problème de production et de transport. Nous pouvons étudier la variante multi-produits dans laquelle différents produits finis nécessitent des sous-ensembles de composants (qui se chevauchent). De plus, nous pouvons ajouter les contraintes énergétiques dans le processus de production en intégrant le problème de sélection des sources d'énergie.

Les différents axes de recherche en dessus peuvent être étudiés aussi dans le cas stochastique du problème. Dans la réalité, la demande des clients aussi la quantité des produits en retour sont souvent incertaines et seule une partie des informations est connue au début de l'horizon de planification. Dans ce dernier, nous pouvons considérer la demande de produits finis et la quantité des composants chez les fournisseurs comme source d'incertitude.

Directions pour les futures recherches sur la méthodologie

La mat-heuristique présentée dans cette thèse peut être adaptée et appliquée à des nombreuses variantes de problème de transport et la gestion de production intégrées. De plus, ce cadre peut être appliqué à d'autres problèmes de chaîne logistique intégrée qui considère la distribution comme faisant partie du processus de prise de décision.

Dans cette thèse, un modèle robuste est proposé pour étudier la nature incertaine de la quantité des composants désassemblés, cependant, la programmation stochastique peut être proposée. Les études futures peuvent aborder ce point en développant par exemple une programmation stochastique à deux

étapes ainsi que des méthodes exactes telles que la décomposition de Benders, la contrainte de hasard, aussi les approches analytiques.

Liste de publications

Articles de revues

- Frifita, S., Afsar, H. M., & Hnaien, F. A robust optimization approach for the disassembly assembly routing problem under uncertain yields. *Expert Systems With Applications* (soumis).
- Frifita, S., Afsar, H. M., & Hnaien, F. An efficient mat-heuristic algorithm for the dynamic disassembly assembly routing problem with returns. *European Journal of Industrial Engineering* (accepté).

Communications dans une conférence internationale avec comité de lecture

- Frifita, S., Afsar, H. M., & Hnaien, F. (2021). Robust Disassembly Assembly Routing Problem With Returns Under Uncertain Yields. *IFAC-PapersOnLine*, 54(2), 354-359.

Communications orales sans actes dans un congrès

- Frifita, S., Afsar, H. M., & Hnaien, F. New formulation for dynamic assembly routing problem with returns the annual conference of the French Society for Operational Research and Decision Aiding, 18 22 February 2019 Le Havre France.
- Frifita, S., Afsar, H. M., & Hnaien, F. A mat-heuristic approach to solve the integrated production and routing problem with returns the annual conference of the French Society for Operational Research and Decision Aiding, 18 22 February 2020 Montpellier France.

Bibliographie

- Absi, N., Archetti, C., Dauzère-Pérès, S., & Feillet, D. (2015a). A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4), 784–795.
- Absi, N., Archetti, C., Dauzère-Pérès, S., & Feillet, D. (2015b). A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4), 784–795.
- Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., & Speranza, M. G. (2018). Comparing sequential and integrated approaches for the production routing problem. *European Journal of Operational Research*, 269(2), 633–646.
- Adulyasak, Y., Cordeau, J.-F., & Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4), 851–867.
- Afsar, H. M., & Hnaien, F. (2020). Formulations and solution algorithms for dynamic assembly routing problem. *International Journal of Production Research*, 58(3), 671–688.
- Agra, A., Requejo, C., & Rodrigues, F. (2018). A hybrid heuristic for a stochastic production-inventory-routing problem. *Electronic Notes in Discrete Mathematics*, 64, 345–354.
- Aksen, D., Altinkemer, K., & Chand, S. (2003). The single-item lot-sizing problem with immediate lost sales. *European Journal of Operational Research*, 147(3), 558–566.
- Alumur, S., & Kara, B. Y. (2007). A new model for the hazardous waste location-routing problem. *Computers & Operations Research*, 34(5), 1406–1423.
- Andersson, H., Hoff, A., Christiansen, M., Hasle, G., & Løkketangen, A. (2010). Industrial aspects and literature survey : Combined inventory management and routing. *Computers & Operations Research*, 37(9), 1515–1536.
- Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation science*, 41(3), 382–391.

