



**HAL**  
open science

# Optimisation de la logistique inverse et planification du désassemblage

Mustapha Hrouga

► **To cite this version:**

Mustapha Hrouga. Optimisation de la logistique inverse et planification du désassemblage. Recherche opérationnelle [math.OC]. Université de Technologie de Troyes, 2016. Français. NNT : 2016TROY0021 . tel-03361913

**HAL Id: tel-03361913**

**<https://theses.hal.science/tel-03361913>**

Submitted on 1 Oct 2021

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

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

Thèse  
de doctorat  
de l'UTT

**Mustapha HROUGA**

# Optimisation de la logistique inverse et planification du désassemblage

**Spécialité :**  
Optimisation et Sécurité des Systèmes

2016TROY0021

Année 2016

---

---

# THESE

*pour l'obtention du grade de*

**DOCTEUR de l'UNIVERSITE  
DE TECHNOLOGIE DE TROYES  
Spécialité : OPTIMISATION ET SURETE DES SYSTEMES**

*présentée et soutenue par*

**Mustapha HROUGA**

*le 24 juin 2016*

---

---

**Optimisation de la logistique inverse  
et planification du désassemblage**

---

---

## JURY

M. N. ESSOUNBOULI	PROFESSEUR DES UNIVERSITES	Président (Rapporteur)
M. L. AMODEO	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. M. GODICHAUD	MAITRE DE CONFERENCES	Directeur de thèse
M. O. GRUNDER	MAITRE DE CONFERENCES - HDR	Examineur
M. Z. SARI	PROFESSEUR TITULAIRE	Rapporteur
Mme A. YALAOUI	MAITRE DE CONFERENCES - HDR	Examineur



# Remerciement

Je crois que la réalisation d'un projet tel une thèse de doctorat demande un effort et une discipline qui sont bien souvent sous-estimés. Ce travail a été pour moi une étape dans ma vie qui m'a permis de mieux connaître mes forces et mes faiblesses. J'ai réussi, par ce travail, à dépasser mes limites. Ce projet est un accomplissement dont je suis extrêmement fier. Ce résultat, je n'aurai pu l'atteindre sans l'aide et les encouragements de personnes bien importantes pour moi.

La première personne est mon directeur de thèse Lionel AMODEO, Professeur à l'Université de Technologie de Troyes (UTT) qui, sereinement, m'a orienté, interpellé, recadré, conseillé et corrigé durant ce projet de longue haleine. Sa rigueur sur la qualité de mes résultats de recherche a été capitale dans l'atteinte de mes objectifs de thèse.

Je remercie aussi mon co-directeur de thèse, Matthieu GODICHAUD, Maître de Conférences à l'Université de Technologie de Troyes (UTT), pour son implication dans mon travail de thèse qui m'a servi à plus d'un titre. Sa totale disponibilité, son suivi permanent et ses critiques fondées et objectives m'ont permis d'avancer correctement dans ma thèse.

Je ne peux oublier Farouk YALAOUI, Professeur à l'Université de Technologie de Troyes (UTT) et Directeur du Laboratoire d'Optimisation des Systèmes Industriels (LOSI) qui a permis des conditions de travail au sein du laboratoire que beaucoup de chercheurs nous envies.

Je tiens à remercier mes rapporteurs de thèse, Najib ESSOUNBOULI, Professeur des Universités, Université de Champagne Ardennes, et Zaki SARI, Professeur à l'Université Aboubekr Belkaïd, Tlemcen-Algérie pour l'honneur qu'ils m'ont fait d'étudier mon travail et pour avoir apporté leurs conseils afin de le rendre meilleur. Leurs interventions ont été, en effet, plus que constructives et m'ont orienté vers des perspectives de recherche encore plus intéressantes.

Je remercie Olivier GRUNDER, Maître de conférences-HDR, Université de Technologie de Belfort-Montbéliard, d'avoir accepté d'examiner mon travail et donner beaucoup des remarques qui m'ont permis d'améliorer le travail.

Je remercie aussi Alice YALAOUI, Maître de conférences-HDR, à l'Université de Technologie de Troyes (UTT) pour l'intérêt qu'elle a porté pour ma thématique de recherche et l'honneur qu'elle m'a fait en acceptant d'examiner mon travail.

Un grand merci également à tous mes amis et à toutes les personnes que j'ai côtoyées durant ma thèse : Amine, Omar, Birome, Jorge, Imad, Ahmed, Ayoub, Mouad, Yassine, Xixi, Mourad, Mohammed, Taha et Taoufik.

Enfin, j'adresse mes remerciements les plus profonds à ma conjointe Meriem et ma petite princesse Salima ainsi qu'à toute ma famille qui m'entoure et me soutient en permanence, malgré la distance qui nous sépare. Je leur dédie ce travail et leur exprime tout mon amour.

*Tout ce qui ne tue pas rend plus fort*

**Résumé :** Dans cette thèse, nous traitons essentiellement des problèmes de lot sizing en désassemblage avec une structure de produits à désassembler à deux niveaux sans composants communs. Nous traitons deux problèmes différents. Dans le premier problème, nous considérons un seul produit et la contribution porte sur le développement de deux modèles de programmation en nombres entiers. Le premier modèle est considéré sans ventes perdues où toutes les demandes doivent être satisfaites, et le deuxième est considéré avec ventes perdues où les demandes peuvent ne pas être satisfaites. Pour la résolution de ce problème, nous développons d'abord une approche analytique permettant de calculer les stocks de surplus (avant la résolution du problème) à la fin de l'horizon de planification. Ensuite, nous adaptons trois heuristiques connues pour leurs performances et largement utilisées dans le problème lot sizing en production « Silver Meal, Part Period Balancing et Least Unit Cost ». Dans le deuxième problème, nous considérons plusieurs produits avec contraintes de capacité et la contribution porte sur l'extension des deux modèles précédents. Le premier est également considéré sans ventes perdues et le deuxième avec ventes perdues. En ce qui concerne la résolution de ce problème et compte tenu de sa complexité, un algorithme génétique est d'abord proposé. Ensuite, afin d'améliorer cet algorithme, nous intégrons une heuristique Fix-and-Optimize dans ce dernier tout en proposant une approche hybride. Finalement, des tests sont effectués sur de nombreuses instances de la littérature afin de montrer l'efficacité et les limites de chaque approche de résolution.

**Mots clés :** *Distribution logistique, Planification, Optimisation mathématique, Recherche opérationnelle, Programmation heuristique, Programmation linéaire.*

**Abstract :** In this thesis, we mainly deal with lot sizing problems by disassembling with a structure of products to disassemble with two levels and without commonality components. We treat two different problems. In the first problem, we consider a single product whose contribution focuses on developing the two programming models integers. The first model is considered without lost sales where all demands must be satisfied, and the second one is considered with lost sales where demands may not be met. To solve this problem, we first develop an analytical approach to calculate the surplus stocks (before solving the problem) at the end of the planning horizon. Then we adapt three heuristics known for their performance and widely used in the lot sizing problem of production "Silver Meal, Part Period Balancing and Least Unit Cost". In the second problem, we consider a number of products with capacity constraint, and the contribution relates to the extension of the two previous models. The first is considered without lost sales and the second with lost sales. Regarding the resolution of this problem and given its complexity, a genetic algorithm is first proposed. Then, to improve this algorithm, we integrate a Fix-and-Optimize heuristic in the latter while offering a hybrid approach. Finally, various tests are performed on different literature instances to demonstrate the effectiveness and limitations of each solving approach.

**Keywords :** *Logistic distribution, Mathematical optimization, Planning, Operations research, Heuristic programming, Linear programming.*

# Table des matières

<b>Introduction générale et présentation du problème.....</b>	<b>1</b>
1 Contexte.....	3
1.1 Problème de désassemblage .....	3
1.1.1 Assemblage vs désassemblage .....	3
1.1.2 Objectifs et décisions de désassemblage .....	4
2 Problématique.....	5
3 Objectifs et contributions .....	6
4 Organisation de la thèse.....	7
<b>Chapitre 1 : État de l'art sur l'optimisation de la logistique inverse et planification du désassemblage .....</b>	<b>9</b>
1 Définitions de la logistique inverse .....	10
2 Planification en logistique inverse.....	17
2.1 Conception de réseaux logistiques inverses .....	18
2.2 Planification et gestion d'approvisionnement .....	20
2.3 Ordonnancement et conception des lignes de désassemblage.....	20
3 Planification du désassemblage .....	22
4 Modèles pour l'optimisation de la planification du désassemblage .....	25
4.1 Planification des besoins .....	27
4.1.1 Méthodes MRP.....	27
4.1.2 Optimisation des coûts .....	27
4.1.3 Incertitudes .....	28
4.2 Problèmes de lot sizing en désassemblage .....	30
4.2.1 Problème mono produit à deux niveaux .....	31
4.2.2 Problème multiniveaux (avec ou sans pièce commune).....	32
4.2.3 Problème multi-produit avec contraintes de capacité.....	34
4.3 Problème de lot sizing en production .....	38
4.3.1 Articles des états de l'art .....	40
4.3.2 Problèmes mono-produit et un niveau.....	40



4.3.3	Problèmes à un niveau et plusieurs produits .....	42
4.3.4	Problèmes à plusieurs niveaux .....	43
4.3.5	Nouvelles approches.....	44
4.3.6	Différences entre le problème de lot sizing en production et en désassemblage.....	45
<b>Chapitre 2 : Problème de lot sizing en désassemblage à un seul produit .....</b>		<b>49</b>
1	Problème de lot sizing en désassemblage sans ventes perdues .....	50
1.1	Modélisation du problème .....	51
1.2	Problématique d'accumulation de stocks .....	52
1.3	Méthodes de résolution .....	54
1.3.1	Méthodes exactes.....	54
1.3.2	Heuristiques.....	55
1.3.3	Résultats expérimentaux.....	61
2	Problème de lot sizing en désassemblage avec ventes perdues .....	66
2.1	Modélisation du problème .....	66
2.2	Équivalence entre modèles .....	68
2.3	Méthodes de résolution .....	69
2.3.1	Méthodes exactes.....	70
2.4	Résultats expérimentaux.....	71
2.5	Heuristiques.....	74
2.5.1	Expérimentations.....	77
<b>Chapitre 3 : Problème de lot sizing en désassemblage de plusieurs produits à deux niveaux.....</b>		<b>83</b>
1	Problème de lot sizing en désassemblage sans ventes perdues .....	84
1.1	Modélisation du problème .....	84
1.2	Méthodes de résolution .....	87
1.2.1	Relaxation lagrangienne .....	87
1.2.2	Méta-heuristique : algorithme génétique .....	88
1.2.3	Heuristique Fix-and-Optimize (FO).....	94
1.2.4	Approche hybride (AH).....	98
1.3	Résultats expérimentaux.....	99

2	Modèle de lot sizing avec ventes perdues .....	106
2.1	Méthodes de résolutions .....	107
2.1.1	Algorithme génétique .....	107
2.1.2	Fix-and-Optimize .....	109
2.2	Résultats expérimentaux.....	109
	<b>Conclusion générale et perspectives.....</b>	<b>115</b>
	<b>Bibliographie.....</b>	<b>119</b>

## Table des figures

Figure 1 : Problèmes liés au désassemblage (Ilgin et Gupta 2010).....	4
Figure 2 : Schéma de la définition de logistique inverse proposée par (Salema et al. 2007).....	12
Figure 3 : Flux de la logistique inverse (Chouinard 2007).....	16
Figure 4 : Nombre d'articles publiés par année sur la logistique inverse (Agrawal et al. 2015) .....	16
Figure 5 : Structure de désassemblage à deux niveaux (structure de base).....	22
Figure 6 : Structure du désassemblage à deux produits avec pièces communes .....	23
Figure 7 : Structure du désassemblage à plusieurs niveaux .....	23
Figure 8 : Structure du désassemblage à plusieurs niveaux (Barba-Gutiérrez et Adenso-Díaz 2009)..	24
Figure 9 : Structure du désassemblage à plusieurs niveaux (Godichaud 2009) .....	24
Figure 10 : Structure du désassemblage à deux produits et deux niveaux (Langella 2007).....	24
Figure 11 : Nombre d'articles publiés par année.....	26
Figure 12 : Nombre d'articles publiés par type du journal.....	26
Figure 13 : Classification des problèmes de lot sizing en désassemblage.....	37
Figure 14 : Nombre d'articles publiés par année.....	39
Figure 15 : Nombre d'articles publiés par type du journal.....	39
Figure 16 : Classification des problèmes de lot sizing (Merzouk 2007) .....	47
Figure 17 : Structure de désassemblage à deux niveaux .....	50
Figure 18 : Exemple d'accumulation de stocks.....	65
Figure 19 : Structure d'un produit à deux niveaux.....	70
Figure 20 : Exemple de calcul de la quantité de désassemblage .....	76
Figure 21 : Exemple de croisement sur les périodes .....	90
Figure 22 : Exemple de croisement sur les produits.....	90
Figure 23 : Exemple de mutation sur les périodes .....	92
Figure 24 : Exemple de mutation sur les produits.....	92
Figure 25 : Exemple de décomposition d'un produit/plusieurs périodes .....	95
Figure 26 : Exemple de décomposition de plusieurs produits/plusieurs périodes.....	96
Figure 27 : Exemple de décomposition d'un produit/horizon glissant.....	97
Figure 28 : Pseudocode de l'approche hybride .....	98
Figure 29 : Algorithme de l'approche hybride (AH).....	99

## Table des tableaux

Tableau 1 : Classification par type de produit.....	14
Tableau 2 : Classification par type de réseau .....	15
Tableau 3 : Classification des problèmes de lot sizing en désassemblage .....	30
Tableau 4 : Complexité du problème USILS .....	41
Tableau 5 : Comparaison entre l'assemblage et le désassemblage (Brennan et <i>al.</i> 1994) et (Gungor et Gupta 1999).....	46
Tableau 6 : Données de l'exemple .....	53
Tableau 7 : Demandes .....	57
Tableau 8 : Plan obtenu par l'heuristique Silver Meal .....	58
Tableau 9 : Plan obtenu par l'heuristique Least Unit Cost.....	60
Tableau 10 : Plan obtenu par l'heuristique Part Period Balancing.....	61
Tableau 11 : Demandes croissantes et décroissantes.....	63
Tableau 12 : Demandes variables.....	64
Tableau 13 : données d'entrées .....	71
Tableau 14 : Plan optimal de l'exemple .....	71
Tableau 15 : Résultats des tests pour des demandes constantes et variables.....	73
Tableau 16 : Résultats des tests pour demandes croissantes et décroissantes .....	74
Tableau 17 : Demandes croissantes et décroissantes.....	78
Tableau 20 : Stratégie S1P/SPP.....	102
Tableau 21 : Stratégie S1PHG/SPPHG .....	103
Tableau 22 : Algorithme génétique/ Approche hybride .....	104
Tableau 23 : Résultats de l'AG et approche hybride.....	111



# Introduction générale et présentation du problème

La réglementation environnementale devient de plus en plus stricte suite à la mise en place de différentes directives européennes imposant aux industriels de collecter, valoriser et recycler leurs produits en fin de vie ainsi que les déchets qui en résultent. Par ailleurs, différentes filières sont successivement apparues sur le terrain : piles et accumulateurs en 1991, véhicules hors usage fin 2000 et déchets d'équipements électriques et électroniques (DEEE) en 2003. Toutes ces contraintes réglementaires ont donné naissance au marché de la « reverse logistics » aussi appelée « logistique des retours ou logistique inverse ». Au début de son apparition, elle se focalisait principalement sur le recyclage de certains produits contenant des substances toxiques telles que les piles, ou encore la récupération des matières premières réutilisables comme l'or des composants électroniques, ou encore certains métaux précieux ou dangereux pour l'environnement. Dans les 10 dernières années, la logistique inverse a pris en compte le cas du retour des produits en fin de vie (produits défectueux) ou encore produits non désirés par les clients. À l'heure actuelle, la logistique inverse est devenue un marché multifacettes, elle inclut les retours des emballages et les retours des produits en fin de vie. Elle répond également à diverses problématiques auxquelles font faces les industriels telles que le développement durable, le service après-vente, la gestion des invendus, etc. La logistique inverse a un rôle fondamental à jouer dans l'optimisation du retour des produits en fin de vie ou usés pour une réutilisation, un désassemblage ou un recyclage, et en participant ainsi à l'économie générale des ressources. En outre, l'importance de cette dernière varie évidemment d'un secteur à un autre, dans lequel les taux de retour des produits en fin de vie sont de plus en plus importants. Dans ce contexte, une étude a été effectuée par (Rogers et Tibben-Lembke 1998) (*Going backwards : Reverse Logistics Trends and Practices*, University of Nevada) sur le taux de retour des produits en pourcentage pour quelques secteurs d'activités. Durant cette étude, les chercheurs ont montré que le taux de retour en pourcentage des journaux et magazines représente 20 à 30%, les ventes par correspondance 18 à 35%, les distributeurs de livres 10 à 20%, les ordinateurs 10 à 20%, les distributeurs de composants électroniques 10 à 12%, la grande distribution 4 à 15%, les pièces de rechanges des voitures 4 à 6%, le grand public 4 à 5% et les produits chimiques ménagers représentent 2 à 3% du taux de retour.

En plus des avantages de la logistique inverse et son importance dans les retours des produits en fin de vie, il existe d'autres raisons encourageant les entreprises à mettre en place ce système de logistique. Ces raisons sont à la fois économiques, sociétales et environnementales. En effet, les raisons économiques sont dues aux profits apportés par la valorisation des produits en fin de vie nécessitant une structuration du retour de l'utilisateur vers l'entreprise. Les raisons écologiques et environnementales sont exigées surtout par la norme ISO 14000, elles visent à minimiser l'impact environnemental des activités logistiques inverses générées par les déchets de produits en fin de vie. Malgré les motivations et les avantages de cette chaîne logistique inverse, sa mise en place reste encore compliquée. En effet, il existe toujours certains niveaux d'incertitudes notamment sur la disponibilité des produits en fin de vie ainsi que la qualité et la quantité des produits retournés. Les producteurs ont souvent moins de contrôle sur le temps et la quantité des produits retournés. Pour faire face à ce problème et afin de valoriser les retours des produits, les industriels doivent développer des méthodes de résolution efficaces permettant d'optimiser les performances de la chaîne logistique inverse. Ces méthodes doivent également 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 de matières à recycler d'autre part. Pour atteindre cet objectif, il nous semble important de développer un système de

planification du désassemblage efficace avec une flexibilité suffisante afin de faire face aux incertitudes liées à l'état des produits retournés.

Par ailleurs, avec la mise en place des nouvelles normes de législation environnementales, le problème du désassemblage et de recyclage des produits a pris une place importante dans la logistique inverse. En effet, le désassemblage est un processus important permettant la valorisation des produits en fin de vie. En donnant la possibilité de réutilisation, de re-fabrication du produit et de ses composants, ou encore la préparation des pièces au recyclage. Ce dernier a été défini par (Gupta et Taleb 1994) comme une méthode systématique pour la séparation du produit en parties, composants, sous-ensembles ou autres groupements. Dans cette thèse, nous nous intéressons à la planification des opérations de désassemblage pour les produits en fin de vie. L'idée est de valoriser les produits en fin de vie par la vente des composants désassemblés tout en optimisant les différents coûts liés aux opérations de désassemblage et de stockage des composants désassemblés. Pour atteindre cet objectif, nous développons un système de planification pour toutes les opérations de désassemblage.

La planification des opérations de désassemblage est un domaine qui date des années 90. Elle vise à organiser dans le temps une succession d'actions ou d'événements afin de désassembler une quantité de composants d'un produit retourné tout en satisfaisant ses demandes sur un horizon de planification périodique ou continu. À l'apparition de la planification du désassemblage, les chercheurs ne considéraient pas l'optimisation des différents coûts liés aux opérations, ils cherchaient seulement à déterminer la quantité du désassemblage des produits en fin de vie pour répondre aux demandes de leurs composants constituants. Au début des années 2000, les chercheurs ont commencé à intégrer d'autres coûts tels que le coût de setup relatif au produit en fin de vie. Ce dernier implique le travail nécessaire à la préparation de l'opération de désassemblage. Quant au coût de stockage, il implique les composants désassemblés et stockés pour répondre aux demandes futures. À cet effet, la planification du désassemblage est traitée dans un problème de décision tactique. Notre analyse de la littérature nous a permis de relever que les auteurs se limitent à la minimisation des coûts tandis que la quantité et la qualité des produits en fin de vie ou encore leur état n'est pas considéré. Dans le problème du désassemblage, il arrive que les coûts soient importants par rapport aux revenus qui peuvent être générés par la vente des composants. Dans ce cas, les demandes peuvent ne pas être satisfaites et entraîne un coût appelé « coût de ventes perdues ». Ce type de problème n'a pas encore été traité dans la littérature. C'est pourquoi dans cette étude nous intégrons le coût de ventes perdues dans le cas du problème de planification du désassemblage.

Cette introduction est organisée en 4 parties : dans une première partie, nous présentons le contexte dans lequel s'inscrit cette thèse. La deuxième partie présente la problématique de planification du désassemblage. Les objectifs de la thèse et les contributions sont élaborées dans la troisième partie. Finalement, nous présentons l'organisation et le déroulement de la thèse.

# 1 Contexte

Dans le cadre de cette thèse, nous traitons un problème de désassemblage. Nous présentons dans un premier temps la définition et le rôle du désassemblage dans la valorisation des produits en fin de vie. Puis dans un deuxième temps, nous nous intéressons aux objectifs, aux décisions et relevons les principales différences entre le problème du désassemblage et de l'assemblage.

## 1.1 Problème de désassemblage

Le désassemblage est un processus permettant d'obtenir des composants et/ou des produits qui peuvent être réutilisés ou recyclés avec des taux de récupérations importants à partir de produits défectueux ou usés (Brennan *et al.* 1994), (Moyer et Gupta 1997). Le désassemblage joue un rôle fondamental dans la valorisation des produits retournés. C'est un processus planifié de récupération des éléments constitutifs d'un produit manufacturé pour lequel les caractéristiques mécaniques, fonctionnelles et de fin de vie sont identifiées. Il s'agit d'une extraction méthodique des pièces et des matériaux de produits mis au rebut après une ou plusieurs opérations de production. Il peut être considéré comme une stratégie de récupération des produits en fin de vie. Les composants récupérés peuvent être réutilisés comme des produits neufs, recyclés ou encore réutilisés comme pièces de rechanges. Les composants sont ensuite vendus en vue de générer des revenus et de minimiser l'impact sur l'environnement. La possibilité de désassembler un produit en fin de vie est dépendante de paramètres techniques et environnementaux. Selon ces paramètres, nous pouvons décider du mode de désassemblage qui peut être destructif ou non destructif. Dans le cas du désassemblage destructif, les composants désassemblés qui sont défectueux peuvent être éliminés avec un coût d'élimination qui appartient au coût total de désassemblage. Pour qu'une opération de désassemblage soit efficace, elle doit prendre en compte le prolongement de la durée de vie des composants désassemblés. Elle devrait également permettre la récupération des composants utiles et la séparation des matériaux décontaminés pour le retraitement. Dans les opérations de désassemblage, l'optimisation est nécessaire afin de rendre le processus de désassemblage efficace et économique.

### 1.1.1 Assemblage vs désassemblage

Le processus de désassemblage est caractérisé par des propriétés physiques et opérationnelles plus différentes et complexes que celles de l'assemblage. (Duta 2006) a relevé plusieurs différences entre la problématique de l'assemblage et du désassemblage. La première différence est que le désassemblage est généralement partiel ce qui entraîne une variable de décision sur la profondeur (jusqu'à quel niveau de composant élémentaire désassembler un produit). Dans le désassemblage il existe une incertitude très forte sur l'état des produits en fin de vie (produits retournés). Il y a également une forte variation sur la demande pour les composants récupérables (composants à désassembler). Une autre différence fondamentale est que les demandes se situent au niveau des composants élémentaires constituant le produit en fin de vie. Ceci engendre une différence de gestion des flux : le désassemblage est un processus divergent où un produit est décomposé en plusieurs composants et sous-ensembles, contrairement à l'assemblage où plusieurs composants convergent vers un seul produit. Pour des raisons techniques et économiques, le désassemblage est essentiellement partiel, car certains composants peuvent ne pas être séparés à cause de leur état de dégradation représentant une difficulté technique, alors que l'assemblage est complet. Le coût d'un désassemblage est parfois plus important que les revenus des composants récupérés. Selon (Lambert 2003), les différences citées auparavant peuvent être résumées comme suit : le désassemblage n'est pas nécessairement complet. La valeur ajoutée par le désassemblage est généralement plus faible que celle ajoutée par l'assemblage. Le

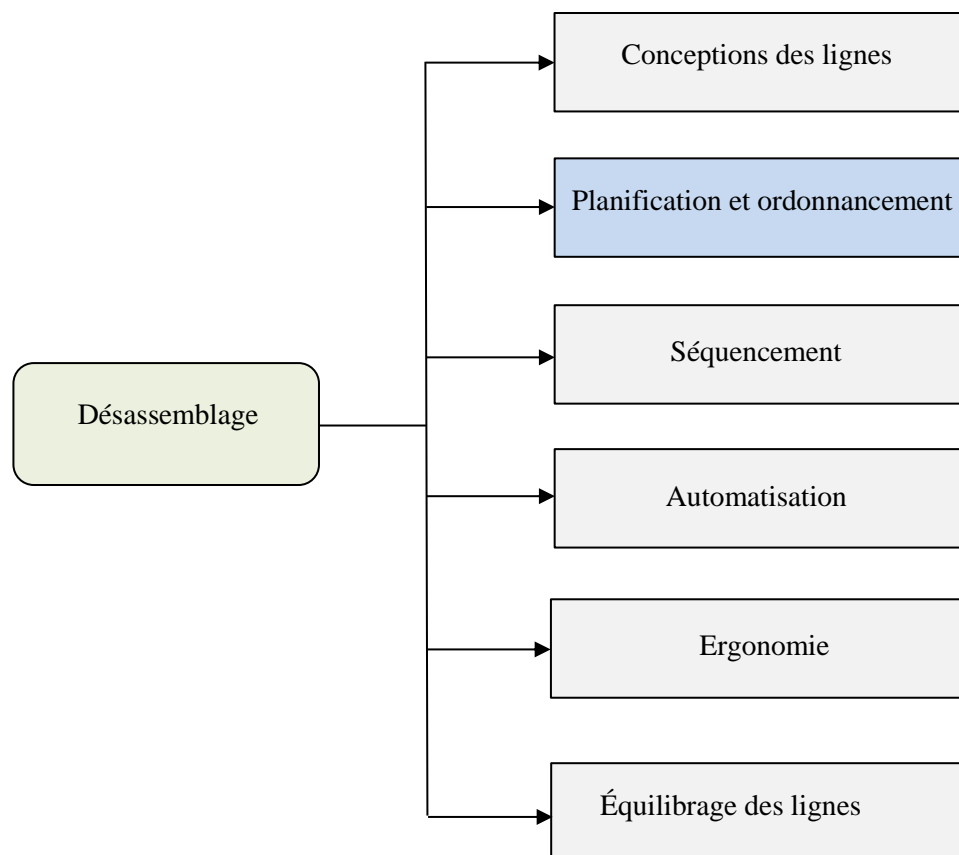


désassemblage est réalisé par une main d'œuvre humaine quant à l'assemblage est plutôt automatisé. Finalement, le désassemblage peut prendre la qualité d'être destructif ou non destructif en raison de ses composants et liaisons.

### 1.1.2 Objectifs et décisions de désassemblage

Après une analyse des besoins de désassemblage, le décideur peut passer à la planification des opérations de désassemblage en vue de répondre aux demandes de composants et de sous-ensembles. L'étape de planification doit aussi permettre de déterminer une stratégie optimale pour le désassemblage d'un produit complexe. Un désassemblage efficace consiste à enlever les pièces les plus précieuses et arrêter le désassemblage lorsque l'opération du désassemblage devient rentable.

Il existe divers problèmes de décisions dans le domaine de désassemblage que nous pouvons classer principalement dans 5 catégories (Ilgin et Gupta 2010), comme nous pouvons le voir dans la Figure 1, on retrouve la conception des lignes de désassemblage facilitant le désassemblage des produits. Il y a aussi l'équilibrage des lignes de désassemblage qui répartit les tâches de désassemblage aux postes de travail en équilibrant la ligne de désassemblage. On trouve également la planification des processus de désassemblage permettant de désassembler les produits en fin de vie et de leur sous-ensemble, le séquençement qui permet de trouver le meilleur ordre pour désassembler des composants et finalement l'automatisation du processus de désassemblage et l'ergonomie. Ces problématiques sont illustrées dans la Figure 1.



**Figure 1** : Problèmes liés au désassemblage (Ilgin et Gupta 2010)

## 2 Problématique

Dans un contexte de la logistique inverse, l'activité de désassemblage permet de récupérer des composants valorisables à partir des produits en fin de vie ou non utilisés. La planification du désassemblage consiste à déterminer sur un horizon à moyen terme la quantité des produits à désassembler afin de satisfaire des demandes concernant les composants constituants. La problématique de la planification du désassemblage est caractérisée par plusieurs paramètres. Parmi ces paramètres, il y a notamment la durée de l'horizon de planification, la gestion et les prévisions des demandes, la gestion et les prévisions des retours et la gestion (continue ou discrète) du temps au cours de l'horizon. Le plan établi au début de l'horizon de planification (quantité de produits à désassembler) doit répondre à des objectifs de satisfaction des clients, de minimisation des coûts (de stockage et de setup) et éventuellement de maximisation des profits. La planification du désassemblage présente un certain nombre de similitudes avec la planification de production mais aussi des spécificités nécessitant le développement d'approches adaptées.

Une première spécificité du désassemblage est relative à la structure ou à la nomenclature des produits à traiter. En effet, dans le désassemblage on travaille sur des nomenclatures inversées des produits puisque les demandes concernent les composants d'un ou plusieurs produits finis. Autrement dit, un même produit est une source à approvisionner pour satisfaire plusieurs demandes différentes. La prise en compte de la gestion des stocks dans la planification devient alors primordiale. De plus, un composant pour lequel nous avons une demande peut être généré à partir de différents produits. Il s'agit alors de déterminer lequel de ces produits il faut désassembler, et à quel moment le désassemblage est rentable. Ensuite, les demandes peuvent se situer aux différents niveaux de la nomenclature du produit et il s'agit alors de déterminer la profondeur du désassemblage afin de répondre à la demande des pièces ou des composants désassemblés tout au long de l'horizon de planification. Une autre spécificité du désassemblage concerne les incertitudes inhérentes à ce type d'activité. Elles sont principalement liées à la qualité des produits récupérés (usure, composant dangereux, composant endommagé...). Nous avons également la possibilité de prévoir les retours et la composition des produits. Par ailleurs, la structure des produits qui n'est pas toujours adaptée au désassemblage ainsi qu'à l'état aléatoire des demandes et du temps de désassemblage. La problématique est, d'une part, de caractériser les incertitudes qui ont un impact sur la planification du désassemblage et, d'autre part, de les prendre en compte dans les modèles en autorisant les ventes perdues.

Compte tenu de la structure des produits en fin de vie à désassembler, le problème de planification du désassemblage peut être généralement défini comme suit : *Pour une structure de désassemblage (nomenclature des produits à désassembler) donnée, l'objectif est de déterminer les types et la quantité des produits en fin de vie, les sous-ensembles et les composants à désassembler, ainsi que les délais de livraison associés en satisfaisant la demande sur un horizon de planification avec des périodes en temps discret.* Dans le problème de planification du désassemblage, nous pouvons rencontrer des risques relatifs aux opérations de désassemblage (comme par exemple la manipulation de néons contenant des gaz, la découpe de véhicules contenant des substances dangereuses ou encore l'arrêt d'une opération de désassemblage en raison de la corrosion d'une vis). Ces risques peuvent se caractériser par des ruptures ou des stocks de surplus, des délais opératoires variables, des arrêts de process, des pertes de lots ou encore des flux réentrants. Ces derniers non pas été pris en compte dans les travaux de littérature. L'objectif est de prendre en considération les risques inhérents au problème de planification du désassemblage. En effet, dans le cas où les quantités et la qualité du produit retourné ne peuvent pas être ni connues ni maîtrisées (un arrêt de processus par exemple), les

demandes peuvent ne pas être livrées au temps prévu. Dans le cadre de cette étude, nous n'autorisons pas les retards sur les demandes, puisqu'il arrive que le client ne tolère aucun retard dans la livraison. Dans ces situations, le décideur est contraint de perdre la demande avec un coût de vente perdue. Il arrive également parfois que nous avons un surplus pour certains composants désassemblés, notamment ceux qui ont un rendement de désassemblage important. L'objectif final est de minimiser les coûts de setup, les coûts de stockage et les coûts de ventes perdues, il faut également gérer le surplus restant à la fin de l'horizon de planification.

La planification du désassemblage dépend principalement de la structure de produits en fin de vie. Parmi les structures rencontrées dans la littérature, on trouve notamment des structures à un seul produit à deux niveaux sans pièces communes. Plusieurs produits à deux niveaux avec ou sans pièces communes. Un ou plusieurs produits et plusieurs niveaux sans ou avec pièces communes, ainsi que l'existence des contraintes de capacité. Dans une structure de désassemblage à deux niveaux, le premier niveau représente le produit en fin de vie et le deuxième niveau représente les composants obtenus directement par l'opération de désassemblage à partir du produit en fin de vie. Dans le cas d'une structure du désassemblage à plusieurs niveaux, le premier niveau représente le produit en fin de vie (éléments parents), le deuxième niveau représente les éléments intermédiaires ou les éléments obtenus directement par l'opération de désassemblage à partir de ce produit, et le dernier niveau représente les composants (éléments enfants) ou pièces qui sont les éléments obtenus directement par l'opération de désassemblage à partir des éléments intermédiaires. Certains parents peuvent avoir les mêmes enfants ce qu'on appelle produits ayant des pièces communes. Le produit en fin de vie peut avoir un ou plusieurs éléments intermédiaires qui peuvent ainsi avoir un ou plusieurs composants. Dans une structure à plusieurs niveaux, nous avons deux types des demandes, des demandes pour les éléments intermédiaires, et des demandes pour les composants ou éléments désassemblés à partir des éléments intermédiaires.

Dans cette thèse, nous nous intéressons à la planification des opérations de désassemblage avec une structure de produits à deux niveaux, un et plusieurs produits sans pièces communes et avec contraintes de capacité. Il s'agit de répondre aux besoins de décideurs (besoin d'aide à la décision) dans la mise en place de la logistique inverse et d'optimiser les performances des aspects industriels impliqués dans une activité de logistique inverse. Plus précisément, il s'agit de déterminer la quantité à désassembler et la quantité à stocker, aussi que celle à perdre à chaque période de l'horizon de planification. Dans le cas où les ventes perdues sont autorisées, il faut déterminer l'instant où il est profitable de perdre les demandes à chaque période de l'horizon. Les quantités de stock et de ventes perdues dépendent de la quantité de désassemblage, du nombre de composants à désassembler (rendement de désassemblage) à partir du produit en fin de vie et la demande des composants.

### **3 Objectifs et contributions**

Le projet de thèse vise à développer des modèles et méthodes d'optimisation des problèmes de désassemblage dans un contexte de la logistique inverse et dans les phases de conception et de gestion opérationnelle. L'objectif est de minimiser le coût total composé du coût de setup, du coût de stockage et du coût de désassemblage. Il s'agit de déterminer à quel instant il faut désassembler le produit en fin de vie et en quelle quantité afin de répondre à la demande des composants désassemblés à partir du produit en fin de vie. Les stocks de produits valorisables permettent de répondre sans attente aux demandes de composants de ces produits. En effet, les stocks de sous-ensembles intermédiaires permettent d'adapter le système aux perturbations sur les opérations de désassemblage liées à l'état

incertain des produits en fin de vie. La non satisfaction de la demande ou l'autorisation de ventes perdues est un aspect important dans le désassemblage. Les demandes peuvent ne pas être satisfaites si par exemple le prix de ventes relatif à la demande est faible, ou ces dernières ne sont pas rentables pour amortir les coûts de stockage et de setup. Si cet aspect a été pris en compte dans des problèmes de lot sizing en production, il n'a pas encore été pris en compte dans le problème de lot sizing en désassemblage.

Les objectifs scientifiques et techniques de cette thèse visent à développer des modèles pour l'analyse de performances et l'optimisation des problèmes de désassemblage d'une part, et de développer des approches permettant de résoudre les modèles proposés d'autre part. Pour atteindre ces objectifs, nous allons traiter différents problèmes de lot sizing en désassemblage. Dans le premier problème, nous considérons un seul produit avec une structure de produit à deux niveaux. Dans le deuxième problème, nous considérons plusieurs produits avec la même structure de produit (deux niveaux), sans pièces communes et avec contraintes de capacité. Ensuite, chaque problème est traité avec et sans ventes perdues. Finalement, pour chaque problème nous proposons des approches différentes de résolution. Nos principales contributions dans cette thèse portent sur :

- Le développement d'une approche analytique permettant de calculer le surplus avant la résolution du modèle (sans passage par l'optimisation). Il s'agit d'une problématique d'accumulation de stocks à la fin de l'horizon de planification qui ne sert à satisfaire aucune demande. Il s'agit de déterminer le stock de surplus pour chaque composant à la fin de l'horizon de planification.
- Le développement de deux nouveaux modèles de programmation linéaire en nombres entiers avec et sans ventes perdues. Dans le premier modèle, nous considérons un seul produit. Dans le deuxième modèle, nous considérons plusieurs produits avec contraintes de capacité. Nous intégrons ensuite les différentes caractéristiques spécifiques au problème de planification du désassemblage. Il s'agit de minimiser les coûts de setup, de stockage, de ventes perdues et les coûts de pénalité de dépassement du temps de désassemblage.
- Le développement de différentes approches et techniques d'optimisation permettant de résoudre les modèles proposés. Plus précisément, nous adaptons des techniques d'optimisation dans le problème de planification en production (lot sizing en production).

## **4 Organisation de la thèse**

Cette thèse est organisée comme suit : dans le premier chapitre, un état de l'art sera présenté sur l'optimisation de la logistique inverse et planification du désassemblage. Plus précisément, nous présenterons une étude bibliographique moins exhaustive notamment sur les flux logistiques inverses et les problèmes de désassemblage (techniques de modélisation et d'optimisation). Ce chapitre est composé de quatre points fondamentaux : le premier point aborde les définitions de la logistique inverse issues de la littérature ainsi que les motivations poussant les entreprises à mettre en place un système de logistique inverse. Dans le deuxième point, nous nous intéressons aux différents types de planification en logistique inverse notamment la conception des réseaux, la conception des lignes de désassemblage, l'approvisionnement et l'ordonnancement. Le troisième et le quatrième point se concentrent respectivement sur le problème de planification du désassemblage et lot sizing en production.

Dans le deuxième chapitre de cette thèse, nous traitons un problème de lot sizing en désassemblage avec une structure à deux niveaux et un seul produit. Le problème est considéré sans et avec ventes perdues où les demandes peuvent être satisfaites ou perdues. Ce chapitre est composé de trois parties : dans la première partie, nous traitons un problème sans ventes perdues et nous étudions la problématique d'accumulation de stocks. Nous développons une analyse du modèle permettant de calculer le stock de surplus avant la résolution du modèle (sans passage par l'optimisation). Dans la deuxième partie, nous proposons trois heuristiques largement connues et utilisées dans la résolution de problème de lot sizing en production « Silver Meal, Least Unit Cost et Part Period Balancing » pour résoudre le problème étudié. Ensuite, nous effectuons différents tests sur plusieurs instances en vue de comparer les performances desdites heuristiques. Dans la troisième partie de ce chapitre, nous traitons un problème de lot sizing en désassemblage avec ventes perdues où les demandes peuvent ne pas être satisfaites. Le modèle est ensuite résolu de la même manière que celui sans ventes perdues (nous utilisons les mêmes heuristiques précédentes). Finalement, nous testons les performances des heuristiques pour la résolution des deux problèmes en fonction des différents profils de demandes (demandes croissantes, demandes décroissantes et demandes variables).

Dans le troisième chapitre de cette thèse, nous traitons un problème de lot sizing en désassemblage à deux niveaux, plusieurs produits et contraintes de capacité. Le problème est considéré sans et avec ventes perdues où toutes les demandes peuvent être satisfaites ou perdues. Ce chapitre est composé de deux parties : la première partie traite le problème sans ventes perdues et la deuxième avec ventes perdues. Pour la résolution de ces problèmes nous proposons d'abord un algorithme génétique. Ensuite, nous proposons une heuristique Fix-and-Optimize avec différentes stratégies de décomposition. Finalement, pour améliorer l'algorithme génétique, nous intégrons l'heuristique Fix-and-Optimize dans ce dernier en proposant de la sorte une approche hybride.