- Archetti, C., Bertazzi, L., Paletta, G., & Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, *38*(12), 1731–1746.
- Armentano, V. A., Shiguemoto, A. L., & Løkketangen, A. (2011). Tabu search with path relinking for an integrated production–distribution problem. *Computers & Operations Research*, *38*(8), 1199–1209.
- Babagolzadeh, M., Shrestha, A., Abbasi, B., Zhang, Y., Woodhead, A., & Zhang, A. (2020). Sustainable cold supply chain management under demand uncertainty and carbon tax regulation. *Transportation Research Part D : Transport and Environment*, *80*, 102245.
- Baddeley, A. D. (1966). The capacity for generating information by randomization. *The Quarterly journal of experimental psychology*, *18*(2), 119–129.
- Bai, C., & Sarkis, J. (2013). Flexibility in reverse logistics : a framework and evaluation approach. *Journal of Cleaner Production*, *47*, 306–318.
- Bai, X., Davis, J. S., Kanet, J. J., Cantrell, S., & Patterson, J. W. (2002). Schedule instability, service level and cost in a material requirements planning system. *International Journal of Production Research*, *40*(7), 1725–1758.
- Baki, M. F., Chaouch, B. A., & Abdul-Kader, W. (2014). A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery. *Computers & Operations Research*, *43*, 225–236.
- Bal, A., & Satoglu, S. I. (2018). A goal programming model for sustainable reverse logistics operations planning and an application. *Journal of cleaner production*, *201*, 1081–1091.
- Baldacci, R., Hadjiconstantinou, E., & Mingozzi, A. (2004). An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations research*, *52*(5), 723–738.
- Barba-Gutiérrez, Y., & Adenso-Díaz, B. (2009). Reverse mrp under uncertain and imprecise demand. *The International Journal of Advanced Manufacturing Technology*, *40*(3-4), 413–424.
- Barba-Gutiérrez, Y., Adenso-Díaz, B., & Hopp, M. (2008). An analysis of some environmental consequences of european electrical and electronic waste regulation. *Resources, Conservation and Recycling*, *52*(3), 481–495.
- Bard, J. F., & Nananukul, N. (2009). The integrated production–inventory–distribution–routing problem. *Journal of Scheduling*, *12*(3), 257–280.

- Bard, J. F., & Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, *37*(12), 2202–2217.
- Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G., & Prutzman, P. J. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, *13*(6), 4–23.
- Ben-Tal, A., & Nemirovski, A. (1998). Robust convex optimization. *Mathematics of operations research*, *23*(4), 769–805.
- Benton, W. (1991). Safety stock and service levels in periodic review inventory systems. *Journal of the operational research society*, *42*(12), 1087–1095.
- Berman, O., & Wang, Q. (2006). Inbound logistic planning : minimizing transportation and inventory cost. *Transportation science*, *40*(3), 287–299.
- Billington, P. J., McClain, J. O., & Thomas, L. J. (1983). Mathematical programming approaches to capacity-constrained mrp systems : review, formulation and problem reduction. *Management Science*, *29*(10), 1126–1141.
- Bookbinder, J. H., & Tan, J.-Y. (1988). Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, *34*(9), 1096–1108.
- Boudia, M., Louly, M. A. O., & Prins, C. (2007). A reactive grasp and path relinking for a combined production–distribution problem. *Computers & Operations Research*, *34*(11), 3402–3419.
- Boudia, M., & Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production–distribution problem. *European Journal of Operational Research*, *195*(3), 703–715.
- Brahimi, N., Absi, N., Dauzère-Pérès, S., & Nordli, A. (2017). Single-item dynamic lot-sizing problems : An updated survey. *European Journal of Operational Research*, *263*(3), 838–863.
- Brahimi, N., & Aouam, T. (2016). Multi-item production routing problem with backordering : a milp approach. *International Journal of Production Research*, *54*(4), 1076–1093.
- Brennan, L., Gupta, S. M., & Taleb, K. N. (1994). Operations planning issues in an assembly/disassembly environment. *International Journal of Operations & Production Management*.

- Carter, C. R., & Ellram, L. M. (1998). Reverse logistics : a review of the literature and framework for future investigation. *Journal of business logistics*, 19(1), 85.
- Carvalho, D. M., & Nascimento, M. C. (2018). A kernel search to the multi-plant capacitated lot sizing problem with setup carry-over. *Computers & Operations Research*, 100, 43–53.
- Chand, S., Hsu, V. N., Sethi, S., & Deshpande, V. (2007). A dynamic lot sizing problem with multiple customers : customer-specific shipping and backlogging costs. *IIE Transactions*, 39(11), 1059–1069.
- Chandra, P. (1993). A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society*, 44(7), 681–692.
- Chandra, P., Fisher, M. L., et al. (1994). Coordination of production and distribution planning. *European journal of operational research*, 72(3), 503–517.
- Chitsaz, M., Cordeau, J.-F., & Jans, R. (2019). A unified decomposition matheuristic for assembly, production, and inventory routing. *INFORMS Journal on Computing*, 31(1), 134–152.
- Chitsaz, M., Cordeau, J.-F., & Jans, R. (2020). A branch-and-cut algorithm for an assembly routing problem. *European Journal of Operational Research*, 282(3), 896–910.
- Chouinard, M., D'Amours, S., & Aït-Kadi, D. (2007). Design of reverse logistics networks for multiproducts, multistates, and multiprocessing alternatives. *Trends in Supply Chain Design and Management*, (pp. 181–211).
- Clark, A. R., & Armentano, V. A. (1995). A heuristic for a resource-capacitated multi-stage lot-sizing problem with lead times. *Journal of the Operational research Society*, 46(10), 1208–1222.
- Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C : Emerging Technologies*, 24, 270–287.
- Corso, L. L., & Wallace, M. (2015). A hybrid method for transportation with stochastic demand. *International journal of logistics research and applications*, 18(4), 342–354.

- Cunha, J. O., Kramer, H. H., & Melo, R. A. (2019). Effective matheuristics for the multi-item capacitated lot-sizing problem with remanufacturing. *Computers & Operations Research*, *104*, 149–158.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, *6*(1), 80–91.
- Dat, L. Q., Linh, D. T. T., Chou, S.-Y., & Vincent, F. Y. (2012). Optimizing reverse logistic costs for recycling end-of-life electrical and electronic products. *Expert systems with applications*, *39*(7), 6380–6387.
- De Brito, M. P., & Van Der Laan, E. A. (2009). Inventory control with product returns : The impact of imperfect information. *European journal of operational research*, *194*(1), 85–101.
- Devoto, C., Fernández, E., & Piñeyro, P. (2020). The economic lot-sizing problem with remanufacturing and inspection for grading heterogeneous returns. *Journal of Remanufacturing*, (pp. 1–17).
- Díaz-Madroñero, M., Mula, J., & Peidro, D. (2017). A mathematical programming model for integrating production and procurement transport decisions. *Applied Mathematical Modelling*, *52*, 527–543.
- Dinler, E., & Güngör, Z. (2017). Planning decisions for recycling products containing hazardous and explosive substances : A fuzzy multi-objective model. *Resources, Conservation and Recycling*, *117*, 93–101.
- Dolgui, A., & Prodhon, C. (2007). Supply planning under uncertainties in mrp environments : A state of the art. *Annual reviews in control*, *31*(2), 269–279.
- Duta, L. (2006). *Contribution à l'étude de la conduite des systèmes de désassemblage..* Ph.D. thesis, Université de Franche-Comté ; Université de Bucarest.
- Eppen, G. D., & Martin, R. K. (1987). Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, *35*(6), 832–848.
- Farhat, M., Akbalik, A., Hadj-Alouane, A. B., & Sauer, N. (2019). Lot sizing problem with batch ordering under periodic buyback contract and lost sales. *International Journal of Production Economics*, *208*, 500–511.
- Federgruen, A., & Tzur, M. (1999). Time-partitioning heuristics : Application to one warehouse, multiitem, multiretailer lot-sizing problems. *Naval Research Logistics (NRL)*, *46*(5), 463–486.

- Ferrer, G., & Whybark, D. C. (2001). Material planning for a remanufacturing facility. *Production and Operations Management*, 10(2), 112–124.
- Finke, G. (1984). A two-commodity network flow approach to the traveling salesman problem. *Congresses Numeration*, 41, 167–178.
- Fiorotto, D. J., Jans, R., & de Araujo, S. A. (2018). Process flexibility and the chaining principle in lot sizing problems. *International Journal of Production Economics*, 204, 244–263.
- Florian, M., & Klein, M. (1971). Deterministic production planning with concave costs and capacity constraints. *Management Science*, 18(1), 12–20.
- Florio, A. M., Hartl, R. F., & Minner, S. (2020). Optimal a priori tour and restocking policy for the single-vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 285(1), 172–182.
- Frifita, S., & Masmoudi, M. (2020). Vns methods for home care routing and scheduling problem with temporal dependencies, and multiple structures and specialties. *International Transactions in Operational Research*, 27(1), 291–313.
- Frifita, S., Mathlouthi, I., Masmoudi, M., & Dammak, A. (2020). Variable neighborhood search based algorithms to solve a rich k-travelling repairmen problem. *Optimization Letters*, (pp. 1–15).
- Gao, N., & Chen, W. (2008). A genetic algorithm for disassembly scheduling with assembly product structure. In *2008 IEEE International Conference on Service Operations and Logistics, and Informatics*, vol. 2, (pp. 2238–2243). IEEE.
- Gendreau, M., Jabali, O., & Rei, W. (2016). 50th anniversary invited article—future research directions in stochastic vehicle routing. *Transportation Science*, 50(4), 1163–1173.
- Godichaud, M., Amodeo, L., & Hrouga, M. (2015). Metaheuristic based optimization for capacitated disassembly lot sizing problem with lost sales. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)*, (pp. 1329–1335). IEEE.
- Goodson, J. C., Thomas, B. W., & Ohlmann, J. W. (2017). A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research*, 258(1), 216–229.

- Gruler, A., Panadero, J., de Armas, J., Pérez, J. A. M., & Juan, A. A. (2020). A variable neighborhood search simheuristic for the multiperiod inventory routing problem with stochastic demands. *International Transactions in Operational Research*, 27(1), 314–335.
- Guide Jr, V. D. R. (2000). Production planning and control for remanufacturing : industry practice and research needs. *Journal of operations Management*, 18(4), 467–483.
- Guide Jr, V. D. R., & Srivastava, R. (2000). A review of techniques for buffering against uncertainty with mrp systems. *Production Planning & Control*, 11(3), 223–233.
- Gungor, A., & Gupta, S. M. (1997). An evaluation methodology for disassembly processes. *Computers & Industrial Engineering*, 33(1-2), 329–332.
- Gupta, S., & Taleb, K. (1994). Scheduling disassembly. *The International Journal of Production Research*, 32(8), 1857–1866.
- Gurkan, M. E., & Tunc, H. (2021). A fix-and-optimize heuristic for the capacitated multi-item stochastic lot-sizing problem. *An International Journal of Optimization and Control : Theories & Applications (IJOCTA)*, 11(1), 41–51.
- Gutiérrez, J., Puerto, J., & Sicilia, J. (2004). The multiscenario lot size problem with concave costs. *European Journal of Operational Research*, 156(1), 162–182.
- Haase, K., & Kimms, A. (2000). Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, 66(2), 159–169.
- Habibi, M. K., Battaïa, O., Cung, V.-D., & Dolgui, A. (2017). Collection-disassembly problem in reverse supply chain. *International Journal of Production Economics*, 183, 334–344.
- Habibi, M. K. K., Battaïa, O., Cung, V.-D., Dolgui, A., & Tiwari, M. K. (2019). Sample average approximation for multi-vehicle collection-disassembly problem under uncertainty. *International Journal of Production Research*, 57(8), 2409–2428.
- Hardin, J. R., Nemhauser, G. L., & Savelsbergh, M. W. (2007). Analysis of bounds for a capacitated single-item lot-sizing problem. *Computers & operations research*, 34(6), 1721–1743.