Finalement, nous terminons par une conclusion générale qui résumera l'ensemble de nos travaux. Nous discuterons ensuite les principaux résultats obtenus dans cette recherche ainsi que leurs interprétations. Nous aborderons également nos contributions et la valeur ajoutée dans cette thèse. Nous terminerons enfin par des perspectives de nos travaux de recherches.

# Chapitre 1 : État de l'art sur l'optimisation de la logistique inverse et planification du désassemblage

La logistique inverse (Reverse Logistics) a pris une place importante dans la logistique globale des entreprises, notamment suite à la mise en place des législations environnementales qui obligent les industriels à recycler leurs produits en fin de vie. L'objectif de ces réglementations est de préserver l'environnement en obligeant les industriels à valoriser leurs produits en fin de vie. Les produits retournés sont valorisés à travers d'un réseau de création de valeur en intégrant les processus de récupération, de traitement, de réparation, de désassemblage, de recyclage et de distribution ou encore d'élimination propre. En effet, l'utilisation de matériels valorisés permet de réduire la production de matériels neufs et par conséquent, de diminuer la consommation en ressources non renouvelables ainsi que son impact sur l'environnement. De plus, des gains directs peuvent être obtenus par la réutilisation des composants. Ces derniers consistent dans l'amélioration de l'image de l'entreprise et les attentes de ses clients. En outre, la logistique inverse permet de gérer le retour des produits défectueux, la gestion des produits retournés ou invendus, les surstocks ou encore les produits en fin de vie. Ses secteurs sont diverses, on trouve notamment les produits électroniques, les appareils électroménagers, l'automobile, les contenants retournables...etc. Ces produits sont de plus en plus conçus afin de réduire leurs impacts sur l'environnement quand ils arrivent en fin de cycle de vie, ainsi que la durée de vie des composants électroniques est souvent très courte.

Par ailleurs, la montée des préoccupations environnementales et la réduction des déchets encouragent les entreprises à mettre en place un système de logistique inverse. En effet, du point de vue environnemental, les standards ISO 14000, la chaîne d'approvisionnement verte et le «*Total Quality Environmental Management*» sont des notions qui deviennent de plus en plus importantes. En outre, les entreprises sont de plus en plus concernées par l'aspect environnemental. Ce dernier joue un rôle important dans les stratégies des entreprises. Du fait de ces réglementations environnementales qui sont de plus en plus strictes, les entreprises sont contraintes à récupérer une quantité minimale de déchets auprès de leurs clients. Ces dernières doivent faire face à une augmentation de reprises de leurs produits et ont besoin d'organiser ces retours. Pour atteindre ces objectifs, les entreprises doivent développer des outils d'aides à la décision afin de réduire les coûts liés à la chaîne logistique inverse.

D'une manière plus générale, l'aspect le plus important de la logistique inverse est la récupération des produits en fin de vie. Ainsi, le désassemblage est devenu un processus important pour faciliter la valorisation de ces produits. En effet, le désassemblage des produits en fin de vie est un procédé en croissance, il est dû à l'augmentation du besoin des entreprises en considérant les facteurs environnementaux à la fin du cycle de vie du produit. En plus, il est l'un des aspects les plus importants de la logistique inverse et de la récupération du produit. Il présente ainsi une étape principale dans le processus de recyclage d'un produit en fin de vie. Il est en fait la première étape essentielle du recyclage et de la remise à neuf, puisqu'il permet d'une part de récupérer les sous-ensembles et les constituants réutilisables et d'autre part, le tri des matériaux et l'extraction des matières dangereuses. Finalement, la récupération des produits en fin de vie devient rentable tout en mettant en place un système de planification du désassemblage.

Comme le problème de la planification de production, la planification du désassemblage est un problème de décision tactique. Il a fait récemment l'objet de plusieurs études dans la pratique industrielle. Le but de cette planification est de trouver la gamme optimale de désassemblage par rapport aux critères économiques. Les paramètres les plus importants dans la planification du désassemblage sont les coûts du processus et les revenus obtenus par la valorisation des composants désassemblés. Pour établir un plan optimal de désassemblage, il faut trouver un compromis entre ces deux paramètres. Son objectif est de déterminer les gammes de produits en fin de vie à désassembler afin de satisfaire les demandes pour leurs constituants sur un horizon temporel donné. C'est pourquoi dans cette étude nous nous sommes intéressés à ce type de problème de planification.

Ce chapitre est composé de quatre parties. Dans la première partie, nous exposons les définitions de la logistique inverse et les différentes motivations poussant les entreprises à mettre en place ce type de logistique. Ensuite, nous présentons les différents échelons de la logistique inverse et les types de produits retournés. Dans la deuxième partie, nous traitons le problème de planification en logistique inverse, plus précisément les travaux menés aux niveaux de décision stratégique, tactique et opérationnelle. Ils sont principalement la conception des réseaux logistiques, l'approvisionnement, la gestion des stocks et l'ordonnancement. Dans la troisième partie, nous présentons le problème de planification du désassemblage. Dans la quatrième et dernière partie de ce chapitre, nous étudions les différents modèles et méthodes d'optimisation de lot sizing en désassemblage et en production. Finalement, nous terminons ce chapitre par une conclusion générale de notre état de l'art.

## 1 Définitions de la logistique inverse

Pour des raisons économiques, sociétales et environnementales évoquées dans l'introduction de ce chapitre, la logistique inverse reçoit de plus en plus d'attention dans les organisations industrielles. Dans la littérature, nous rencontrons plusieurs définitions de la logistique inverse, mais le concept en lui-même reste flou. Chacun n'en a pas la même définition et il n'inclut pas les mêmes prestations, mais à la fin les définitions se rejoignent sur de nombreux points. Notre objectif dans cette partie est de retenir les définitions les plus souvent utilisées afin de se positionner par rapport à ces dernières. Nous abordons d'abord les différentes définitions de la logistique inverse proposées dans la littérature. Ensuite, nous analysons chaque définition en vue de donner une définition qui correspond au travail développé dans la suite de cette thèse. La littérature définit de nombreuses appellations pour désigner sensiblement la même chose. Les termes rencontrés dans la littérature sont : la logistique inversée, la logistique inverse, la gestion de retours, la distribution verte et la logistique verte. Ces termes sont utilisés pour référer essentiellement le même concept.

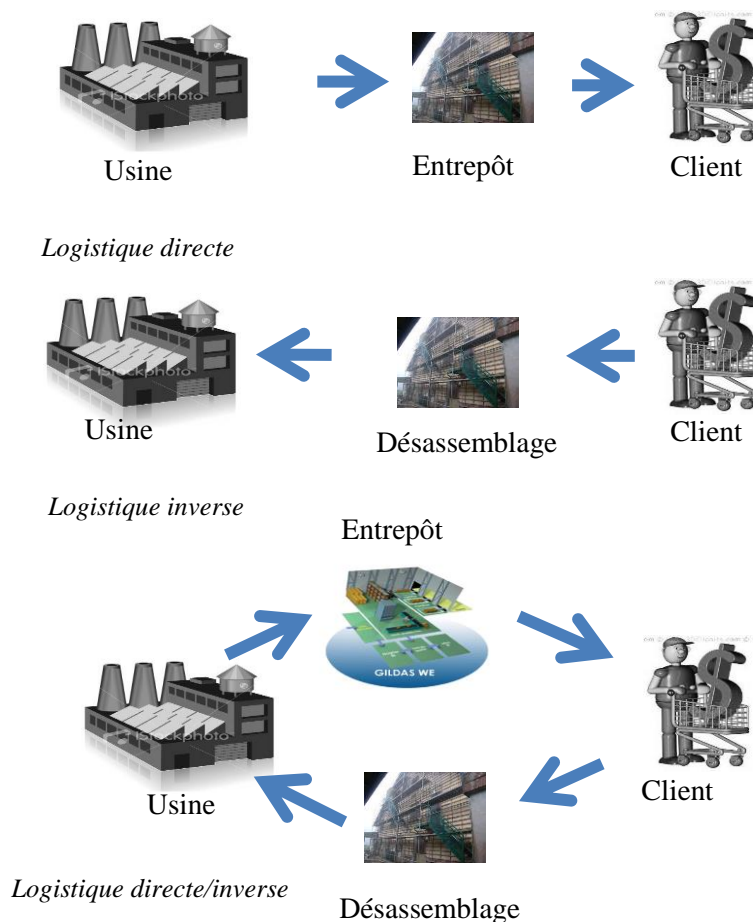
*Le dictionnaire (Dictionnaire AIPCR de la Route) définit la logistique inverse comme les Services logistiques se rapportant à la réutilisation et au retour des produits et matériaux : le terme "logistique inversée" est réservé au cas où le système de logistique existant est modifié pour accommoder les retours de marchandises. Lorsque les marchandises sont retournées au moyen d'un système de logistique distinct, on parle plutôt de "logistique inverse", de "logistique des retours" ou de "logistique à rebours".*

Le terme de logistique inverse est apparu dans les années 90. Ce dernier, pour la plupart des auteurs, semble regrouper sous une même définition de la logistique verte et de la distribution inverse. Les auteurs se sont ensuite intéressés à la logistique inverse afin d'élargir sa définition en se basant sur

celles de la chaîne d'approvisionnement traditionnelle. Plusieurs définitions ont été proposées pour définir la logistique inverse dans les années 90. La définition qui a été souvent utilisée est celle de (Thierry et *al.* 1995) en définissant la logistique inverse comme étant la gestion de la récupération des produits, des composants et des matériels usés ou éliminés par des entreprises manufacturières. L'objectif de la récupération des produits est de retirer la quantité extrême de déchets et réduire l'impact sur l'environnement. À la fin des années 90, une autre définition a été proposée par (Rogers et Tibben-Lembke 1999). Ces auteurs ont défini la logistique inverse comme étant un processus de planification, des en-cours de production, des produits finis, et d'informations y afférentes, du point de consommation jusqu'au point d'origine, dont le but de récupérer ou de créer de la valeur et d'améliorer l'élimination des déchets. Cette définition est la plus couramment utilisée dans la littérature. Il semble que la plupart des travaux de la littérature s'inspirent de cette définition pour définir la logistique inverse (leur article qui définit la logistique inverse a été cité plus de 1472 fois).

Dans les années 2000, (Chouinard 2007) commente la définition de (Rogers et Tibben-Lembke 1999), il précise que la logistique inverse n'est pas seulement une gestion unidirectionnelle de flux de matières et d'information. Il propose donc sa propre définition, et selon lui : la logistique inverse consiste à récupérer des biens à partir des consommateurs, et ensuite traiter les produits retournés dans le but d'en tirer le maximum de valeur en les réintégrant sur le marché. Par son champ d'action, il faut assurer la gestion et la planification des activités de collecte, d'évaluation, de tri, de désassemblage et de redistribution. D'après ces définitions, nous remarquons que certains auteurs qui traitent la logistique inverse s'entendent sur le fait qu'il est un domaine qui s'intéresse à la gestion des produits récupérés. D'autres auteurs traitent à la fois la logistique traditionnelle ou la logistique verte et la logistique inverse/directe. Très peu d'auteurs mettent un lien entre la logistique directe ou logistique verte et la logistique inverse/directe. Dans ce contexte, nous citons les travaux de (Salema et *al.* 2007) qui ont défini la logistique directe comme une chaîne de transmission reliant les usines vers les clients en passant par les entrepôts. Ils ont ensuite défini la logistique inverse comme une chaîne de retour reliant les clients vers les usines en passant par des centres de désassemblage. Finalement, la logistique inverse/directe a été définie comme l'association des flux directs à d'autres inverses, elle consiste à la distribution optimale et à la récupération directe et inverse comme le montre la Figure 2. Selon (Grellier 2008), la logistique inverse, comme son nom l'indique, se réfère aux activités de logistique d'une organisation mais dans un sens inversé à celui de la logistique traditionnelle ou directe. (Bai et Sarkis 2013) définissent la logistique inverse comme l'inverse de la logistique directe. Par conséquent, la logistique inverse peut prendre les termes suivants : logistique inversée, reverse logistics, et gestion de la récupération des produits.





**Figure 2 :** Schéma de la définition de logistique inverse proposée par (Salema et al. 2007)

En conclusion, nous pouvons remarquer que les définitions de la logistique inverse peuvent varier selon le type du produit, le type d'activité ou encore les motivations poussant les entreprises à mettre en place un système de logistique inverse.

Dans la littérature, de nombreux auteurs ont montré que les facteurs économiques, environnementaux, législatifs et écologiques sont les facteurs qui poussent le plus les entreprises à mettre en place un système de la logistique inverse. Selon (Rogers et Tibben-Lembke 1999) et (Carter et Ellram 1998) : les législations, l'aspect marketing et l'intérêt des consommateurs désirant des produits sains pour l'environnement, l'aspect économique et écologique sont les principales motivations qui poussent les entreprises à mettre en place le système de la logistique inverse.

#### a. Aspect économique

Le système de la logistique inverse peut apporter des gains directs aux entreprises lorsqu'une organisation est en mesure de valoriser un produit, habituellement caractérisé par un cycle de vie court (Chouinard 2007). L'étude menée par (Gupta et Wassenhove 2003) sur le recyclage des composants électroniques aux États-Unis a montré que ces derniers arrivent en fin de vie utile dans des périodes assez courtes avec des composants ayant une valeur intrinsèque. Selon (Bennekrouf et al. 2010), la reprise d'un produit pourra éventuellement générer des économies par rapport à la fabrication ou à l'achat d'un produit neuf. (Fleischmann 2000) confirme que la reprise d'un produit utilisé permet généralement de générer des économies par rapport à la fabrication ou l'achat d'un

produit neuf. (Louwers et al. 1999) soulignent que les entreprises peuvent s'impliquer dans la récupération comme une étape stratégique pour se préparer à la future législation. Finalement, l'aspect économique englobe des gains directs comme la matière première, la réduction du coût et la valeur ajoutée par la récupération. Il englobe également des gains indirects comme l'image des entreprises, l'amélioration de la relation client/fournisseur et la protection de l'environnement.

*b. Aspect écologique et environnemental*

Cet aspect est exigé surtout par la norme ISO 14000 et le «*Total Quality Environmental Management*». Ce sont deux notions qui deviennent de plus en plus importantes. Ces notions visent à sensibiliser les entreprises à développer des réseaux de logistique inverse afin de gérer le problème de pollution en préservant l'environnement. (Sarkis 1995) montre que les entreprises sont de plus en plus concernées par l'aspect environnemental. Ce dernier est important dans la protection de l'environnement et l'image de l'entreprise. (Carter et Ellram 1998) montrent que la mise en place des normes relatives au management environnemental dans les entreprises est indispensable. Cette mise en place pousse les responsables logistiques à prendre en compte l'impact sur l'environnement des activités logistiques. Ensuite, sensibiliser ces derniers à la prise en charge des systèmes de logistique inverse.

Il est également important de souligner que la mise en vigueur des réglementations environnementales sensibilise de plus en plus les entreprises à utiliser la logistique inverse. Ce type d'activité est assez critique pour les entreprises, car elles peuvent se retrouver face à des réseaux de logistique inverse à plusieurs échelons. Par ailleurs, la logistique inverse peut varier selon le type du retour de produits, selon les activités et selon le type de réseau.

Dans la suite de ce travail, nous classons les travaux de la littérature traitant la logistique inverse en se basant sur les définitions proposées par (Salema et al. 2007). En ce qui concerne les types de réseaux, nous trouvons des réseaux logistiques directes, des réseaux logistiques inverses et des réseaux logistiques directes/inverses. D'autres classifications sont basées sur le type de retours du produit que nous pouvons les trouver dans les travaux de (Chouinard 2007). Les classifications sont données dans les tableaux suivants.

**Tableau 1** : Classification par type de produit

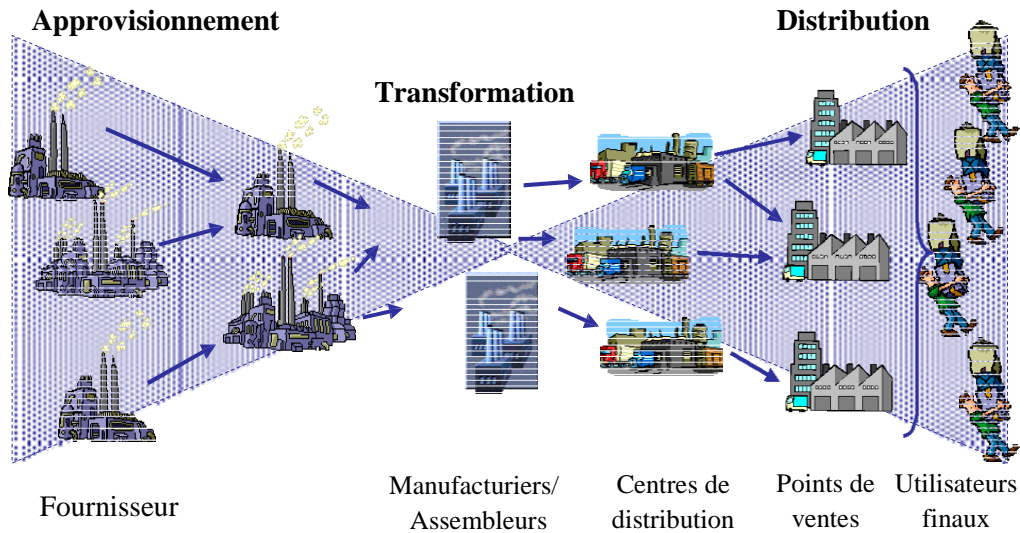
Auteurs	Éclectiques et Électroniques	Recyclage des déchets	Modèles théoriques
(Shih 2000)	✓		
(Dat et <i>al.</i> 2012)	✓		
(S.Sodhi et Reimer 2001)	✓		
(Meng 2008)	✓		
(Louwers et <i>al.</i> 1999)		✓	
(Krikke et <i>al.</i> 1999)			✓
(Tiwari et <i>al.</i> 2002)	✓		
(Barros et <i>al.</i> 1998)		✓	✓
(Hu et <i>al.</i> 2002)		✓	
(Alumur et Kara 2007)		✓	
(Jayaraman et <i>al.</i> 2003)		✓	
(Logozar et <i>al.</i> 2006)		✓	✓
(Pishvae et <i>al.</i> 2009)			✓
(Chen et <i>al.</i> 2007)			✓
(Si-Bo et Wei-Lai 2008)			✓
(Barker et Zabinsky 2011)			✓
(Lee et <i>al.</i> 2009)			✓
(Ramezani et <i>al.</i> 2013)			✓
(Achillas et <i>al.</i> 2010)	✓		✓
(Kim et <i>al.</i> 2006)			✓
(Kannan et <i>al.</i> 2009)		✓	✓
(Zhou et Zhou 2015)		✓	
(Salema et <i>al.</i> 2007)			✓

**Tableau 2** : Classification par type de réseau

Références	Type de réseau		
	Logistique directe	Logistique inverse	Directe/inverse
(Shih 2000)		✓	
(Dat et al. 2012)	✓	✓	
(S.Sodhi et Reimer 2001)		✓	
(Meng 2008)		✓	
(Krikke et al. 1999)		✓	
(Louwers et al. 1999)		✓	
(Barros et al. 1998)		✓	
(Hu et al. 2002)		✓	
(Jayaraman et al. 2003)	✓		
(Logozar et al. 2006)		✓	
(Chen et al. 2007)		✓	
(Si-Bo et Wei-Lai 2008)		✓	
(Lee et al. 2009)		✓	✓
(Salema et al. 2007)	✓	✓	
(Kannan et al. 2009)	✓	✓	
(El-Sayed et al. 2010)	✓	✓	✓
(Ramezani et al. 2013)	✓	✓	✓
(Lee et al. 2007)		✓	✓
(Hu et Sheu 2013)		✓	
(Zhou et Zhou 2015)		✓	
(Das et Chowdhury 2012)		✓	
(Kim et al. 2006)		✓	
(Pishvae et al. 2009)	✓	✓	✓
(Achillas et al. 2010)		✓	
(Assavapokee et Wongthatsanekorn 2012)		✓	
(Devika et al. 2012)		✓	
(Vahdani et al. 2012)	✓	✓	✓

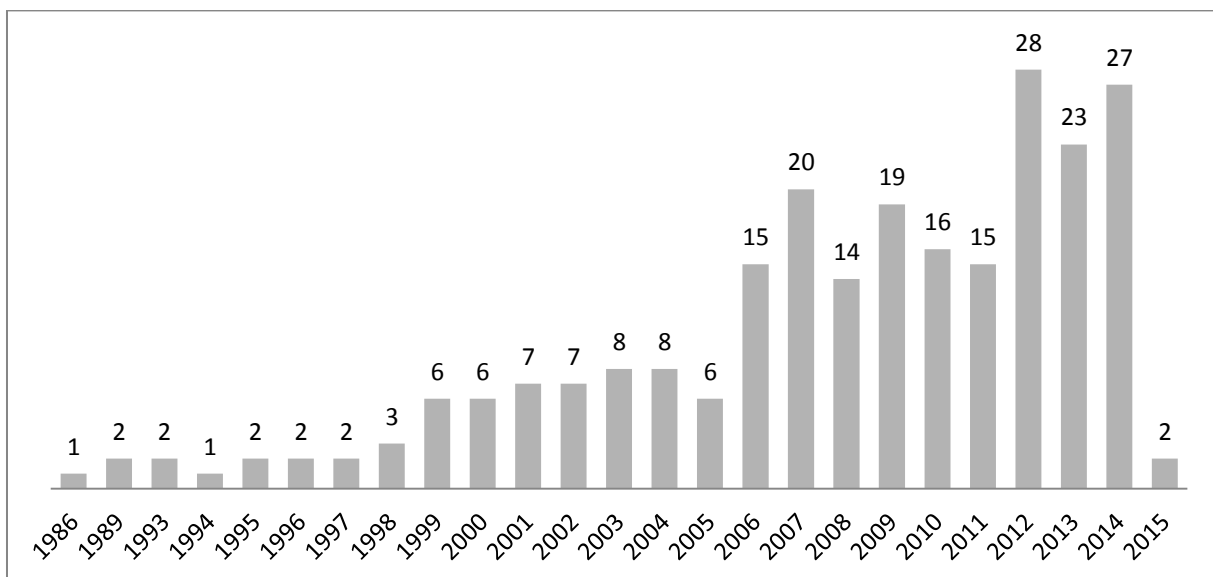
Suite aux définitions et classifications précédentes, nous pouvons remarquer que la définition de la logistique inverse la plus couramment utilisée dans la littérature est celle de (Rogers et Tibben-Lembke 1998), puisqu'elle fait référence à plusieurs auteurs. Nous pouvons conclure que la logistique inverse peut être définie comme étant une gestion des flux de retours des produits en fin de vie à partir

des consommateurs (clients) vers l'un des facteurs d'une chaîne d'approvisionnement. Le flux de retour comprend plusieurs étapes : la collecte, le transport, le traitement, le tri, le désassemblage, l'élimination et la distribution (Figure 3). Ainsi, la définition que nous proposons dans cette thèse est la suivante : « la logistique inverse regroupe toutes les opérations relatives à la réutilisation de produits en fin de vie ou usés avec des activités incluant : la collecte, le tri, le désassemblage et le traitement de produits et/ou de matériaux déjà utilisés en vue d'une utilisation, d'une réparation ou d'un recyclage ». Dans notre définition, nous intégrons également l'activité du désassemblage car elle présente une étape importante dans la reproduction (remise à neuf). En effet, le désassemblage est un processus important du moment qu'il tente de remettre le produit usagé dans un état standard de qualité.



**Figure 3 :** Flux de la logistique inverse (Chouinard 2007)

Les travaux de la logistique inverse ont été classés par nombre d'articles publiés par année (Agrawal *et al.* 2015). D'après la Figure 4, nous remarquons que la logistique inverse croît en fonction des années.



**Figure 4 :** Nombre d'articles publiés par année sur la logistique inverse (Agrawal *et al.* 2015)

Dans cette thèse, nous nous positionnons dans une chaîne de logistique inverse en prenant en compte l'activité de désassemblage. En effet, dans un système de logistique inverse, nous cherchons à valoriser les produits ou les composants retournés. L'objectif est de récupérer une quantité importante de produits en fin de vie afin de maximiser le gain de ces derniers. En outre, le désassemblage permet d'une part de récupérer les sous-ensembles et les constituants réutilisables et d'autre part, le tri des matériaux et l'extraction des composants qui peuvent être encore réutilisables.

Dans la suite de cette recherche, nous étudions un système de planification du désassemblage dans une chaîne de logistique inverse. Il s'agit de développer des modèles et méthodes de résolution en vue de minimiser les différents coûts liés à la gestion de stock et des opérations de désassemblage. L'objectif principal de la logistique inverse est d'assurer le retour des produits/matières de l'utilisateur vers le producteur. Cette dernière comprend de nombreuses activités qui sont principalement la collecte, le tri, la transformation, le reconditionnement, l'enfouissement et l'incinération. Par ailleurs, le concept de la logistique inverse ne se limite pas seulement aux activités de récupération et de recyclage des matières (rebut ou rejets de production ou retours d'emballage), il intègre également les activités de retour des produits pour cause de défautuosité ou de non-conformité aux besoins (retours sous garantie, produits en fin de vie, retours commerciaux, ...). Ces activités nécessitent une planification sur différents horizons temporels.

Dans cette première section, nous avons analysé les travaux de la littérature sur la logistique inverse. Nous avons montré l'importance de la logistique inverse et ses principales définitions proposées dans la littérature. Ensuite, nous avons étudié les principales raisons poussant les entreprises à mettre en place la logistique inverse. Finalement, nous avons classé les travaux de la littérature selon différents types de réseaux et des retours de produits. Dans la prochaine partie, nous allons étudier la planification en logistique inverse ainsi que les travaux menés au niveau de décision stratégique, tactique et opérationnelle.

## **2 Planification en logistique inverse**

La logistique inverse vise la valorisation des produits en fin de vie via un réseau de création de valeur intégrant les processus de récupération, de traitement, de recyclage, de désassemblage de distribution ou d'élimination propre. Pour concevoir un réseau de logistique inverse robuste et durable, les prises de décisions au niveau des activités des différents éléments constituant ce réseau doivent être cohérentes (Bennekrouf et *al.* 2010). Ainsi, les activités de la logistique inverse doivent suivre des changements hiérarchiques par ordre de priorité sur un horizon de planification, une classification regroupant trois niveaux de décisions stratégiques, tactiques et opérationnelles.

Comme pour la planification de production, la planification de la logistique inverse peut se décomposer en trois niveaux de décisions selon l'horizon du temps sur lequel elle s'applique : stratégique, tactique et opérationnelle. En effet, dans le processus de planification de la production nous devons décider quels produits sont nécessaires et quand. Le problème de désassemblage est donc la grande variété de produits qui doivent être désassemblés et leurs états incertains.

- a. Niveau stratégique* : ce niveau a trait au long terme, nous nous focalisons sur la conception de la chaîne de retour, la localisation de l'installation, l'ouverture ou la fermeture des sites industriels et la sélection des sites de retour.
- b. Niveau tactique* : ce niveau concerne le moyen terme et donne à l'entreprise les moyens de réaliser le niveau stratégique.

- c. *Niveau opérationnel* : assure le suivi et le contrôle de flux physique afin d'assurer la mise à disposition des produits à chaque maillon selon les conditions définies au niveau tactique (Schmidt et Wilhelm 2000). D'une manière analogique, ces niveaux peuvent être également appliqués aux chaînes logistiques inverses et à la planification du désassemblage.

## 2.1 Conception de réseaux logistiques inverses

Les travaux menés aux niveaux des décisions stratégiques se concentrent principalement sur la conception de réseaux logistiques inverses. Il s'agit des problèmes de localisation et d'allocation avec prise en compte de l'optimisation des coûts totaux. Plusieurs auteurs ont conçu des modèles mathématiques afin de concevoir des réseaux de récupération de produits à coût minimal. Certains modèles traitent à la fois la conception de réseaux globaux de la logistique inverse et ensuite intègrent la logistique inverse dans un réseau existant. La localisation et la taille des usines constituent le réseau et les coûts totaux composés du coût d'ouverture de nouveaux sites, du coût de transport, du coût de collecte...etc. D'autres modèles cherchent à optimiser seulement les coûts totaux dans un réseau logistique inverse.

L'objectif de la conception du réseau logistique inverse est de déterminer les emplacements et leur capacité en minimisant les flux entre les installations. Ce type de problème est généralement appelé par l'allocation-localisation. Ce dernier implique les investissements dans les installations, les coûts d'exploitation, de transport et de stockage. Ces coûts sont particulièrement intéressants pour les grandes entreprises à grande échelle. Nous regroupons ainsi les travaux de la littérature traitant la conception de réseaux logistiques inverses par type de produit.

### a. *Produits électriques et électroniques*

Dans le contexte de recyclage des produits électriques et électroniques en fin de vie, le travail de (Shih 2000) cherche à déterminer le nombre des sites de collectes, de recyclages et d'allocation des flux entre ces sites. Il s'agit d'optimiser le coût total dans un réseau de récupération d'appareils électriques. L'étude menée par (Dat et al. 2012) est également réalisée sur le recyclage des produits électriques et électroniques en fin de vie à Taiwan. Le but étant de minimiser le coût total du processus de recyclage et de déterminer ensuite le nombre des centres de collectes. Les auteurs ont proposé un modèle de programmation linéaire afin de minimiser les coûts de transport, d'exploitation et d'élimination. Dans le même contexte, (Sodhi et Reimer 2001) ont présenté des différentes étapes du processus de recyclage des produits en proposant un modèle correspondant à chaque étape. L'objectif étant de déterminer la quantité des produits à traiter dans chaque centre afin de maximiser le bénéfice total du processus de recyclage des produits électriques aux États-Unis. (Meng 2008) a également travaillé sur le recyclage des déchets électriques composés de plusieurs métaux et éléments toxiques. Il a proposé un modèle de programmation linéaire mixte en nombres entiers afin de déterminer la localisation des centres de traitement en minimisant les coûts fixes de traitement. (Krikke et al. 1998) traitent un problème de retour de photocopieurs du Pays-Bas à la république Tchèque. Les auteurs ont également proposé un modèle de programmation linéaire en nombres entiers. La modélisation des processus de retours des photocopieurs passe par les étapes suivantes : les machines retournées sont d'abord réparées, stockées, transportées et ensuite démantelées. L'objectif de ce modèle cherche à optimiser le processus de retour des photocopieurs dont le but de minimiser les coûts de traitement, de transport et de stockage.

#### *b. Recyclage des tapis*

(Louwers et *al.* 1999) ont développé un modèle pour le recyclage des tapis. Leur modèle vise à déterminer les localisations des installations, les auteurs ne prennent pas en considération les restrictions sur l'emplacement géographique des installations. Le modèle proposé comprend la quantité de produit résiduel transporté annuellement de la source préliminaire à la fin du réseau de la chaîne de recyclage. Ils cherchent à minimiser le coût de transport et le coût annuel de traitement, et ensuite déterminer la quantité de produit entreposé ainsi que la quantité maximale du produit résiduel à vendre.

#### *c. Industrie automobile*

(Krikke et *al.* 1999) traitent un problème de retour dans l'industrie automobile. L'objectif est de concevoir et optimiser un réseau de retour des produits retournés, et ensuite déterminer la localisation des installations et du routage. Le modèle est composé de fournisseurs de produits, des centres de traitement et de recyclage. La fonction objectif minimise le coût total composé du coût de transport et du coût d'investissement.

#### *d. Recyclage du sable*

Les travaux de (Barros et *al.* 1998) se concentrent sur le traitement du problème de retours du sable. L'objectif étant de déterminer le nombre et la localisation des installations ainsi que leurs capacités de stockage dont le but de minimiser le coût de chaîne de recyclage du sable. Les paramètres de coûts considérés sont les coûts de collecte, de stockage, de nettoyage, de distribution, de transport et les coûts fixes de l'ouverture des sites d'installation. L'objectif du modèle proposé est de minimiser les coûts fixes et de transport. Les auteurs ont supposé que les autres coûts n'ont pas une influence significative sur le coût total de la chaîne de logistique inverse.

#### *e. Recyclage de déchets dangereux*

Une étude a été réalisée par (Hu et *al.* 2002) sur le recyclage des déchets dangereux. Les auteurs proposent un modèle de programmation linéaire en nombres entiers afin de minimiser le coût total du retour de déchets dangereux, et ensuite déterminer la quantité de collecte, de stockage et de traitement. (Alumur et Kara 2005) proposent également un modèle afin de déterminer les localisations des centres d'élimination et l'itinéraire du transport de déchets dangereux à la destination du centre de stockage. L'objectif est de minimiser le coût de transport, le coût de traitement et le coût d'élimination ainsi que la minimisation de risque sur l'environnement provoqué par le transport de déchets dangereux. (Jayaraman et *al.* 2003) proposent également un modèle de programmation linéaire en vue de minimiser le coût d'ouverture des centres de collecte et le coût du transport.

#### *f. Recyclage de l'aluminium*

(Logozar et *al.* 2006) ont réalisé une étude approfondie sur le processus de recyclage de l'aluminium. Les auteurs proposent deux modèles afin de minimiser le coût annuel de transport. Le but étant de réduire le déchet de l'aluminium interne et minimiser le coût de transport. Le premier modèle est composé de cinq sites de collectes qui sont situés à côté de chaque processus de l'unité de production, le deuxième modèle est composé d'un seul site de collecte de déchets d'aluminium. Dans leur étude, les auteurs cherchent à déterminer le nombre de sites de traitement et de recyclage, ainsi que leurs capacités de traitement, et ensuite minimiser les coûts de transport, de traitement et d'élimination.



Contrairement aux problèmes de niveau stratégique, le niveau tactique regroupe les problèmes de décisions s'appliquent sur un horizon à moyen terme (entre 1 et 2 ans). Ces derniers ont trait à des problèmes aussi divers comme la gestion de stock, la planification du désassemblage et de transport ou encore le besoin en ressources humaines. Dans la suite de cette partie, nous allons voir en détail les concepts et les travaux menés au niveau de décision tactique ainsi que les méthodes et les modèles d'optimisation développés dans la littérature au niveau de décision tactique.

## **2.2 Planification et gestion d'approvisionnement**

Cette partie se concentre sur une analyse détaillée des modèles et des méthodes appliquées au problème de planification tactique des opérations de désassemblage et de logistique inverse. L'objectif de planification du désassemblage est de déterminer sur un horizon à moyen terme la quantité de produits ou des pièces/composants à désassembler afin de satisfaire la demande de ces derniers. La problématique de planification comprend plusieurs paramètres : l'horizon de planification, la demande de clients, la quantité désassemblée, la quantité en stock, la satisfaction des demandes ainsi que la gestion des retours et des approvisionnements. D'autres paramètres comme les coûts de setup et de stockage sont également considérés.

Dans le désassemblage axé sur la demande ou le désassemblage à la demande, les produits en fin de vie sont désassemblés une fois qu'il y a une demande (Langella 2007). Dans le désassemblage à l'offre, les produits doivent être désassemblés, mais il n'est pas nécessaire de répondre à toutes les demandes (Duta 2006). Les problèmes de planification du désassemblage traités dans la littérature peuvent être classés en déterministes (Lee et al. 2004), (Lee et Xirouchakis 2004), (Xanthopoulos et Iakovou 2009), (Kim et al. 2005), (Kim et al. 2006a), (Kim et al. 2006b), (Kim et al. 2006c), (Langella 2007), (Ilgin et Gupta 2009), (Kim et Lee 2011) et (Ji et al. 2015), et en stochastiques, (Inderfurth et Langella 2005), (Imtavanich et Gupta 2006), (Barba-Gutiérrez et Adenso-Díaz 2009) et (Kim et Xirouchakis 2010). Dans ce type de modèles, nous cherchons généralement à minimiser le coût de setup et de stockage sur un horizon de planification donné. (Spengler 2003) a développé un outil d'aide à la décision de recyclage de déchets électroniques dans une entreprise allemande avec une chaîne d'approvisionnement en boucle fermée. L'auteur a proposé un modèle de programmation linéaire afin de maximiser les revenus totaux et réduire le coût de stockage des composants.

En conclusion, le niveau tactique s'applique sur un horizon à moyen terme (1 à 2 ans), il consiste dans la planification et les opérations de désassemblage. Les décisions tactiques dans la logistique inverse se concentrent également sur la planification des opérations de désassemblage/recyclage ainsi que les retours de produits. Finalement, au niveau tactique de la logistique inverse, les décisions s'appliquent surtout dans la gestion des retours de produits en fin de vie.

## **2.3 Ordonnancement et conception des lignes de désassemblage**

Le niveau opérationnel regroupe principalement les problématiques de l'ordonnancement des opérations de désassemblage où les décisions s'appliquent sur un horizon à court terme (des opérations journalières d'ordonnancement par exemple). Cette décision se concentre d'une manière globale sur le contrôle des niveaux de stocks ainsi que l'ordonnancement.

Dans une décision opérationnelle de logistique inverse, (Prakash et al. 2011) proposent une approche de recuit simulé pour résoudre le problème d'ordonnancement de désassemblage. Il s'agit de planifier les opérations de désassemblage afin de réduire les stocks de produits désassemblés avec

pièces communes. Dans la structure d'un seul produit à plusieurs niveaux, les pièces communes présentent un composant commun qui peut être obtenu à partir de deux ou plusieurs éléments intermédiaires ou d'au moins de deux produits. (Duhaime et *al.* 2001) ont également travaillé sur le niveau opérationnel, le but étant de minimiser le coût de stockage et de pénalité. Les auteurs montrent que la rupture sur les stocks peut être évitée si les conteneurs utilisés retournent plus rapidement au centre de distribution après leurs utilisations. Un modèle d'optimisation de la séquence de désassemblage des produits en fin de vie dans l'industrie automobile est présenté par (Go et *al.* 2012).

(Nakashima et Gupta 2012) ont réalisé une étude sur un système industriel en utilisant un système Kanban qui permet de faciliter l'ordonnancement dans un processus de désassemblage. Finalement, (Shih et Lee 2007) proposent un modèle en vue de minimiser la séquence d'un processus de désassemblage. L'objectif étant d'une part, de minimiser le coût total composé du coût de désassemblage, y compris le coût de la main d'œuvre, le coût de séparation, le coût d'élimination des déchets ordinaires et de déchets dangereux, et d'autre part, de maximiser le bénéfice total du processus de recyclage et minimiser l'impact total environnemental de ce processus.

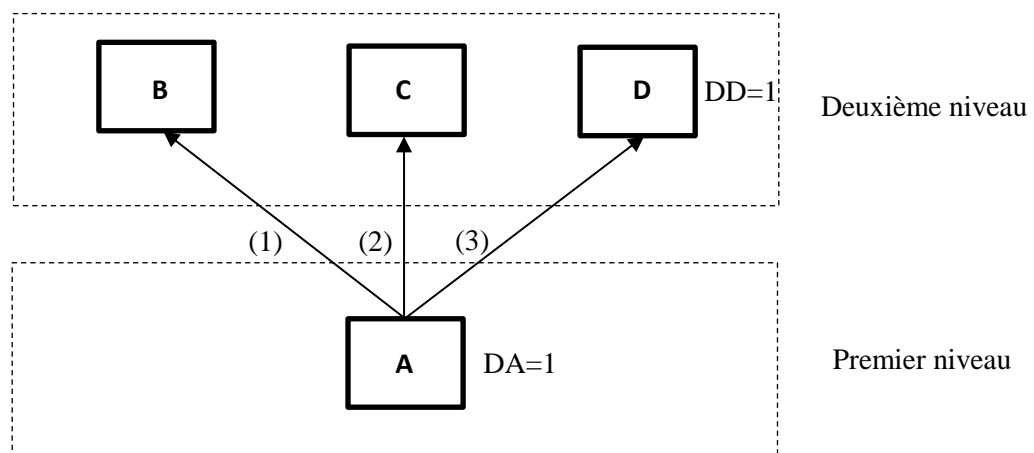
Les spécificités de la logistique inverse par rapport à la hiérarchisation des décisions sont diverses. La logistique inverse se focalise sur la conception de réseau logistique inverse, la stratégie de récupération des produits en fin de vie, le transport, la gestion des stocks, etc.

Dans cette section, nous avons analysé les travaux de la littérature menés dans un système de logistique inverse. Nous avons constaté que les problèmes au niveau de décision tactique liés à la logistique inverse et au désassemblage sont nombreux et diverses. En effet, la planification tactique du désassemblage sur un horizon de planification donné consiste à minimiser le coût de stockage et de setup. D'autres travaux cherchent à minimiser les coûts de setup et d'élimination. Ils cherchent également à déterminer la quantité à produire ainsi que la capacité des installations. D'une manière globale, l'objectif de la planification tactique est de trouver un plan permettant de répondre aux demandes afin de satisfaire les besoins des clients dans les temps prévus. Les modèles deviennent compliqués si certains paramètres sont stochastiques (demandes ou rendement de désassemblage par exemple) ainsi qu'avec contraintes de capacité. Le problème de lot sizing est également traité dans le cadre d'une décision tactique. Finalement, les décisions tactiques sont planifiées sur des horizons à moyens termes environ un à deux ans, les décisions opérationnelles sont planifiées sur des horizons à court terme environ quelques semaines ou un jour par exemple. Ces dernières consistent à gérer la planification et le fonctionnement quotidien de l'entreprise notamment l'ordonnancement et la gestion des stocks.