- Hein, F., & Almeder, C. (2016). Quantitative insights into the integrated supply vehicle routing and production planning problem. *International Journal of Production Economics*, 177, 66–76.
- Hein, F., Almeder, C., Figueira, G., & Almada-Lobo, B. (2018). Designing new heuristics for the capacitated lot sizing problem by genetic programming. *Computers & Operations Research*, 96, 1–14.
- Heizer, J., & Render, B. (2013). *Operations management : sustainability and supply chain management*. Pearson Higher Ed.
- Helber, S., Sahling, F., & Schimmelpfeng, K. (2013). Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR spectrum*, 35(1), 75–105.
- Hervani, A. A., Helms, M. M., & Sarkis, J. (2005). Performance measurement for green supply chain management. *Benchmarking : An international journal*.
- Hnaien, F., & Afsar, H. M. (2017). Robust single-item lot-sizing problems with discrete-scenario lead time. *International Journal of Production Economics*, 185, 223–229.
- Hrouga, M., Godichaud, M., & Amodeo, L. (2016). Heuristics for multi-product capacitated disassembly lot sizing with lost sales. *IFAC-PapersOnLine*, 49(12), 628–633.
- Hu, C., Lu, J., Liu, X., & Zhang, G. (2018). Robust vehicle routing problem with hard time windows under demand and travel time uncertainty. *Computers & Operations Research*, 94, 139–153.
- Hui, W., Dong, X., & Guanhong, D. (2008). A genetic algorithm for product disassembly sequence planning. *Neurocomputing*, 71(13-15), 2720–2726.
- Iasya, A., & Handayati, Y. (2015). Material requirement planning analysis in micro, small and medium enterprise case study : grooveline-an apparel outsourcing company final project. *vol*, 4, 317–329.
- Inderfurth, K. (2003). *Analytical solution for a single period production inventory problem with uniformly distributed yield and demand*. Otto-von-Guericke-Univ., Fak. für Wirtschaftswiss.
- Inderfurth, K. (2009). How to protect against demand and yield risks in mrp systems. *International Journal of Production Economics*, 121(2), 474–481.

- Inderfurth, K., & van der Laan, E. (2001). Leadtime effects and policy improvement for stochastic inventory control with remanufacturing. *International Journal of Production Economics*, 71(1-3), 381–390.
- Inderfurth, K., Vogelgesang, S., & Langella, I. M. (2015). How yield process misspecification affects the solution of disassemble-to-order problems. *International Journal of Production Economics*, 169, 56–67.
- Jahangir, H., Mohammadi, M., Pasandideh, S. H. R., & Nobari, N. Z. (2019). Comparing performance of genetic and discrete invasive weed optimization algorithms for solving the inventory routing problem with an incremental delivery. *Journal of Intelligent Manufacturing*, 30(6), 2327–2353.
- Ji, X., Zhang, Z., Huang, S., & Li, L. (2016). Capacitated disassembly scheduling with parts commonality and start-up cost and its industrial application. *International Journal of Production Research*, 54(4), 1225–1243.
- Jiao, W., Zhang, J.-L., & Yan, H. (2017a). The stochastic lot-sizing problem with quantity discounts. *Computers & Operations Research*, 80, 1–10.
- Jiao, W., Zhang, J.-L., & Yan, H. (2017b). The stochastic lot-sizing problem with quantity discounts. *Computers & Operations Research*, 80, 1–10.
- Kadipasaoglu, S. N., & Sridharan, V. (1995). Alternative approaches for reducing schedule instability in multistage manufacturing under demand uncertainty. *Journal of Operations Management*, 13(3), 193–211.
- Kim, H.-J., Lee, D.-H., Xirouchakis, P., & Kwon, O. (2009). A branch and bound algorithm for disassembly scheduling with assembly product structure. *Journal of the Operational Research Society*, 60(3), 419–430.
- Kim, H.-J., & Xirouchakis, P. (2010). Capacitated disassembly scheduling with random demand. *International Journal of Production Research*, 48(23), 7177–7194.
- Koken, P., Raghavan, V. A., & Yoon, S. W. (2018). A genetic algorithm based heuristic for dynamic lot sizing problem with returns and hybrid products. *Computers & Industrial Engineering*, 119, 453–464.
- Kongar, E., & Gupta, S. M. (2006). Disassembly to order system under uncertainty. *Omega*, 34(6), 550–561.
- Krarpup, J., & Bilde, O. (1977). Plant location, set covering and economic lot size : An 0 (mn)-algorithm for structured problems. In *Numerische methoden bei optimierungsaufgaben band 3*, (pp. 155–180). Springer.