En se positionnant à un niveau tactique, nous pensons que l'étude de planification du désassemblage, selon la nature, la qualité et la quantité de produits retournés dans lequel le désassemblage est rentable, peut faire partie des préoccupations importantes dans une chaîne de logistique inverse. À ce sujet, nous allons étudier le problème de planification du désassemblage notamment les modèles et les techniques/méthodes de résolution et d'optimisation utilisées pour la résolution de ces problèmes.

### 3 Planification du désassemblage

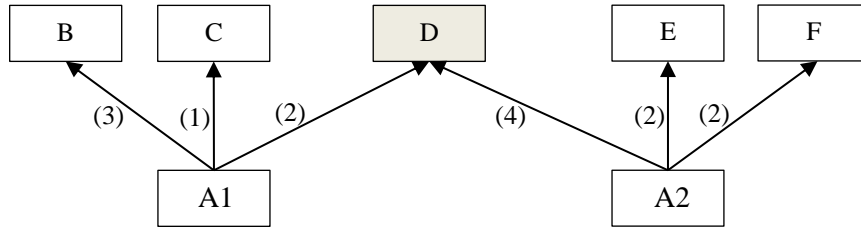
La planification du désassemblage vise à déterminer le type et le nombre de produits en fin de vie. Ensuite, déterminer les sous-ensembles et les composants à désassembler ainsi que les délais de livraison associés pour satisfaire la demande sur un horizon de planification donné. Dans la littérature, la plupart des auteurs supposent qu'il n'y a pas de limite sur le produit en fin de vie qui peut être désassemblé. Les demandes peuvent être satisfaites par le désassemblage de produits en fin de vie (Godichaud 2009). La structure du produit en fin de vie peut être représentée sous forme de nomenclature, cette dernière est composée du produit en fin de vie et ses composants constituants. Nous donnons un exemple de nomenclature pour une structure à un produit et deux niveaux comme le montre la Figure 5.



**Figure 5 :** Structure du désassemblage à deux niveaux (structure de base)

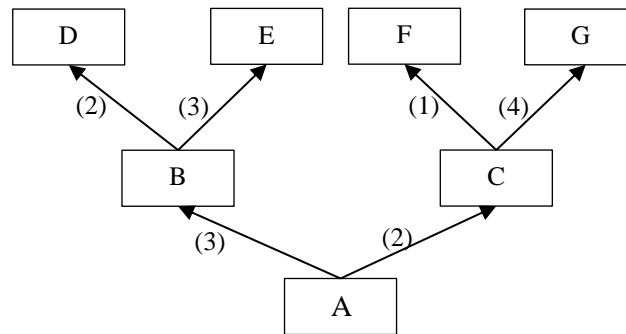
La Figure 5 illustre la nomenclature de désassemblage du produit A, ce produit représente le produit en fin de vie ou (root item en anglais), ce dernier forme le premier niveau. Les lettres B, C et D représentent les «composants» ou (leaf items en anglais), ces derniers forment le deuxième niveau. Le nombre entre parenthèse représente le rendement de désassemblage ou (yield en anglais), ou encore le nombre de composants obtenu par l'opération de désassemblage d'une unité de produit A. DA représente le délai d'approvisionnement et DD représente le délai du désassemblage. D'après l'exemple, le désassemblage du produit A permet d'obtenir un seul composant de B, deux composants de C et trois composants de D. Les demandes se situent aux niveaux des composants B, C et D. Nous notons que dans la nomenclature de désassemblage, nous pouvons rencontrer des structures de produits constituées de plusieurs niveaux.

Nous notons également que dans le cadre de cette thèse, les délais de désassemblage ne sont pas pris en considération (ils sont supposés négligeables). La structure du désassemblage peut également exister en plusieurs produits, plusieurs niveaux ainsi qu'avec pièces communes.



**Figure 6 :** Structure du désassemblage à deux produits avec pièces communes

D'après la Figure 6, le composant D est un composant commun entre les produits A1 et A2. Pour satisfaire la demande de ce dernier, il faut désassembler 2 composants du produit A1 et 4 composants du produit A2.

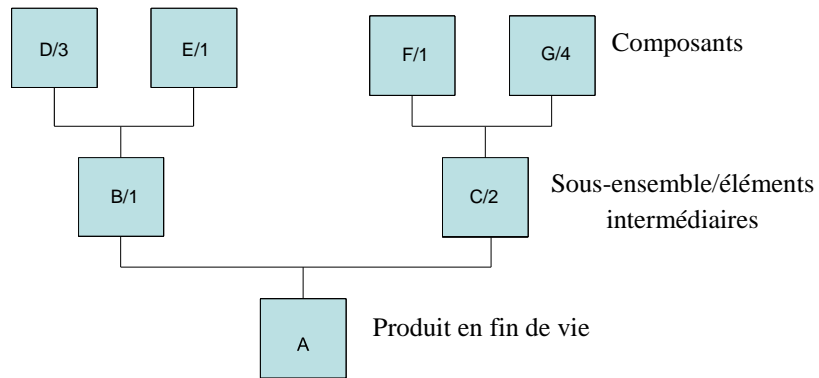


**Figure 7 :** Structure du désassemblage à plusieurs niveaux

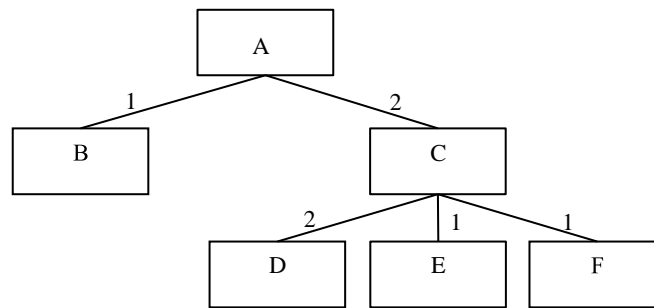
D'après la Figure 7, les composants B et C représentent les éléments intermédiaires, les demandes peuvent être également situées au niveau de ces composants.

La question qui se pose dans une structure du désassemblage à deux niveaux (Figure 5) : combien du produit A doit être désassemblé pour répondre à une demande  $X$  de chacun des composants A, B et C avec un coût minimal. Pour répondre à cette question, nous mettons en place un système de planification du désassemblage permettant de calculer la quantité  $X$ . Cette dernière est calculée à partir des quantités en stocks, des rendements de désassemblage et des demandes.

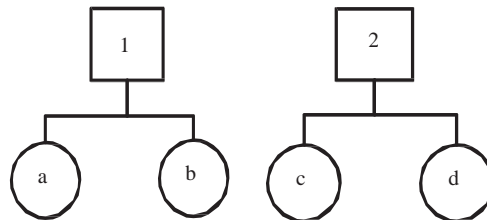
Les 3 premières structures que nous avons présenté avant sont les structures les plus souvent utilisées dans la littérature. Cependant, nous pouvons également rencontrer d'autres nomenclatures de produits.



**Figure 8 :** Structure du désassemblage à plusieurs niveaux (Barba-Gutiérrez et Adenso-Díaz 2009)



**Figure 9 :** Structure du désassemblage à plusieurs niveaux (Godichaud 2009)



**Figure 10 :** Structure du désassemblage à deux produits et deux niveaux (Langella 2007)

Dans la structure de la Figure 8, la numérotation de désassemblage commence de bas vers le haut, c'est-à-dire nous commençons par le produit en fin de vie et nous partons vers les composants. Les rendements de désassemblage sont indiqués directement sur les composants. Contrairement aux Figures 9 et 10, la numérotation commence de haut vers le bas. Les rendements de désassemblage sont indiqués sur les arcs. Dans ce présent travail, nous utilisons la structure la plus souvent rencontrée dans la littérature comme celle présentée dans la Figure 5.

## 4 Modèles pour l'optimisation de la planification du désassemblage

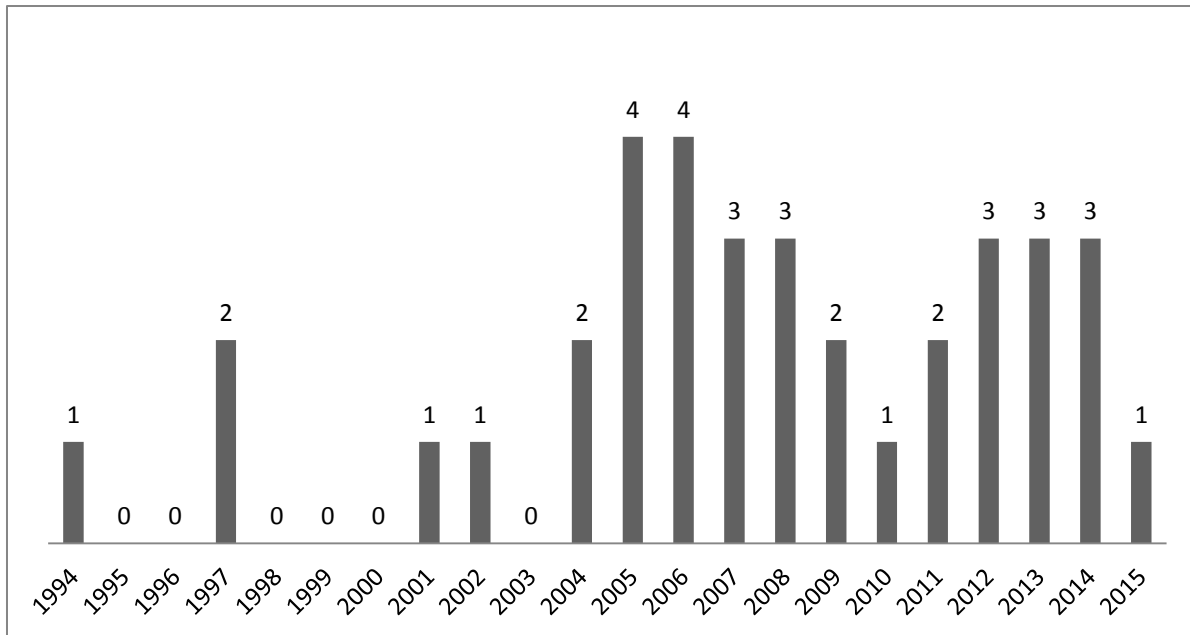
Comme la planification de la production, la planification du désassemblage s'applique sur un horizon à moyen terme (environ 12 à 24 mois), il s'agit d'une décision tactique. Dans cette section, nous présentons les modèles de planification du désassemblage rencontrés dans la littérature.

L'objectif des modèles de planification du désassemblage vise à déterminer les quantités des produits en fin de vie à désassembler en vue de satisfaire les demandes pour leurs constituants sur un horizon périodique ou continu. Dans la plupart des modèles rencontrés, il n'y a pas de contrainte sur la quantité de produit en fin de vie qui peut être désassemblé. Toutes les demandes de composants peuvent être satisfaites en traitant la quantité de produit en fin de vie nécessaire.

Le premier modèle de la planification du désassemblage à notre connaissance est celui proposé par (Gupta et Taleb 1994). La structure du désassemblage est composée d'un seul produit, plusieurs niveaux et sans pièces/composants communs. En 1997, (Taleb et Gupta 1997) et (Taleb et *al.* 1997) ont étendu leur modèle en considèrent la même structure mais avec prise en compte des pièces communes. À partir des années 2000, Kim, Xirouchakis et Lee ont largement travaillé sur le problème de lot sizing, leur dernier article a été publié en 2011. Entre 2012 et 2015, (Prakash et *al.* 2012), (Ullrich 2013) et (Sung et Jeong 2014) ont également étudié ce problème. Finalement, le dernier article sur le désassemblage vient d'être publié par (Ji et *al.* 2015). Le nombre d'articles publiés par année est présenté dans la Figure 11.

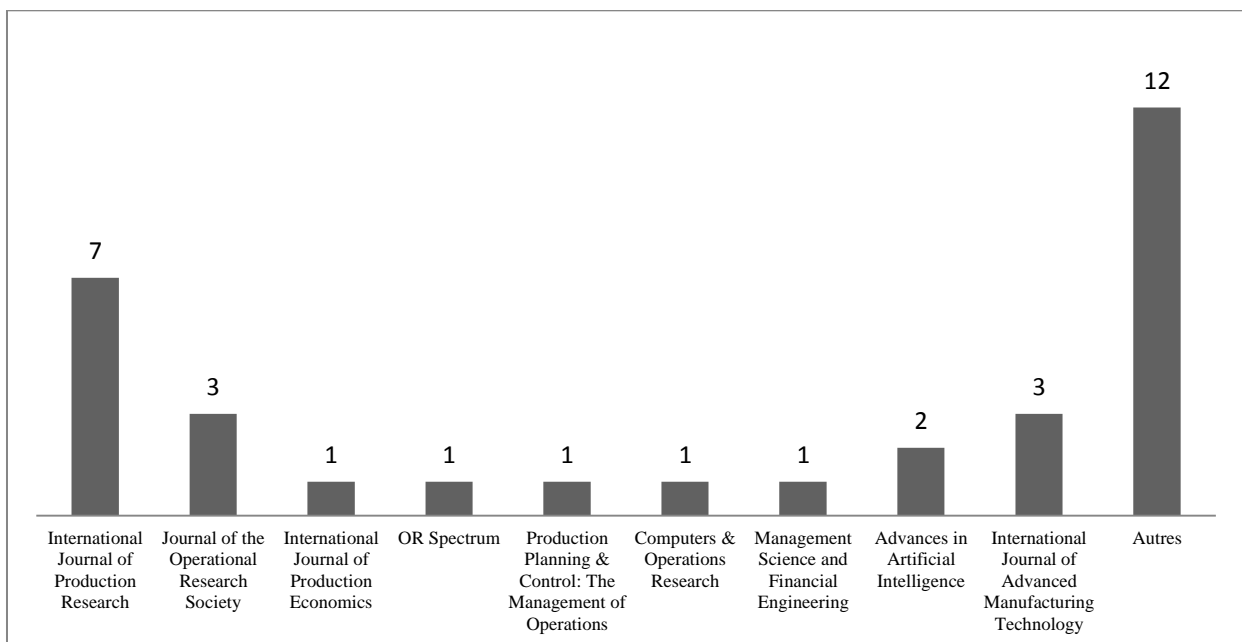
Par ailleurs, dans les premiers modèles de planification du désassemblage, les auteurs cherchent à calculer les besoins en retours des produits en déterminant seulement la quantité de produit à désassembler. Ils cherchent seulement à satisfaire les demandes de composants. Ensuite, ils ont intégré la problématique d'optimisation suivant différentes fonctions de coûts (coûts de stockage, de désassemblage et d'achat) comme il est proposé par (Lee et *al.* 2002). Ils ont intégré également de nouvelles contraintes (conservation du flux de stockage et des conditions sur les opérations de désassemblage). Finalement, les chercheurs ont intégré la contrainte de capacité et des paramètres stochastiques tels que les demandes et les rendements de désassemblage. Dans les deux dernières années, d'autres décisions ont été prises en compte sur l'achat des composants (Ji et *al.* 2015).

Ces problèmes peuvent être classés en fonction de nombre d'articles publiés par année et par type du journal.



**Figure 11 :** Nombre d'articles publiés par année

Les problèmes de planification du désassemblage sont également classés par type du journal (Figure 12). D'après cette figure, nous remarquons que la plupart des articles sont publiés dans le journal «International Journal of Operational Research». En effet, la majorité des travaux sur le lot sizing en désassemblage sont des problèmes d'optimisation traités avec les techniques de la recherche opérationnelle.



**Figure 12 :** Nombre d'articles publiés par type du journal

Dans la suite de ce travail, nous analysons les travaux de la littérature menés sur la planification du désassemblage. Nous nous intéressons plus précisément aux modèles proposés et aux approches et

techniques d'optimisation et de résolution utilisées, puisque certains auteurs utilisent des approches basées sur la planification notamment le MRP (*Material Requirements Planning - MRP*). D'autres proposent des modèles de programmation linéaire qui sont généralement connus dans le problème de lot sizing en désassemblage.

## **4.1 Planification des besoins**

La planification des besoins matières (*Material Requirements Planning - MRP*) est une étape de la planification des opérations. Dans la planification du désassemblage, cette approche permet de déterminer la date et la quantité de produits en fin de vie à désassembler afin de minimiser le coût lié à la gestion des stocks.

### **4.1.1 Méthodes MRP**

Le premier modèle de planification du désassemblage des produits en fin de vie a été proposé par (Gupta et Taleb 1994). Les auteurs ont proposé une approche MRP inversée. L'horizon de planification est décomposé en périodes et les temps de désassemblage sont proportionnels à une période. Sur chaque période, les quantités de sous-ensembles à désassembler sont déterminées pour satisfaire les demandes de composants. Concernant le désassemblage du produit en fin de vie, en remontant de bas vers le haut de la nomenclature afin de déterminer la quantité de produit à désassembler. Cet algorithme ne prend pas en compte les coûts liés à la gestion des stocks. Il fournit seulement une première solution au problème de planification du désassemblage. (Taleb et Gupta 1997) et (Taleb et al. 1997) ont également utilisé une approche MRP. L'objectif étant de déterminer les quantités et les dates de produits à désassembler pour satisfaire les demandes de composants. Pour résoudre ce problème, ils proposent deux algorithmes : «Core Algorithm» et «Allocation Algorithm». Le premier algorithme vise à déterminer la quantité de produits en fin de vie (produits retournés) à désassembler sur les différentes périodes de l'horizon de planification. Le deuxième vise à répartir les quantités désassemblées sur tout l'horizon de l'horizon afin de déterminer la quantité de composants. (Spengler et al. 2003) proposent un modèle de programmation linéaire pour la planification du désassemblage et de recyclage des produits en fin de vie. L'objectif étant de maximiser le revenu total du produit généré par la vente des pièces désassemblées. Les variables de décisions sont la quantité de déchets à recycler en interne et celle à recycler en externe. L'outil informatique LINGO est utilisé pour résoudre le problème. Les résultats ont montré que la décision concernant le recyclage des déchets en interne ou en externe est influencée par la situation du marché. En outre, le bénéfice augmente si une grande quantité de pièces est vendue et les pièces volumineuses doivent être recyclées en interne.

En conclusion, nous remarquons qu'avec l'approche MRP, les auteurs cherchent seulement à satisfaire la demande en déterminant la quantité à désassembler, ils ne prennent pas en considération l'optimisation des différents coûts de setup et de désassemblage. Cependant, l'optimisation de ces derniers est un critère important dans le problème de planification du désassemblage.

### **4.1.2 Optimisation des coûts**

Dans les problèmes de planification du désassemblage, les auteurs utilisent des approches MRP afin d'optimiser le coût d'acquisition du produit en fin de vie, les coûts de stockage et de setup. Ces derniers ne sont pas forcément des problèmes d'optimisation, ils sont plutôt des problèmes de planification. Dans ce contexte, (Barba-Gutiérrez et Adenso-Díaz 2008) ont proposé un modèle de planification du désassemblage basé sur l'approche MRP, ce modèle est une extension de celui proposé par (Gupta et Taleb 1994). Cette approche permet d'optimiser les coûts de stockage par



rapport aux coûts de réalisation des opérations de désassemblage. (Langella 2007) a également proposé un modèle de programmation en nombres entiers afin de minimiser les coûts de stockage, les coûts de mise à disposition des produits en fin de vie, les coûts de mise en décharge et les coûts de désassemblage. L'auteur a amélioré l'algorithme proposé par (Taleb et Gupta 1997) en vue de minimiser le coût total. Il a ensuite proposé un «Integral Algorithm» afin de déterminer la quantité de désassemblage pour chaque produit. (Xanthopoulos et Iakovou 2009) proposent un modèle de programmation linéaire en nombres entiers pour la récupération de produits en fin de vie. Ils ont utilisé une analyse multicritère pour identifier les principaux composants rejetés afin de sélectionner les composants à récupérer de chaque type du produit. Deux fonctions objectives ont été proposées : la première fonction permet de sélectionner les meilleurs composants qui peuvent être rentables à récupérer, et la deuxième cherche à maximiser le bénéfice total des produits récupérés.

Le problème d'optimisation multi-objectif est également traité dans le problème de planification du désassemblage. Dans ce contexte, nous citons les travaux de ( Krarup et Gupta 2002) qui étudient un système où les produits en fin de vie sont achetés auprès des utilisateurs, ensuite les produits collectés sont désassemblés dans une usine de désassemblage. Ils considèrent par la suite le processus de désassemblage destructif et non-destructif. Dans le cas d'un désassemblage destructif, les composants sont réutilisés, stockés et recyclés ou éliminés. L'objectif étant de déterminer la meilleure combinaison de plusieurs produits désassemblés afin de répondre à la demande pour les composants. Les auteurs cherchent à maximiser le profit et minimiser le nombre de pièces à éliminer, à stocker et celles à disposer. Un modèle de programmation en nombres entiers est proposé par (Lee et *al.* 2002). L'objectif est de minimiser les coûts de stockage, de production et d'achat. Une étude de cas sur le désassemblage des imprimantes a été effectuée. Les résultats montrent que CPLEX ne trouve pas de solution pour des grandes instances (30 périodes et 20 produits).

D'autres auteurs utilisent des modèles de programmation linéaire pour minimiser les coûts de stockage et de désassemblage (Lee et *al.* 2002). Ces modèles ont été également utilisés dans la minimisation des coûts fixes et variables ainsi que le coût de stockage. Ici, les problèmes d'optimisation sont formulés comme une programmation linéaire en nombres entiers (Kim et *al.* 2003), (Lee et *al.* 2004), (Kim 2005) et (Prakash et *al.* 2012). Très peu de travaux cherchent à minimiser les coûts de pénalité, de ventes perdues et d'achat respectivement (Kim et Xirouchakis 2010) et (Ji et *al.* 2015).

En conclusion, dans le problème de planification du désassemblage, de nombreux auteurs cherchent à minimiser le coût de stockage et de setup. Dans très peu de cas, le coût de recyclage et d'élimination sont également pris en compte. Dans le problème de désassemblage, il existe une incertitude très forte sur l'état de produits en fin de vie, puisqu'il arrive parfois que les pièces désassemblées soient défectueuses. Dans cette figure de cas, la prise en compte des incertitudes sur les demandes, sur le temps de désassemblage ou encore sur les rendements de désassemblage doivent être traités comme des paramètres incertains.

### **4.1.3 Incertitudes**

La prise en compte des incertitudes dans le problème de planification du désassemblage est une problématique compliquée. Cette dernière n'est pas souvent abordée dans la littérature, alors que c'est une option inhérente à la gestion des produits en fin de vie. En effet, dans un système de logistique inverse, il arrive que la qualité d'un produit retourné ne soit pas connue à l'avance, ainsi que celle des pièces récupérées à partir du produit retourné est soumise à des incertitudes. Dans ce cas, nous parlons d'un rendement de désassemblage incertain. Il arrive également que le temps de désassemblage et la

demande soient stochastiques. Par conséquent, les incertitudes inhérentes au problème de planification du désassemblage sont les rendements de désassemblage et les demandes incertaines. Dans le cas où le nombre de pièces à désassembler (rendement de désassemblage) ou la demande n'est pas connue à l'avance, les demandes peuvent ne pas être satisfaites, cela peut engendrer un surplus ou une perte de la demande non satisfaction de la demande).

(Inderfurth et Langella 2005) ont traité un problème avec rendement de désassemblage stochastique selon une loi de distribution normale. Pour modéliser ce problème, les auteurs ont utilisé deux méthodes pour satisfaire les demandes : la première méthode consiste à désassembler une quantité importante de composants en supposant que les rendements de désassemblage sont connus. La deuxième méthode consiste à augmenter cette quantité sur la base de rendement de désassemblage moyennement observé. Ces deux méthodes permettent de satisfaire la demande de tous les composants. Finalement, ils ont proposé deux heuristiques pour résoudre ce problème.

(Kim et Xirouchakis 2010) ont supposé que la demande est stochastique suivant une loi de distribution uniforme. Les auteurs supposent que la demande peut ne pas être satisfaite, la demande qui n'est pas satisfaite est pénalisée avec un coût de rupture. Ils ont proposé une heuristique lagrangienne en relaxant la contrainte de capacité. L'heuristique proposée a été performante en termes de qualité de solution et en temps d'exécution.

(Barba-Gutiérrez et Adenso-Díaz 2009) ont également proposé un modèle prenant en compte des incertitudes sur les demandes. Le modèle proposé est identique à celui proposé par (Barba-Gutiérrez et Adenso-Díaz 2008), il s'agit de déterminer les besoins sur chaque période. Les auteurs ont fait une estimation de la demande. Cette dernière est générée par l'introduction de bruit dans la demande réelle correspondante avec une probabilité de 80%. Le bruit étant positif ou négatif avec une probabilité de 50%. Ils ont ensuite utilisé deux approches de résolution : MRP inversée et MRP basée sur la logique floue. Puisque les demandes sont stochastiques, ces dernières peuvent ne pas être satisfaites sur certaines périodes de l'horizon. Ensuite, les demandes qui ne sont pas satisfaites sont reportées sur la période suivante comme des demandes déterministes. Les résultats ont montré que l'approche MRP basée sur la logique floue a été meilleure dans le cas d'une simple structure de produits à désassembler (deux niveaux sans pièces communes).

Les modèles rencontrés dans la littérature sur la planification du désassemblage cherchent à déterminer, sur un horizon de planification, les quantités de produits en fin de vie et leurs dates d'obtention afin de satisfaire les demandes au niveau des composants. Les chercheurs considèrent qu'il n'y a pas de limitation sur le nombre de produits en fin de vie qui peut être désassemblé. Toutes les demandes au niveau des composants peuvent être satisfaites en traitant la quantité de produit en fin de vie nécessaire. Toutefois, les approches MRP ne tiennent pas compte le coût de setup dans un système de planification du désassemblage. Pour minimiser les coûts de stockage, de setup et de désassemblage, les auteurs proposent de formuler le problème sous forme d'une programmation linéaire qui est également connue dans le problème de lot sizing en désassemblage. Comme le problème de planification du désassemblage, le problème de lot sizing en désassemblage est un problème d'optimisation dont l'objectif est de minimiser les coûts de stockage et de setup.

## 4.2 Problèmes de lot sizing en désassemblage

Dans un problème de lot sizing en désassemblage, nous cherchons à déterminer les quantités de produits en fin de vie à désassembler afin de satisfaire les demandes pour leurs composants constituants sur un horizon temporel donné. Les problèmes de lot sizing en désassemblage peuvent être classés en problèmes déterministes et stochastiques qui intègrent des facteurs aléatoires dans les modèles. Ils peuvent être également classés en problèmes mono et multi-produits, avec et sans pièces communes ainsi qu'avec l'existence des contraintes de capacité (tableau 3).

**Tableau 3 :** Classification des problèmes de lot sizing en désassemblage

Articles	Structures		Produits		Capacités		Type de modèle	
	APC <sup>1</sup>	SPC <sup>2</sup>	Un seul	Plusieurs	AC <sup>3</sup>	SC <sup>4</sup>	Déterministe	Stochastique
(Gupta et Taleb 1994)		✓	✓			✓	✓	
(Taleb et al. 1997)	✓		✓			✓	✓	
(Taleb et Gupta 1997)	✓			✓		✓	✓	
(Lee et al. 2002)		✓	✓		✓		✓	
(Kim et al. 2003)	✓			✓		✓	✓	
(Lee et Xirouchakis 2004)	✓		✓			✓	✓	
(Lee et al. 2004)	✓	✓	✓	✓		✓	✓	
(Kim et al. 2005)		✓	✓		✓		✓	
(Inderfurth et Langella 2005)	✓			✓		✓		✓
(Kim et al. 2006b)	✓			✓		✓	✓	
(Kim et al. 2006a)		✓	✓		✓		✓	
(Kim et al. 2006d)		✓	✓		✓		✓	
(Kim et al. 2006c)	✓			✓		✓	✓	
(Gao et Chen 2008)		✓	✓			✓	✓	
(Langella 2007)	✓			✓		✓	✓	
(Barba-Gutiérrez et Adenso-Díaz 2009)		✓	✓		✓			✓
(Kim et al. 2009)		✓	✓			✓	✓	
(Kim et Xirouchakis 2010)		✓		✓	✓			✓
(Kim et Lee 2011)	✓			✓		✓	✓	
(Prakash et al. 2012)	✓		✓			✓	✓	
(Sung et Jeong 2014)		✓	✓			✓	✓	
(Ullerich 2013)		✓		✓	✓		✓	
(Ji et al. 2015)	✓			✓	✓		✓	

<sup>1</sup> Avec pièces communes

<sup>2</sup> Sans pièces communes

<sup>3</sup> Avec capacité

<sup>4</sup> Sans capacité

## 4.2.1 Problème mono produit à deux niveaux

Dans cette section, nous présentons la modélisation du problème de lot sizing en désassemblage, sa complexité et ses techniques et méthodes de résolution. Ce problème est généralement appelé par le problème de base de lot sizing en désassemblage. L'objectif est de minimiser le coût de setup correspondant au produit en fin de vie, et le coût de stockage correspondant aux composants obtenus par l'opération de désassemblage. Dans la suite de ce travail, nous allons utiliser la notation SIDLS (Single Item Disassembly Lot Sizing) pour représenter ce problème.

### 4.2.1.1 Modélisation du problème

Afin de décrire le modèle mathématique, nous définissons les paramètres et les variables de ce problème comme suit :

#### Paramètres

$T$  : nombre de périodes de l'horizon de planification

$t$  : indices pour les périodes

$N$  : nombre de composants

$i$  : indices pour les composants

$a_i$  : rendement de désassemblage de composant  $i$

$d_{it}$  : demande pour le composant  $i$  à la période  $t$

$h_{it}$  : coût de stockage du composant  $i$  à la période  $t$

$s_t$  : coût de setup à la période  $t$

$M$  : nombre entier assez grand

#### Variables de décisions

$I_{it}$  : niveau de stock du composant  $i$  à la période  $t$

$X_t$  : quantité de désassemblage à la période  $t$

$Y_t$  : variable binaire égale à 1 s'il y a un désassemblage à la période  $t$ , et 0 sinon

En utilisant les paramètres et les variables ci-dessus, le modèle de lot sizing en désassemblage peut être formulé comme suit :

$$\text{Min } P1 \quad \sum_{t=1}^T s_t \cdot Y_t + \sum_{i=1}^N \sum_{t=1}^T h_{it} \cdot I_{it}$$

Sous contraintes :

$$I_t = I_{t-1} + a_i \cdot X_t - d_{it} \quad \text{pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (1.1)$$

$$X_t \leq M \cdot Y_t \quad \text{pour } t = 1, \dots, T \quad (1.2)$$

$$X_t, I_{it} \geq 0 \text{ et } X_t, I_{it} \in \mathbb{N} \text{ pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (1.3)$$

$$Y_t \in \{0,1\} \text{ pour } t = 1, \dots, T \quad (1.4)$$

La fonction objectif minimise la somme des coûts de setup et de stockage. La contrainte (1.1) représente la conservation des flux de stockage pour les composants. La contrainte (1.2) assure le lancement de désassemblage où la quantité produite ne doit pas dépasser le paramètre  $M$  ( $M$  est une valeur assez grande) qui peut représenter la somme maximale des demandes d'un composant sur l'horizon de planification. La contrainte (1.3) signifie que les variables  $X_t$  et  $I_{it}$  sont discrètes non négatives. La dernière contrainte (1.4) représente une variable binaire qui est égale à 1 si on produit à la période  $t$  et 0 sinon.

#### 4.2.1.2 Approches de résolution

Le problème SIDLS est un problème de base qui peut être résolu facilement à l'aide d'une méthode exacte. Dans la littérature traitant ce type de problème, on trouve seulement les travaux de (Kim 2005) qui a proposé une programmation dynamique basée sur l'approche de Wagner et Whitin pour résoudre le problème à des petites instances. L'auteur a montré ensuite que le problème peut être résolu avec une heuristique pour des grandes instances.

#### 4.2.2 Problème multiniveaux (avec ou sans pièce commune)

Le problème de base de lot sizing en désassemblage peut être étendu en tenant compte de plusieurs types de produits et plusieurs niveaux. Ce dernier a le même objectif que celui d'un seul produit. Dans la suite de ce travail, nous utilisons la notation MIDLS (Multi-Items Disassembly Lot Sizing) pour représenter ce problème.

##### 4.2.2.1 Modélisation du problème

Le modèle de lot sizing en désassemblage à plusieurs niveaux est similaire à celui d'un seul produit et deux niveaux en ce qui concerne la fonction objectif. Pour formuler ce problème, tous les éléments de la structure du désassemblage sont numérotés dans l'ordre topologique (de bas vers le haut et de gauche à droite ou selon la structure de produits à désassembler) en commençant par le produit. Les éléments sont numérotés par des entiers  $1, 2, \dots, i_l, i_l + 1, \dots, N$ , où  $i_l + 1$  est l'indice du premier composant. Les indices qui sont plus grands que  $i_l$  représentent les composants (Lee et al. 2004). Afin de décrire le modèle mathématique, nous définissons les paramètres et les variables de ce problème comme suit :

##### Paramètres

$\varphi(i)$  : parent de l'élément  $i$

$a_{ji}$  : rendement de désassemblage du composant  $i$  obtenu par le désassemblage de l'élément  $j$  ( $i < j$ )

$d_{it}$  : demande pour le composant  $i$  à la période  $t$

$h_{it}$  : coût de stockage du composant  $i$

$s_{it}$  : coût de setup du composant  $i$  à la période  $t$

$M$  : nombre entier assez grand

### Variables de décisions

$I_{it}$  : niveau de stock du composant  $i$  à la période  $t$

$X_{it}$  : quantité de désassemblage du composant  $i$  à la période  $t$

$Y_{it}$  : variable binaire égale à 1 s'il y a le désassemblage à la période  $t$ , et 0 sinon.

En utilisant les paramètres et les variables ci-dessus, le modèle de lot sizing en désassemblage peut être formulé comme suit :

$$\text{Min } P2 \quad \sum_{t=1}^T \sum_{i=1}^{i_l} s_{it} \cdot y_{it} + \sum_{t=1}^T \sum_{i=1}^N h_{it} \cdot I_{it}$$

Sous contraintes :

$$I_{i1} = I_{i,0} + a_{\varphi(i)i} \cdot X_{\varphi(i)1} - d_{i1} \quad \text{pour } i = 1, \dots, N \quad (1.5)$$

$$I_{it} = I_{i,t-1} + a_{\varphi(i)i} \cdot X_{\varphi(i)t} - X_{it} \quad \text{pour } i = 2, 3, \dots, i_l \text{ et } t = 1, \dots, T \quad (1.6)$$

$$I_{it} = I_{i,t-1} + a_{\varphi(i)i} \cdot X_{\varphi(i)t} - d_{it} \quad \text{pour } i = i_l + 1, \dots, N \text{ et } t = 1, \dots, T \quad (1.6')$$

$$X_{it} \leq M \cdot Y_{it} \quad \text{pour } i = 1, \dots, i_l \text{ et } t = 1, \dots, T \quad (1.7)$$

$$X_{it} \geq 0 \quad X_{it} \in \mathbb{N} \quad \text{pour } i = 1, \dots, i_l \text{ et } t = 1, \dots, T \quad (1.8)$$

$$I_{it} \geq 0 \quad I_{it} \in \mathbb{N} \quad \text{pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (1.8')$$

$$Y_{it} \in \{0,1\} \quad \text{pour } i = 1, \dots, i_l \text{ et } t = 1, \dots, T \quad (1.9)$$

La fonction objectif minimise la somme des coûts de setup et de stockage. Les contraintes (1.5), (1.6) et (1.6') représentent la conservation des flux de stockage pour les produits en fin de vie et les composants respectivement. La contrainte (1.7) assure le lancement du désassemblage où la quantité produite ne doit pas dépasser le paramètre  $M$  ( $M$  est une valeur assez grande) qui peut représenter la somme maximale des demandes parmi tous les composants sur l'horizon de planification. Les contraintes (1.8) et (1.8') signifient que les variables  $X_{it}$  et  $I_{it}$  sont discrètes non négatives. La dernière contrainte (1.9) représente une variable binaire égale à 1 si on produit à la période  $t$  et 0 sinon.

#### 4.2.2.2 Approches de résolution

(Kim et al. 2006c) et (Kim et al. 2009) ont montré que le problème MIDLS est NP-difficile. Pour résoudre ce dernier, plusieurs auteurs ont utilisé des méthodes approchées.

Dans (Kim et al. 2006d), les auteurs cherchent à minimiser le nombre de produits à désassembler. Ils ont proposé un algorithme en relâchant la contrainte des variables entières (quantité de désassemblage et de niveau de stock). (Lee et Xirouchakis 2004) considèrent le même problème que celui traité par (Lee et al. 2004). Ils ont développé une heuristique afin de résoudre les problèmes de grande taille. (Kim et al. 2006a) et (Kim et al. 2008) ont développé une méthode exacte pour résoudre le problème à un seul produit et plusieurs niveaux.

(Kim et al. 2009) proposent un Branch and Bound pour résoudre le problème à plusieurs produits. Ensuite, la solution optimale est obtenue sous la vérification d'une condition. En effet, si le produit désassemblé a au moins un composant dans le stock, ce dernier ne peut pas être désassemblé à la prochaine période. (Kim et Lee 2011) proposent un modèle pour le problème de lot sizing en désassemblage afin de minimiser les coûts de stockage et de setup. Ils ont proposé ensuite une heuristique basée sur la règle de priorité. Cette dernière donne de bonnes solutions dans le cas où le pourcentage de pièces communes est inférieur ou égal à 2%. (Kang et al. 2012) cherchent à minimiser les coûts de setup et de désassemblage. Ils ont traité deux problèmes différents. Le premier problème est considéré sans pièces communes qui est résolu à l'aide d'un algorithme en temps polynomial. Le second problème est considéré avec pièces communes qui est ensuite résolu en utilisant l'heuristique de (Kim et Lee 2011). Les résultats ont montré que l'heuristique a été efficace dans le cas où le pourcentage de pièces communes est faible (moins 2%). (Kim et al. 2005) ont proposé un algorithme pour résoudre le problème à plusieurs produits avec contraintes de capacité. (Kim et al. 2006b) ont proposé dans un premier temps une relaxation lagrangienne en relaxant la contrainte de variables entières (quantité de désassemblage et niveau de stock), puis dans un deuxième temps, ils ont proposé une programmation dynamique. (Kim et al. 2006c) ont également proposé une relaxation lagrangienne pour résoudre le problème sans contraintes de capacité.

Très peu d'auteurs ont utilisé des méta-heuristiques pour résoudre le problème MIDLS. Ces méta-heuristiques sont utilisées surtout dans des cas où CPLEX ne trouve pas de solution dans un temps raisonnable. (Gao et Chen 2008) ont proposé un algorithme génétique afin de minimiser les coûts de désassemblage, de stockage et de setup. La variable de setup compose le chromosome. Ensuite, cette variable a été générée initialement comme une variable binaire (0,1). Les résultats ont montré que l'algorithme permet de résoudre des problèmes de grande taille dans un temps relativement court (moins d'une minute). (Prakash et al. 2011) ont proposé un recuit simulé afin de minimiser les coûts de setup, de stockage, d'achat et de désassemblage.

#### 4.2.3 Problème multi-produit avec contraintes de capacité

Comme dans les problèmes de planification de la production, la contrainte de capacité des ressources est une considération importante dans la planification du désassemblage. En effet, la limitation de capacité dans une période est considérée comme un temps des opérations de désassemblage effectué durant cette période. Autrement dit, il existe une limite supérieure sur le temps disponible à chaque période de l'horizon de planification, chaque opération de désassemblage affectée à une période consomme une partie du temps disponible à cette période.

Dans la suite de cette partie, nous supposons que  $C_t$  la contrainte disponible à la période  $t$ . Soit  $R$  le nombre de produits en fin de vie et  $g_r$  le temps d'une opération de désassemblage du produit  $r$ . La contrainte de capacité est représentée comme suit :

$$\sum_{i=1}^{i_t} g_i \cdot X_{it} \leq C_t \text{ pour } t = 1, \dots, T \quad (1.10)$$

Ce problème peut être modélisé comme suit :

$$\text{Min } P3 \quad \sum_{t=1}^T \sum_{i=1}^{i_t} s_{it} \cdot Y_{it} + \sum_{t=1}^T \sum_{i=1}^N h_{it} \cdot I_{it}$$

Sous contraintes :

(1.5), (1.6) (1.6'), (1.7), (1.8) (1.8'), (1.9) et (1.10)

La contrainte (1.10) représente la contrainte de capacité dans chaque période. Le temps total nécessaire pour effectuer les opérations de désassemblage à chaque période doit être inférieur ou égal à la capacité de désassemblage disponible à cette période.

En ce qui concerne les travaux qui ont traité le problème avec contraintes de capacité, les chercheurs utilisent souvent des approches de relaxation lagrangienne. En effet, dans ces problèmes en relaxant les contraintes de ressource, nous trouvons plusieurs sous-problèmes de type SIDLS plus faciles à résoudre. (Kim et al. 2005), (Kim et al. 2006c) et (Kim et al. 2006d) ont traité le même problème que celui de (Lee et al. 2002) en intégrant le coût de setup. (Kim et Xirouchakis 2010) ont également traité le problème avec contraintes de capacité et incertitude sur la demande, ils ont ensuite relaxé la contrainte de capacité. (Kim et al. 2006d) ont proposé une heuristique de relaxation lagrangienne pour résoudre le problème avec contraintes de capacité. Ils ont relaxé la contrainte de capacité et ensuite calculé la borne inférieure. Ainsi, (Ji et al. 2015) ont traité le problème avec pièces communes, ils ont également relaxé la contrainte de capacité et celle de conservation des flux. L'objectif étant de minimiser les coûts de setup, de démarrage, de désassemblage, de stockage et d'achat. Ils supposent que si la demande ne peut pas être satisfaite, une quantité de composants doit être achetée afin de satisfaire cette dernière.

Les modèles de planification du désassemblage présentés précédemment considèrent les quantités de produits en fin de vie illimitées. Les demandes se situent au niveau des composants (le plus bas/haut niveau de la nomenclature), ces dernières doivent être satisfaites avec un coût minimal. La plupart des modèles rencontrés dans la littérature cherchent à déterminer les dates et les quantités de produits en fin de vie à désassembler. Par ailleurs, la problématique de non satisfaction de la demande ou de ventes perdues est un aspect important dans la planification du désassemblage. Cette dernière n'a pas encore été prise en compte dans les modèles traités dans la littérature.

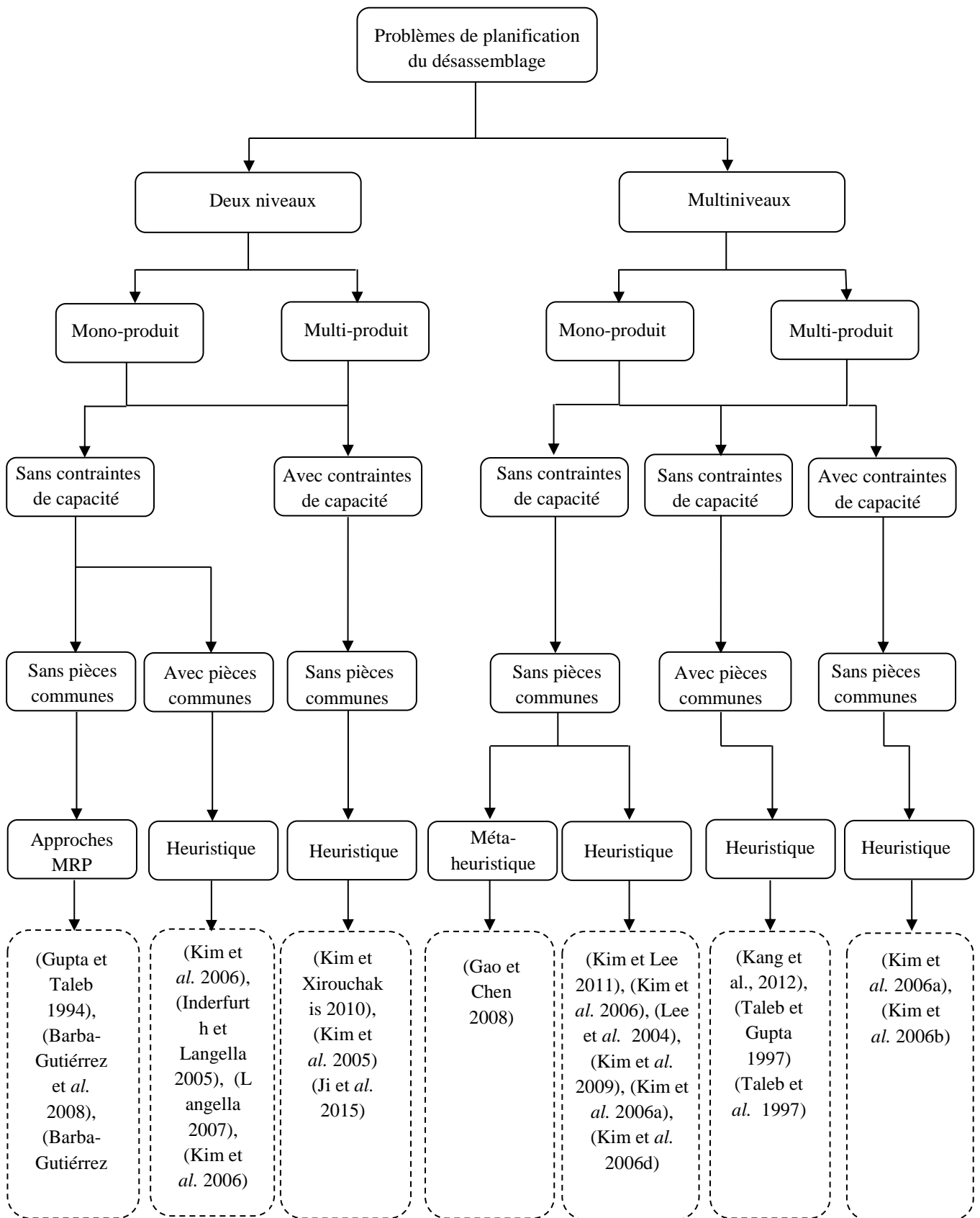
D'après l'analyse bibliographique, nous avons constaté que le problème de lot sizing en désassemblage à un seul produit et plusieurs niveaux est généralement résolu à l'aide des heuristiques. Dans très peu de travaux, les auteurs utilisent des méta-heuristiques notamment pour les problèmes de grande taille. Par contre, dans le cas des problèmes à plusieurs produits, les méthodes de résolution les plus souvent utilisées sont les relaxations lagrangiennes. Les contraintes relaxées sont souvent la contrainte de capacité et celles de variables entières notamment la quantité de désassemblage et le niveau de stock.

Dans cette partie, nous avons analysé des travaux de la littérature plus précisément des modèles et des approches et des techniques d'optimisation sur le problème de planification du désassemblage. L'objectif le plus fréquent de ces problèmes consiste à minimiser les coûts d'acquisition, de setup, de stockage et de désassemblage des produits en fin de vie. Ces problèmes peuvent être classés selon plusieurs catégories : par exemple les produits en fin de vie peuvent être classés de même type comme



il a été étudié par (Gupta et Taleb 1994), (Lee et Xirouchakis 2004) et (Barba-Gutiérrez et Adenso-Díaz 2008), plusieurs niveaux sans prise en compte des pièces communes (Kang et *al.* 2012), avec prise en compte des pièces communes (Taleb et Gupta 1997), (Taleb et *al.* 1997), (Kim et *al.* 2003), (Kim et Lee 2011) et (Kang et *al.* 2012). Certains travaux considèrent la contrainte de capacité (Lee et *al.* 2002), (Kim et *al.* 2006a), (Kim et *al.* 2006d) et (Ji et *al.* 2015). La plupart des chercheurs considèrent des modèles déterministes et incertains (Inderfurth et Langella 2005), (Barba-Gutiérrez et Adenso-Díaz 2009) et (Godichaud et *al.* 2010) et (Kim et Xirouchakis 2010). Ces travaux peuvent être résumés dans la Figure 13.

Pour vérifier si d'autres méthodes de lot sizing en production peuvent être utilisées dans la résolution de notre problème. Nous nous ne limitons pas seulement aux méthodes et techniques de résolution utilisées dans le lot sizing en désassemblage. Nous analysons également les travaux menés au niveau des problèmes de lot sizing en production. Nous présentons donc les modèles, les techniques et les méthodes de résolution utilisés dans ce domaine. L'objectif de cette étude est de sélectionner des techniques et des approches de résolution qui peuvent être applicables aux problèmes de lot sizing en désassemblage.



**Figure 13 :** Classification des problèmes de lot sizing en désassemblage

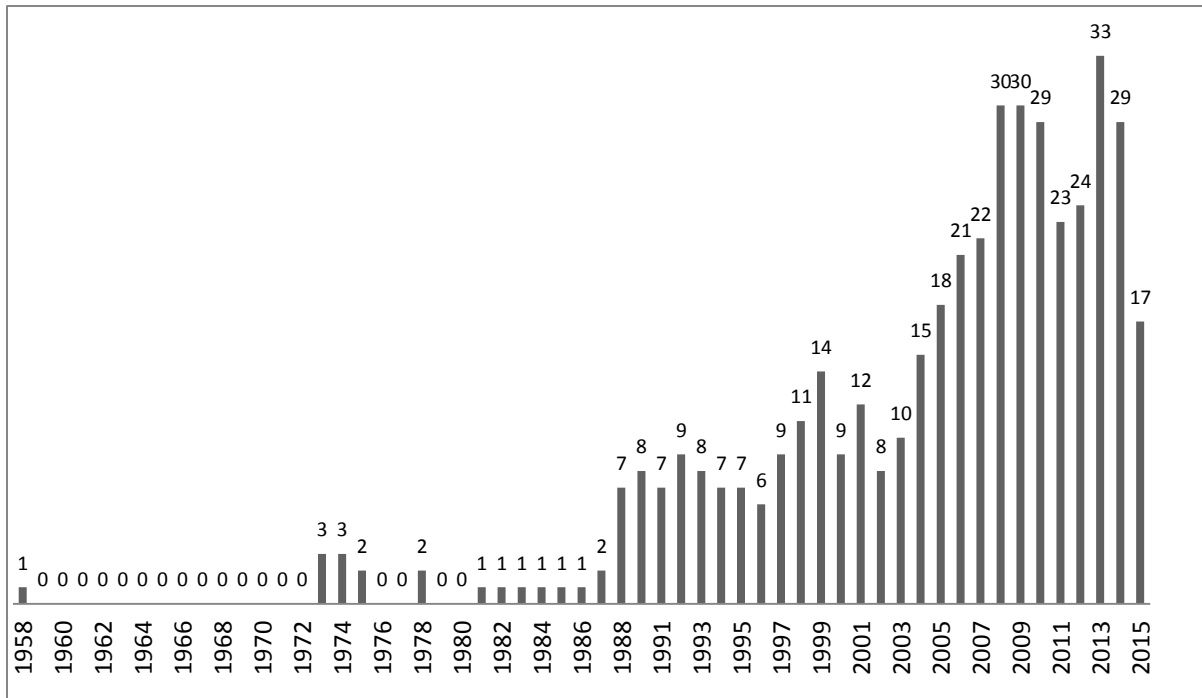
### 4.3 Problème de lot sizing en production

Cette partie se focalise sur le problème de lot sizing en production notamment les modèles et les méthodes de résolution. Il s'agit en fait de vérifier si certaines méthodes peuvent être utilisées ou adaptées pour la résolution du problème de lot sizing en désassemblage.

Le problème de lot sizing en production est l'un des plus importants et aussi les plus difficiles problèmes dans le cas de la planification de production. Il nécessite des méthodes de résolution spécifiques. Dans ce problème, nous cherchons à déterminer la taille optimale des lots de production. En effet, il est souvent trop coûteux de produire un produit à chaque période puisqu'il existe des coûts ou des temps de lancement associés à chaque produit. Contrairement, produire une grande quantité de produits dans une longue période peut engendrer des coûts de stockage importants. L'objectif des problèmes de lot sizing est donc de trouver un compromis entre les différents coûts afin de calculer une taille de production optimale. Ce sujet a été largement étudié dans la littérature. Par ailleurs, il existe certaines spécificités entre ce problème et celui de lot sizing en désassemblage.

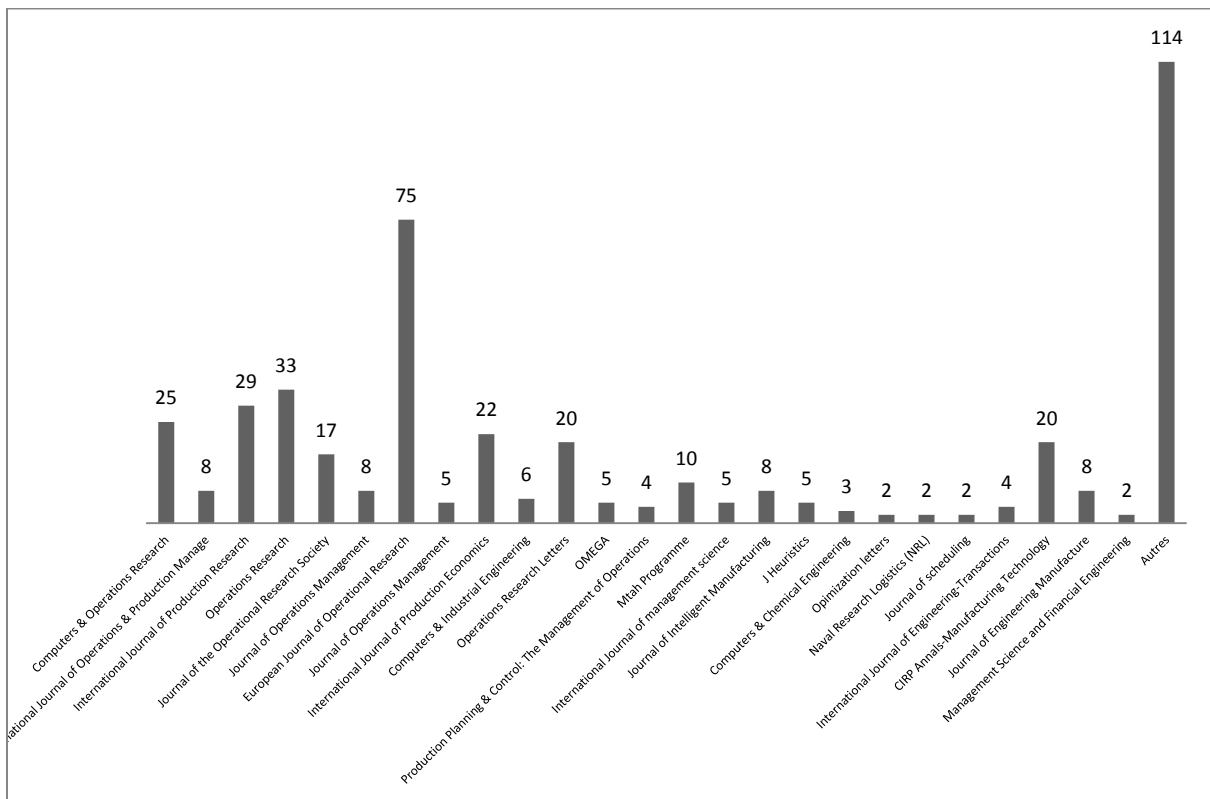
Selon (Duta 2006), il existe plusieurs différences entre la problématique de l'assemblage et de désassemblage. En ce qui concerne le problème de désassemblage, ce dernier possède des caractéristiques plus complexes que celles de l'assemblage. En effet, il existe d'abord un aspect de divergence pour le désassemblage c'est-à-dire un produit est composé de plusieurs composants. Contrairement à l'assemblage, plusieurs composants convergent vers un seul produit. En outre, dans le désassemblage, il existe une incertitude très forte sur l'état des produits en fin de vie, c'est-à-dire sur leur valeur et sur la difficulté opératoire pour les récupérer. De plus, dans le désassemblage les demandes se situent au niveau des composants élémentaires (plus haut niveau de la nomenclature). Suite à ces différences, les méthodes utilisées dans la résolution du problème de lot sizing en production ne peuvent pas être utilisées directement dans la résolution du problème de lot sizing en désassemblage. Cependant, ces dernières nécessitent des adaptations spécifiques pour résoudre ce dernier.

Les travaux sur lot sizing en production datent fin des années 50. Le premier modèle à notre connaissance est celui de Wagner et Whitin qui a été proposé en 1958. Pendant presque vingt-ans, il n'y a pas eu d'évolution sur ce problème. En 1974 (Adler et Nanda 1974) ont proposé un modèle à un seul produit en se basant sur les propriétés de Wagner et Whitin. Dans les années 80, il y avait très peu d'articles qui ont été publiés sur ce problème. Ensuite, à partir des années 90, on a eu une évolution significative jusqu'aux années 2000. Finalement, entre les années 2000 et 2015, nous remarquons qu'il y a une hausse d'articles publiés (Figure 14).



**Figure 14 :** Nombre d'articles publiés par année

Nous pouvons aussi classer les problèmes de lot sizing en fonction du type de journal (Figure 15). D'après cette figure, nous remarquons que la plupart des articles sont publiés dans le journal «European Journal of Operational Research».



**Figure 15 :** Nombre d'articles publiés par type du journal

Dans la suite de ce travail, nous allons présenter des articles de l'état de l'art sur le problème de lot sizing en production. Ensuite, nous classons les modèles par type du problème notamment aux problèmes à un seul niveau et un produit, aux problèmes à un seul niveau et plusieurs produits, et aux problèmes à plusieurs niveaux. Finalement, nous allons présenter pour chaque type du problème, sa modélisation, sa complexité ainsi que ses méthodes de résolution (méthodes exactes, méthodes approchées ou autres).

#### 4.3.1 Articles des états de l'art

Dans cette partie, nous présentons des travaux de l'état de l'art sur le problème de lot sizing en production. Nous commençons par les travaux de (Maes et Wassenhove 1988) qui ont présenté un état de l'art sur les problèmes de lot sizing à un seul niveau et plusieurs produits avec les techniques de résolution proposées dans la littérature. (Yano 1995) a proposé un état de l'art sur le problème de lot sizing en production avec rendement d'approvisionnement incertain. (Drexl et Kimms 1997), (Staggemeier et Clark 2001), (Jans et Degraeve 2008) et (Quadt et Kuhn 2008) ont étudié différents modèles de lot sizing, par contre ils n'ont présenté aucune approche de résolution. (Wolsey 1995) et (Brahimi et al. 2006) ont proposé un état de l'art sur le problème de lot sizing à un seul produit. (Karimi et al. 2003) ont proposé également un état de l'art pour le problème à un seul niveau avec contraintes de capacité, ils ont ensuite proposé différentes méthodes de résolution (méthodes exactes et approchées). Dans (Glock et al. 2014), les auteurs ont proposé un état de l'art permettant aux chercheurs d'identifier leurs travaux par rapport à ceux de la littérature en présentant les problèmes sous formes d'arborescences. Ils ont ensuite classé les problèmes de lot sizing en nombre d'articles publiés par année et par type de journal. Un état de l'art sur des méta-heuristiques adaptées aux problèmes de lot sizing en production a été proposé par (Jans et Degraeve 2007). Les auteurs ont montré que pour résoudre le problème de lot sizing à des grandes instances, les méta-heuristiques sont jugées très efficaces. Parmi les méta-heuristiques utilisées, on trouve souvent l'algorithme génétique et le recuit simulé.

Nous allons maintenant détailler les travaux de l'état de l'art en classant les problèmes à un seul niveau et un produit, à un seul niveau et plusieurs produits et aux problèmes à plusieurs niveaux.

#### 4.3.2 Problèmes mono-produit et un niveau

Dans cette partie, nous commençons d'abord par une simple représentation du modèle de lot sizing proposé par (Wagner et Whitin 1958). Ensuite, nous présentons la modélisation du problème, sa complexité, et les méthodes de résolution exactes et approchées qui sont adaptées pour ce type du modèle et/ou à des modèles similaires étendus. Dans la suite de ce travail, nous allons utiliser la notation USILS (Uncapacitated Single Item Lot Sizing) pour représenter ce problème.

Le premier modèle du problème de lot sizing à un seul produit à notre connaissance est celui de (Wagner et Whitin 1958). L'indice  $t = 1, \dots, T$  représente les périodes discrètes de l'horizon de planification avec  $T$  le nombre total de périodes. Les auteurs cherchent à minimiser le coût de stockage et de production. Sur tout l'horizon de planification  $T$ , les variables de décisions sont  $X_t$  qui représente la quantité produite à la période  $t$ ,  $Y_t$  est une variable binaire qui est égale à 1 si  $X_t > 0$  et 0 sinon, et  $I_t$  représente le niveau de stock.

#### 4.3.2.1 Complexité du problème USILS

(Bitran et Yanasse 1982) et (Florian et al. 1980) ont montré que le problème de lot sizing à capacité finie est NP-difficile. Ainsi, (Chen et al. 1994) ont montré que le problème n'est pas NP-difficile au sens fort en proposant un algorithme pseudo-polynomial. Les complexités de problème sont présentées dans la tableau 4.

**Tableau 4 :** Complexité du problème USILS

Problèmes	Complexité	Références
USILS	$O(T^2)$ ; $O(T \log T)$	(Wagner et Whitin 19958), (Aggarwal et Park 1993), (Federgruen et Tzur 1991), (Wagelmans et al. 1992).
Problèmes à capacité constante	$O(T^3)$	(Hoesel et Wagelmans 2001).

Les méthodes utilisées pour résoudre le problème USILS sont souvent la programmation dynamique. (Wagner et Whitin 1958) ont proposé un algorithme de programmation dynamique pour résoudre le problème USILS. Ce problème a été généralisé pour résoudre des modèles étendus avec contraintes de capacité. Les auteurs ont démontré qu'il existe une solution optimale telle que  $X_t \cdot I_{t-1} = 0 \forall t$ . Cette propriété signifie que pour une solution optimale, on ne produit jamais à une période ayant un stock non nul provenant de la période précédente. Cette dernière s'appelle la propriété de Wagner et Whitin. On se basant sur cette propriété, les auteurs ont résolu le problème USILS à l'aide d'une programmation dynamique en  $O(T^2)$ . (Martel et Gascon 1998) proposent également une programmation dynamique pour résoudre le problème USILS. (Federgruen et Tzur 1991) proposent une programmation dynamique basée sur l'approche de Wagner et Whitin avec une complexité en  $O(T \log T)$ .

(Wagelmans et al. 1992) ont proposé des algorithmes différents, mais avec la même complexité. L'objectif étant d'améliorer la complexité de l'algorithme de (Wagner et Whitin 1958), en passant son temps de calcul à  $O(T \log T)$ . (Loparic et al. 2001) ont également proposé une programmation dynamique pour traiter le problème de lot sizing à un seul produit, sans contraintes de capacité, avec des variables représentant les ventes au lieu des demandes fixes et des bornes inférieures sur les variables de stock, ensuite le problème a été résolu en  $O(T^2)$ . (Brahimi 2004) et (Dauzère-Pérès et al. 2005) ont étudié le problème de lot sizing avec des fenêtres de temps. Ils ont ensuite proposé un algorithme polynomial en  $O(T^4)$  pour résoudre le problème. (Absi et al. 2011) ont autorisé les backlogs, et ils ont ensuite proposé un algorithme de programmation dynamique pour résoudre le problème en  $O(T^2)$ . (Chen et al. 1994) et (Beltran et Krass 2002) ont également proposé un algorithme de programmation dynamique afin de résoudre le problème de lot sizing à un seul produit avec contraintes de capacité. Un problème NP-difficile avec backlog est étudié par (Hoesel et Wagelmans 2001), ce problème est également résolu à l'aide d'une programmation dynamique. Les auteurs ont résolu le problème en  $O(T^3)$ , ils ont ensuite démontré que le problème peut être résolu en  $O(T^2)$ .

D'après l'analyse des travaux de la littérature, on trouve que la méthode la plus appropriée pour résoudre les problèmes de lot sizing à un seul produit est celle de la programmation dynamique. (Beltran et Krass 2002) montrent que cet algorithme peut être résolu en  $O(T^3)$ , ils ont ensuite réduit la complexité en  $O(T^2)$  où le problème peut être résolu facilement. Comme la plupart des approches

exactes, l'algorithme de Wagner et Whitin a des difficultés pour résoudre des problèmes de grande taille.

Certains auteurs ont utilisé des heuristiques pour résoudre le problème USILS. (Barany et al. 1984) et (Aggarwal et Park 1993) ont proposé une heuristique pour résoudre ce problème. Les chercheurs ont généré les instances données par (Graves et Orlin 1985) et le modèle a été résolu en  $O(T^2)$ , ils ont ensuite adapté une heuristique de Silver Meal pour résoudre le problème. (Pochet et al. 1992) proposent deux modèles de lot sizing, ils ont ensuite proposé une heuristique pour la résolution de ces derniers. (Chu et al. 2013) ont proposé un algorithme polynomial pour le problème de lot sizing avec backlog. (Alfieri et al. 2002) proposent une heuristique pour résoudre le problème avec contraintes de capacité.

Dans le problème de lot sizing avec ventes perdues ou backlog, nous citons les travaux de (Blackburn et Kunreuther 1974). Le problème est résolu à l'aide d'une programmation dynamique. (Shaw et Wagelmans 1998) ont étudié le problème USILS avec une fonction linéaire par morceaux pour les coûts de production et de backlog. (Sandbothe et Thompson 1990) ont étudié le problème avec rupture sur la demande et contraintes de capacité. Ils ont proposé un algorithme en  $O(T^3)$ . (Aksen et al. 2003) ont également proposé une programmation dynamique pour résoudre le problème de lot sizing avec rupture sur la demande sans contraintes de capacité. Dans leur cas, la demande peut être perdue mais ne peut pas être retardée. Ils ont ensuite proposé une programmation dynamique en  $O(T^2)$ . (Liu et Tu 2008) ont traité le problème avec rupture sur la demande, ils ont également proposé une programmation dynamique pour résoudre le problème en  $O(T^2)$ . (Federgruen et Tzur 1993) proposent une heuristique afin de résoudre le problème étudié par (Federgruen et Tzur 1991) avec backlog en  $O(T \log T)$ .

### 4.3.3 Problèmes à un niveau et plusieurs produits

Dans le problème de lot sizing à plusieurs produits avec contraintes de capacité, les techniques de résolution les plus souvent utilisées sont les méthodes basées sur la relaxation lagrangienne, les méthodes de Branch and Cut, les heuristiques ainsi que les méta-heuristiques. Dans la suite de ce travail, nous allons utiliser la notation CMILS (Capacitated Multi-Items Lot Sizing) pour représenter ce problème.

Pour résoudre le problème CMILS, la plupart des auteurs utilisent des heuristiques basées sur une relaxation lagrangienne. (Manne 1958) a été parmi les premiers à proposer une modélisation du problème CMILS. L'objectif de modèle étant de déterminer un plan de production d'un ensemble de produits sur un horizon de planification constitué de plusieurs périodes. (Absi et Kedad-Sidhoum 2008) ont traité le problème avec temps de setup et coût de rupture sur la demande, ils ont résolu le problème à l'aide d'un Branch and Cut. (Absi et Kedad-Sidhoum 2006) ont étudié un problème CMILS avec temps de setup, stock de sécurité et rupture sur la demande. (Chen et Thizy 1990) ont traité le problème de lot sizing à plusieurs produits avec contraintes de capacité. Ils ont proposé plusieurs approches de relaxation lagrangienne ainsi qu'une formulation basée sur le problème de plus court chemin. (Buschkühl et al. 2010) ont également proposé un état de l'art sur le problème de lot sizing avec contraintes de capacité, ils ont présenté plusieurs problèmes de lot sizing et différentes techniques de résolution. (González-Ramírez et al. 2011) proposent une approche heuristique afin de résoudre le problème de lot sizing à plusieurs produits. (Trigeiro et al. 1989) et (Van Vyve et Wolsey 2014) ont proposé une relaxation lagrangienne sur la contrainte de capacité pour obtenir une borne inférieure. (Suerie et Stadtler 2003) proposent une heuristique basée sur la décomposition du temps pour résoudre le problème avec contraintes de capacité.

Les problèmes de lot sizing avec ventes perdues sont également considérés comme des problèmes de lot sizing à plusieurs produits avec contraintes de capacité. Dans ce contexte, nous citons le travail de (Brahimi et Dauzère-Pérès 2014) qui proposent une heuristique lagrangienne pour résoudre le problème CMILS. Ils ont montré ensuite que l'heuristique proposée peut être utilisée pour résoudre les problèmes de planification de production globale. (Chu et Chu 2007) ont également proposé une relaxation lagrangienne, en relaxant les contraintes de capacité, nous tombons sur des problèmes de type USILS faciles à résoudre. (Robinson et al. 2009) ont proposé une heuristique de relaxation pour résoudre le problème de lot sizing. La complexité mathématique du problème traité est NP-difficile, ensuite ils ont montré que le problème peut être résolu à l'aide d'un algorithme en temps polynomiale. (Brahimi et al. 2010) ont proposé une relaxation lagrangienne en relaxant la contrainte de capacité. (Absi et al. 2013) ont également proposé une relaxation lagrangienne pour résoudre le problème avec ventes perdues. Ils ont relaxé la contrainte de capacité et ensuite le problème est résolu en  $O(T^2)$ . Dans (Sambasivan et Yahya 2005) et (Absi et Kedad-sidhoum 2007), les auteurs traitent le même problème que celui de (Absi et al. 2013) mais sans vente perdues, la contrainte de capacité a été également relaxée.

D'autres auteurs proposent des méta-heuristiques afin de résoudre le problème CMILS. Parmi les méta-heuristiques proposées, nous trouvons souvent l'algorithme génétique et le recuit simulé. (Dellaert et al. 2000) et (Wu et al. 2011) ont proposé un algorithme génétique, ils ont utilisé un codage sur la variable de setup afin de concevoir un algorithme efficace. Une brève présentation de recuit simulé appliqué au problème à plusieurs produits a été proposé par (Tang 2004). (Kianfar et Mokhtari 2009) ont également proposé un algorithme génétique pour résoudre le problème avec des délais de livraison stochastiques. Ces derniers suivent une loi de distribution discrète. (Berretta et Fernando 2004) ont proposé un algorithme mémétique afin de déterminer la quantité à produire dans chaque période de l'horizon de planification. Ils ont présenté différentes méta-heuristiques telles que l'algorithme génétique, le recuit simulé et la recherche tabou. Finalement, ils ont proposé une heuristique de relaxation lagrangienne.

#### **4.3.4 Problèmes à plusieurs niveaux**

Dans le problème à plusieurs niveaux, les produits sont fabriqués à partir des éléments intermédiaires. Ce type de problème nécessite une nomenclature pour chacun des produits fabriqués, qui permet de prendre en compte les différentes structures possibles de composition des produits. Ce problème peut être considéré avec ou sans capacité. Dans la suite de ce travail, nous allons utiliser la notation MLLSP (Multi-Levels Lot Sizing Problem) pour représenter ce problème

Pour résoudre le problème MLLSP, l'une des méthodes approchées les plus efficaces et les plus souvent utilisée est celle basée sur la relaxation lagrangienne. Très peu d'auteurs proposent des techniques de résolution basées sur la décomposition de niveaux (résolution niveau par niveau) (Coleman et McKnew 1991), ou période par période (Afentakis 1987). (Afentakis et Gavish 1986) ont proposé une relaxation lagrangienne engendrant des sous-problèmes constitués de problèmes multi-niveaux avec structure en série. (Tempelmeier et Derstroff 1996) ont proposé une heuristique lagrangienne où les contraintes de capacité et les équations d'équilibrage de flux ont été relaxées. (Özdamar et Barbarosoglu 2000) ont proposé une méthode en combinant la relaxation lagrangienne et le recuit simulé. À chaque itération de la relaxation lagrangienne, le recuit simulé améliore la solution du problème relaxé.

D'autres auteurs utilisent des heuristiques pour résoudre le problème MLLSP. Dans ce contexte, nous citons le travail de (Harrison et Lewis 1996) qui ont proposé une heuristique d'approximation des



variables binaires de lancement de la production. (Cattrysse et *al.* 1990) et (Jans et Degraeve 2004) ont proposé une heuristique basée sur la génération de colonnes. (Hindi 1995) ont proposé une approche hybride combinant la relaxation lagrangienne, la génération de colonnes, le flot à coût minimum et le recuit simulé. (Maes et *al.* 1991) ont proposé une heuristique basée sur la programmation linéaire pour résoudre le problème de lot sizing à capacité finie et plusieurs niveaux de production. Les auteurs ont utilisé la formulation basée sur le problème de localisation de (Krarup et Bilde 1977).

Nous venons de proposer une étude bibliographique non exhaustive sur les travaux de littérature de problèmes de lot sizing en production. Nous avons constaté que la plupart des états de l'art proposés dans la littérature ne sont pas exhaustifs. Il existe toujours de nouveaux travaux qui n'ont pas encore été traités. Pour que cette étude soit exhaustive, nous allons analyser des nouveaux articles qui ne sont pas encore été analysés ou qu'ils viennent d'être publiés.

#### **4.3.5 Nouvelles approches**

(Helber et *al.* 2010) et (Levi et Shi 2013) ont proposé deux modèles de programmation linéaire en nombres entiers qui sont résolus à l'aide d'une heuristique Fix-and-Optimize. Ils ont ensuite comparé l'heuristique avec CPLEX. Les résultats montrent que sur 1296 instances, CPLEX ne trouve aucune solution dans une heure de calcul, alors que l'heuristique permet de trouver la solution dans moins de 400 secondes. Le problème de lot sizing avec des demandes stochastiques est difficile à résoudre. Afin de traiter ce type de problème, (Brandimarte 2006) a également utilisé une heuristique de type Fix-and-Relax proposée par (Dillenberger 1994). (Goren et *al.* 2012) ont proposé un algorithme génétique combiné avec une heuristique Fix-and-Optimize. (Chen 2015) a également proposé une heuristique Fix-and-Optimize pour résoudre le problème à plusieurs produits et contraintes de capacité. Il a généré une solution de départ avec un coût de setup à chaque période, il a appliqué ensuite une recherche locale pour minimiser le coût de setup et ensuite minimiser le coût total. (Tempelmeier et Hilger 2015) a traité le problème à plusieurs produits avec incertitude sur le délai d'approvisionnement. Ils ont également proposé une heuristique Fix-and-Optimize. (Toledo et *al.* 2015) ont utilisé une heuristique Fix-and-Relax pour construire une solution initiale. Ils ont ensuite proposé une Fix-and-Optimize pour améliorer la solution trouvée. L'heuristique Fix-and-Relax commence par fixer une fenêtre de décision au début de l'horizon de planification et une fenêtre d'approximation sur le reste de l'horizon, et ensuite toutes les contraintes de problème sont relaxées.

(De Reyck 2015) ont proposé un algorithme basé sur la génération de colonnes et une relaxation lagrangienne pour résoudre le problème avec contraintes de capacité. (Akbalik et *al.* 2015) ont traité le problème à plusieurs produits sans contraintes de capacité avec une limitation sur le stock. Ils supposent qu'à chaque période la quantité qui peut être stockée est limitée. Pour résoudre ce problème, ils ont proposé une programmation dynamique. Un problème à deux niveaux à plusieurs produits est étudié par (Verma et Sharma 2015). Ils ont proposé dans un premier temps une relaxation lagrangienne, puis dans un deuxième temps, ils ont proposé un Branch and Bound. (Karimi-Nasab et Modarres 2015) et (Gicquel et Minoux 2015) ont également proposé une nouvelle famille d'inégalités valides permettant de représenter l'aspect multi-produits dans la formulation du problème. Ensuite, ils ont proposé un Branch and Bound pour accélérer le processus de recherche par solveurs communs. Un algorithme d'énumération exhaustive et une heuristique ont été récemment proposés par (Cárdenas-Barrón et *al.* 2015).

(Xiao et *al.* 2015) ont proposé une méta-heuristique telle que le recuit simulé. Ensuite, ils ont analysé différents types de voisinages basés sur la relaxation de différentes contraintes. Les contraintes relaxées sont la contrainte de capacité et celle d'équilibre des flux. (Wolosewicz et *al.* 2015) ont

proposé une heuristique de relaxation lagrangienne pour calculer la borne inférieure du problème. (Mazdeh et *al.* 2015) ont proposé un algorithme basé sur la technique de «Fordyce–Webster». Cet algorithme permet de résoudre des problèmes avec des paramètres de certitude.

Nous venons d'analyser les différents problèmes de lot sizing dans le domaine de la production. Nous avons présenté les différents modèles et techniques de résolution. Nous allons maintenant vérifier si certaines méthodes peuvent être utilisées pour résoudre le problème de lot sizing en désassemblage. Afin de vérifier si ces méthodes peuvent être utilisées dans la résolution du problème de lot sizing en désassemblage, nous allons définir les principales différences entre ce dernier et celui de lot sizing en production.

#### **4.3.6 Différences entre le problème de lot sizing en production et en désassemblage**

Suite aux différences proposées par (Duta 2006). Nous pouvons constater que la différence la plus évidente entre le problème de désassemblage et d'assemblage consiste dans le processus de flux. En effet, le processus de désassemblage est divergent c'est-à-dire un produit en fin de vie est composé de plusieurs composants (les composants divergents, c'est-à-dire que l'on peut satisfaire plusieurs types de demandes à partir d'une même source) (Kim et *al.* 2007). Quant au processus d'assemblage est convergent c'est-à-dire les composants convergent à une source unique des demandes du produit final. De plus, dans le problème de désassemblage, la quantité et la qualité de produits retournés ne sont pas connus. Le problème de l'assemblage est forcément complet alors que le désassemblage est principalement partiel (il arrive parfois qu'un composant ne peut pas être séparé).

Il existe ainsi une forte incertitude sur les structures et la qualité de produits en fin de vie dans un problème de désassemblage. Contrairement à celui d'assemblage, les composants du produit sont connus et contrôlés avant qu'ils soient assemblés. Contrairement au désassemblage, la qualité des produits retournés n'est pas souvent connue (il arrive parfois que les produits retournés sont défectueux). Finalement, la différence qui nous semble pertinente par rapport à notre travail est la suivante : dans l'assemblage, la demande est située au niveau d'un produit fini, alors que dans le désassemblage cette dernière est située à la fois au niveau des composants, éléments intérimaires ou encore le produit entier. De plus, le coût de stockage implique seulement les composants dans un problème de désassemblage, et le stockage des produits en fin de vie n'est pas pris en compte. Les différences entre le problème de désassemblage et d'assemblage sont présentées dans le tableau 5 comme il est donné par (Brennan et *al.* 1994) et (Güngör et Gupta 1999) :

**Tableau 5 :** Comparaison entre l'assemblage et le désassemblage (Brennan *et al.* 1994) et (Gungor et Gupta 1999)

<b>Caractéristique de système</b>	<b>Assemblage</b>	<b>Désassemblage</b>
Type de la demande	Unique	Multiple
Nécessité de prévision	Entité unique	Entité multiple
Horizon de planification	Cycle de vie du produit	Non défini
Orientation de la conception	Conception pour l'assemblage	Conception pour le désassemblage
Planification des installations et des capacités	Simple	Complexe
Complexité des opérations	Moyenne	Élevée
Flux des produits	Convergent	Divergent
Flux des matières	Direct	Inverse
Gestions des stocks des sous-ensembles	Aucun	Potentielle
Existence d'outils de planification	Plusieurs	Aucun

Suite à ces différences, les méthodes qui peuvent être appliquées ou qui ont déjà été appliquées aux problèmes de lot sizing en désassemblage sont des méta-heuristiques notamment l'algorithme génétique et le recuit simulé. Des heuristiques de relaxation lagrangienne ainsi que des méthodes exactes notamment la programmation dynamique (Kim 2005) et l'approche Branch and Bound proposée par (Kim *et al.* 2009). D'autres heuristiques notamment Silver Meal, Part Period Balancing, Fix-and-Optimize peuvent également être utilisées dans la résolution des problèmes de lot sizing en désassemblage. En effet, quel que soit le problème de lot sizing en désassemblage, les techniques d'optimisation utilisées dans le lot sizing en production ne peuvent pas être appliquées directement, mais elles nécessitent des adaptations spécifiques pour résoudre ce type de problèmes.