- Krikke, H. R., Van Harten, A., & Schuur, P. (1999). Business case roteb : recovery strategies for monitors. *Computers & Industrial Engineering*, 36(4), 739–757.
- Kuhn, H., & Liske, T. (2011). Simultaneous supply and production planning. *International Journal of Production Research*, 49(13), 3795–3813.
- Kuhn, H., & Liske, T. (2014). An exact algorithm for solving the economic lot and supply scheduling problem using a power-of-two policy. *Computers & operations research*, 51, 30–40.
- Kumar, V. V., & Chan, F. T. (2011). A superiority search and optimisation algorithm to solve rfid and an environmental factor embedded closed loop logistics model. *International Journal of Production Research*, 49(16), 4807–4831.
- Kuşakcı, A. O., Ayvaz, B., Cin, E., & Aydın, N. (2019). Optimization of reverse logistics network of end of life vehicles under fuzzy supply : A case study for istanbul metropolitan area. *Journal of cleaner production*, 215, 1036–1051.
- Lambert, A., & Gupta, S. M. (2005). A heuristic solution for the disassembly line balancing problem incorporating sequence dependent costs. In *Environmentally Conscious Manufacturing V*, vol. 5997, (p. 59970A). International Society for Optics and Photonics.
- Langella, I. M. (2007). Heuristics for demand-driven disassembly planning. *Computers & Operations Research*, 34(2), 552–577.
- Lee, C., Lee, K., & Park, S. (2012). Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63(9), 1294–1306.
- Lee, D., Kang, J., & Xirouchakis, P. (2001). Disassembly planning and scheduling : review and further research. *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture*, 215(5), 695–709.
- Lee, D., Kim, H., Choi, G., & Xirouchakis, P. (2004). Disassembly scheduling : integer programming models. *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture*, 218(10), 1357–1372.

- Lee, D.-H., & Xirouchakis, P. (2004). A two-stage heuristic for disassembly scheduling with assembly product structure. *Journal of the Operational Research Society*, 55(3), 287–297.
- Lei, H., Laporte, G., & Guo, B. (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research*, 38(12), 1775–1783.
- Li, C.-L., Hsu, V. N., & Xiao, W.-Q. (2004). Dynamic lot sizing with batch ordering and truckload discounts. *Operations Research*, 52(4), 639–654.
- Li, S. (2020). Constant approximation algorithm for nonuniform capacitated multi-item lot sizing via strong covering inequalities. *Mathematics of Operations Research*, 45(3), 947–965.
- Li, X., Tian, P., & Leung, S. C. (2010). Vehicle routing problems with time windows and stochastic travel and service times : Models and algorithm. *International Journal of Production Economics*, 125(1), 137–145.
- Li, Z., Çil, Z. A., Mete, S., & Kucukkoc, I. (2020). A fast branch, bound and remember algorithm for disassembly line balancing problem. *International Journal of Production Research*, 58(11), 3220–3234.
- Lieckens, K., & Vandaele, N. (2012). Multi-level reverse logistics network design under uncertainty. *International Journal of Production Research*, 50(1), 23–40.
- Liu, K., & Zhang, Z.-H. (2018). Capacitated disassembly scheduling under stochastic yield and demand. *European Journal of Operational Research*, 269(1), 244–257.
- Liu, M., Liu, X., Chu, F., Zheng, F., & Chu, C. (2020). An exact method for disassembly line balancing problem with limited distributional information. *International Journal of Production Research*, (pp. 1–18).
- Lmariouh, J., El Hachemi, N., Jamali, A., Bouami, D., Rousseau, L. M., et al. (2019). An integrated production and distribution problem with direct shipment : a case from moroccan bottled-water market. *International Journal of Operational Research*, 34(1), 144–160.
- Louly, M.-A., & Dolgui, A. (2013). Optimal mrp parameters for a single item inventory with random replenishment lead time, poq policy and service level constraint. *International Journal of Production Economics*, 143(1), 35–40.

- Louly, M.-A., Dolgui, A., & Hnaien, F. (2008). Optimal supply planning in mrp environments for assembly systems with random component procurement times. *International Journal of Production Research*, 46(19), 5441–5467.
- Mabert, V. A. (2007). The early road to material requirements planning. *Journal of Operations Management*, 25(2), 346–356.
- Maes, J., McClain, J. O., & Van Wassenhove, L. N. (1991). Multilevel capacitated lotsizing complexity and lp-based heuristics. *European Journal of Operational Research*, 53(2), 131–148.
- Marinaki, M., & Marinakis, Y. (2016). A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands. *Expert Systems with Applications*, 46, 145–163.
- Mehra, A., Minis, I., & Proth, J. (1996). Hierarchical production planning for complex manufacturing systems. *Advances in Engineering Software*, 26(3), 209–218.
- Melo, R. A., & Wolsey, L. A. (2012). Mip formulations and heuristics for two-level production-transportation problems. *Computers & Operations Research*, 39(11), 2776–2786.
- Mete, S., Çil, Z. A., Özceylan, E., Ağpak, K., & Battaïa, O. (2018). An optimisation support for the design of hybrid production lines including assembly and disassembly tasks. *International Journal of Production Research*, 56(24), 7375–7389.
- Miller, A. J., Nemhauser, G. L., Savelsbergh, M. W., et al. (2000). Solving multi-item capacitated lot-sizing problems with setup times by branch-and-cut. Tech. rep., Université Catholique de Louvain. Center for Operations Research and
- Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4), 326–329.
- Miranda, P. L., Cordeau, J.-F., Ferreira, D., Jans, R., & Morabito, R. (2018). A decomposition heuristic for a rich production routing problem. *Computers & Operations Research*, 98, 211–230.
- Mjirda, A., Jarbouï, B., Macedo, R., Hanafi, S., & Mladenović, N. (2014). A two phase variable neighborhood search for the multi-product inventory routing problem. *Computers & Operations Research*, 52, 291–299.

- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11), 1097–1100.
- Montagné, R., Gamache, M., & Gendreau, M. (2019). A shortest path-based algorithm for the inventory routing problem of waste vegetable oil collection. *Journal of the Operational Research Society*, 70(6), 986–997.
- Moon, I., Ha, B.-H., & Kim, J. (2012). Inventory systems with variable capacity. *European Journal of Industrial Engineering*, 6(1), 68–86.
- Munasinghe, M. (1993). *Environmental economics and sustainable development*. The World Bank.
- Ndanguza, D., Nyirahabinshuti, A., & Sibosiko, C. (2020). Modeling the effects of toxic wastes on population dynamics. *Alexandria Engineering Journal*, 59(4), 2713–2723.
- Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.-F., Guimarães, L., & Jans, R. (2019). Solving a large multi-product production-routing problem with delivery time windows. *Omega*, 86, 154–172.
- Nourelfath, M. (2011). Service level robustness in stochastic production planning under random machine breakdowns. *European Journal of Operational Research*, 212(1), 81–88.
- Özceylan, E., Demirel, N., Çetinkaya, C., & Demirel, E. (2017). A closed-loop supply chain network design for automotive industry in turkey. *Computers & industrial engineering*, 113, 727–745.
- Pineyro, P., & Viera, O. (2010). The economic lot-sizing problem with remanufacturing and one-way substitution. *International Journal of Production Economics*, 124(2), 482–488.
- Pochampally, K. K., & Gupta, S. M. (2008). A multiphase fuzzy logic approach to strategic planning of a reverse supply chain network. *IEEE Transactions on Electronics Packaging Manufacturing*, 31(1), 72–82.
- Pochampally, K. K., Nukala, S., & Gupta, S. M. (2009). Eco-procurement strategies for environmentally conscious manufacturers. *International Journal of Logistics Systems and Management*, 5(1-2), 106–122.
- Pour-Massahian-Tafti, M., Godichaud, M., & Amodeo, L. (2020). New models and efficient methods for single-product disassembly lot-sizing problem with surplus inventory decisions. *International Journal of Production Research*, (pp. 1–21).

- Prahinski, C., & Kocabasoglu, C. (2006). Empirical research opportunities in reverse supply chains. *Omega*, *34*(6), 519–532.
- Prakash, P., Ceglarek, D., & Tiwari, M. (2012). Constraint-based simulated annealing (cbsa) approach to solve the disassembly scheduling problem. *The International Journal of Advanced Manufacturing Technology*, *60*(9–12), 1125–1137.
- Qiu, Y., Ni, M., Wang, L., Li, Q., Fang, X., & Pardalos, P. M. (2018). Production routing problems with reverse logistics and remanufacturing. *Transportation Research Part E : Logistics and Transportation Review*, *111*, 87–100.
- Qiu, Y., Qiao, J., & Pardalos, P. M. (2017). A branch-and-price algorithm for production routing problems with carbon cap-and-trade. *Omega*, *68*, 49–61.
- Rapine, C., Penz, B., Gicquel, C., & Akbalik, A. (2018). Capacity acquisition for the single-item lot sizing problem under energy constraints. *Omega*, *81*, 112–122.
- Rizk, N., Martel, A., & D'Amours, S. (2006). Multi-item dynamic production-distribution planning in process industries with divergent finishing stages. *Computers & operations research*, *33*(12), 3600–3623.
- Rodoplu, M., Arbaoui, T., & Yalaoui, A. (2020). A fix-and-relax heuristic for the single-item lot-sizing problem with a flow-shop system and energy constraints. *International Journal of Production Research*, *58*(21), 6532–6552.
- Rogers, D. S., & Tibben-Lembke, R. S. (1998). Going backwards : reverse logistics trends and practices, university of nevada. *Reno Center for Logistics Management*.
- Rogers, D. S., Tibben-Lembke, R. S., et al. (1999). *Going backwards : reverse logistics trends and practices*, vol. 2. Reverse Logistics Executive Council Pittsburgh, PA.
- Rossi, R., Kilic, O. A., & Tarim, S. A. (2015). Piecewise linear approximations for the static–dynamic uncertainty strategy in stochastic lot-sizing. *Omega*, *50*, 126–140.
- Ruokokoski, M., Solyali, O., Cordeau, J.-F., Jans, R., & Süral, H. (2010). Efficient formulations and a branch-and-cut algorithm for a production-routing problem. *Les Cahiers du GERAD*.