Les problèmes de lot sizing en production peuvent être divisés en problèmes à deux niveaux ou plusieurs niveaux, à un seul ou plusieurs produits avec ou sans contraintes de capacité. Ces derniers sont résumés par (Merzouk 2007) dans la Figure 16 :

D'après notre étude bibliographique, nous avons constaté que la programmation dynamique a été largement utilisée pour résoudre les différents problèmes de lot sizing en production à un seul produit. Nous avons vu que les heuristiques basées sur la relaxation lagrangienne sont très efficaces en termes de qualité de solution et en temps d'exécution pour les deux problèmes de lot sizing en production et en désassemblage. En effet, ces dernières sont connues pour leurs efficacités à obtenir de borne inférieure et pour leur simplicité à construire des solutions réalisables de bonne qualité dans un temps de résolution raisonnable. Des méta-heuristiques ont été également efficaces dans le cas d'une structure du désassemblage à plusieurs niveaux et plusieurs produits. Elles donnent un choix énorme en termes de solution et en temps de calcul. Outre que, des heuristiques telles que « Fix-and-Optimize » ont été également connues par leurs importances suite à leurs efficacités de résolution de modèles d'optimisation. Finalement, nous avons constaté qu'aucun auteur n'est intéressé à la problématique de lot sizing en désassemblage avec accumulation de stocks et ventes perdues. Seulement (Ji *et al.* 2015) qui ont proposé un modèle prenant en compte le coût d'achat qui est connu comme les ventes perdues de point de vue du modèle. Pour ces raisons, il nous a semblé intéressant d'autoriser les ventes perdues dans le problème de lot sizing en désassemblage afin de limiter l'accumulation de stocks.

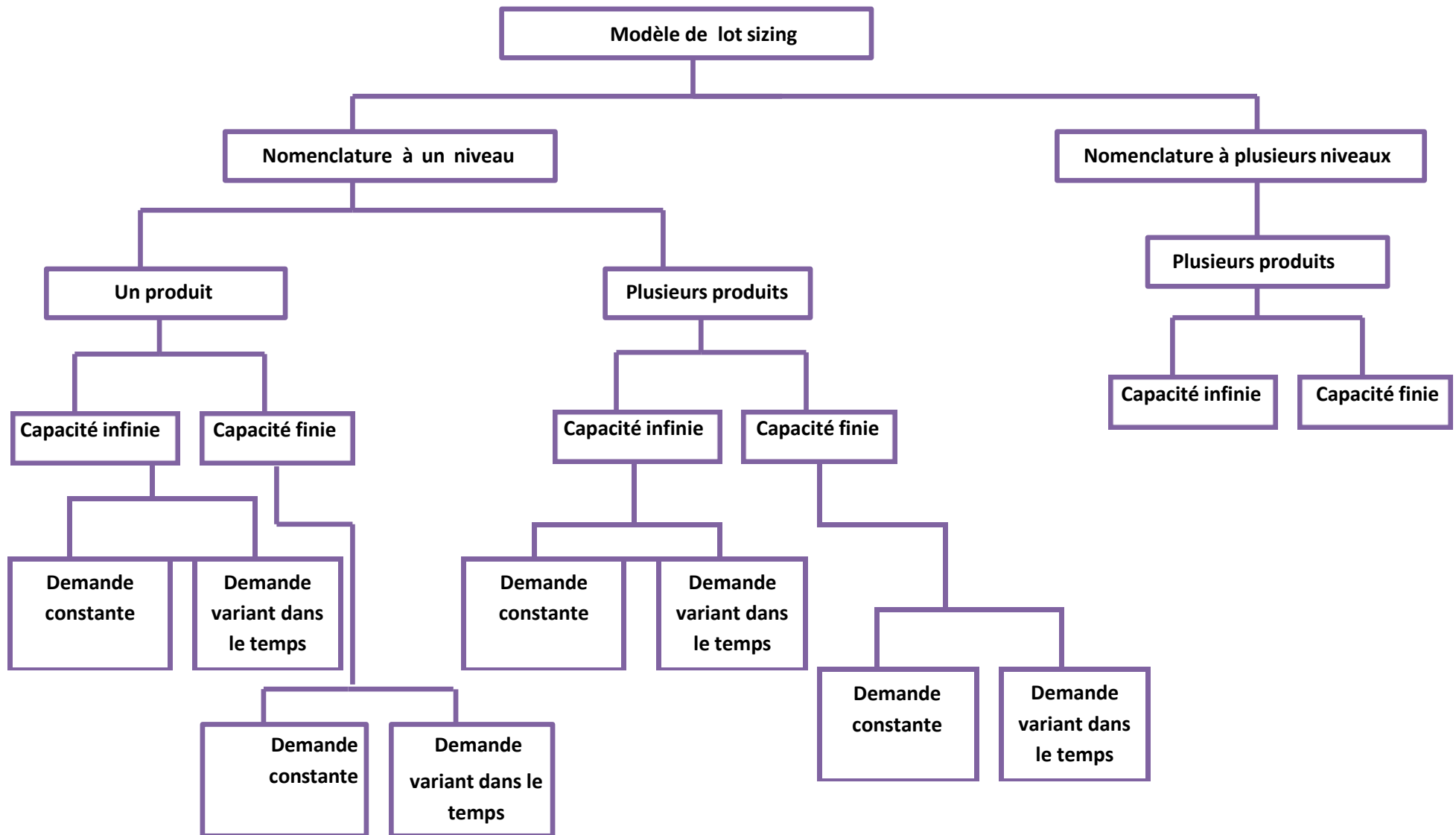


Figure 16 : Classification des problèmes de lot sizing (Merzouk 2007)

## Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur l'optimisation de la logistique inverse et la planification du désassemblage. Dans un premier temps, nous avons montré l'importance de la logistique inverse ainsi que ses principales définitions présentées dans la littérature. Nous avons ensuite regroupé les définitions en proposant notre propre définition de la logistique inverse. Nous avons également montré l'importance de désassemblage dans la logistique inverse. Dans un deuxième temps, nous nous sommes intéressés à la problématique de planification du désassemblage. Nous avons analysé différents modèles et méthodes de résolution, ensuite nous avons proposé une nouvelle classification des problèmes. Dans un troisième temps, nous avons proposé un état de l'art sur le problème de lot sizing en production en analysant les modèles et leurs techniques de résolution proposées. Ensuite, nous avons sélectionné certaines méthodes qui peuvent être utilisées dans la résolution du problème de lot sizing en désassemblage.

Après notre analyse des travaux de la littérature, nous avons constaté qu'aucun chercheur n'est intéressé au problème de lot sizing en désassemblage avec ventes perdues. Cette problématique est inhérente à ce problème puisque nous avons une forte incertitude sur le produit retourné. En effet, il arrive souvent que les produits désassemblés soient défectueux ou ne sont pas disponibles pour la collecte. Dans ces circonstances les demandes peuvent ne pas être satisfaites et l'autorisation de ventes perdues devient indispensable. C'est pourquoi nous nous sommes intéressés au problème de lot sizing en désassemblage avec ventes perdues.

Dans la suite de cette thèse, nous étudierons plusieurs problèmes de lot sizing en désassemblage à deux niveaux, à un seul et plusieurs produits avec et sans ventes perdues et contraintes de capacité. Nous proposons ensuite des approches heuristiques et méta-heuristiques pour résoudre ces problèmes. Ce type d'approches a été déjà utilisé dans la résolution du problème de lot sizing en production. Dans le cas d'un seul produit, nous utilisons des heuristiques largement connues et utilisées dans le problème de lot sizing en production « Silver Meal, Least Unit Cost et Part Period Balancing ». Dans le cas du problème avec plusieurs produits et contraintes de capacité, nous allons proposer dans un premier temps un algorithme génétique, ensuite pour améliorer cet algorithme, nous intégrons une heuristiques Fix-and-Optimize dans ce dernier en proposant de la sorte une approche hybride. Finalement, plusieurs tests seront effectués pour tester les performances des approches proposées.

## Chapitre 2 : Problème de lot sizing en désassemblage à un seul produit

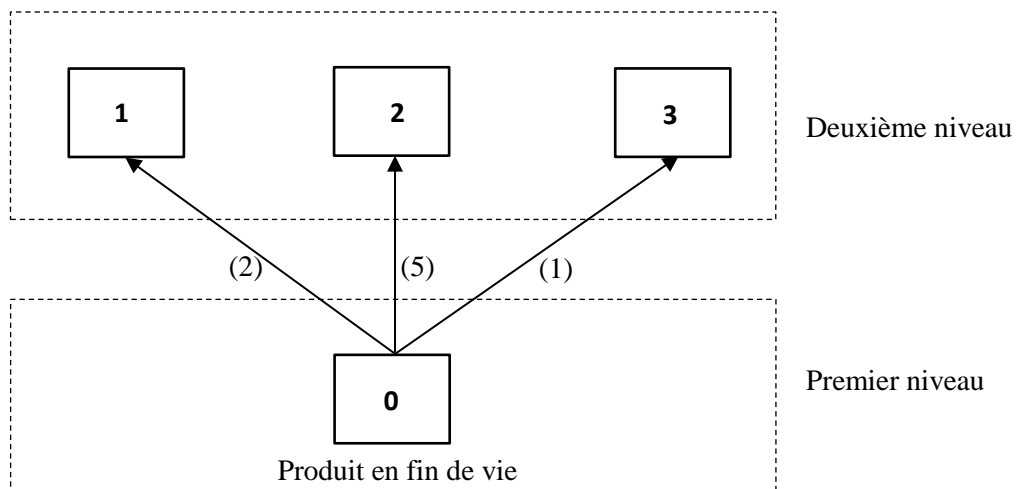
Dans ce chapitre, nous considérons le problème de planification du désassemblage à un seul produit et deux niveaux. En effet, dans une structure de produit à deux niveaux, il n'y a pas d'éléments intermédiaires et l'opération de désassemblage permet d'obtenir directement les composants désassemblés. Le problème peut être formulé comme un modèle de programmation linéaire en nombres entiers avec un horizon décomposé en périodes discrètes. Le but est de déterminer la quantité de produits en fin de vie à désassembler sur chaque période afin de satisfaire les demandes des différents composants désassemblés.

Deux problèmes sont traités dans ce chapitre. Dans le premier problème, toutes les demandes doivent être satisfaites (aucune rupture ou retard sur la demande n'est accepté). L'objectif est de minimiser la somme des coûts fixes de désassemblage et de stockage des composants. Le modèle est analysé afin de calculer le stock de surplus à la fin de l'horizon de planification pour chaque composant (le surplus est la quantité accumulée à la fin de l'horizon qui ne sert à satisfaire aucune demande) qui peut être calculé avant la résolution du modèle. Pour la résolution de ce problème, nous adaptons trois heuristiques connues pour leurs performances dans le lot sizing en production « *Silver Meal, Part Period Balancing et Least Unit Cost* ». Dans le deuxième problème, les ventes perdues sont autorisées et les demandes ne sont pas forcément satisfaites. La demande non satisfaite entraîne un coût correspondant à la perte appelé «coût de ventes perdues». L'objectif est de minimiser la somme des coûts de setup, de stockage et de ventes perdues. L'intérêt est de perdre la demande si les coûts sont importants (par exemple le coût de setup est important par rapport au coût de stockage) et de limiter le stock de surplus restant à la fin de l'horizon de planification. Pour résoudre ce problème, nous utilisons les mêmes heuristiques précédentes en les adaptant de manière à prendre en compte les ruptures. Finalement, pour tester les performances des heuristiques adaptées, nous générons des instances sur différents profils de demandes telles que demandes variables, demandes croissantes et demandes décroissantes. En effet, dans le désassemblage il arrive que pour certains produits en fin de vie, nous avons des demandes importantes ou moins importantes sur certaines saisons de l'année (par exemple indisponibilité ou excès de produits en fin de vie), cela dépend également de type du produit en fin de vie à collecter.

# 1 Problème de lot sizing en désassemblage sans ventes perdues

Dans cette étude, nous considérons des produits à désassembler structurés en deux niveaux. Le premier niveau représente le produit qui doit être désassemblé, et le deuxième niveau représente les composants obtenus directement par l'opération de désassemblage de ce produit. Un exemple de produit avec trois composants est donné dans la Figure 17. Les demandes se situent au niveau des composants 1, 2 et 3 et les quantités qui peuvent être obtenues à partir du produit en fin de vie sont indiquées sur les arcs. D'après la Figure 17, le composant 0 représente le produit en fin de vie, et les composants 1, 2 et 3 représentent les composants à désassembler. Les nombres indiqués sur les arcs représentent les rendements de désassemblage ou (Yields en anglais) qui sont relatifs à chaque composant. D'après la figure, le désassemblage de 0 va générer 2 composants de type 1, 5 composants de type 2 et un seul composant de type 3.

Dans la suite de ce problème, l'horizon de planification est décomposé en  $T$  périodes et un produit est constitué de  $N$  composants pour lesquels des demandes sont prévues sur chaque période. L'objectif est de minimiser la somme de coûts fixes de désassemblage (aussi appelés setup) et de stockage. Le coût fixe de désassemblage est relatif à la préparation de l'opération de désassemblage ou à la commande des produits à désassembler. Ce coût est pris en compte sur une période si une opération de désassemblage d'un produit est effectuée sur cette période, sinon il est égal à 0. Les coûts de stockage sont proportionnels aux composants stockés pour satisfaire la demande future sur l'horizon de planification.



**Figure 17 :** Structure du désassemblage à deux niveaux

## 1.1 Modélisation du problème SIDLS

Le problème d'optimisation peut être formulé sous forme d'un modèle de programmation linéaire en nombres entiers. Les notations ci-dessous seront utilisées dans tout le reste de ce chapitre.

### Indices

$i$  : indices pour les composants,  $i = 1, \dots, N$

$t$  : indices pour les périodes,  $t = 1, \dots, T$

### Paramètres

$a_i$  : nombre de composants à désassembler à partir du produit en fin de vie appelé aussi rendement de désassemblage ou (Yield en anglais)

$d_{it}$  : demande pour l'élément  $i$  à la période  $t$

$D_i = \sum_{t=1}^T d_{i,t}$  : demande cumulée du composant  $i$  sur tout l'horizon de planification  $T$

$h_i$  : coût de stockage du composant  $i$

$I_{i0}$  : niveau de stock initial du composant  $i$

$s_t$  : coût de setup à la période  $t$  (correspond au produit en fin de vie)

$M$  : nombre entier assez grand

### Variables de décisions

$I_{it}$  : niveau de stock du composant  $i$  à la fin de la période  $t$

$X_t$  : quantité de produit à désassembler à la période  $t$

$Y_t$  : variable binaire égale à 1 s'il y a un désassemblage à la période  $t$ , et 0 sinon

En utilisant ces variables et paramètres, nous formulons le problème d'optimisation comme suit :

$$\text{Min} \sum_{t=1}^T s_t \cdot Y_t + \sum_{t=1}^T \sum_{i=1}^N h_{it} \cdot I_{it}$$

Sous contraintes :

$$I_{i,t} = I_{i,t-1} + a_i \cdot X_t - d_{it} \text{ pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (2.1)$$

$$X_t \leq M \cdot Y_t \text{ pour } t = 1, \dots, T \quad (2.2)$$

$$I_{it}, X_t \geq 0 \text{ et } X_t, I_{it} \in \mathbb{N} \text{ pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (2.3)$$

$$Y_t \in \{0,1\} \text{ pour } t = 1, \dots, T \quad (2.4)$$

La fonction objectif minimise le coût total encouru sur l'horizon de planification. Ce coût est composé des coûts fixes ou (de setup) et de stockage. La contrainte (2.1) représente la conservation des flux de stockage pour les composants. Le niveau de stock d'un composant à la fin de la période  $t$



est égal à la somme de niveau du stock à la période  $t - 1$  plus les entrées moins les sorties à la fin de la période  $t$ . Les entrées sont la quantité générée par le désassemblage et les sorties sont les demandes. La contrainte (2.2) assure le lancement de désassemblage où la quantité produite ne doit pas dépasser le paramètre  $M$  ( $M$  est une valeur assez grande) qui peut représenter la somme maximale des demandes d'un composant sur l'horizon de planification. La contrainte (2.3) implique que les variables discrètes (la quantité de désassemblage et le niveau de stock) sont discrètes non négatives. La dernière contrainte (2.4) représente une variable binaire égale à 1 si on désassemble à la période  $t$  et 0 sinon.

L'une des principales différences entre les modèles de planification du désassemblage et de production/assemblage est relative aux flux qui sont divergent (un produit est décomposé en plusieurs composants et/ou sous-ensembles). Dans les problèmes de désassemblage le flux part d'un produit et divergent vers chaque composant, contrairement à l'assemblage qui est convergent où plusieurs composants convergent vers un seul produit. Par opposition aux problèmes d'assemblage, les problèmes de désassemblage sont divergents c'est-à-dire que l'on peut satisfaire plusieurs types de demandes à partir d'une même source (un produit par exemple). Un stock non nul peut ainsi exister à la fin de l'horizon pour certains composants qui ne sert à satisfaire aucune demande, il est appelé «stock de surplus». Ce surplus s'accumule tout au long de l'horizon de planification générant des coûts de stockage supplémentaires.

## 1.2 Problématique d'accumulation de stocks

En se basant sur les modèles et méthodes de la littérature, nous constatons que l'on doit faire des surplus de stocks en raison des écarts de demandes entre les composants ainsi que les rendements de désassemblage peuvent être aussi différents d'un composant à un autre. Ces surplus de stockage peuvent augmenter pour certains composants ce qui peut devenir un problème au niveau opérationnel car il ne sert à satisfaire aucune demande. Ce stock de surplus à la fin de chaque période est identique quelle que soit la solution de planification étant donné la contrainte de satisfaction des demandes de tous les composants.

Dans un problème de lot sizing en désassemblage à un seul produit et deux niveaux, par le désassemblage d'un produit nous pouvons satisfaire l'ensemble des demandes des composants de ce dernier. Si nous supposons que le niveau de stock initial de tous les composants est nul ( $I_{i0} = 0$ ) pour faciliter le raisonnement (il faut enlever le stock initial de la demande pour se ramener à ce problème). Le stock de surplus peut être calculé à l'aide de la formule suivante :

$$I_{i,T} = \left( a_i \cdot \frac{D_{i^*}}{a_{i^*}} \right) - D_i \text{ pour } i = 1, \dots, N \text{ avec } i^* = \operatorname{argmax}_{i=1, \dots, N} \left[ \frac{D_i}{a_i} \right] \quad (2.5)$$

La quantité de produit à désassembler pour satisfaire les demandes  $d_{it}$  est calculée à partir de l'équation de conservation des flux (2.6) :

$$a_i \cdot X_t = I_{it} - I_{i,t-1} + d_{it} \text{ pour } i = 1, \dots, T \text{ et } t = 1, \dots, T \quad (2.6)$$

D'après l'équation (2.6), le stock à la fin de l'horizon de planification est calculé comme suit :

$$I_{i,T} = a_i \sum_{t=1}^T \left( X_t - \frac{\sum_{t=1}^T d_{it}}{a_i} \right) \text{ pour } i = 1, \dots, N \quad (2.7)$$

$$I_{i,T} = a_i \cdot \sum_{t=1}^T X_t - D_i \text{ pour } i = 1, \dots, N \quad (2.8)$$

Comme nous l'avons déjà défini, dans le problème de lot sizing en désassemblage à un seul produit et deux niveaux, nous cherchons à satisfaire la somme de la demande maximale parmi tous les composants. La somme de la quantité de désassemblage doit couvrir toutes les demandes sur l'horizon de planification. Dans le cas de plusieurs composants, la somme de la quantité de désassemblage est calculée par la formule suivante :

$$\sum_{t=1}^T X_t = \max_{i=1, \dots, N} \left\lceil \frac{\sum_{t=1}^T d_{it} - I_{i0}}{a_i} \right\rceil \quad (2.9)$$

Où  $\lceil \bullet \rceil$  est la plus petite valeur entière supérieure ou égale à  $\bullet$ .

Nous donnons ensuite un exemple avec la même structure du désassemblage de la Figure 17. L'horizon de planification est composé de 10 périodes. Les demandes sont déterministes et elles sont toutes satisfaites. Le stock initial est supposé nul à chaque période. Les rendements de désassemblage sont  $a_1 = 2$ ,  $a_2 = 5$  et  $a_3 = 1$  pour les composants 1, 2 et 3 respectivement. Le coût de stockage est supposé constant pour chaque composant et à chaque période. Les ratios de désassemblage sont :  $\frac{D_1}{a_1} = 327$ ,  $\frac{D_2}{a_2} = 110$  et  $\frac{D_3}{a_3} = 716$  avec les  $\frac{D_i}{a_i}$  (les ratios sont des arrondis supérieurs pour permettre de satisfaire les demandes). Le coût de setup est supposé constant à chaque période  $s = 1000$  et le coût de stockage est supposé fixe pour tous les composants  $h_i = 1$ . Les données de l'exemple sont résumées dans le tableau 6.

**Tableau 6 :** Données de l'exemple

	Périodes												
	1	2	3	4	5	6	7	8	9	10	$\sum_{t=1}^T d_{1t}$	$\sum_{t=1}^T d_{2t}$	$\sum_{t=1}^T d_{3t}$
$d_{1t}$	48	92	43	69	48	74	48	74	91	66			
$d_{2t}$	51	72	55	78	40	37	40	34	95	46	653	548	716
$d_{3t}$	72	36	87	94	84	77	84	77	44	61			

Par la résolution de l'exemple, nous constatons qu'il y a un stock de surplus de 779 et 3032 pour les composants 1 et 2 respectivement.

Nous allons maintenant calculer ce stock de surplus à la fin de l'horizon de planification en utilisant la méthode analytique. Nous exposons ci-dessous la démarche pour calculer le surplus.

Pour  $i = 1, 2, 3$  ;  $I_{1,0} = I_{2,0} = I_{3,0} = 0$  et  $T = 10$

Calcul de la valeur maximale du rendement de désassemblage :

$$\max \left[ \frac{D_1}{a_1}, \frac{D_2}{a_2}, \frac{D_3}{a_3} \right] = \max \left[ \frac{653}{2}, \frac{548}{5}, \frac{716}{1} \right] = 716$$

Calcul de la somme de la quantité de désassemblage sur tout l'horizon de planification :

$$\sum_{t=1}^T X_t = \max \left[ \frac{D_i}{a_i} \right] = 716$$

$$i^* = \operatorname{argmax} \left\lfloor \frac{D_i}{a_i} \right\rfloor = 3$$

Ici, c'est le composant 3 qui a un niveau de surplus nul ( $I_{3T} = 0$ )

Les niveaux de surplus sont calculés à l'aide de la formule suivante :

$$I_{i,T} = \left( a_i \cdot \frac{D_3}{a_3} \right) - D_i$$

Donc les niveaux de stocks pour les composants 1, 2 et 3 sont :

$$I_{1,T} = \left( a_1 \cdot \frac{D_3}{a_3} \right) - D_1 = (2.716) - 653 = 779$$

$$I_{2,T} = \left( a_2 \cdot \frac{D_3}{a_3} \right) - D_2 = (5.716) - 548 = 3032$$

$$I_{3,T} = \left( a_3 \cdot \frac{D_3}{a_3} \right) - D_3 = (1.716) - 716 = 0$$

Dans cette section, nous avons montré que dans le problème de lot sizing en désassemblage, il existe un stock de surplus pour certains composants qui ne sert à satisfaire aucune demande. Ce stock de surplus entraîne un coût de stockage important et par conséquent le coût total devient également important. Pour minimiser ce coût total, il faut proposer des solutions pour la gestion de ce surplus. Nous avons fait également l'hypothèse que les retours de produits en fin de vie sont supérieurs aux demandes et par conséquent les quantités de ces produits ne sont pas limitées. Alors que dans la réalité, les produits ne sont pas toujours une variable de décision dans le problème de planification du désassemblage.

### 1.3 Méthodes de résolution

Il est possible de résoudre le problème ci-dessus de façon exacte à l'aide d'un logiciel de programmation linéaire en nombres entiers. (Kim 2005) a proposé un algorithme en  $O(N.T^2)$  pour résoudre de manière exacte le même problème où  $T$  est le nombre de périodes et  $N$  le nombre de composants. La programmation dynamique s'applique lorsque la propriété  $I_{i,t-1}.X_t = 0$  est vérifiée (c'est le cas pour les solutions optimales lorsque le plan de désassemblage n'est pas remis en cause entre la première période et la dernière). Cette propriété signifie que l'on ne produit jamais à une période ayant un stock non nul provenant de la période précédente. Afin d'adapter une méthode de résolution efficace aux problèmes étudiés, nous comparons entre les méthodes exactes et les heuristiques.

#### 1.3.1 Méthodes exactes

(Gicquel et al. 2008) ont proposé un état de l'art et ils ont montré que plusieurs auteurs ont utilisé des méthodes exactes pour résoudre le problème de lot sizing en production. Parmi les méthodes exactes souvent utilisées pour résoudre le problème à un seul produit et un niveau, nous trouvons celle de la programmation dynamique proposée par (Wagner et Whitin 1958).

Les travaux de (Kim 2005) permettent d'utiliser une approche basée sur l'algorithme de Wagner et Whitin pour résoudre le problème de lot sizing en désassemblage à un seul produit et deux niveaux. La complexité de son algorithme est  $O(N.T^2)$ .

### 1.3.2 Heuristiques

Pour résoudre ce problème, nous adaptons des heuristiques existantes et largement connues et utilisées en pratique pour résoudre le problème de lot sizing en production. Elles sont construites de manière à déterminer les tailles de lots en équilibrant entre les coûts de setup et de stockage. Ces dernières sont des approches « Myopes » qui sont souvent utilisées dans la pratique. Plusieurs travaux dans la littérature ont montré les performances des heuristiques Silver Meal, Least Unit Cost et Part Period Balancing dans un problème de lot sizing en production. Dans (Kropp et al. 1983), les auteurs ont étudié le problème de lot sizing à un seul niveau et un seul produit et ils ont montré que les heuristiques Silver Meal et Part Period Balancing ont été très performantes sur toutes les instances. (Jeunet et Jonard 2000) ont également testé les performances de Silver Meal et Least Unit Cost sur le problème de lot sizing à un seul niveau et un seul produit, ils ont montré que les heuristiques ne dépassent pas 10% par rapport à la solution optimale sur des demandes variables. (Teunter et al. 2006) et (Schulz 2011) ont considéré le cas des retours de produits, ils ont ensuite testé les trois heuristiques sur différents profils de demandes (demandes stationnaires, demandes saisonnières, demandes croissantes et demandes décroissantes). Leurs résultats ont montré que Part Period Balancing n'a pas été performante par rapport à Silver Meal et Least Unit Cost. Finalement, (Toy et Berk 2013) ont testé les performances des trois heuristiques pour la résolution du problème de lot sizing en production à un seul niveau et un seul produit, et ils ont montré que les heuristiques donnent des solutions optimales sur des petites instances ainsi que sur des horizons glissants. L'objectif de notre étude est de tester les performances de ces heuristiques dans la résolution du problème de lot sizing en désassemblage en apportant les modifications nécessaires.

#### a. Silver Meal (SM)

Le principe de l'heuristique de Silver Meal (Silver et Meal 1973) cherche à minimiser le coût moyen par période  $C_{u,t}$  si une opération de désassemblage est réalisée à la période  $u$  pour satisfaire les demandes de la période  $u$  à la période  $t$  (c'est-à-dire il n'y a pas d'opération de désassemblage lors des périodes  $u + 1$  à  $t$ ). Cela signifie qu'une demande pour un composant  $i$  pour une période  $k$  telle que  $u \leq k \leq t$  sera stockée  $k - u$  périodes et que le surstock produit en  $u$  sera stocké  $(t - u + 1)$  périodes. D'où l'équation (2.10) où  $SS_{it}$  est le surstock du composant  $i$  qui sera accumulé si l'horizon se terminait en  $t$ .

$$C_{u,t} = \frac{s_t + \sum_{i=1}^N \sum_{k=u}^t (k - u) \cdot h_i \cdot d_{ik} + \sum_{i=1}^N h_i \cdot (t - u + 1) \cdot SS_{it}}{t - u + 1} \quad (2.10)$$

Les valeurs de  $SS_{i,t}$  peuvent être précalculées pour chaque période et chaque composant en déterminant dans un premier temps le besoin cumulé  $XC_t$  pour chaque période :

$$XC_t = \max_i \left\{ \left\lceil \frac{\sum_{k=1}^t d_{ik} - I_{i,0}}{a_i} \right\rceil \right\} \text{ puis } SS_{i,t} = a_i \cdot XC_t - \sum_{k=1}^t d_{ik}.$$

b. *Least Unit Cost (LUC)*

Nous calculons le coût global (coût de setup + coût de stockage), et nous ramenons ensuite cette valeur au coût unitaire (coût global/la somme des niveaux de stocks). Le coût moyen  $C_{u,t}$  de désassemblage à la période  $u$  pour satisfaire les demandes jusqu'à la période  $t$  est donné par la formule suivante (2.11).

$$C_{u,t} = \frac{s_t + \sum_{i=1}^N \sum_{k=u}^t (k-u) \cdot h_i \cdot d_{ik} + \sum_{i=1}^N h_i \cdot (t-u+1) \cdot SS_{it}}{\sum_{i=1}^N \sum_{k=u}^t d_{ik}} \quad (2.11)$$

c. *Part Period Balancing (PPB)*

L'heuristique PPB consiste à désassembler sur une période  $u$  pour satisfaire les demandes d'une période  $t$  tant que le coût de stockage est inférieur au coût de setup en partant de  $t = u$  (le coût de stockage étant incrémenté par rapport aux demandes des composants et aux surstocks à chaque fois que l'on avance d'une période). Si on note ici  $C_{u,t-1}$  le coût de stockage si on désassemble en période  $u$  pour satisfaire les demandes des périodes de  $u$  à  $t-1$ , les demandes de la période  $t$  seront satisfaites à partir du désassemblage en période  $u$  si :

$$C_{u,t-1} + \sum_{i=1}^N (t-u) \cdot h_i \cdot d_{it} + h_i \cdot (t-u+1) \cdot SS_{it} \leq s_t \quad (2.12)$$

La complexité de nos approches de résolution est polynomiale  $O(N.T)$ . Pour démontrer les performances des heuristiques proposées, nous réalisons plusieurs tests sur différents profils de demandes. Avant de réaliser ces tests sur des demandes croissantes et décroissantes ainsi que des demandes variables, nous testons d'abord les performances des heuristiques sur des petites instances.

Avant de tester les performances des heuristiques sur des grandes instances, nous allons donner d'abord un exemple didactique (petites instances).

**Exemple didactique :**

Nous considérons une structure de produit à désassembler constituée d'un seul produit et trois composants. L'horizon de planification est  $T=10$ . Les coûts de setup et de stockage sont supposés constants  $s=800$  et  $h_i=1$ . Les nombres de composants à désassembler sont  $a_1 = 1$ ,  $a_2 = 2$  et  $a_3 = 3$  pour les composants 1, 2 et 3 respectivement. Le stock initial est supposé nul  $I_{i0} = 0$ . Les demandes sont données dans le tableau 7 :

**Tableau 7 : Demandes**

	Périodes									
	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
$d_{1t}$	20	30	25	35	30	40	20	50	20	40
$d_{2t}$	45	45	50	60	70	70	70	80	50	65
$d_{3t}$	55	90	65	120	90	120	60	150	60	120

Le détail de l'application de l'heuristique de Silver Meal aux données de l'exemple est le suivant.

**Application de SILVER MEAL**

Calculs préliminaires

a. Détermination du besoin cumulé  $XC_t$  pour chaque période  $t$

$XC_t$	23	50	75	110	140	180	205	250	270	310
--------	----	----	----	-----	-----	-----	-----	-----	-----	-----

b. Calcul des surstocks à la fin de chaque période  $SS_{i,t}$

$SS_{1,t}$	3	0	0	0	0	0	5	0	0	0
$SS_{2,t}$	1	10	10	20	10	20	0	10	0	15
$SS_{3,t}$	14	5	15	0	0	0	15	0	0	0

Voici en détail l'algorithme de cette heuristique.

Itérations principales

$$u = 1$$

$$t = 1 : C_{1,1} = \frac{s_t + SS_{1,1} \cdot h_1 + SS_{2,1} \cdot h_2 + SS_{3,1} \cdot h_3}{1} = \frac{800 + 3 \cdot 1 + 1 \cdot 1 + 14 \cdot 1}{1} = 818$$

$$t = 2 : C_{1,2} = \frac{s_t + d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + SS_{1,2} \cdot 2 \cdot h_1 + SS_{2,2} \cdot 2 \cdot h_2 + SS_{3,2} \cdot 2 \cdot h_3}{2}$$

$$C_{1,2} = \frac{800 + 30 \cdot 1 + 45 \cdot 1 + 90 \cdot 1 + 0 \cdot 2 \cdot 1 + 10 \cdot 2 \cdot 1 + 5 \cdot 2 \cdot 1}{2} = 497.5$$

$$t = 3 : C_{1,3} = \frac{s_t + d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + d_{1,3} \cdot 2 \cdot h_1 + d_{2,3} \cdot 2 \cdot h_2 + d_{3,3} \cdot 2 \cdot h_3 + SS_{1,3} \cdot 3 \cdot h_1 + SS_{2,3} \cdot 3 \cdot h_2 + SS_{3,3} \cdot 3 \cdot h_3}{3}$$

$$C_{1,3} = \frac{800 + 30 \cdot 1 + 45 \cdot 1 + 90 \cdot 1 + 25 \cdot 2 + 50 \cdot 2 + 65 \cdot 2 + 0 \cdot 3 \cdot 1 + 10 \cdot 3 \cdot 1 + 15 \cdot 3 \cdot 1}{3} = 440$$

$$t = 4 :$$

$$C_{1,4} =$$

$$\frac{s_t + d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + d_{1,3} \cdot 2 \cdot h_1 + d_{2,3} \cdot 2 \cdot h_2 + d_{3,3} \cdot 2 \cdot h_3 + d_{1,4} \cdot 3 \cdot h_1 + d_{2,4} \cdot 3 \cdot h_2 + d_{3,4} \cdot 3 \cdot h_3 + SS_{1,4} \cdot 4 \cdot h_1 + SS_{2,4} \cdot 4 \cdot h_2 + SS_{3,4} \cdot 4 \cdot h_3}{4}$$

$$C_{1,4} = \frac{800 + 30 \cdot 1 + 45 \cdot 1 + 90 \cdot 1 + 25 \cdot 2 + 50 \cdot 2 + 65 \cdot 2 + 35 \cdot 3 + 60 \cdot 3 + 120 \cdot 3 + 0 \cdot 4 \cdot 1 + 20 \cdot 4 \cdot 1 + 0 \cdot 4 \cdot 1}{4} = 492.5$$

$C_{1,4} > C_{1,3}$  : le désassemblage en période 1 permet de satisfaire les demandes de la période 1 à la période 3. Nous avons alors  $X_1 = 75$ ,  $X_2 = 0$ ,  $X_3 = 0$ .

$$u = 4$$

$$t = 4 : C_{4,4} = \frac{s_t + SS_{1,4} \cdot h_1 + SS_{2,4} \cdot h_2 + SS_{3,4} \cdot h_3}{1} = \frac{800 + 0.1 + 20.1 + 0.1}{1} = 820$$

$$t = 5 : C_{4,5} = \frac{s_t + d_{1,5} \cdot h_1 + d_{2,5} \cdot h_2 + d_{3,5} \cdot h_3 + SS_{1,5} \cdot 2 \cdot h_1 + SS_{2,5} \cdot 2 \cdot h_2 + SS_{3,5} \cdot 2 \cdot h_3}{2}$$

$$C_{4,5} = \frac{800 + 30.1 + 70.1 + 90.1 + 0.2.1 + 10.2.1 + 0.2.1}{2} = 505$$

$$t = 6 : C_{4,6} = 503.3$$

$$t = 7 : C_{4,7} = 495$$

$$t = 8 : C_{4,8} = 614$$

$C_{4,8} > C_{4,7}$  : le désassemblage en période 4 permet de satisfaire les demandes de la période 1 à la période 7. Nous avons alors  $X_4 = 205 - 75 = 130$ ,  $X_5 = X_6 = X_7 = 0$ .

$$u = 8$$

$$t = 8 : C_{8,8} = 810$$

$$t = 9 : C_{8,9} = 465$$

$$t = 10 : C_{8,10} = 475$$

$C_{8,10} > C_{8,9}$  : le désassemblage en période 8 permet de satisfaire les demandes de la période 8 à la période 9. Nous avons alors  $X_8 = 270 - 205 = 65$ ,  $X_9 = 0$ .

$$u = 10$$

$$t = 10 : C_{10,10} = 815$$

$$X_{10} = 310 - 270 = 40$$

Le plan obtenu par l'heuristique Silver Meal est résumé dans le tableau 8 :

**Tableau 8** : Plan obtenu par l'heuristique Silver Meal

	Périodes									
	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
$I_{1t}$	55	25	0	95	65	25	5	20	0	0
$I_{2t}$	105	60	10	210	140	70	0	50	0	15
$I_{2t}$	170	80	15	285	195	75	15	60	0	0
$X_t$	75	0	0	130	0	0	0	65	0	40

Le coût total obtenu par l'heuristique est égal 4785.

### Application de LEAST UNIT COST :

Par principe, nous calculons le coût global (coût de setup + coût de stockage). On ramène ensuite cette valeur au coût unitaire (coût moyen par période). Il s'agit de connaître si en augmentant dans la quantité de la période  $t$ , les besoins de la période  $t+1$ , on bénéficie d'un coût unitaire minimum. En cumulant ainsi les périodes, nous finissons par retenir le plan qui offre un coût moyen le plus bas. Puis le calcul reprend.

Voici en détail l'algorithme de cette heuristique.

$$u = 1$$

$$t = 1 : C_{1,1} = \frac{s_t + SS_{1,1} \cdot h_1 + SS_{2,1} \cdot h_2 + SS_{3,1} \cdot h_3}{d_{1,1} + d_{2,1} + d_{3,1}} = \frac{800 + 3.1 + 1.1 + 14.1}{20 + 45 + 55} = 6.81$$

$$t = 2 : C_{1,2} = \frac{s_t + d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + SS_{1,2} \cdot 2 \cdot h_1 + SS_{2,2} \cdot 2 \cdot h_2 + SS_{3,2} \cdot 2 \cdot h_3}{d_{1,1} + d_{2,1} + d_{3,1} + d_{1,2} + d_{2,2} + d_{3,2}}$$

$$C_{1,2} = \frac{800 + 30.1 + 45.1 + 90.1 + 0.2 \cdot 1 + 10.2 \cdot 1 + 5.2 \cdot 1}{20 + 45 + 55 + 30 + 45 + 90} = 3.49$$

$$t = 3 : C_{1,3} = \frac{s_t + d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + d_{1,3} \cdot 2 \cdot h_1 + d_{2,3} \cdot 2 \cdot h_2 + d_{3,3} \cdot 2 \cdot h_3 + SS_{1,3} \cdot 3 \cdot h_1 + SS_{2,3} \cdot 3 \cdot h_2 + SS_{3,3} \cdot 3 \cdot h_3}{d_{1,1} + d_{2,1} + d_{3,1} + d_{1,2} + d_{2,2} + d_{3,2} + d_{1,3} + d_{2,3} + d_{3,3}}$$

$$C_{1,3} = \frac{800 + 30.1 + 45.1 + 90.1 + 25.2 + 50.2 + 65.2 + 0.3 \cdot 1 + 10.3 \cdot 1 + 15.3 \cdot 1}{20 + 45 + 55 + 30 + 45 + 90 + 25 + 50 + 65} = 3.10$$

$$t = 4 :$$

$$C_{1,4} =$$

$$\frac{s_t + d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + d_{1,3} \cdot 2 \cdot h_1 + d_{2,3} \cdot 2 \cdot h_2 + d_{3,3} \cdot 2 \cdot h_3 + d_{1,4} \cdot 3 \cdot h_1 + d_{2,4} \cdot 3 \cdot h_2 + d_{3,4} \cdot 3 \cdot h_3 + SS_{1,4} \cdot 4 \cdot h_1 + SS_{2,4} \cdot 4 \cdot h_2 + SS_{3,4} \cdot 4 \cdot h_3}{d_{1,1} + d_{2,1} + d_{3,1} + d_{1,2} + d_{2,2} + d_{3,2} + d_{1,3} + d_{2,3} + d_{3,3} + d_{1,4} + d_{2,4} + d_{3,4}}$$

$$C_{1,4} = \frac{800 + 30.1 + 45.1 + 90.1 + 25.2 + 50.2 + 65.2 + 35.3 + 60.3 + 120.3 + 0.4 \cdot 1 + 20.4 \cdot 1 + 0.4 \cdot 1}{20 + 45 + 55 + 30 + 45 + 90 + 25 + 50 + 65 + 35 + 60 + 120} = 3.08$$

$$t = 5 : C_{1,5} = 3.25$$

$C_{1,5} > C_{1,4}$  : le désassemblage en période 1 permet de satisfaire les demandes de la période 1 à la période 4. Nous avons alors  $X_1 = 110$ ,  $X_2 = X_3 = X_4 = 0$ .

$$u = 5$$

$$t = 5 : C_{5,5} = \frac{s_t + SS_{1,5} \cdot h_1 + SS_{2,5} \cdot h_2 + SS_{3,5} \cdot h_3}{d_{1,5} + d_{2,5} + d_{3,5}} = \frac{800 + 0.1 + 10.1 + 0.1}{30 + 70 + 90} = 4.26$$

$$t = 6 : C_{5,6} = 2.54$$

$$t = 7 : C_{5,7} = 2.43$$

$$t = 8 : C_{5,8} = 2.60$$

$C_{5,8} > C_{5,7}$  : le désassemblage en période 5 permet de satisfaire les demandes de la période 5 à la période 7. Nous avons alors  $X_5 = 205 - 110 = 95$ ,  $X_6 = X_7 = 0$ .

$$u = 8$$

$$t = 8 : C_{8,8} = 2.89$$

$$t = 9 : C_{8,9} = 2.27$$

$$t = 10 : C_{8,10} = 2.24$$

Le désassemblage en période 8 permet de satisfaire les demandes de la période 8 à la période 10. Nous avons alors  $X_8 = 310 - 205 = 105$ ,  $X_9 = X_{10} = 0$ .



Le plan obtenu par l'heuristique Least Unit Cost est résumé dans le tableau 9 :

**Tableau 9** : Plan obtenu par l'heuristique Least Unit Cost

	Périodes									
	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
$I_{1t}$	90	60	35	0	65	25	5	60	40	0
$I_{2t}$	175	130	80	20	140	70	0	130	80	15
$I_{2t}$	275	185	120	0	195	75	15	180	120	0
$X_t$	110	0	0	0	95	0	0	105	0	0

Le coût total obtenu par l'heuristique est égal 4785.

### **Application de PART PERIOD BALANCING :**

L'heuristique de Part Period Balancing cherche à minimiser la différence entre le coût de stockage et le coût de setup. Il faut tout d'abord calculer la différence entre le coût de stockage  $CS$  et le coût de setup  $s$  jusqu'à ce que  $t = T$  ou que le coût de stockage soit supérieur au coût de setup. Ensuite, choisir la différence absolue minimale.

Voici en détail l'algorithme de cette heuristique.

$$u = 1$$

$$t = 1 : C_{1,1} = SS_{1,1} \cdot h_1 + SS_{2,1} \cdot h_2 + SS_{3,1} \cdot h_3 = 18 < 800$$

$$t = 2 : C_{1,2} = d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + SS_{1,2} \cdot 2 \cdot h_1 + SS_{2,2} \cdot 2 \cdot h_2 + SS_{3,2} \cdot 2 \cdot h_3 = 195 < 800$$

$$t = 3 : C_{1,3} = d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + d_{1,3} \cdot 2 \cdot h_1 + d_{2,3} \cdot 2 \cdot h_2 + d_{3,3} \cdot 2 \cdot h_3 + SS_{1,3} \cdot 3 \cdot h_1 + SS_{2,3} \cdot 3 \cdot h_2 + SS_{3,3} \cdot 3 \cdot h_3 = 520 < 800$$

$$t = 4 : C_{1,4} = d_{1,2} \cdot h_1 + d_{2,2} \cdot h_2 + d_{3,2} \cdot h_3 + d_{1,3} \cdot 2 \cdot h_1 + d_{2,3} \cdot 2 \cdot h_2 + d_{3,3} \cdot 2 \cdot h_3 + d_{1,4} \cdot 3 \cdot h_1 + d_{2,4} \cdot 3 \cdot h_2 + d_{3,4} \cdot 3 \cdot h_3 + SS_{1,4} \cdot 4 \cdot h_1 + SS_{2,4} \cdot 4 \cdot h_2 + SS_{3,4} \cdot 4 \cdot h_3 = 1170 > 800$$

Le désassemblage en période 1 doit permettre de satisfaire les demandes de la période 1 à la période 3. Nous avons alors  $X_1 = 75$ ,  $X_2 = X_3 = 0$ .

$$u = 4$$

$$t = 4 : C_{4,4} = 20 < 800$$

$$t = 5 : C_{4,5} = 210 < 800$$

$$t = 6 : C_{4,6} = 710 < 800$$

$$t = 7 : C_{4,7} = 1180 > 800$$

Le désassemblage en période 4 doit permettre de satisfaire les demandes de la période 4 à la période 6. Nous avons alors  $X_4 = 180 - 75 = 105$ ,  $X_5 = X_6 = 0$ .

$$u = 7$$

$$t = 7 : C_{7,7} = 20 < 800$$

$$t = 8 : C_{7,8} = 300 < 800$$

$$t = 9 : C_{7,9} = 540 < 800$$

$$t = 10 : C_{7,10} = 1275 > 800$$

Le désassemblage en période 7 doit permettre de satisfaire les demandes de la période 7 à la période 9. Nous avons alors  $X_7 = 270 - 180 = 90$ ,  $X_8 = X_9 = 0$ .

$$u = 10$$

$$t = 10 : C_{10,10} = 15$$

Le désassemblage en période 10 doit permettre de satisfaire les demandes de la période 10. Nous avons alors  $X_{10} = 310 - 270 = 40$ .

Le plan obtenu par l'heuristique Part Period Balancing est résumé dans le tableau 10 :

**Tableau 10** : Plan obtenu par l'heuristique Part Period Balancing

	Périodes									
	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
$I_{1t}$	55	25	0	70	40	0	70	20	0	0
$I_{2t}$	105	60	10	160	90	20	130	50	0	15
$I_{2t}$	170	80	15	210	120	0	210	60	0	0
$X_t$	75	0	0	105	0	0	90	0	0	40

Le coût total obtenu par l'heuristique est égal 4985.

### 1.3.3 Résultats expérimentaux

Afin de tester les performances des heuristiques proposées, nous générons des instances caractérisées par le nombre de période, le nombre de composant et le profil des demandes (évolution au cours de l'horizon de planification). Le nombre de période peut être 10, 20 ou 30. Si  $DU(a, b)$  est une distribution uniforme discrète entre  $a$  et  $b$ , trois classes du nombre de composant sont considérées : petite, moyenne et grande :  $DU(1,10)$ ,  $DU(10,50)$  et  $DU(50,100)$  pour la petite classe, moyenne et grande respectivement. Dix instances de chaque type sont ainsi générées.

Les instances sont générées de la même manière que (Teunter et *al.* 2006). Nous supposons ensuite trois profils de demandes : demandes croissantes, demandes décroissantes et demandes variables. L'objectif est de déterminer sur quel profil des demandes les heuristiques sont performantes. En effet, dans le problème de désassemblage, il arrive que les produits en fin de vie ne soient pas disponibles ou qu'il y a parfois un excès de ces derniers. Les données sont les suivantes :

- le coût de setup est supposé fixe à chaque période  $s = 1000$ ,
- le coût de stockage est généré suivant  $DU(1, 5)$ , il est supposé fixe à chaque période,
- le rendement de désassemblage est généré suivant  $DU(1, 4)$ ,
- les demandes variables sont générées suivant  $DU(50,200)$ ,

- le stock initial est généré suivant  $DU(20,100)$ .

Dans cette étude, l'écart en pourcentage (gap) de la solution obtenue par chaque heuristique par rapport à la solution optimale est calculé pour chaque type d'instance. L'impact des différents profils de demandes sur les performances de l'heuristique est analysé. La programmation dynamique présentée dans (Kim 2005) fournit la solution optimale mais avec des temps de calcul différents de CPLEX. Les résultats des tableaux 11 et 12 présentent le gap moyen et le gap maximal pour chaque heuristique et pour chaque classe des composants. Ils représentent également les temps de calcul moyen et maximal de chaque heuristique, de programmation dynamique ainsi que celui de CPLEX.

Pour chaque instance, les valeurs de la fonction objectif obtenues par les heuristiques sont comparées avec celles de la solution optimale obtenue par l'algorithme de (Kim 2005). Les gaps moyen et maximal ainsi que le temps de calcul de chaque heuristique sont présentés dans les tableaux 11 et 12.

$$Gap = \frac{\text{Solution heuristique} - \text{Solution optimale}}{\text{Solution optimale}}$$

### 1.3.3.1 Demandes croissantes et décroissantes

La demande est générée aléatoirement selon une distribution normale avec une tendance croissante ou décroissante (sur la moyenne) selon la méthode présentée dans (Teunter et al. 2006). En effet, dans une période  $t$ , la demande est égale :  $d_{it} = \mu + \epsilon_t + \tau(t - 1)$  pour  $i = 1, \dots, N$  et  $t = 1, \dots, T$ , avec  $\epsilon_t$  une variable aléatoire qui suit une distribution normale centrée d'écart type  $\sigma = 5$ ,  $\mu$  est la demande de départ et  $\tau$  le facteur de tendance avec  $\mu = 100$ , et  $\tau = 10$ . Dans le cas des demandes décroissantes  $\tau = -10$  et  $\mu = 210$ .

Les résultats des demandes croissantes et décroissantes sont résumés dans le tableau 11. D'après ces résultats, nous remarquons que les heuristiques donnent des gaps d'environ 8% pour les petites et les moyennes instances et environ 26.90% pour les grandes instances. Étant donné que les demandes sont croissantes, celles-ci sont importantes aux dernières périodes de l'horizon et le coût de stockage devient important à cause d'une accumulation de stock à la fin de l'horizon de planification. Ainsi, le stock augmente proportionnellement avec la demande, ce qui entraîne un coût total important. Il est à noter également que PPB est toujours meilleure par rapport aux heuristiques SM et LUC (en calculant la moyenne de sa moyenne).

Nous remarquons également que les heuristiques sont plus performantes en comparant avec le cas des demandes croissantes. Dans les deux cas, comme nous pouvons le constater sur le tableau 11, Part Period Balancing est meilleure par rapport aux autres heuristiques. Il est à noter également que ces heuristiques ne sont pas très performantes pour les grandes instances.

À partir du tableau 12, nous constatons que les heuristiques donnent des solutions moins performantes dans le cas où l'horizon de planification est long ainsi qu'avec l'augmentation du nombre de composants. Dans ce cas, le coût de stockage total est important car le stock des composants s'accumule et augmente tout au long de l'horizon de planification par opposition aux horizons de planification plus court où le coût de stockage a moins d'influence (les surplus sont stockés sur des durées plus courtes).

**Tableau 11** : Demandes croissantes et décroissantes

*Demandes croissantes*

N	T	Gaps(%)						CPU (secondes)									
		PPB		SM		LUC		Moyen			Maximal			Moyen	Maximal	Moyen	Maximal
		Moy	Max	Moy	Max	Moy	Max	SM	LUC	PPB	SM	LUC	PPB	CPLEX		Programmation dynamique	
Petit	10	<b>4.37</b>	<b>15.81</b>	6.34	16.92	8.53	17.14	0.001	0.001	0.001	0.008	0.008	0.008	0.010	0.050	0.002	0.014
	20	8.41	<b>13.21</b>	9.39	18.27	<b>6.43</b>	20.50	0.002	0.002	0.002	0.010	0.009	0.012	0.020	0.070	0.004	0.015
	30	<b>7.57</b>	<b>13.46</b>	9.61	17.50	9.80	21.66	0.003	0.003	0.003	0.014	0.015	0.015	0.030	0.100	0.007	0.021
Moyen	10	<b>6.54</b>	<b>15.96</b>	9.73	16.04	10.91	19.25	0.003	0.002	0.004	0.012	0.011	0.012	0.040	0.080	0.006	0.018
	20	<b>5.44</b>	<b>16.76</b>	7.51	18.10	10.75	20.65	0.014	0.015	0.017	0.050	0.067	0.059	0.070	0.160	0.028	0.069
	30	5.19	<b>16.54</b>	<b>3.36</b>	19.37	10.51	22.29	0.018	0.018	0.023	0.067	0.056	0.066	0.080	0.270	0.033	0.089
Grande	10	9.50	<b>23.81</b>	<b>8.43</b>	27.32	12.27	31.93	0.029	0.036	0.036	0.091	0.103	0.117	2.650	7.570	0.105	0.282
	20	<b>7.89</b>	<b>20.90</b>	11.12	24.03	11.27	28.19	0.039	0.037	0.051	0.192	0.208	0.206	6.330	13.090	0.138	0.539
	30	12.00	26.90	14.02	32.79	<b>11.90</b>	30.60	0.047	0.056	0.060	0.212	0.218	0.218	7.110	14.560	0.169	0.617

*Demandes décroissantes*

N	T	PPB		SM		LUC		Moyen			Maximal			Moyen	Maximal	Moyen	Maximal
		Moy	Max	Moy	Max	Moy	Max	SM	LUC	PPB	SM	LUC	PPB	CPLEX		Programmation dynamique	
		Petit	10	<b>5.04</b>	17.71	7.46	<b>11.21</b>	7.65	17.19	0.001	0.001	0.001	0.008	0.008	0.009	0.010	0.040
20	<b>8.40</b>		<b>14.33</b>	9.63	18.80	9.58	22.41	0.002	0.001	0.002	0.011	0.011	0.012	0.020	0.060	0.003	0.015
30	8.87		<b>17.85</b>	10.95	18.21	<b>8.41</b>	24.81	0.003	0.003	0.003	0.014	0.015	0.015	0.020	0.090	0.004	0.019
Moyen	10	<b>8.94</b>	13.02	9.34	<b>12.19</b>	8.98	19.83	0.003	0.002	0.004	0.012	0.011	0.012	0.030	0.140	0.006	0.017
	20	<b>7.37</b>	<b>17.91</b>	8.72	19.35	8.97	19.07	0.014	0.018	0.020	0.049	0.065	0.059	0.060	0.230	0.028	0.068
	30	6.10	<b>17.68</b>	<b>5.28</b>	20.70	6.44	23.83	0.018	0.018	0.023	0.073	0.056	0.067	0.070	0.210	0.032	0.101
Grande	10	<b>12.09</b>	<b>23.81</b>	12.39	27.32	13.60	31.93	0.029	0.035	0.032	0.098	0.112	0.129	2.650	7.570	0.112	0.289
	20	<b>13.05</b>	23.49	13.77	<b>21.39</b>	13.91	28.19	0.030	0.033	0.054	0.197	0.223	0.226	6.330	13.090	0.141	0.589
	30	13.29	26.90	<b>11.38</b>	<b>22.22</b>	13.23	30.60	0.046	0.062	0.069	0.216	0.212	0.232	7.110	14.560	0.163	0.652

### 1.3.3.2 Demandes variables

Nous rappelons que les demandes sont générées aléatoirement suivant  $DU(50,200)$ .

**Tableau 12 :** Demandes variables

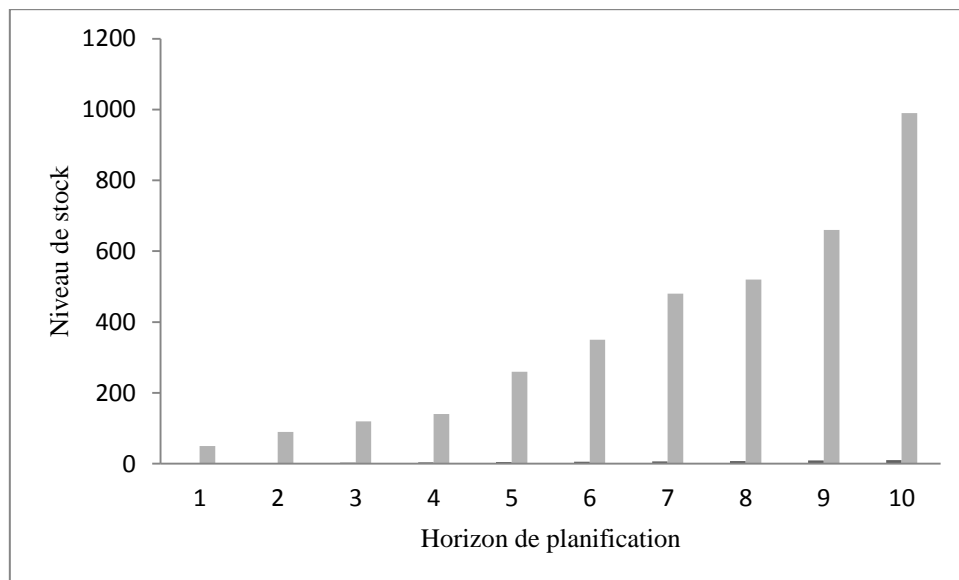
#### *Demandes variables*

N	T	Gaps (%)						CPU (secondes)									
		PPB		SM		LUC		Moyen			Maximal			Moyen	Maximal	Moyen	Maximal
		Moy	Max	Moy	Max	Moy	Max	PPB	SM	LUC	PPB	SM	LUC	CPLEX		Programmation dynamique	
Petit	10	<b>8.78</b>	12.76	9.78	<b>11.02</b>	9.14	16.40	0.001	0.001	0.001	0.010	0.010	0.011	0.020	0.060	0.001	0.014
	20	<b>7.89</b>	16.54	9.89	<b>16.40</b>	8.95	21.18	0.002	0.002	0.002	0.014	0.015	0.014	0.020	0.080	0.003	0.019
	30	<b>6.23</b>	17.04	7.31	<b>15.16</b>	7.71	13.51	0.004	0.003	0.004	0.019	0.019	0.019	0.040	0.110	0.006	0.026
Moyen	10	8.19	<b>16.06</b>	<b>6.57</b>	17.22	9.94	18.67	0.003	0.003	0.003	0.017	0.016	0.018	0.040	0.180	0.004	0.024
	20	9.37	18.91	<b>9.12</b>	<b>17.35</b>	10.97	22.07	0.014	0.017	0.016	0.048	0.053	0.054	0.070	0.290	0.022	0.067
	30	10.10	<b>17.68</b>	<b>9.28</b>	20.70	10.44	23.83	0.017	0.017	0.020	0.057	0.054	0.063	0.080	0.380	0.028	0.079
Grande	10	9.14	22.89	<b>7.92</b>	25.67	11.53	30.00	0.03	0.04	0.03	0.105	0.129	0.137	2.760	7.880	0.129	0.393
	20	<b>7.89</b>	<b>20.90</b>	11.12	24.03	11.27	28.19	0.03	0.03	0.06	0.217	0.236	0.244	6.590	13.630	0.158	0.676
	30	12.00	<b>26.90</b>	14.02	32.79	<b>11.90</b>	30.60	0.05	0.06	0.07	0.229	0.221	0.248	7.400	15.160	0.176	0.681

D'une manière générale, les heuristiques, dans le cas où les demandes sont croissantes, donnent un gap inférieur par rapport à celui où les demandes sont décroissantes. En effet, les demandes sont faibles dans les premières périodes et l'accumulation de stocks n'est pas assez importante. Par conséquent, l'accumulation de stocks impacte moins le coût. Puisque l'effet de l'accumulation de stocks est limité par l'horizon de planification, les résultats des trois heuristiques ne sont pas très différents par rapport à la valeur d'erreur moyenne (gap moyen). Dans le cas des demandes décroissantes, les surstocks générés au début de l'horizon de planification sont en général plus persistants.

Le stock de surplus s'accumule sur l'horizon de planification comme le montre la Figure 18. D'après l'équation de conservation des flux nous avons :

$$I_{it} = I_{i,t-1} + a_i \cdot X_t - d_{it} \text{ pour } t = 1, \dots, T \text{ et } i = 1, \dots, N \quad (2.13)$$



**Figure 18 :** Exemple d'accumulation de stocks

Finalement, la solution optimale et le temps de calcul dépendent également du nombre de composants. Ainsi, nous obtenons des solutions quasi-optimales dans le cas où le nombre de composants est petit (ne dépasse pas 10 composants). Dans le cas contraire, nous obtenons un gap d'environ 26.90% pour des instances où le nombre de composants est très important (environ 1000 composants). Il en est de même pour un horizon de planification très long (30 périodes par exemple). En effet, dans ce cas, nous avons une accumulation de stocks pour les composants puisque les heuristiques ne tiennent pas compte de la dernière période de l'horizon. Par conséquent, le coût de stockage devient très élevé. En conclusion, les heuristiques utilisées dans le problème de lot sizing en production peuvent être également utilisées dans la résolution du problème de lot sizing en désassemblage notamment pour les petites et les moyennes instances. Par contre, pour les grandes instances les heuristiques ne sont pas jugées performantes.

Dans cette étude, nous avons constaté qu'il y a un problème d'accumulation de stocks influençant directement le coût total. Cette dernière augmente avec le nombre de périodes (plus l'horizon de

planification est long plus le stock est élevé). Pour minimiser le coût total, il faut tenir compte du surplus qui ne sert à satisfaire aucune demande à la fin de l'horizon de planification.

L'accumulation de stocks jusqu'à la fin de l'horizon de planification est un problème rencontré en pratique dans les centres de désassemblage. Les modèles actuels ne tiennent pas compte des décisions qui doivent être prises par rapport à ce stock. Un modèle pour diminuer le surstock consiste à sélectionner les demandes qui doivent être satisfaites en autorisant les ventes perdues.

## 2 Problème de lot sizing en désassemblage avec ventes perdues

Dans la partie précédente, nous avons traité le problème de lot sizing en désassemblage avec satisfaction de toutes les demandes en minimisant les coûts de setup et de stockage. Ce type de modèle génère un stock de surplus à la fin de l'horizon qui ne sert à satisfaire aucune demande. Ce surplus correspond aux composants restants à la fin de l'horizon. Les ventes perdues impliquent que certaines demandes ne sont pas satisfaites qui engendrent un coût de «ventes perdues». Le modèle avec ventes perdues est intéressant dans la pratique pour limiter le surplus. En effet, les demandes peuvent être perdues si les coûts sont importants par rapport aux profits en trouvant un équilibre entre trois types de coûts : les coûts de setup, de stockage et de ventes perdues.

### 2.1 Modélisation du problème

Dans ce problème, nous considérons la même structure de produits de la section précédente. Il s'agit de déterminer un plan optimal de désassemblage d'un produit constitué de  $N$  composants sur un horizon de planification constitué de  $T$  périodes. Puisque les demandes peuvent être perdues, nous introduisons un coût de ventes perdues pour chaque composant. Les notations suivantes sont utilisées pour formuler le modèle mathématique.

Pour formuler le modèle mathématique, nous utilisons les notations suivantes :

#### *Indices*

$i$  : indice pour les composants,  $i = 1, \dots, N$

$t$  : indice pour les périodes,  $t = 1, \dots, T$

#### *Paramètres*

$a_i$  : rendement de désassemblage

$d_{it}$  : demande pour le composant  $i$  à la période  $t$

$D_i = \sum_{t=1}^T d_{it}$  : Demande cumulée du composant  $i$  sur tout l'horizon de planification  $T$

$h_i$  : coût de stockage du composant  $i$

$p_i$  : coût de ventes perdues du composant  $i$

$I_{i0}$  : niveau de stock initial du composant  $i$

$s_t$  : coût de setup de produit en fin de vie à la période  $t$

$M$  : nombre entier assez grand

### *Variables de décisions*

$I_{it}$  : niveau de stock du composant  $i$  à la période  $t$

$L_{it}$  : quantité de ventes perdues de composant  $i$  à la période  $t$

$X_t$  : quantité de désassemblage à la période  $t$

$Y_t$  : variable binaire égale à 1 s'il y a un désassemblage à la période  $t$ , et 0 sinon

En utilisant les paramètres et les variables ci-dessus, le modèle avec ventes perdues peut être formulé comme suit :

$$\text{Min} \sum_{t=1}^T s_t \cdot Y_t + \sum_{t=1}^T \sum_{i=1}^N (h_i \cdot I_{it} + p_i \cdot L_{it})$$

Sous contraintes :

$$I_{it} = I_{i,t-1} + L_{it} + a_i \cdot X_t - d_{it} \text{ pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (2.14)$$

$$X_t \leq M \cdot Y_t \text{ pour } t = 1, \dots, T \quad (2.15)$$

$$L_{it} \leq d_{it} \text{ pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (2.16)$$

$$Y_t \in \{0,1\} \text{ pour } t = 1, \dots, T \quad (2.17)$$

$$I_{it}, L_{it}, X_t \geq 0 \text{ et } X_t, L_{it}, I_{it} \in \mathbb{N} \text{ pour } i = 1, \dots, N \text{ et } t = 1, \dots, T \quad (2.18)$$

La fonction objectif minimise la somme du coût total encouru sur l'horizon de planification, ce dernier est composé des coûts de stockage, de setup et de ventes perdues. La contrainte (2.14) représente la conservation des flux pour les composants. La contrainte (2.15) assure le lancement de désassemblage où la quantité produite ne doit pas dépasser le paramètre  $M$  ( $M$  est une valeur assez grande) qui peut représenter la somme maximale des demandes d'un composant sur l'horizon de planification. La contrainte (2.16) assure que les quantités de ventes perdues de composant  $i$  à la période  $t$  soient inférieures à la demande du même composant à la même période. La contrainte (2.17) est une variable binaire égale à 1 si on produit à la période  $t$  et 0 sinon. Finalement, la contrainte (2.18) signifie que les variables  $I_{it}$ ,  $L_{it}$  et  $X_t$  sont discrètes non négatives pour tous les composants  $i$  et à chaque période  $t$ .

Les hypothèses du modèle proposé sont les suivantes :

- les demandes de composants sont données et déterministes ;
- la vente perdue est autorisée (la demande peut ne pas être satisfaite) ;
- toutes les pièces sont dans un état parfait et aucune pièce n'est défectueuse ;
- il n'y a pas de stockage pour les produits en fin de vie, les produits peuvent être obtenus une fois qu'ils sont demandés.



Nous allons ensuite montrer que les techniques d'optimisation utilisées dans la résolution du problème de lot sizing en production ne peuvent pas être appliquées directement au problème de lot sizing en désassemblage. En effet, des adaptations spécifiques sont nécessaires si les ventes perdues sont autorisées.

Avant de passer à la résolution du problème proposé, nous allons présenter d'abord d'autres modèles équivalents avec le lot sizing en désassemblage avec ventes perdues. L'objectif est de démontrer l'originalité de notre modèle proposé.

## 2.2 Équivalence entre modèles

À notre connaissance, le problème de lot sizing en désassemblage avec ventes perdues n'a pas été traité dans la littérature. Le coût de ventes perdues peut représenter la marge réalisée sur la vente d'une unité ou encore le coût additionnel pour assurer rapidement une unité de produit au client si la rupture sur la demande est autorisée. Tout récemment, (Ji et al. 2015) ont proposé un modèle similaire à celui que nous avons proposé. Dans leur problème, les auteurs supposent que si la demande ne peut pas être satisfaite, une quantité des composants peut être achetée pour satisfaire cette demande. Ce coût d'achat est équivalent au coût de ventes perdues dans notre modèle. En outre, le modèle de maximisation des ventes se ramène également à celui de ventes perdues. Dans ce type de modèles, nous cherchons à maximiser le profit total qui est la différence entre le revenu total et le coût total sur un horizon de planification. Le revenu total est obtenu par la vente de composants. Ce coût est composé du coût d'acquisition de produits en fin de vie, du coût de stockage, du coût de setup et du coût de désassemblage.

Pour représenter les deux modèles, nous utilisons les paramètres et les variables suivants :

$C_t$  : coût de désassemblage

$\alpha_i$  : coût d'achat des composants

$\beta_t$  : coût d'achat de produits en fin de vie

### *Variables des décisions*

$Q_{it}$  : quantité achetée du composant  $i$  à la période  $t$

$R_{it}$  : quantité vendue du composant  $i$  à la période  $t$

En utilisant ces paramètres et variables, le modèle peut être formulé comme suit :

#### *a. Modèle de maximisation des ventes*

$$P1 = \text{Max} \left( \sum_{t=1}^T \sum_{i=1}^N p_i \cdot R_{it} - \left( \sum_{t=1}^T s_t \cdot Y_t + \sum_{t=1}^T \beta_t \cdot X_t + \sum_{t=1}^T \sum_{i=1}^N h_i \cdot I_{it} \right) \right)$$

Nous transformons ce modèle pour se ramener à celui avec ventes perdues. Dans ce problème, le revenu à la période  $t$  est réalisé est obtenu en multipliant les ventes réalisées par le prix de vente

unitaire correspondant. La vente réalisée dans la période  $t$  est donnée par  $(d_{it} - L_{it})$ . Par conséquent, le problème P1 peut être écrit comme suit :

$$P1 = \text{Max} \left( \sum_{t=1}^T \sum_{i=1}^N p_i \cdot (d_{it} - L_{it}) - \left( \sum_{t=1}^T S_t \cdot Y_t + \sum_{t=1}^T \beta_t \cdot X_t + \sum_{t=1}^T \sum_{i=1}^N h_i \cdot I_{it} \right) \right)$$

Ou encore :

$$P1 = \text{Max} \left( \sum_{t=1}^T \sum_{i=1}^N p_i \cdot d_{it} - \left( \sum_{t=1}^T S_t \cdot Y_t + \sum_{t=1}^T \beta_t \cdot X_t + \sum_{t=1}^T \sum_{i=1}^N (h_i \cdot I_{it} + p_i \cdot L_{it}) \right) \right)$$

Le terme  $\sum_{t=1}^T \sum_{i=1}^N p_i \cdot d_{it}$  est une constante qui peut être omise de la fonction objectif. Puisqu'il n'y a pas de différence entre la maximisation d'une fonction objectif  $f(x)$  et la minimisation de  $-f(x)$  en ce qui concerne les valeurs optimales de variable de décision  $x$ , nous pouvons donc écrire P1 comme suit :

$$-P1 = \text{Max} \left( \left( \sum_{t=1}^T S_t \cdot Y_t + \sum_{t=1}^T \beta_t \cdot X_t + \sum_{t=1}^T \sum_{i=1}^N (h_i \cdot I_{it} + p_i \cdot L_{it}) \right) - \text{constante} \right)$$

$\text{constante} = \sum_{t=1}^T \sum_{i=1}^N p_i \cdot d_{it}$ . La fonction objectif devient la minimisation des coûts de setup, de stockage, de désassemblage et de vente perdue :

$$P'1 = \text{Min} \left( \sum_{t=1}^T S_t \cdot Y_t + \sum_{t=1}^T \beta_t \cdot X_t + \sum_{t=1}^T \sum_{i=1}^N (h_i \cdot I_{it} + p_i \cdot L_{it}) \right)$$

Sous contraintes :

(2.14), (2.15), (2.16), (2.17) et (2.18)

Ce modèle devient similaire à celui que nous avons proposé (modèle avec ventes perdues). Par conséquent, le modèle de maximisation des ventes se ramène à celui avec ventes perdues.

Nous rappelons qu'il existe certaines spécificités pour le problème de lot sizing en désassemblage. En effet, le désassemblage est un processus divergent c'est-à-dire un produit est composé en plusieurs composants, contrairement à l'assemblage qui est convergent où plusieurs composants convergent vers un seul produit.

### 2.3 Méthodes de résolution

Des méthodes de résolution ont été proposées pour les problèmes de lot sizing en production avec ventes perdues. La méthode de la programmation dynamique utilisée dernièrement dans (Absi et al. 2013) ne donnent pas une solution optimale dans le cas du désassemblage comme le montre le contre-exemple présenté dans la section 2.3.1. Ceci est dû à la divergence des flux présenté avant.

### 2.3.1 Méthodes exactes

Plusieurs auteurs ont utilisé des méthodes exactes pour résoudre le problème de lot sizing en production avec ventes perdues. (Sandbothe et Thompson 1990) sont parmi les premiers chercheurs à l'avoir étudié, ils considèrent une version du problème avec capacité limitée, ils ont résolu le problème à l'aide d'une programmation dynamique. Lorsque la contrainte de capacité est relâchée, (Aksen et al. 2003) ont proposé un algorithme de résolution quadratique pour le problème basé encore une fois sur la programmation dynamique. (Absi 2005) et (Absi et Kedad-Sidhoum 2008) ont également traité le problème de ventes perdues lorsque plusieurs commandes de préparation sont envisagées. (Absi et al. 2011) ont considéré à la fois les backlogs et les ventes perdues. Dans (Absi et al. 2013), les auteurs ont traité le problème avec ventes perdues et contraintes de capacité. Ils ont proposé une relaxation lagrangienne en relaxant la contrainte de capacité et ils ont ensuite résolu le problème à un seul produit à l'aide d'une programmation dynamique.

Nous montrons avec un exemple que la programmation dynamique utilisée par (Absi et al. 2013) ne peut pas être utilisée directement dans la résolution du problème que nous avons proposé. Ces auteurs ont traité le problème de lot sizing en production avec ventes perdues et contraintes de capacité. Ils ont supposé que la demande peut être totalement ou partiellement perdue ou totalement satisfaite.

Nous rappelons maintenant les propriétés qui permettent d'avoir un algorithme de programmation dynamique polynomiale pour le problème de lot sizing en production.

**Propriété 1 :** Dans le problème de lot sizing avec ventes perdues, il existe une solution optimale telle que la demande dans une période  $t$  est entièrement satisfaite si l'approvisionnement est effectué durant cette période :  $L_t \cdot X_t = 0$ .

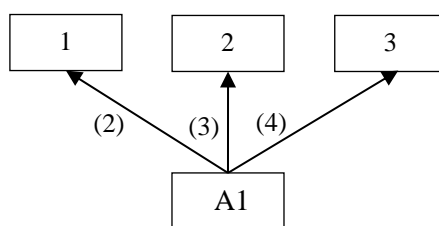
**Propriété 2 :** Il existe une solution optimale telle que : si une demande est perdue à une période donnée  $t$ , cette demande doit être totalement perdue.  $L_t(d_t - L_t) = 0$ .

**Propriété 3 :** Il existe également une propriété qui signifie que pour une solution optimale, on ne produit jamais à une période ayant un stock non nul provenant de la période précédente  $X_t \cdot I_{t-1} = 0 \forall t$ .

Afin de mieux comprendre la différence entre les propriétés du problème de lot sizing en production et en désassemblage, nous allons donner un exemple.

**Exemple :**

Nous considérons une structure du produit à deux niveaux et un seul produit. Le produit est composé de trois composants (Figure 19). Les données de l'exemple sont résumées dans le tableau 13.



**Figure 19 :** Structure d'un produit à deux niveaux

**Tableau 13** : données d'entrées

$h_i$	$p_i$	$a_i$	$d_{i1}$	$d_{i2}$	$d_{i3}$	$d_{i4}$	$d_{i5}$
5	10	2	40	100	70	50	20
5	10	3	100	50	25	80	100
5	10	4	70	30	50	20	80

Le coût de setup est supposé constant tout au long de l'horizon de planification :  $s = 1000$ .

La solution optimale de l'exemple est donnée dans la table 14:

**Tableau 14** : Plan optimal de l'exemple

	$T$				
	1	2	3	4	5
$Y_t$	1	1	0	1	0
$x_t$	28	14	17	0	0
$I_{1t}$	16	0	42	0	34
$I_{2t}$	26	0	17	23	1
$I_{3t}$	0	48	62	13	1
$L_{1t}$	0	0	0	2	0
$L_{2t}$	0	16	0	0	0
$L_{3t}$	29	0	0	0	0

D'après les résultats de tableau 14, nous remarquons que pour les périodes 4 et 5 aucun composant ne satisfait les propriétés 1 et 2 à la fois. Pour les composants 2, 1 et 3, nous ne perdons pas la totalité de la demande sur les périodes 1, 2 et 4 respectivement. Pour le composant 3, nous avons un surplus en périodes 2 et 3. Les périodes 1, 2 et 4 sont des périodes de setup. Notre stratégie permet de trouver un compromis entre le coût de stockage et de ventes perdues.

Nous remarquons également que :  $L_{31} \cdot (d_{11} - L_{31}) = 29 \cdot (70 - 29) \neq 0$ , donc cela ne vérifie pas la propriété 1 où  $L_t \cdot (d_t - L_t) = 0$ . Nous avons également  $L_{31} \cdot X_1 = 812 \neq 0$ , donc la propriété 2 où  $L_t \cdot X_t = 0$  n'est pas vérifiée dans le problème de lot sizing en désassemblage. Par conséquent, la programmation dynamique ne peut pas être utilisée directement pour résoudre le problème de lot sizing en désassemblage avec ventes perdues.

Par contre, le problème que nous proposons peut être résolu à l'aide d'un logiciel d'optimisation tel que CPLEX, Solveur ou autre. Pour résoudre ce problème pour toutes les instances, nous utilisons d'abord CPLEX.

## 2.4 Résultats expérimentaux

Nous allons générer plusieurs instances en faisant varier le nombre de composant et le nombre de périodes en générant aléatoirement 10 instances de chaque type. Le nombre de composant est généré suivant  $DU(1,10)$ ,  $DU(10,100)$ ,  $DU(100,1000)$  et  $DU(1000,10000)$  respectivement pour les types d'instances petite, moyenne grande et X-Grande. Des instances de 10, 20, 30, 40 et 50 périodes sont générées pour chaque plage de nombre de composants.

Les instances sont générées de la même manière que (Teunter et *al.* 2006). Nous supposons ensuite trois profils de demandes : demandes croissantes, demandes décroissantes et demandes variables. L'objectif est de déterminer sur quel profil des demandes les heuristiques sont performantes. En effet, dans le problème de désassemblage, il arrive que les produits en fin de vie ne soient pas disponibles ou qu'il y a parfois un excès de ces derniers.

- le coût de setup est supposé fixe à chaque période  $s = 1000$ ,
- le coût de stockage est généré suivant  $DU(1, 5)$ , il est supposé fixe à chaque période,
- le rendement de désassemblage est généré suivant  $DU(1, 4)$ ,
- les demandes variables sont générées suivant  $DU(50, 200)$ ,
- le stock initial est supposé nul,
- le coût de ventes perdues est généré suivant  $DU(10, 20)$ ,
- le stock initial est généré suivant  $DU(20, 50)$ .
- dans le cas où les demandes sont supposées constantes, les caractéristiques de celle-ci sont les mêmes pour toutes les périodes. Dans cette étude la demande est générée selon une distribution normale avec une moyenne  $\mu = 100$  et un écart type  $\sigma = 10$ .

Les demandes croissantes et décroissantes sont générées de la même manière que celles de la section 1.3.3.1.

**Tableau 15** : Résultats des tests pour des demandes constantes et variables

Classes	Périodes	Demandes constantes		Demandes variables	
		CPU (secondes)		CPU (secondes)	
		Moy	Max	Moy	Max
Petit	10	0.11	0.29	0.31	0.48
	20	0.18	0.65	0.32	0.77
	30	1.54	2.11	1.68	3.11
	40	1.59	2.55	1.69	2.89
	50	1.65	2.84	1.89	3.11
Moyen	10	1.02	1.88	1.11	2.15
	20	2.07	3.11	2.21	3.56
	30	2.11	3.55	2.29	3.89
	40	2.24	3.98	2.36	4.56
	50	2.68	4.29	3.12	5.67
Grande	10	1.34	2.63	1.36	3.01
	20	1.54	4.63	1.88	11.99
	30	3.75	7.85	4.05	13.31
	40	8.87	17.32	9.66	22.51
	50	17.63	37.57	21.56	45.61
X-Grande	10	12.49	31.20	13.00	32.89
	20	18.15	59.48	21.32	63.81
	30	53.96	112.99	94.22	132.32
	40	96.09	313.08	102.33	411.23
	50	125.65	723.21	162.32	821.33

Le tableau 15 résume les temps de calcul trouvés par CPLEX pour les demandes variables et constantes. D'après ce tableau, nous remarquons que le temps de calcul augmente avec l'augmentation de périodes et du nombre de composants. Nous remarquons également que pour la petite classe, nous ne dépassons pas 4 secondes même pour le plus long horizon ( $T = 50$ ). Par contre dans la plus grande classe où le nombre de composants varie entre 1000 et 10000, ce temps devient très long sur le même horizon environ 820 secondes. Nous pouvons donc constater que le temps de calcul est également très sensible en fonction de nombre de composants.

Il est important de souligner que le temps de calcul dépend également de la distribution ou profil des demandes. Comme le montre le tableau 16, le temps de calcul dans le cas des demandes variables est plus grand que celui des demandes constantes.

**Tableau 16** : Résultats des tests pour demandes croissantes et décroissantes

Classes	Périodes	Demandes croissantes		Demandes décroissantes	
		CPU (secondes)		CPU (secondes)	
		Moy	Max	Moy	Max
Petit	10	0.12	0.36	0.11	0.31
	20	0.23	0.63	0.12	0.42
	30	0.25	0.72	0.14	0.72
	40	0.67	1.22	0.18	0.81
	50	0.68	1.43	0.19	0.82
Moyen	10	0.13	0.56	0.14	0.51
	20	0.55	1.22	0.18	0.89
	30	0.61	2.89	0.24	1.66
	40	0.69	3.11	0.33	2.11
	50	0.71	3.28	0.42	2.62
Grande	10	0.49	2.36	0.23	2.35
	20	2.65	7.57	1.20	4.68
	30	6.33	13.09	7.21	15.11
	40	7.11	14.56	10.13	25.45
	50	12.63	28.74	13.44	30.19
X-Grande	10	8.11	23.91	6.59	19.42
	20	23.22	61.62	44.15	127.78
	30	211.21	393.78	421.02	640.47
	40	246.66	581.84	351.33	709.23
	50	345.12	1215.11	463.13	1494.52

Le problème avec ventes perdues ne peut pas être résolu avec les méthodes de programmation dynamique proposées dans la littérature (Absi et *al.* 2013) et les temps de calcul avec l'utilisation de CPLEX sont très variables ce qui limite son utilisation dans le cadre de résolution de sous-problèmes dans des problèmes multi-produits. Par ailleurs, l'utilisation de Solveur de programmation mathématique n'est pas toujours développée dans les outils de planification en pratique. La programmation dynamique n'étant pas adaptée à ce problème, les heuristiques présentées dans la première partie de ce chapitre sont adaptées pour résoudre le problème avec ventes perdues.