- Rusdiansyah, A., & Tsao, D.-b. (2005). An integrated model of the periodic delivery problems for vending-machine supply chains. *Journal of Food Engineering*, 70(3), 421–434.
- Sahling, F. (2013). A column-generation approach for a short-term production planning problem in closed-loop supply chains. *Business Research*, 6(1), 55–75.
- Salameh, M., & Jaber, M. (2000). Economic production quantity model for items with imperfect quality. *International journal of production economics*, 64(1-3), 59–64.
- Salini, P. (1997). Road transport : questions about a durable growth. do we have to be afraid of the“external effects“ ? ; transport routier : questions sur un developpement durable. faut-il craindre les“effets externes“ ?
- Satoglu, S. I., & Sahin, I. E. (2013). Design of a just-in-time periodic material supply system for the assembly lines and an application in electronics industry. *The International Journal of Advanced Manufacturing Technology*, 65(1-4), 319–332.
- Savelsbergh, M., & Song, J.-H. (2007). Inventory routing with continuous moves. *Computers & operations research*, 34(6), 1744–1763.
- Savelsbergh, M., & Song, J.-H. (2008). An optimization algorithm for the inventory routing problem with continuous moves. *Computers & operations research*, 35(7), 2266–2282.
- Schmidt, M., Münzberg, B., & Nyhuis, P. (2015). Determining lot sizes in production areas—exact calculations versus research based estimation. *Procedia CIRP*, 28, 143–148.
- Schultmann, F., Zumkeller, M., & Rentz, O. (2006). Modeling reverse logistic tasks within closed-loop supply chains : An example from the automotive industry. *European journal of operational research*, 171(3), 1033–1050.
- Shih, L.-H. (2001). Reverse logistics system planning for recycling electrical appliances and computers in taiwan. *Resources, conservation and recycling*, 32(1), 55–72.
- Shokouhyar, S., & Aalirezai, A. (2017). Designing a sustainable recovery network for waste from electrical and electronic equipment using a genetic algorithm. *International Journal of Environment and Sustainable Development*, 16(1), 60–79.

- Shuang, Y., Diabat, A., & Liao, Y. (2019). A stochastic reverse logistics production routing model with emissions control policy selection. *International Journal of Production Economics*, 213, 201–216.
- Sifaleras, A., Konstantaras, I., & Mladenović, N. (2015). Variable neighborhood search for the economic lot sizing problem with product returns and recovery. *International Journal of Production Economics*, 160, 133–143.
- Slama, I., Ben-Ammar, O., Dolgui, A., & Masmoudi, F. (2020). New mixed integer approach to solve a multi-level capacitated disassembly lot-sizing problem with defective items and backlogging. *Journal of Manufacturing Systems*, 56, 50–57.
- Solyalı, O., & Süral, H. (2009). A relaxation based solution approach for the inventory control and vehicle routing problem in vendor managed systems. In *Modeling, computation and Optimization*, (pp. 171–189). World Scientific.
- Solyalı, O., & Süral, H. (2012). The one-warehouse multi-retailer problem : Reformulation, classification, and computational results. *Annals of Operations Research*, 196(1), 517–541.
- Solyalı, O., & Süral, H. (2017). A multi-phase heuristic for the production routing problem. *Computers & Operations Research*, 87, 114–124.
- Sungur, I., Ordóñez, F., & Dessouky, M. (2008). A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *Iie Transactions*, 40(5), 509–523.
- Suzanne, E., Absi, N., Borodin, V., & van den Heuvel, W. (2020). A single-item lot-sizing problem with a by-product and inventory capacities. *European Journal of Operational Research*, 287(3), 844–855.
- Taleb, K. N., & Gupta, S. M. (1997). Disassembly of multiple product structures. *Computers & Industrial Engineering*, 32(4), 949–961.
- Taş, D., Gendreau, M., Jabali, O., & Jans, R. (2019). A capacitated lot sizing problem with stochastic setup times and overtime. *European Journal of Operational Research*, 273(1), 146–159.
- Tempelmeier, H., & Herpers, S. (2011). Dynamic uncapacitated lot sizing with random demand under a fillrate constraint. *European Journal of Operational Research*, 212(3), 497–507.