## 2.5 Heuristiques

Comme nous l'avons déjà montré dans la section (1.2.2), l'heuristique *Silver Meal* (SM), l'heuristique *Least Unit Cost* (LUC) et l'heuristique *Part Period Balancing* (PPB) sont rapides et elles donnent des solutions acceptables notamment pour les petites et les moyennes instances. En plus, ces dernières n'ont pas encore été utilisées dans le problème de lot sizing en désassemblage avec ventes perdues.

a. *Silver Meal*

Comme pour le problème sans ventes perdues, l'heuristique consiste à désassembler une quantité de produits  $X_u$  sur une période  $u$  pour satisfaire des demandes de la période  $u$  à la période  $t$ .  $u$  étant fixée,  $t$  est choisie en partant de  $u$  et en l'incrémentant d'une période tant que  $C_{u,t}$  décroît avec :

$$C_{u,t} = \frac{s_u + \sum_{i=1}^N \sum_{k=u}^t (h_i \cdot I_{ik} + p_i \cdot L_{ik})}{t - u + 1} \text{ pour } i = 1, \dots, N \quad (2.19)$$

Les coûts de stockage  $CS_{u,t} = \sum_{i=1}^N \sum_{k=u}^t (h_i \cdot I_{ik} + p_i \cdot L_{ik})$  ne dépendent que de la quantité  $X_u$  de produits désassemblés lors de la période  $u$ . En effet, nous avons :

$$I_{ik} = \max\left(0, a_i \cdot X_u - \sum_{j=u}^k d_{ij}\right); \quad L_{ik} = \max\left(0, \sum_{j=u}^k d_{ij} - a_i \cdot X_u\right) \text{ pour } k = u, \dots, t \quad (2.20)$$

Si nous notons  $CS_{u,t}(X)$  les coûts de stockage entre  $u$  et  $t$  avec  $X_u = X$ , nous pouvons noter que  $X$  est compris entre 0 et  $\max_i \left\{ \left\lfloor \frac{\sum_{k=u}^t d_{ik} - I_{i,u-1}}{a_i} \right\rfloor \right\}$ . Dans le premier cas de la limite inférieure toutes des demandes sont perdues exceptées celles satisfaites à partir  $I_{i,u-1}$  et dans le deuxième cas de la limite supérieure toutes les demandes sont satisfaites (on se ramène alors au problème présenté dans la première partie). Il n'est pas nécessaire de considérer des quantités supérieures car cela génère des solutions avec des coûts de stockage plus importants. L'heuristique étant appliquée à partir de la première période de l'horizon de planification, les valeurs de  $I_{i,u-1}$  sont connues à chaque itération.

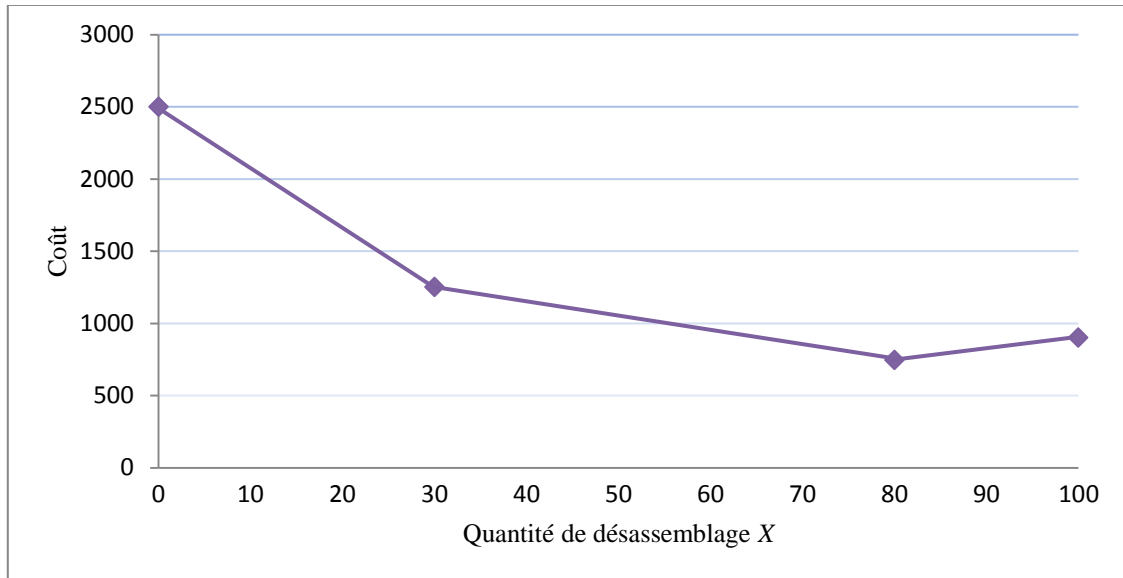
La fonction  $CS_{u,t}(X)$  est linéaire par morceaux et les changements de pentes correspondent au passage en vente perdue ( $I_{ik} = 0$  et  $L_{ik} > 0$ ) ou en stockage ( $I_{ik} > 0$  et  $L_{ik} = 0$ ) d'un composant  $i$  sur une période  $k$ . En un point de changement de pente,  $X$  est alors égal à la somme des demandes d'un composant  $i$  de la période  $u$  à une période  $k$  (c'est-à-dire toutes les demandes de  $i$  sont entièrement satisfaites de la période  $u$  à une période  $k$  et totalement perdues de la période  $k + 1$  à la période  $t$ ). Nous pouvons ainsi déterminer la quantité  $X^*$  qui minimise  $CS_{u,t}(X)$  en déterminant le couple optimal  $(i, k)$  correspond à un changement de pente. La fonction étant convexe, nous pouvons dans un premier temps déterminer toutes les valeurs possibles de  $X^*$  correspond à des points où il y a un changement de pente, puis, en ordonnant les valeurs obtenues dans un ordre croissant passé d'un point à un autre tant le coût diminue.

**Exemple de calcul de  $X$  :**

Nous supposons une structure du désassemblage d'un seul produit constitué de trois composants. Les demandes pour les composants 1,2 et 3 sont respectivement 100, 60 et 80, les rendements de désassemblage sont 1, 2 et 1 pour les composants 1, 2 et 3 respectivement. Les coûts de stockage et de ventes perdues sont  $h_i = 5$  et  $p_i = 10$  respectivement, ils sont supposés égaux pour chaque composant, le coût de setup  $s=45$ . L'horizon de planification est composé d'une seule période  $t=T=1$ .



La Figure 20 montre la variation du coût en fonction de la valeur de la quantité de désassemblage. Il s'agit d'une fonction convexe où le coût minimum est égal 745 correspond à la valeur de  $X = 80$ . La solution obtenue est un optimum local, pour calculer le coût total sur un horizon de planification composé de plusieurs périodes, nous allons répéter cette procédure à chaque période de l'horizon jusqu'à l'obtention du coût total optimal.



**Figure 20 :** Exemple de calcul de la quantité de désassemblage

*b. Least Unit Cost*

La méthode est adaptée au problème de lot sizing de la même manière que celle de Silver Meal. Son coût moyen par période est donné par la formule suivante :

$$C_{u,t} = \frac{s + \sum_{i=1}^N \sum_{k=u}^t (h_i \cdot I_{ik} + p_i \cdot L_{ik})}{\sum_{i=1}^N \sum_{k=u}^t (d_{ik})} \quad (2.21)$$

*c. Part Period Balancing (PPB)*

Par analogie avec le problème sans vente perdue, l'heuristique PPB réalise une opération de désassemblage sur la période  $u$  pour satisfaire des demandes entre  $u$  et une période  $t$  (l'opération de désassemblage suivante aura lieu à  $t + 1$ ) tant que :

$$CS_{u,t}(X^*) < s_t \quad (2.22)$$

### 2.5.1 Expérimentations

Nous considérons les mêmes instances de la section 2.3.1. Le coût de setup est supposé constant à chaque période de l'horizon  $s = 1000$ , le coût de stockage est généré suivant  $DU(1,5)$ , le coût de ventes perdues est généré suivant  $DU(10,20)$ , et les rendements de désassemblage sont générés suivant  $DU(1,4)$ . Les demandes sont générées de la même manière que celles de la section 1.3.3.1. Nous allons ensuite tester les performances des heuristiques sur différents profils de demandes : demandes croissantes, demandes décroissantes et demandes variables.

Les résultats des demandes croissantes et décroissantes ainsi que des demandes variables sont donnés dans les tableaux 17 et 18. D'après ces résultats, nous remarquons que les heuristiques donnent des solutions proches de l'optimale pour les petites instances (nous ne dépassons pas 15% par rapport à la solution optimale). Pour les moyennes instances où le nombre de composants ne dépasse pas 50, nous sommes presque à 20% par rapport à la solution optimale. Finalement, pour les grandes instances avec 100 composants, nous sommes presque à 25% par rapport à la solution optimale. En revanche, dans le cas des demandes croissantes, nous avons un gap maximal de 27%, contrairement à celui des demandes décroissantes nous avons un gap de 25%. L'accumulation de stocks est limitée par les ventes perdues, le coût de stockage est moins important à la fin de l'horizon dans le cas des demandes croissantes. Par conséquent, le coût total des demandes décroissantes est important par rapport à celui des demandes croissantes. Dans le cas des demandes variables, nous sommes presque à 28% par rapport à la solution optimale pour les grandes instances. Nous sommes très proches aux solutions obtenues dans le cas des demandes décroissantes. Par ailleurs, les performances des trois heuristiques sont presque les mêmes pour toutes les instances, nous avons une différence d'environ 3% pour les petites instances entre PPB et LUC. Pour les grandes instances, nous avons environ 4% de différence entre SM et LUC. Concernant PPB et SM, nous avons une différence presque négligeable pour toutes les instances. En ce qui concerne le temps de calcul des heuristiques, ces dernières sont très rapides par rapport au CPLEX. Nous remarquons que leurs temps d'exécution augmentent en fonction du nombre de composants et du nombre de périodes. Finalement, les heuristiques proposées dans cette étude peuvent être utilisées également dans la résolution du problème de lot sizing en désassemblage avec ventes perdues. Elles seront utiles notamment pour les petites instances où le nombre de composants ne dépasse pas 10 composants avec un horizon de planification composé de 10 périodes.

**Tableau 17 : Demandes variables**

<i>Demandes croissantes</i>															
N	T	Gaps (%)						CPU (seconde)							
		PPB		SM		LUC		Moyen			Maximal			Moyen	Maximal
		Moy	Max	Moy	Max	Moy	Max	PPB	SM	LUC	PPB	SM	LUC	CPLEX	
Petit	10	<b>3.93</b>	17.85	7.16	<b>13.46</b>	9.63	13.83	0.004	0.004	0.004	0.028	0.028	0.032	0.150	0.410
	20	<b>6.49</b>	<b>14.91</b>	8.60	21.63	7.26	23.14	0.007	0.004	0.007	0.039	0.039	0.042	0.250	0.690
	30	7.55	18.19	<b>6.85</b>	<b>17.76</b>	11.06	24.45	0.011	0.011	0.011	0.049	0.053	0.053	0.290	0.780
Moyen	10	7.38	<b>18.02</b>	<b>6.98</b>	23.11	12.32	21.73	0.011	0.007	0.014	0.042	0.039	0.042	0.180	0.620
	20	<b>6.14</b>	20.92	8.48	<b>19.43</b>	12.14	23.31	0.049	0.063	0.070	0.173	0.229	0.208	0.540	1.250
	30	<b>5.86</b>	<b>18.67</b>	7.79	21.87	11.86	25.16	0.063	0.063	0.081	0.257	0.197	0.236	0.640	2.930
Grande	10	9.65	19.47	<b>9.21</b>	21.84	8.24	<b>20.52</b>	0.039	0.035	0.035	0.166	0.176	0.204	0.600	2.900
	20	<b>8.34</b>	<b>17.10</b>	9.73	25.21	8.42	25.53	0.060	0.060	0.063	0.328	0.349	0.363	0.880	3.210
	30	<b>9.65</b>	<b>23.00</b>	9.66	23.41	12.40	24.46	0.074	0.085	0.095	0.381	0.384	0.486	0.920	3.660
<i>Demandes décroissantes</i>															
N	T	PPB		SM		LUC		Moyen			Maximal			Moyen	Maximal
		Moy	Max	Moy	Max	Moy	Max	PPB	SM	LUC	PPB	SM	LUC	CPLEX	
		Petit	10	6.36	15.79	<b>5.94</b>	<b>14.93</b>	8.68	15.34	0.004	0.004	0.004	0.028	0.028	0.028
20	<b>5.52</b>		<b>16.53</b>	10.75	23.99	8.05	25.66	0.007	0.007	0.007	0.035	0.032	0.042	0.230	0.620
30	9.48		21.84	<b>9.03</b>	<b>20.02</b>	9.26	27.11	0.011	0.011	0.011	0.049	0.053	0.053	0.260	0.710
Moyen	10	<b>8.18</b>	23.98	7.18	<b>23.63</b>	13.66	24.10	0.011	0.007	0.014	0.042	0.039	0.042	0.170	0.610
	20	6.81	<b>20.98</b>	<b>6.40</b>	22.65	10.46	25.85	0.049	0.053	0.060	0.176	0.236	0.208	0.510	1.100
	30	<b>6.50</b>	<b>20.70</b>	8.64	24.25	13.15	25.90	0.063	0.063	0.081	0.236	0.197	0.233	0.630	1.960
Grande	10	9.89	<b>24.79</b>	<b>8.78</b>	28.45	12.78	33.25	0.054	0.073	0.074	0.175	0.192	0.225	6.150	17.570
	20	<b>8.22</b>	<b>21.76</b>	11.58	25.02	11.73	29.35	0.071	0.072	0.092	0.396	0.382	0.382	14.620	30.390
	30	12.49	<b>28.01</b>	14.60	34.14	<b>12.39</b>	31.86	0.091	0.108	0.118	0.397	0.404	0.406	16.550	33.850

**Tableau 18 :** Demandes variables

<i>Demandes variables</i>															
N	T	Gaps(%)						CPU (secondes)							
		PPB		SM		LUC		Moyen			Maximal			Moyen	Maximal
		Moy	Max	Moy	Max	Moy	Max	PPB	SM	LUC	PPB	SM	LUC	CPLEX	
Petit	10	<b>4.40</b>	18.97	8.01	15.06	10.78	15.48	0.004	0.004	0.004	0.035	0.035	0.039	0.290	0.480
	20	<b>6.61</b>	21.68	7.85	<b>21.20</b>	8.12	25.89	0.007	0.007	0.007	0.049	0.053	0.049	0.390	0.690
	30	9.56	<b>16.99</b>	<b>9.14</b>	23.22	12.37	27.35	0.014	0.011	0.014	0.067	0.067	0.067	1.620	3.050
Moyen	10	8.25	<b>20.16</b>	<b>6.29</b>	25.86	13.78	24.31	0.011	0.011	0.011	0.060	0.056	0.063	1.100	2.060
	20	6.87	<b>21.17</b>	<b>6.48</b>	22.85	13.58	26.08	0.049	0.060	0.056	0.169	0.187	0.190	1.550	2.780
	30	7.56	<b>20.88</b>	<b>6.72</b>	24.47	13.27	27.13	0.060	0.060	0.070	0.201	0.190	0.222	1.880	2.970
Grande	10	<b>12.59</b>	<b>24.79</b>	12.90	28.45	14.16	33.25	0.052	0.063	0.063	0.185	0.212	0.247	6.150	17.570
	20	<b>13.59</b>	24.46	14.34	<b>22.27</b>	14.48	29.35	0.061	0.062	0.102	0.367	0.416	0.429	14.690	30.390
	30	13.84	28.01	<b>11.85</b>	<b>23.14</b>	13.78	31.86	0.082	0.117	0.135	0.409	0.393	0.436	16.510	33.800

## Conclusion

Dans ce chapitre, nous avons traité le problème de lot sizing en désassemblage à un seul produit à deux niveaux. Nous avons proposé deux modèles de programmation linéaire en nombres entiers. Le premier modèle est considéré sans ventes perdues, toutes les demandes doivent être satisfaites, aucune demande ne peut être perdue ou retardée. Nous avons montré ensuite que le problème de lot sizing en désassemblage sans ventes perdues entraîne une accumulation de stocks à la fin de l'horizon de planification. Ce surplus ne sert à satisfaire aucune demande et engendre un coût de stockage important. Nous avons ensuite identifié et calculé ce stock de surplus en développant une approche analytique permettant de calculer le surplus avant la résolution du modèle. Finalement, pour résoudre ce problème, nous avons proposé une adaptation de trois heuristiques largement connues et utilisées dans le problème de lot sizing en production (Silver Meal, Least Unit Cost et Part Period Balancing). Pour tester les performances des heuristiques, nous avons analysé l'écart d'erreur en pourcentage par rapport à la solution optimale. Nous avons analysé les performances des heuristiques sur l'impact des différents profils de demandes. Nous avons remarqué que l'heuristique de Part Period Balancing a été toujours meilleure dans la majorité des cas en termes de solution et du temps d'exécution. Le temps d'exécution des heuristiques ne dépasse pas 0.4 seconde pour toutes les instances.

Pour limiter le stock de surplus, nous avons autorisé les ventes perdues dans le deuxième modèle. L'importance de ce modèle est intéressante dans la pratique, dans le cas où les coûts sont importants par rapport aux profits, ainsi que dans des modèles contenant une contrainte de capacité, cela permet de respecter cette contrainte. Ensuite, nous avons résolu le modèle à l'aide de CPLEX dans un premier temps. Nous avons effectué plusieurs tests sur différents profils de demandes (demandes variables, demandes croissantes et demandes décroissantes). Nous avons constaté que le temps de calcul dépend du nombre de demandes et de composants puisque sur des grandes instances (50 périodes et environ 10000 composants) le temps de calcul de CPLEX dépasse 1400 secondes. Contrairement aux heuristiques, le temps de calcul ne dépasse pas 0.4 seconde pour les grandes instances.

Nous rappelons que le problème considéré dans cette étude est un problème de lot sizing en désassemblage à un seul produit et deux niveaux. Ce problème a été résolu avec une méthode approchée en temps polynomial. Cependant, le problème devient compliqué dans le cas de plusieurs produits avec contraintes de capacité. Dans ce cas, il faut proposer une méthode approchée pour résoudre le problème dans un temps raisonnable pour toutes les instances. Le cas de plusieurs produits et contraintes de capacité sera traité dans le prochain chapitre. Ce chapitre a fait objet d'un article de conférence internationale.

**M., Hrouga,** M. Godichaud, L. Amodeo. Disassembly lot-sizing problem: Detection and quantification of surplus. *5th International conference on Information Systems, Logistics and Supply Chain (ILS 2014)*, August 2014, Breda, The Netherlands.



# **Chapitre 3 : Problème de lot sizing en désassemblage de plusieurs produits à deux niveaux**

Le problème de planification du désassemblage de plusieurs produits avec contraintes de capacité est traité dans ce chapitre. Cette contrainte vise à limiter la quantité de produits pouvant être désassemblée sur chaque période de l'horizon de planification. Comme pour le chapitre précédent, chaque produit est constitué de plusieurs composants. Une solution du problème détermine les quantités de produits en fin de vie à désassembler sur chaque période de l'horizon de planification en vue de satisfaire des demandes pour leurs composants. Une solution est évaluée en fonction des coûts fixes de désassemblage (setup), des coûts de stockage et, dans une deuxième partie, des coûts de ventes perdues.

Deux problèmes sont abordés : avec et sans ventes perdues. Dans le premier problème, toutes les demandes doivent être satisfaites (aucune rupture sur la demande n'est acceptée). L'objectif est de minimiser le coût de setup et le coût de stockage. Nous montrons que le modèle sans ventes perdues peut être résolu avec une méthode exacte pour les petites instances, mais le temps de calcul croît en fonction de l'augmentation de la taille des données d'entrées. Pour les grandes instances, plusieurs méthodes sont proposées. Un algorithme génétique est tout d'abord développé. Il permet de résoudre le problème pour toutes les instances dans un temps de calcul raisonnable. Ensuite nous proposons une heuristique Fix-and-Optimize. Finalement, nous intégrons l'heuristique Fix-and-Optimize dans l'algorithme génétique en proposant de sorte une approche hybride. Les résultats des tests sont comparés avec les bornes inférieures obtenues par une méthode de relaxation lagrangienne proposée dans la littérature.

Dans le deuxième problème, les ventes perdues sont autorisées et les demandes ne sont pas forcément satisfaites. En effet, les demandes qui ne sont pas satisfaites entraînent un coût correspondant à la perte. L'objectif de ce modèle est de minimiser le coût de setup, le coût de stockage et le coût de ventes perdues. Comme dans la première partie, ce modèle est également résolu à l'aide d'un algorithme génétique et une approche hybride. Enfin, dans la dernière partie de ce chapitre, nous analysons les résultats des tests.



# 1 Problème de lot sizing en désassemblage sans ventes perdues

Comme dans le chapitre précédent, les produits sont structurés en deux niveaux. Le premier niveau représente le produit qui doit être désassemblé et le deuxième niveau représente les composants qui sont obtenus directement par l'opération de désassemblage. L'horizon de planification est constitué de  $T$  périodes et  $R$  produits différents peuvent être désassemblés pour satisfaire des demandes de  $N$  composants. L'objectif est de minimiser le coût total sur l'horizon de planification qui est constitué des coûts de setup relatifs aux produits en fin de vie et des coûts de stockage relatifs aux composants. Le modèle est considéré avec contrainte de capacité stricte et avec pénalité du temps de dépassement. Le coût de pénalité du temps de dépassement correspond au dépassement du temps de désassemblage. La restriction de capacité des ressources implique qu'il existe une restriction sur les ressources et les capacités de la main d'œuvre, de l'équipement, des machines, des budgets etc. La quantité totale du produit à désassembler sur chaque période est limitée par les demandes des composants. Ce problème peut être formulé comme une programmation linéaire en nombres entiers.

## 1.1 Modélisation du problème

Il s'agit d'une extension du modèle de programmation linéaire du chapitre 2 précédent en considérant plusieurs produits (chacun ayant toujours une nomenclature à deux niveaux) et un ensemble de contrainte de capacité pour chaque période. En liant les produits entre eux, la contrainte de capacité rend le problème plus difficile à résoudre (Florian et al. 1980).

Afin de décrire le modèle mathématique, nous considérons les hypothèses suivantes :

- Les demandes de composants sont données et déterministes ;
- Toutes les demandes doivent être satisfaites sur chaque période ;
- Tous les composants obtenus sont supposés dans un état permettant de satisfaire la demande ;
- Il n'y a pas de stockage de produits en fin de vie, les produits peuvent être obtenus une fois qu'ils sont demandés ;
- Les temps d'opération de désassemblage sont donnés et déterministes ;
- Les paramètres de coûts sont considérés constants.

Pour faciliter la compréhension de ce modèle, les notations ci-dessous seront utilisées dans tout le reste de ce chapitre.

### *Indices*

$i$  : indices pour les composants  $i = 1, \dots, N$

$t$  : indices pour les périodes  $t = 1, \dots, T$

$r$  : indices pour les produits  $r = 1, \dots, R$

### *Paramètres*

$a_{ri}$  : rendement de désassemblage du composant  $i$  par rapport au produit  $r$

$d_{it}$  : demande pour le composant  $i$  à la période  $t$

$h_i$  : coût de stockage du composant  $i$

$I_{i0}$  : niveau de stock initial du composant  $i$

$s_{rt}$  : coût de setup du produit  $r$  à la période  $t$

$g_r$  : temps de désassemblage pour une unité du produit  $r$

$C_t$  : capacité disponible à la période  $t$

$u$  : coût de pénalité de dépassement du temps de désassemblage

$M$  : nombre entier assez grand

### *Variables de décisions*

$I_{it}$  : niveau de stock du composant  $i$  à la période  $t$

$X_{rt}$  : quantité de désassemblage du produit  $r$  à la période  $t$

$O_t$  : temps supplémentaire de désassemblage à la période  $t$

$Y_{rt}$  : variable binaire égale à 1 si il y a un désassemblage du produit  $r$  à la période  $t$ , et 0 sinon

Le modèle avec seulement contrainte de capacité peut être donnée comme suit :

$$\text{Min} \sum_{r=1}^R \sum_{t=1}^T s_{rt} \cdot Y_{rt} + \sum_{t=1}^T \sum_{i=1}^N (h_i \cdot I_{it})$$

Sous contraintes :

$$I_{it} = I_{i,t-1} + a_{ri} \cdot X_{rt} - d_{i,t} \text{ pour } i = 1, \dots, N; r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.1)$$

$$X_{rt} \leq M * Y_{rt} \text{ pour } r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.2)$$

$$\sum_{r=1}^R g_r * X_{rt} \leq C_t \text{ pour } t = 1, \dots, T \quad (3.3)$$

$$I_{it}, X_{rt}, O_t \geq 0 \text{ et } X_{rt}, I_{it} \in \mathbb{N}, \text{ pour } i = 1, \dots, N; r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.4)$$

$$Y_{rt} \in \{0,1\} \text{ pour } r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.5)$$

Le modèle avec contraintes de capacité et pénalité du temps de dépassement est donné comme suit :

$$\text{Min} \sum_{r=1}^R \sum_{t=1}^T s_{rt} \cdot Y_{rt} + \sum_{t=1}^T \sum_{i=1}^N (h_i \cdot I_{it}) + \sum_{t=1}^T u \cdot O_t$$

Sous contraintes :

(3.1), (3.2), (3.3), (3.4) et (3.5)

$$\sum_{r=1}^R g_r * X_{rt} - O_t \leq C_t \text{ pour } t = 1, \dots, T \quad (3.6)$$

La fonction objectif minimise la somme des coûts de setup et de stockage ainsi que le coût de dépassement du temps de désassemblage. En effet, le coût de setup correspond au désassemblage du produit, et le coût de stockage correspond au stockage de composants. La contrainte (3.1) représente la conservation des flux de stockage pour les produits en fin de vie et les de composants respectivement. Le niveau de stock d'un composant à la fin de la période  $t$  est égal à la somme de niveau de stock à la période  $t - 1$  plus les entrées moins les sorties à la fin de la période  $t$ . Les entrées sont la quantité générée par le désassemblage, et les sorties sont les demandes. La contrainte (3.2) assure le lancement de désassemblage où la quantité produite ne doit pas dépasser le paramètre  $M$  ( $M$  est une valeur assez grande) qui peut représenter la somme maximale des demandes d'un composant sur l'horizon de planification. La contrainte (3.3) représente la contrainte de capacité. La contrainte (3.4) signifie que les variables  $X_{rt}$ ,  $O_t$  et  $I_{it}$  sont discrètes non négatives. La contrainte (3.5), représente une variable binaire qui est égale à 1 si on désassemble à la période  $t$  et 0 sinon. Finalement, la contrainte (3.8) représente la capacité disponible à la période  $t$ .

Le problème de lot sizing en désassemblage avec contraintes de capacité est considéré comme un problème complexe. En effet, la contrainte de capacité engendre un lien entre les produits. Il est possible de résoudre le problème ci-dessus de manière exacte à l'aide d'un logiciel de programmation linéaire en nombres entiers (CPLEX, Solveur ou autres) pour des petites instances. Pour les grandes instances, la résolution nécessite un temps de calcul très élevé. Il est à noter également que depuis les travaux de (Wagner et Whitin 1958) et (Manne 1958), plusieurs recherches ont été menées sur les problèmes de lot sizing avec contraintes de capacité. Ce dernier est l'un des problèmes les plus étudiés en planification de la production. Sa complexité est principalement due à deux raisons. La première est que les variables de setup sont des variables binaires, et le problème est ainsi un problème d'optimisation combinatoire. La deuxième raison de sa complexité liée à la contrainte de capacité qui lit les différents produits. Si la contrainte de capacité n'est pas considérée, le problème de lot sizing peut se réduire à un ensemble des problèmes de lot sizing à un seul produit sans capacités où chaque sous-problème peut être résolu en  $O(T \log T)$ . En outre, le problème de lot sizing à un seul produit a été prouvé NP-difficile dans (Florian et al. 1980). Par contre, par la proposition d'un algorithme pseudo-polynomial, ils ont prouvé que ce dernier n'est pas NP-difficile au sens fort. Ainsi que (Chen et Thizy 1990) ont prouvé que le problème à plusieurs produits est NP-difficile au sens fort. Dans ces circonstances, les méthodes exactes ne sont pas les meilleures techniques de résolution pour le problème traité dans cette étude. En effet, le temps pour résoudre le problème constitue un facteur important dans le processus de décision des gestionnaires d'entreprises. C'est pour ces raisons qu'il est important d'utiliser une méthode approchée (heuristiques ou méta-heuristiques) pour résoudre ce problème.

Dans les problèmes d'optimisation, les algorithmes génétiques ou les méta-heuristiques d'une manière générale sont souvent utilisées pour résoudre des problèmes difficiles. Elles permettent généralement d'obtenir une solution de très bonne qualité pour des problèmes de grande taille. En effet, le rapport entre le temps d'exécution et la qualité de la solution trouvée par une méta-heuristique reste dans la majorité des cas très intéressant par rapport aux différentes approches de résolution. En plus, une méta-heuristique peut être adaptée pour différents types de problèmes, tandis qu'une heuristique est utilisée à un problème donné. Un des enjeux de la conception des méta-heuristiques est de faciliter le choix d'une méthode et le réglage des paramètres pour les adapter à un problème donné. Les méta-heuristiques peuvent être classées de nombreuses façons. Il y a notamment celles qui travaillent avec une population de solutions et d'autres qui ne manipulent qu'une seule solution à la

fois. Parmi les approches méta-heuristiques nécessitant une population initiale figure l'algorithme génétique. En effet, les algorithmes génétiques tiennent une place non négligeable dans la résolution du problème de lot sizing (Goren et al. 2010). Dans le cadre de cette étude, nous utilisons dans un premier temps un algorithme génétique pour résoudre le problème étudié. Ensuite, pour améliorer la solution trouvée par l'algorithme génétique, nous intégrons une heuristique Fix-and-Optimize dans l'algorithme génétique en proposant une approche hybride.

Dans le cadre de cette étude, nous proposons un algorithme génétique qui nous permettra de résoudre le problème dans un délai raisonnable pour toutes les instances. Ensuite, nous proposons une heuristique Fix-and-Optimize, nous utilisons ainsi plusieurs stratégies de décomposition (stratégie à un seul produit et plusieurs périodes, plusieurs produits/plusieurs périodes, stratégie un seul produit/horizon glissant et stratégie plusieurs produits/horizon glissant). Finalement, nous intégrons l'heuristique Fix-and-optimize dans l'algorithme génétique en proposant une approche hybride. D'ailleurs, les algorithmes génétiques parviennent à trouver de bonnes solutions sur des problèmes très complexes, et très éloignés des problèmes combinatoires classiques pour qu'on puisse tirer un profit de certaines propriétés connues. Ils permettent également de déterminer entre deux solutions quelle est la meilleure afin d'opérer leurs sélections. En raison de ces avantages, l'algorithme génétique a été employé avec succès pour résoudre des problèmes difficiles combinatoires dans plusieurs domaines, y compris les problèmes de lot sizing. Le lecteur peut se référer à (Goren et al. 2010) pour d'autres études utilisant des algorithmes génétiques pour la résolution de ce problème. En outre, dans cette étude, l'approche hybride combine l'algorithme génétique avec une heuristique appelée Fix-and-Optimize. La principale motivation de cette hybridation est d'employer l'heuristique Fix-and-Optimize comme une stratégie de diversification de sorte que l'algorithme génétique peut être guidé vers de nouveaux domaines de recherches, en particulier dans un grand espace de solutions et donc les minimums locaux peuvent être surmontés.

## **1.2 Méthodes de résolution**

Le problème considéré dans cette étude peut être résolu d'une manière exacte à l'aide d'un logiciel d'optimisation pour des petites instances. En appliquant une relaxation lagrangienne de la contrainte de capacité, le problème peut aussi être décomposé en sous problème à un seul produit pour obtenir une borne inférieure des solutions. Cette solution sera une base de référence pour évaluer la qualité des solutions obtenues par les autres approches de résolution.

Dans le cas où le rendement de désassemblage est égal à 1, le problème de lot sizing en désassemblage devient similaire à celui de lot sizing en production. Puisque le problème de lot sizing à plusieurs produits avec contraintes de capacité est NP-difficile (Chen et Thizy 1990). Ainsi que, (Kim et Xirouchakis 2010) ont montré que le problème de lot sizing en désassemblage à deux niveaux, plusieurs produits avec contraintes de capacité est NP-difficile. Donc par analogie, le problème considéré dans cette étude est NP-difficile.

### **1.2.1 Relaxation lagrangienne**

La relaxation lagrangienne permet de trouver une borne inférieure à notre problème en relâchant la contrainte de capacité et en pénalisant sa violation dans la fonction objectif. (Kim et Xirouchakis 2010) ont proposé une relaxation lagrangienne en utilisant une approche de programmation dynamique pour résoudre les sous-problèmes relâchés. Pour obtenir une solution faisable, les auteurs ont proposées plusieurs heuristiques permettant de déplacer des quantités de période où la capacité est

dépassée vers des périodes où il reste une marge. Cette approche a tendance à rajouter des périodes de setup sur l'horizon de planification. Plus l'écart entre les coûts de setup et les coûts de stockage sont importants plus son efficacité diminue. Elle n'est aussi pas adaptable au problème avec composants communs au niveau de la méthode de résolution des sous-problèmes relâchés.

### 1.2.2 Méta-heuristique : algorithme génétique

Les algorithmes génétiques (AGs) sont des algorithmes d'optimisation incertaine fondés sur les mécanismes de la sélection naturelle et de la génétique. En partant d'une population de solutions potentielles (chromosomes) initiales arbitrairement choisies, de nouvelles populations de solutions potentielles en utilisant des opérateurs évolutionnaires telles que la sélection, le croisement et la mutation. Un algorithme génétique va faire évoluer une population afin d'améliorer les individus en fonction de leur performance (fitness). Dans cette partie, nous adaptons l'algorithme génétique pour la résolution du problème étudié. Ces adaptations portent sur le codage des solutions, la génération de la solution initiale, les opérateurs de croisement et de mutation et la gestion des contraintes.

#### 1.2.2.1 Codage de solution et évaluation

Chaque individu de la population contient les valeurs des variables  $X_{rt}$  pour  $r = 1, \dots, R$  et  $t = 1, \dots, T$ . Une solution du problème est un plan de désassemblage qui est entièrement caractérisée par ces variables définissant les lots de désassemblage de chaque produit sur chaque période. Les autres variables permettent de modéliser les différentes composantes de coûts dans la fonction à minimiser et de spécifier les contraintes.

Elles peuvent cependant toutes être déterminées à partir des variables  $X_{rt}$ . Une variable  $Y_{rt}$  est égale à 1 si  $X_{rt} > 0$ , à 0 si  $X_{rt} = 0$  pour respecter la contrainte (3.2) calculer les coûts fixes de désassemblage sur tout l'horizon de planification et pour chaque produit. Chaque variable  $I_{it}$  est déterminée par la contrainte de conservation des flux (3.1) en connaissant le stock initial  $I_{i0}$  et la demande  $d_{it}$  (en commençant par  $t = 1$  jusqu'à  $t = T$ ) ou directement par  $I_{it} = I_{i0} + \sum_{r=1}^R \sum_{k=1}^t a_{ri} \cdot X_{rk} - \sum_{k=1}^t d_{ik}$ . Les coûts de stockage sur tout l'horizon de planification pour chaque composant sont ainsi déterminés.

#### 1.2.2.2 Solution initiale

La recherche de solutions est initialisée à partir d'une première population dont les solutions sont générées selon différentes méthodes. Les méthodes retenues sont simples à implémenter, elles génèrent une solution rapidement ou génèrent de bonnes solutions pour des sous-problèmes (par exemple sans contraintes de capacité). L'objectif est de couvrir au mieux l'espace des solutions possibles du problème général. Ces méthodes ne fournissent pas nécessairement des solutions respectant la contrainte de capacité. Après avoir générer une solution, selon le mode de gestion de la contrainte de capacité, soit une méthode de réparation, présentée par la suite est appliquée, soit une pénalisation est ajoutée à la fonction objectif à minimiser de manière à privilégier les solutions respectant la contrainte.

La principale méthode de génération de l'algorithme proposé ici est que la population initiale est aléatoire. Chaque solution est construite en trois étapes. La première étape consiste à générer aléatoirement les périodes de désassemblage (période où  $X_{rt} > 0$  et  $Y_{rt} = 1$ ). La deuxième étape permet de calculer pour ces périodes la quantité de produit à désassembler, valeur exacte d'une

variable  $X_{rt}$ , est déterminée de manière à satisfaire le besoin (différence entre la demande et le stock à la fin de la période précédente) maximal parmi tous les composants du produit considéré jusqu'à la période de désassemblage suivante. La troisième étape consiste, après avoir toutes les quantités  $X_{rt}$  de la solution, à déterminer pour chaque période les dépassements de capacité et d'appliquer une méthode de réparation pour la gestion de la contrainte de capacité avec réparation.

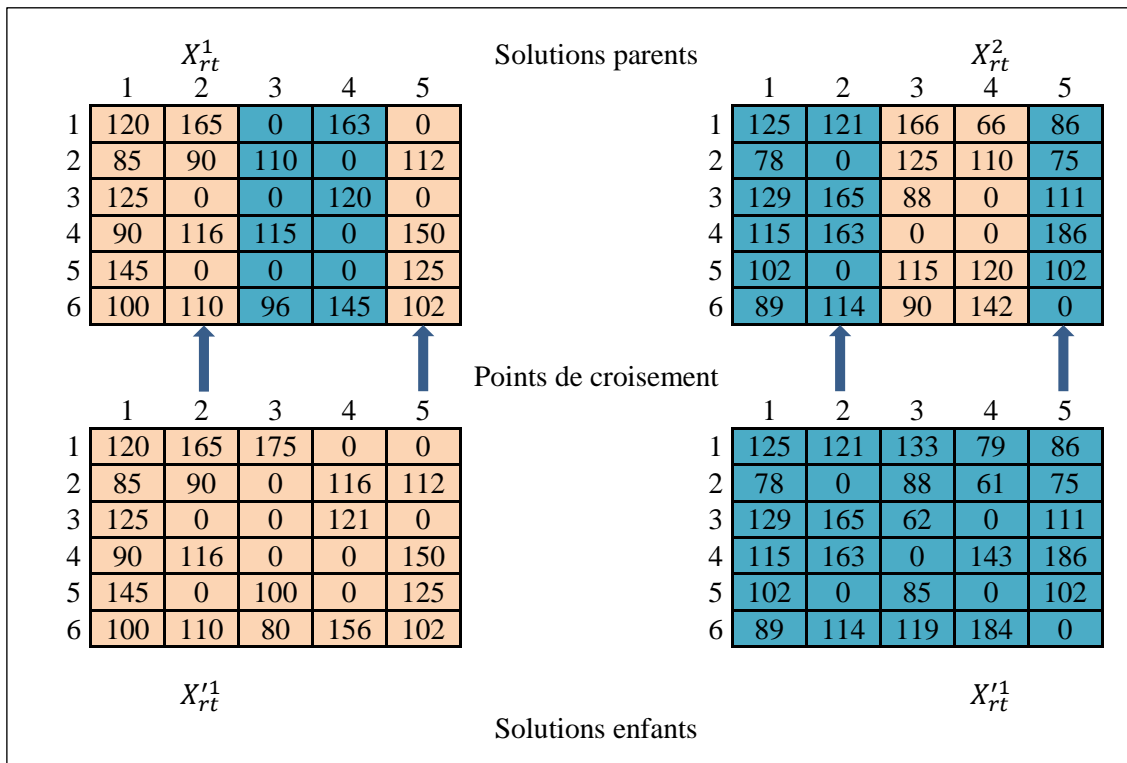
Plusieurs autres méthodes sont aussi utilisées pour générer chacune une solution particulière. Après avoir généré la solution, ces méthodes utilisent la même gestion de la contrainte de capacité que la méthode aléatoire.