- Teunter, R. H., & Flapper, S. D. P. (2011). Optimal core acquisition and remanufacturing policies under uncertain core quality fractions. *European Journal of Operational Research*, 210(2), 241–248.
- Thevenin, S., Adulyasak, Y., & Cordeau, J.-F. (2020). Material requirements planning under demand uncertainty using stochastic optimization. *Production and Operations Management*.
- Thierry, M., Salomon, M., Van Nunen, J., & Van Wassenhove, L. (1995). Strategic issues in product recovery management. *California management review*, 37(2), 114–136.
- Thizy, J., & Van Wassenhove, L. N. (1985). Lagrangean relaxation for the multi-item capacitated lot-sizing problem : A heuristic implementation. *IIE transactions*, 17(4), 308–313.
- Tian, X., & Zhang, Z.-H. (2019). Capacitated disassembly scheduling and pricing of returned products with price-dependent yield. *Omega*, 84, 160–174.
- Tillman, F. A. (1969). The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3), 192–204.
- Ullrich, C., & Buscher, U. (2013). Flexible disassembly planning considering product conditions. *International Journal of Production Research*, 51(20), 6209–6228.
- Veerakamolmal, P., & Gupta, S. M. (2002). A case-based reasoning approach for automating disassembly process planning. *Journal of Intelligent Manufacturing*, 13(1), 47–60.
- Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, 5(1), 89–96.
- Wang, H.-F., & Huang, Y.-S. (2013). A two-stage robust programming approach to demand-driven disassembly planning for a closed-loop supply chain system. *International Journal of Production Research*, 51(8), 2414–2432.
- Widyadana, G. A., & Irohara, T. (2019). Modelling multi-tour inventory routing problem for deteriorating items with time windows. *Scientia Iranica*, 26(2), 932–941.
- Witthayapraphakorn, A., & Charnsethikul, P. (2019). Benders decomposition with special purpose method for the sub problem in lot sizing problem under uncertain demand. *Operations Research Perspectives*, 6, 100096.

- Wolsey, L. A. (1998). *Integer programming*, vol. 42. Wiley Online Library.
- Xu, W., Tang, Q., Liu, J., Liu, Z., Zhou, Z., & Pham, D. T. (2020). Disassembly sequence planning using discrete bees algorithm for human-robot collaboration in remanufacturing. *Robotics and Computer-Integrated Manufacturing*, 62, 101860.
- Youssef, Y., Lee, G., Godinez, C., Sutton, E., Klein, R. V., George, I. M., Seagull, F. J., & Park, A. (2011). Laparoscopic cholecystectomy poses physical injury risk to surgeons : analysis of hand technique and standing position. *Surgical endoscopy*, 25(7), 2168–2174.
- Zangwill, W. I. (1969). A backlogging model and a multi-echelon model of a dynamic economic lot size production system—a network approach. *Management Science*, 15(9), 506–527.
- Zhang, J., Lam, W. H., & Chen, B. Y. (2016). On-time delivery probabilistic models for the vehicle routing problem with stochastic demands and time windows. *European Journal of Operational Research*, 249(1), 144–154.
- Zhang, Q., Sundaramoorthy, A., Grossmann, I. E., & Pinto, J. M. (2017). Multiscale production routing in multicommodity supply chains with complex production facilities. *Computers & Operations Research*, 79, 207–222.
- Zhao, X., Lai, F., & Lee, T. (2001). Evaluation of safety stock methods in multilevel material requirements planning (mrp) systems. *Production Planning & Control*, 12(8), 794–803.
- Zhou, S., Zhou, Y., Zuo, X., Xiao, Y., & Cheng, Y. (2018). Modeling and solving the constrained multi-items lot-sizing problem with time-varying setup cost. *Chaos, Solitons & Fractals*, 116, 202–207.

Sana FRIFITA

Doctorat : Optimisation et Sûreté des Systèmes

Année 2021

Optimisation de problèmes de production et de transport intégré

Dans cette thèse, nous traitons essentiellement un problème intégré de tournées de véhicules avec la gestion de production. Ce problème combine les opérations d'assemblage et de désassemblage, avec la collecte des produits en fin de vie et des matières premières. Premièrement, nous étudions le cas déterministe du problème multi-périodes dans sa version statique et dynamique. Nous proposons un modèle mathématique linéaire en nombres entiers mixtes incluant ces nouvelles contraintes. Une mat-heuristique itérative à deux phases basée sur la programmation mathématique et un algorithme de recherche à voisinage variable, est également développée pour résoudre les instances de plus grande taille. Deuxièmement, nous étudions le cas stochastique où les quantités des composants désassemblés sont incertaines en raison du statut inconnu des produits retournés. Nous proposons une approche basée sur une modélisation robuste définie par des scénarios discrets. Nous abordons différentes hypothèses, où différents niveaux de flexibilité sont donnés aux différentes décisions. Une mat-heuristique à trois étapes pour optimiser le pire cas est développée pour la résolution itérative de différents problèmes dans tous les scénarios. Différents tests sont effectués afin de montrer l'efficacité et les limites de chaque approche dans chaque cas.

Mots clés : planification – production – gestion des stocks – transport – logistique inverse – programmation linéaire mixte – heuristique.

Optimization of Integrated Production and Transport Problems

In this thesis, we study an integrated vehicle routing problem with production management. This problem combines assembly and disassembly operations with the collection of end-of-life products and raw materials. First, we study the deterministic case of the multi-period problem in its static and dynamic version. We propose a mixed integer linear programming model including these new constraints. A two-phase mat-heuristic based on mathematical programming and the variable neighborhood search algorithm is also developed to solve the larger instances. Second, we study the stochastic case where the yield of disassembled components are uncertain due to the unknown status of the returned products. We propose an approach based on robust modeling defined by discrete scenarios. We address different assumptions, where different levels of flexibility are given to different decisions. A three-stage mat-heuristic to optimize the worst case is developed to solve the different problems under all scenarios. Different tests are performed to show the effectiveness and limitations of each approach in each case.

Keywords: planning – production – inventory control – transportation – reverse logistics – mixed linear programming – heuristic.

Thèse réalisée en partenariat entre :