Une solution de la population initiale est générée à partir d'une méthode « lot pour lot » où le désassemblage a lieu sur chaque période. Les quantités sont déterminées à partir du besoin maximal (le maximum de la division de demande par le rendement de désassemblage) parmi tous les composants du produit considéré sur la période considérée. Une autre solution est générée à partir de l'optimisation de la fonction objectif à minimiser de chaque produit séparément (sans considérer la contrainte de capacité).

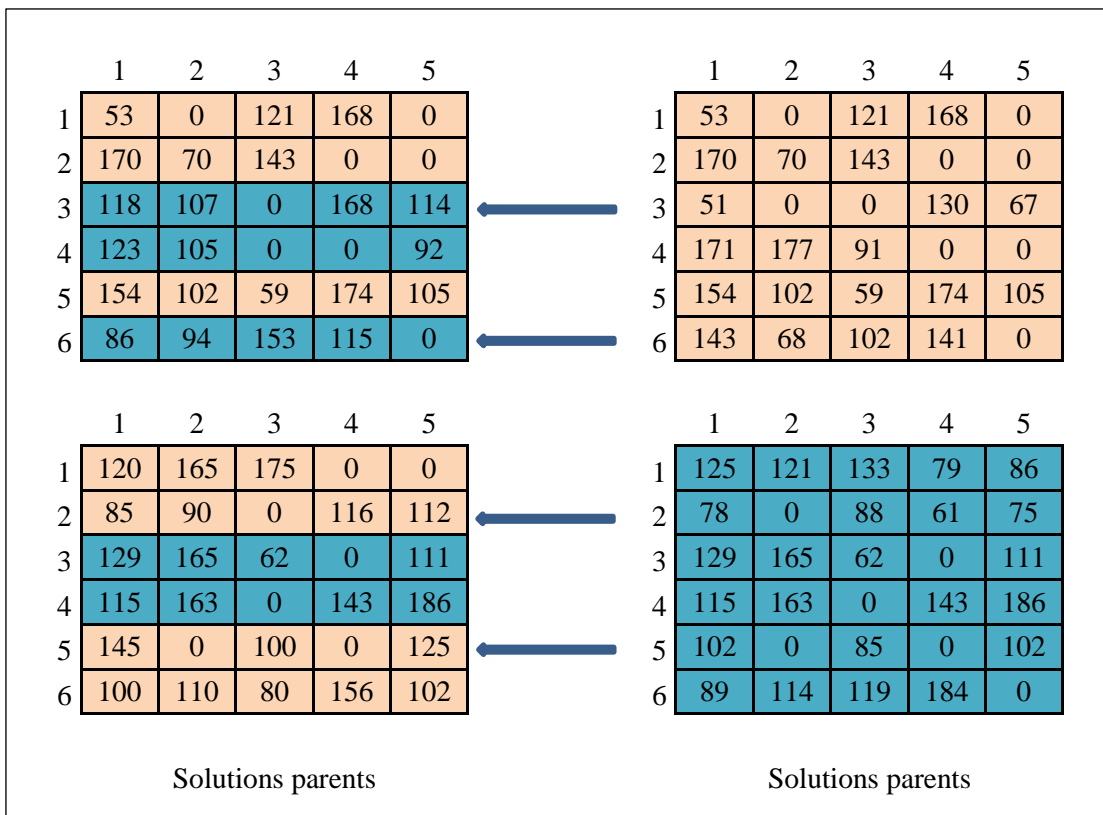
Trois solutions sont générées à partir des heuristiques « Part Periode Balancing », « Silver Meal » et « Least Unit Cost » appliquées à chaque produit. Nous nous rappelons que ce sont des méthodes dites myopes car nous cherchons qu'à positionner les périodes de désassemblage les unes à la suite des autres sans considérer à chaque fois les décisions futures. Elles se différencient pour calculer le nombre de périodes couvertes par une période de désassemblage donnée. Ces heuristiques ont été développées dans le chapitre 2. Elles sont construites de manière à déterminer les tailles de lots en équilibrant entre le coût de setup et le coût de stockage.

### 1.2.2.3 Méthode de croisement

Les méthodes de croisement dans un algorithme génétique permettent de générer un ou plusieurs solution individus, nommés « enfants », à partir de plusieurs individus, nommés « parents », de la population courante sélectionnée en fonction de la valeur de la fonction objectif à minimiser de la solution qu'ils représentent. Des méthodes avec un ou plusieurs points de croisement sont utilisées dans l'algorithme proposé avec l'utilisation de deux individus parents pour créer deux individus enfants. Les individus étant mis sous forme de tableau à deux dimensions avec comme entrées les produits  $r = 1, \dots, R$  et les périodes  $t = 1, \dots, T$  et comme valeurs les variables  $X_{rt}$ , un ou plusieurs points de croisement sont positionnés sur des couples  $(r, t)$ . Entre deux points de croisement, un individu enfant est constitué exclusivement des valeurs d'un des deux individus parents. Trois approches de croisement sont utilisées. La première est un croisement sur les périodes dans laquelle deux périodes  $t_1$  et  $t_2$  sont sélectionnées comme points de croisement. Entre les périodes 0 et  $t_1$  et entre  $t_2$  et  $T$ , les valeurs d'un individu enfant sont issues du deuxième individu parent pour tous les  $r = 1, \dots, R$ . Une deuxième approche de croisement sur les produits reprend le même principe mais en sélectionnant deux produits  $r_1$  et  $r_2$  comme points de croisement. Entre les produits 0 et  $r_1$  et entre  $r_2$  et  $R$ , les valeurs d'un individu enfant sont issues d'un premier individu parent pour toutes les périodes  $t = 1, \dots, T$ . Dans la troisième approche de croisement, deux couples  $(r_1, t_1)$  et  $(r_2, t_2)$  sont choisis aléatoirement. Un individu enfant est constitué des valeurs d'un premier individu parent pour toutes les périodes  $t = 1, \dots, T$  des produits  $r = 1, \dots, r_1 - 1$  et  $r = r_2 + 1, \dots, R$ , les périodes  $t = 1, \dots, t_1 - 1$  du produit  $r_1$  et les périodes  $t = t_2 + 1, \dots, T$  du produit  $r_2$ . Les autres valeurs sont issues du deuxième individu parent. Cette méthode de croisement peut être présentée par les Figures 21 et 22.



**Figure 21** : Exemple de croisement sur les périodes



**Figure 22** : Exemple de croisement sur les produits

#### 1.2.2.4 Méthode de mutation et/ou de recherche locale

Les méthodes de mutation génèrent un nouvel individu en modifiant une ou plusieurs variables d'un individu de la population courante Figure 23. Nous obtenons ainsi dans le voisinage de la solution courante mutée pour tenter l'améliorer. Les méthodes de mutation peuvent aussi permettre de diversifier ou perturber des solutions en appliquant plusieurs fois la mutation sur un même individu et en acceptant la dégradation de la solution de départ. En partant d'un couple produit-période  $(r, t)$ , les méthodes utilisées dans l'algorithme proposé consiste à augmenter ou diminuer la quantité  $X_{rt}$  en déplaçant une quantité  $\Delta$  depuis ou vers une période  $u > t$  (en diminuant ou en augmentant une quantité  $X_{rt}$ ). Nous imposons que les contraintes doivent être respectées par la solution générée alors la quantité maximale  $\Delta_{max}$  qui peut être déplacée est limitée sinon il faut appliquer les méthodes de réparation des contraintes présentée par la suite après avoir effectué la mutation.

L'augmentation d'une variable  $X_{rt}$  se fait en diminuant la  $X_{ru}$  où  $u$  est la période de désassemblage de  $r$  suivant directement  $t$ . La valeur maximale de la quantité de  $\Delta$  dépend de la capacité non utilisée sur la période  $t$  :

$$\Delta_{max} = \min \left\{ X_{ru}; C_t - \left[ \sum_{j=1}^R g_j \cdot X_{jt} \right] \right\} \quad (3.7)$$

Deux variantes de la mutation par augmentation sont utilisées. En effet,  $\Delta$  peut être fixée aléatoirement entre 1 et  $\Delta_{max}$  ou directement à  $\Delta_{max}$  lorsque nous cherchons à enlever un coût fixe de désassemblage (à la période  $u$ ).

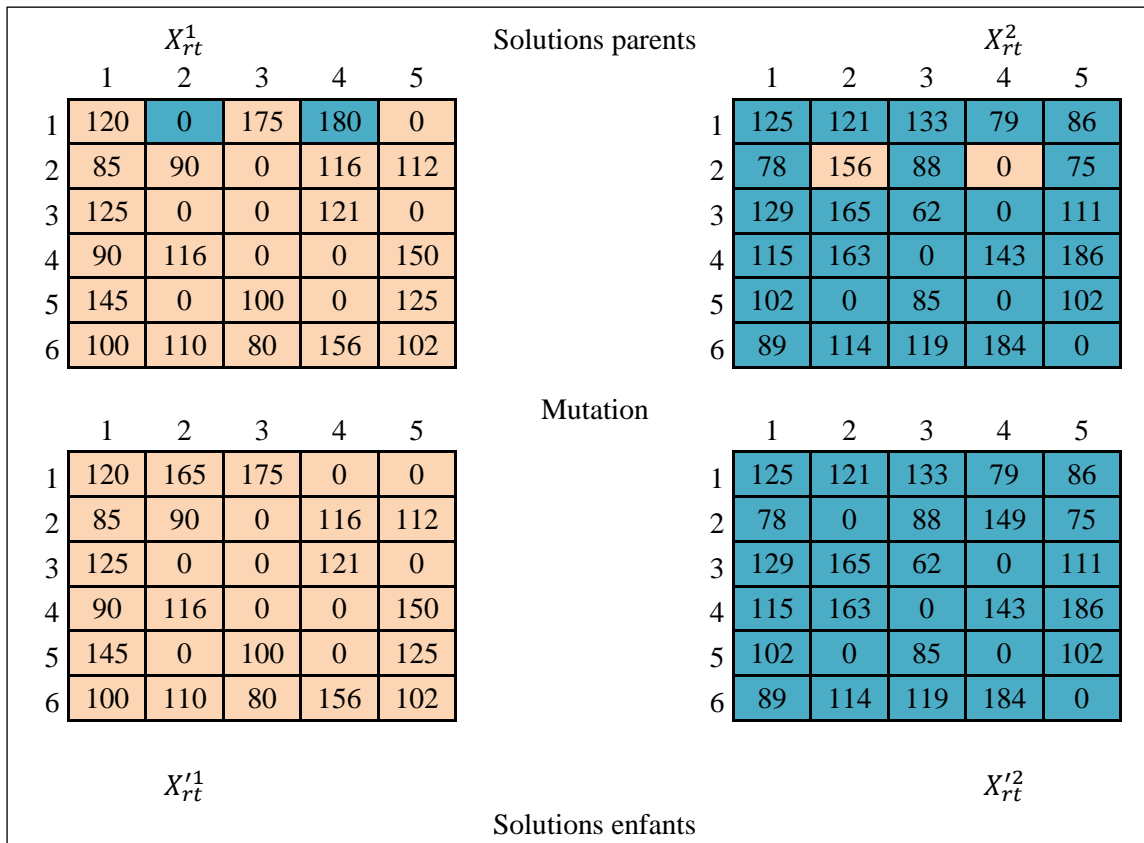
La diminution d'une variable  $X_{rt}$  se déplaçant une quantité  $\Delta$  depuis la période  $t$  vers la période  $t + 1$ . La valeur maximale de la quantité  $\Delta$  dépend de la capacité non utilisée sur la période  $t + 1$  et la demande doit être satisfaite à la période  $t$  :

$$\Delta_{max} = \min \left\{ X_{rt}; C_{t+1} - \left[ \sum_{j=1}^R g_j \cdot X_{jt} \right]; \min_{i/a_{ri} > 0} \left\{ \left\lfloor \frac{I_{it}}{a_{ri}} \right\rfloor \right\} \right\} \quad (3.8)$$

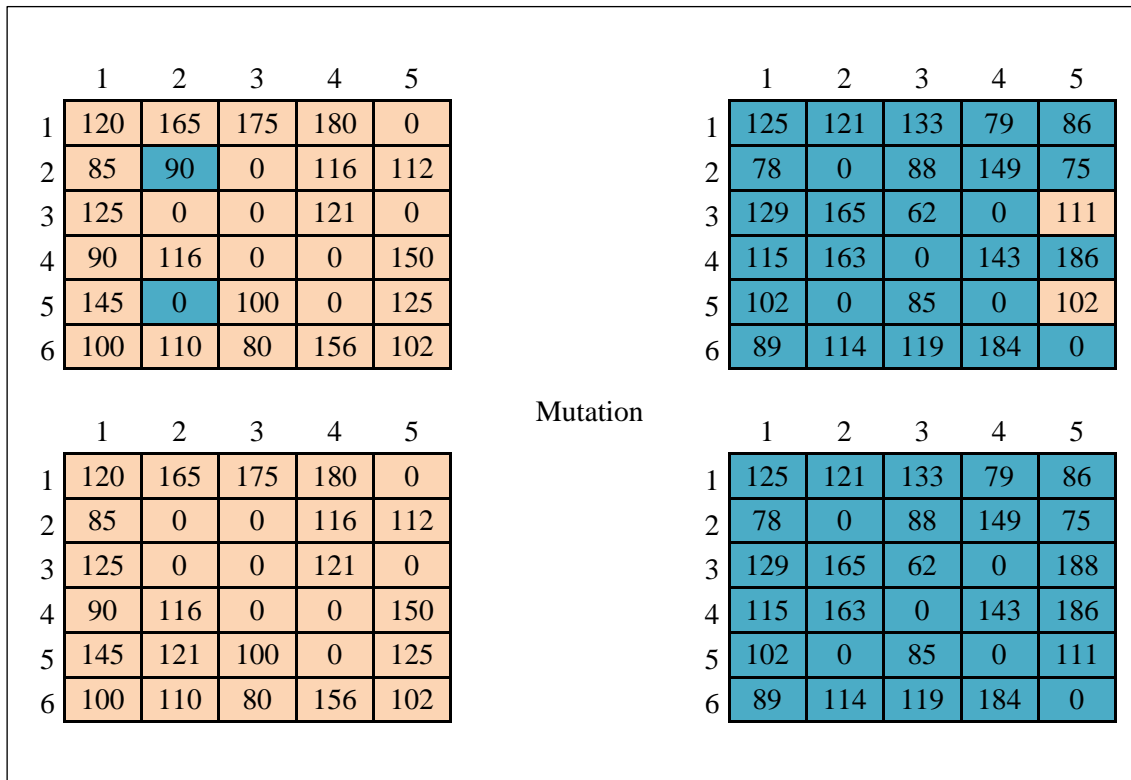
Comme pour l'augmentation, deux variantes de la mutation par diminution sont utilisées : aléatoire ou complète pour chercher à enlever un coût fixe de désassemblage en  $t$ .

La sélection d'un couple produit-période  $(r, t)$  servant de support à la modification qui est réalisée aléatoirement. Une ou plusieurs variables d'une même solution peuvent être modifiées au cours de mutation. Le nombre de modification étant fixé aléatoirement selon une répartition uniforme. Une approche selon une recherche locale par itération est également utilisée. Elle consiste à chaque itération à sélectionner une couple  $(r, t)$  aléatoirement et à arrêter la recherche lorsqu'il y a plus d'amélioration au bout d'un nombre fixé d'itération.





**Figure 23 :** Exemple de mutation sur les périodes



**Figure 24 :** Exemple de mutation sur les produits

### 1.2.2.5 Gestion et méthode de réparation des contraintes

L'application des différentes méthodes présentées avant (initialisation, croisement et mutation) peut générer dans un premier des temps des solutions infaisables par rapport à la contrainte de capacité. La réparation de la solution consiste à modifier les valeurs de variables  $X_{rt}$  de manière à obtenir des solutions faisables. Deux approches de gestion de la violation de cette contrainte sont comparées. La première consiste à ajouter à la fonction évaluant le coût d'une solution une pénalisation proportionnelle au dépassement de la contrainte de capacité. L'avantage de cette première approche est qu'elle est simple à mettre en œuvre et consomme moins de temps de calcul qu'une méthode de réparation. La deuxième approche consiste à réparer la solution en modifiant les valeurs de variables  $X_{rt}$  de manière à obtenir des solutions faisables. La méthode retenue dans l'algorithme proposé, qui est aussi la plus couramment employée dans les divers problèmes de lot sizing, est similaire à celle présentée dans (Kim 2005) et dont le principe est rappelé ici. La réparation consiste en une succession de phases vers l'arrière c'est-à-dire de la dernière période vers la première, et vers l'avant, c'est-à-dire de la première vers la dernière. Pour chaque période  $t$  où la contrainte de capacité est dépassée dans une phase vers l'arrière, des mouvements de quantité sont réalisés de la période  $t$  vers la période  $t - 1$  : pour un produit donné  $r$ , une quantité  $\Delta$  est retirée de la variable  $X_{rt}$  et ajoutée à la variable  $X_{r,t+1}$ . Le coût généré par un mouvement sur chaque période est évalué et le composant générant le coût minimum est sélectionné. L'opération est répétée tant que la contrainte de capacité est dépassée sur la période  $t$ . Dans une phase vers l'avant les mouvements sont réalisés d'une période  $t$ , où la contrainte de capacité est dépassée, vers la période  $t + 1$ . Dans l'algorithme génétique proposé ici, une phase vers l'arrière et une phase vers l'avant sont successivement appliquées pour réparer une solution non faisable.

Après la réalisation d'un croisement aléatoire sur les périodes, la contrainte de conservation des flux peut être violée (la quantité de composant reçu et en stock sur une période ne permet pas de satisfaire la demande). La méthode de réparation appliquée consiste dans une première phase à vérifier période par période de la première période jusqu'à la dernière si toutes les demandes sont satisfaites et, si ce n'est pas le cas, à augmenter la quantité  $X_{rt}$  de la période  $t$  en cours de vérification et pour le produit  $r$  contenant le composant pour lequel la contrainte est violée. À la fin de cette première phase, la quantité totale désassemblée sur tout l'horizon de planification d'un produit peut être supérieure au besoin maximal total (demande cumulée sur le rendement de désassemblage) parmi ces composants. Dans ce cas le surplus est retiré jusqu'à élimination en partant de la dernière période.

### 1.2.2.6 Gestion de la survie

Les meilleurs individus entre la population créée sont gardés pour la génération suivante en respectant la taille de la population. Cette approche permet de converger rapidement vers de bonne solution mais peut générer une perte de diversité de la population. Chaque fois la diversité de la population n'est plus satisfaisante une méthode de perturbation est appliquée sur plusieurs individus en acceptant une dégradation de leur fonction d'évaluation. Cette méthode de perturbation consiste à appliquer la méthode de mutation à plusieurs reprises en acceptant une dégradation de la solution.

### 1.2.3 Heuristique Fix-and-Optimize (FO)

Les heuristiques de décomposition consistent à résoudre successivement des sous-problèmes au moyen des techniques de résolution exactes des programmes linéaires en nombre entiers (en tirant notamment l'avantage des logiciels de résolution tel que CPLEX). Un sous-problème est défini par la séparation de l'ensemble des variables du problème général en deux sous-ensembles. Le premier sous-ensemble est constitué de variables à optimiser (selon la fonction à minimiser et les contraintes du problème général) et le deuxième sous-ensemble est constitué des autres variables dont les valeurs sont fixées en ajoutant des contraintes dans le problème général. Le sous-problème est donc modélisé par un programme linéaire identique à celui modélisant le problème général avec les contraintes supplémentaires permettant de fixer les variables du deuxième sous-ensemble. L'heuristique fonctionne par itération en modifiant à chaque itération le sous ensemble des variables fixées et en intégrant les valeurs des variables obtenues à l'itération précédente. Il est donc nécessaire de générer une solution initiale pour lancer l'heuristique.

L'heuristique Fix-and-Optimize a été largement utilisée dans la littérature pour résoudre le problème de lot sizing. (Sahling et al. 2009) ont tout d'abord proposé une FO pour résoudre le problème de lot sizing à plusieurs niveaux avec contraintes de capacité. Ils ont présenté trois principes pour déterminer les sous-problèmes : la décomposition du produit, la décomposition des ressources et la décomposition du processus. Dans (Helber et Sahling 2010), les auteurs ont appliqué à nouveau la même approche FO pour le cas de plusieurs niveaux avec un délai d'exécution constant. Ils ont montré que pour obtenir une bonne solution, il faut fixer plusieurs variables binaires dans l'algorithme. (Goren et al. 2012) ont également utilisé une heuristique FO pour résoudre le problème avec contraintes de capacité, ils ont ensuite intégré une Fix-and-Optimize dans l'algorithme génétique. Contrairement à (Sahlin et al. 2009) et (Helber et Sahlin 2010) ont fixé toutes les variables de setup pour construire une solution initiale, ils ont ensuite comparé leurs résultats avec (Gopalakrishnan et al. 2001) en utilisant un algorithme génétique.

Le développement d'une heuristique de décomposition pour le problème avec contraintes de capacité est présenté dans cette partie. Pour cela, il faut définir des approches pour générer une solution initiale, définir le sous-problème résolu à chaque itération et le critère d'arrêt des itérations. Les variables fixées dans notre problème de lot sizing en désassemblage sont des variables binaires  $Y_{rt}$  (fixées à 0 ou 1). Toutes les autres types de variables  $X_{rt}$  et  $I_{it}$ , qui sont des quantités, sont optimisées dans chaque sous-problème mais en les considérant comme continues. Différentes méthodes sont possibles pour rétablir les valeurs réelles obtenues par la méthode de résolution du programme linéaire en nombre mixtes en valeurs entières. On notera par la suite  $K$  l'ensemble des variables  $Y_{rt}$  fixées dans un sous-problème avec les valeurs  $Y'_{rt}$  de la solution initiale obtenue à l'itération précédente de l'heuristique,  $K = \{Y_{rt}/Y_{rt} = Y'_{rt}\}$  et le cardinal de  $K$  indique le nombre des contraintes à ajouter au modèle du problème général pour générer le modèle d'un sous-problème.

#### 1.2.3.1 Solution initiale

L'heuristique de décomposition nécessite une méthode pour générer une solution initiale qui permettra de fixer certaines variables à la première itération. La méthode doit être simple, rapide et être faisable par rapport aux contraintes du problème général notamment les contraintes de capacité. La méthode proposée dans le cadre de ce travail est la méthode « lot pour lot » déjà présentée dans les méthodes de génération de solutions initiales de l'algorithme génétique. Le plan de désassemblage

ainsi obtenu ne satisfait pas nécessairement toutes les contraintes de capacité et les méthodes de réparation identiques à celles utilisées dans l'algorithme génétique sont utilisées.

### 1.2.3.2 Stratégies de décomposition

La stratégie de décomposition détermine le choix des variables fixées et optimisées à chaque itération. La taille d'un sous-problème et, par conséquent, son temps de résolution en fonction du nombre de variable à optimiser. L'objectif est de permettre une résolution rapide en cherchant à améliorer la solution courante. Il y a  $R \cdot T$  variables  $Y_{rt}$  dans le problème général, un sous-problème est donc défini par le choix d'un sous-ensemble de produits et d'un sous-ensemble de périodes pour lesquels les variables  $Y_{rt}$  sont optimisées. Différentes stratégies de décomposition sont présentées ici.

#### a. Stratégie un produit/plusieurs périodes (SIP)

Un produit  $r^*$  et une période  $t^*$  sont choisis pour définir un sous-problème et un paramètre  $l$  donné en entrée détermine la plage de périodes sur laquelle les variables correspondant au produit  $r^*$  sont optimisées (Figure 25). Les variables optimisées sont les variables  $Y_{rt}$  telles que  $r = r^*$  et  $t = t^* - l, \dots, t^* + l$ .

			$t^* = t - l$		$t^*$		$t^* = t + l$			
	1	0	0	1	0	1	1	0	0	1
	1	1	0	1	1	0	0	0	1	0
	1	0	0	0	1	1	0	1	1	0
$r^*$	1	0	1	0	1	0	0	1	1	0
	1	0	0	1	0	1	1	0	0	1
	1	1	0	1	1	0	0	0	1	0
		Variables fixées					Variables optimisées			

Figure 25 : Exemple de décomposition d'un produit/plusieurs périodes

#### b. Stratégie plusieurs produits/plusieurs périodes (S2P)

Le principe est le même que la stratégie précédente sauf que plusieurs produits sont choisis pour définir un sous-problème (

Figure 2626). Si nous notons  $K_R$  le sous-ensemble de produit sélectionné, les variables optimisées sont les variables  $Y_{rt}$  telles que  $r \in K_R$  et  $t = t^* - l, \dots, t^* + l$ .

$K_R = \{r_1, r_2, r_3\}$										
			$t^* = t - l$		$t^*$		$t^* = t + l$			
	1	0	0	1	0	1	1	0	0	1
$r_1$	1	1	0	1	0	1	1	0	1	0
	1	0	0	0	1	1	0	1	1	0
$r_2$	1	0	1	0	1	0	0	1	1	0
$r_3$	1	0	0	1	0	1	1	0	0	1
	1	1	0	1	1	0	0	0	1	0
		Variables fixées					Variables optimisées			

**Figure 26 :** Exemple de décomposition de plusieurs produits/plusieurs périodes

*c. Stratégie un produit/horizon glissant (SHG)*

Dans cette stratégie, une succession par rapport aux périodes de sous-problèmes est résolue pour un produit dans un ordre chronologique. Un paramètre  $l$  donné en entrée détermine la plage de périodes sur laquelle les variables correspondant au produit  $r^*$  sont optimisées et un paramètre  $\delta$  définit le nombre de périodes sur lesquelles des variables sont à nouveau optimisées dans un nouveau sous-problème par rapport au précédent c'est-à-dire qui se chevauchent entre deux sous-problèmes (le chevauchement donne de meilleures solutions (Absi 2005)). La succession des sous-problèmes pour un produit  $r$  commence à  $t = 0$  et se termine dès que  $t > T$ . Pour un sous-problème commençant à la période  $t$  les variables optimisées sont les variables  $Y_{rt}$  telles que  $r = r^*$  et  $t = t^* - l, \dots, t^* + l$  et le sous-problème suivant commencera à la période  $t = t + l - \delta$  (Figure 27).

$\Delta = 4, \delta = 1$										
	$t^*$			$t^* + \Delta$						
	1	0	0	1	0	1	1	0	0	1
	1	1	0	1	1	0	0	0	1	0
	1	0	0	0	1	1	0	1	1	0
$r^*$	1	0	0	1	1	0	0	1	1	0
	1	1	1	0	0	0	1	1	0	1
	1	1	0	1	0	0	0	0	1	1
		Variables $Y_{r,t}$ fixées						Variables $Y_{r,t}$ optimisées (et toutes les autres variables $X_{r,t}$ et $I_{i,t}$ )		
Itération suivante :										
				$t^* \leftarrow t + \Delta + 1 - \delta$			$t^* + \Delta$			
	1	0	0	1	0	1	1	0	0	1
	1	1	0	1	1	0	0	0	1	0
	1	0	0	0	1	1	0	1	1	0
$r^*$	1	0	0	1	1	0	0	1	1	0
	1	0	0	1	0	1	1	0	0	1
	1	1	0	1	1	0	0	0	1	0
		Variables $Y_{r,t}$ fixées						Variables $Y_{r,t}$ optimisées (et toutes les autres variables $X_{r,t}$ et $I_{i,t}$ )		

**Figure 27** : Exemple de décomposition d'un produit/horizon glissant

*d. Stratégie plusieurs produits/horizon glissant (S2PHG)*

Le principe est le même que la stratégie précédente sauf que plusieurs produits sont choisis pour définir un sous-problème. Dans chaque sous-problème les périodes pour lesquelles les variables sont optimisées sont les mêmes pour tous les produits.

**1.2.3.3 Schéma général de l'heuristique**

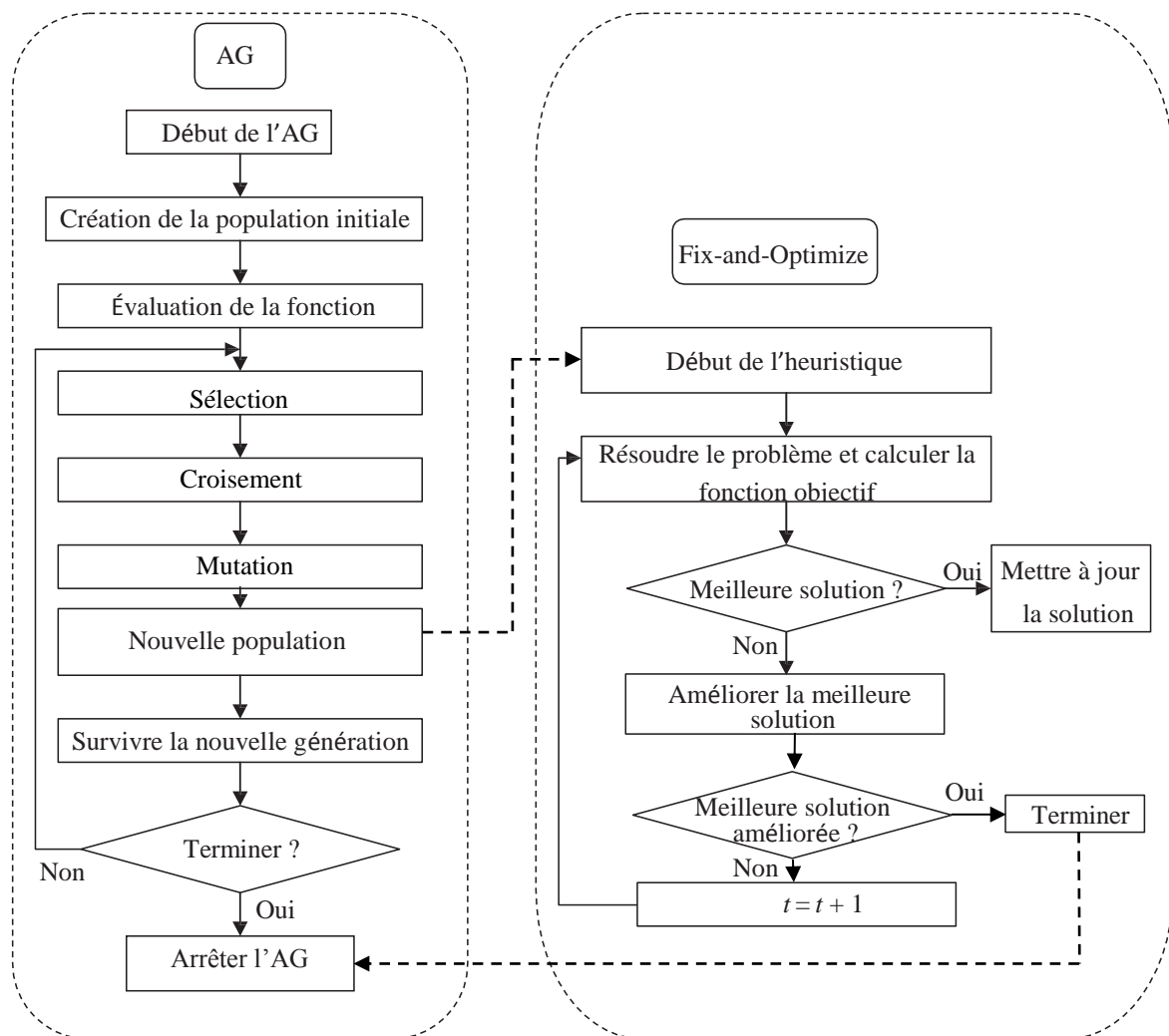
L'heuristique se déroule par itération. À chaque itération un sous-problème est résolu selon la stratégie retenue. Pour toutes les stratégies utilisées un produit ou un sous-ensemble de produits et une période, pour les deux premières stratégies, sont choisies aléatoirement (avec la même probabilité) à chaque itération. L'heuristique s'arrête dès qu'un nombre d'itération, noté *iter*, sans amélioration de la solution est atteint ou qu'une limite de temps, notée *timeLimit*, est atteinte. Les sous-problèmes sont résolus à l'optimum à l'aide du logiciel CPLEX. Une nouvelle solution est acceptée à chaque itération si elle donne un coût inférieur à la précédente.

### 1.2.4 Approche hybride (AH)

L'heuristique Fix-and-Optimize est intégrée dans l'algorithme génétique. Comme le montre la Figure 28, avant de déterminer la survie des individus dans la prochaine génération, un individu est sélectionné de manière aléatoire dans la nouvelle population pour être utilisé comme solution initiale dans l'heuristique Fix-and-Optimize. L'heuristique permet d'affiner les solutions qui sont générées par l'AG. Il s'agit d'une stratégie de diversification qui conduit la recherche dans des régions où il existe des solutions optimales ou quasi-optimales. En particulier, en utilisant l'heuristique Fix-and-Optimize en vue d'améliorer la qualité des solutions obtenues par l'AG. L'intégration de l'heuristique Fix-and-Optimize est résumée dans les Figures 28 et 29.

1. *AG*
2. *Population initiale*
3. **Pour** *Population*  $\leftarrow 1$  à *nPop* **faire**
4. *Sélectionner des parents ()*
5. *Enfant*  $\leftarrow$  *parents croisés*
6. *Muter (enfant)*
7. *Calculer la fitness (enfant)*
8. *Pop[p]. Insérer enfant*
9. **Fin pour**
10. *Mettre à jour la meilleure solution*
11. *Meilleure solution*  $\leftarrow$  *Fix-and-Optimize (Meilleure solution)*
12. *Calculer la fitness (meilleure solution)*
13. *Améliorer la meilleure solution*
14. **Si** *la meilleure solution n'est pas améliorée* **alors**
15. *Passer à la période suivante*
16. **Fin si**
17. *Retourner la meilleure solution*

**Figure 28 :** Pseudocode de l'approche hybride



**Figure 29 :** Algorithme de l'approche hybride (AH)

### 1.3 Résultats expérimentaux

Les instances des tests sont générées en adaptant les travaux de (Kim et Xirouchakis 2010). Les instances sont classées selon le nombre de produits (10, 20, 30), le nombre de composants par produits (généralisé selon une loi uniforme entre 1 et 10 et 10 et 100) et le nombre de périodes (10, 20, 30). Dix instances de chaque type sont ainsi générées. Les paramètres sont générés comme suit :

- $s_r$  est généré entre 200 et 1000 selon une loi uniforme pour chaque produit,
- $a_{ri}$  est généré entre 1 et 4 selon une loi uniforme pour chaque couple produit-composant,
- Les coûts de stockage  $h_i$  sont générés de manière à avoir un rapport  $s_r / \sum_i a_{ri} \cdot h_i$  égal à 200 pour chaque produit,
- $g_r$  est généré entre 1 et 3 pour chaque produit,



- La capacité est de 1000 pour chaque période,
- Les demandes sont générées entre 50 et 200 dans un premier temps puis sont ajustées pour chaque instance soit faisable par rapport à la contrainte de capacité.

Pour ajuster les demandes, chaque instance est résolue sans la contrainte de capacité et sans les coûts et les variables de setup (méthode lot pour lot). Cela permet de déterminer la capacité nécessaire ( $CN$ ) pour satisfaire toutes les demandes. Si nous notons  $TC$  la capacité totale disponible ( $T \cdot 1000$ ) alors les demandes sont modifiées en faisant :  $0,9(TC/CU)d_{it}$ . Le paramétrage de l'algorithme est le suivant :

- Probabilité de croisement : 0.7,
- Probabilité de mutation :  $1/(TC)$ ,
- Taille de la population : 100,
- Nombre de générations : 1000.

L'algorithme s'arrête si le meilleur individu ne s'améliore pas lorsqu'on dépasse 1000 générations ou la meilleure solution n'est pas améliorée après 200 générations.

Pour tester les performances des méthodes proposées, nous comparons avec la borne inférieure obtenue par la relaxation lagrangienne. Le gap de chaque solution est calculé comme suit :

$$Gap = \frac{\text{Solution méthode approchée} - \text{Borne inférieure}}{\text{Borne inférieure}} \cdot 100$$

Les bornes inférieures sont calculées en utilisant la relaxation lagrangienne (Kim 2005). Les résultats des tests sont présentés dans les tableaux 19, 20, 21 et 22.

Les résultats des tests de la meilleure borne supérieure obtenue par la relaxation lagrangienne sont résumés dans le tableau 19. Ils montrent les gaps minimaux, moyens et maximaux ainsi que le temps (CPU seconde). Comme nous pouvons le remarquer dans la table, l'heuristique de la relaxation lagrangienne donne des solutions très proches de l'optimale pour toutes les instances. En effet, les valeurs maximales du gap moyen et maximal sont 2.33% et 4.19% respectivement. En ce qui concerne le temps de calcul, nous remarquons que ce dernier augmente rapidement avec l'augmentation du nombre de périodes et de produits.

Par ailleurs, le gap diminue lorsque le nombre de produits et de composants augmente alors qu'il ne montre pas une tendance particulière sur le nombre de périodes. Ces résultats peuvent être dus à la complexité de notre approche de résolution qui est  $O(R \cdot N \cdot T^2)$  et l'effet de lissage des valeurs de solution où  $R$  est le nombre de produits,  $N$  le nombre de composants dans chaque produit et  $T$  le nombre de périodes.

**Tableau 19 :** Relaxation lagrangienne

Nombre de produits	Classes de composants	Périodes	Relaxation lagrangienne (Borne supérieure)					
			Gap (%)			CPU (seconde)		
			Min	Moy	Max	Min	Moy	Max
10	Petit	10	0.79	2.33	4.19	0.11	0.17	0.25
		20	0.42	1.26	2.49	0.19	0.27	0.38
		30	0.72	1.02	1.54	0.29	0.39	0.55
	Moyen	10	0.50	1.41	2.24	0.62	1.43	2.31
		20	0.41	0.87	1.27	1.46	3.41	6.68
		30	0.21	0.35	0.44	2.03	3.71	7.62
20	Petit	10	0.24	0.81	1.73	0.29	0.55	1.02
		20	0.12	0.49	0.99	0.57	0.94	1.77
		30	0.12	0.33	0.75	0.81	1.51	2.53
	Moyen	10	0.16	0.55	0.94	1.97	3.64	7.04
		20	0.04	0.24	0.63	1.62	9.20	18.78
		30	0.05	0.10	0.16	7.35	11.95	19.92
30	Petit	10	0.14	0.50	0.94	0.58	1.05	1.57
		20	0.13	0.36	0.72	0.74	1.48	2.98
		30	0.07	0.22	0.40	1.41	2.99	4.10
	Moyen	10	0.07	0.25	0.51	3.99	9.35	14.75
		20	0.05	0.13	0.20	8.37	16.16	22.71
		30	0.03	0.09	0.19	2.83	21.31	38.72

**Tableau 20 : Stratégie S1P/SPP**

Nombre de produits	Classes de composants	Périodes	S1P						SPP					
			Gap (%)			CPU (seconde)			Gap (%)			CPU (seconde)		
			Min	Moy	Max	Min	Moy	Max	Min	Moy	Max	Min	Moy	Max
10	Petit	10	2.00	4.66	6.20	0.90	1.33	2.20	<b>0.27</b>	<b>1.58</b>	<b>3.11</b>	1.73	3.18	6.39
		20	0.49	2.42	3.60	4.20	7.99	14.14	<b>0.45</b>	<b>0.70</b>	<b>1.09</b>	6.63	10.73	14.30
		30	0.66	1.22	2.10	10.72	22.10	30.28	<b>0.33</b>	<b>0.59</b>	<b>0.89</b>	17.79	24.22	37.82
	Moyen	10	1.79	4.35	11.06	4.94	8.84	12.30	<b>0.34</b>	<b>0.90</b>	<b>1.68</b>	8.54	14.90	24.45
		20	0.70	1.72	3.41	44.07	58.27	77.97	<b>0.17</b>	<b>0.49</b>	<b>0.86</b>	46.34	70.67	123.23
		30	0.30	0.55	0.99	137.18	196.58	267.26	<b>0.16</b>	<b>0.28</b>	<b>0.48</b>	119.82	186.56	335.72
20	Petit	10	2.60	5.12	6.51	4.46	7.11	12.08	<b>0.47</b>	<b>0.94</b>	<b>1.30</b>	7.23	10.06	14.54
		20	1.00	2.41	4.07	16.08	28.04	39.92	<b>0.42</b>	<b>0.65</b>	<b>1.01</b>	15.39	25.55	36.31
		30	0.49	1.25	2.11	63.63	78.46	101.86	<b>0.27</b>	<b>0.48</b>	<b>0.84</b>	53.50	71.07	90.99
	Moyen	10	1.45	4.06	8.05	23.63	31.78	45.25	<b>0.28</b>	<b>0.74</b>	<b>1.54</b>	30.48	51.09	77.87
		20	1.02	1.55	3.17	148.25	240.43	350.05	<b>0.23</b>	<b>0.43</b>	<b>0.76</b>	155.48	222.54	331.23
		30	0.49	0.69	0.90	505.43	642.47	833.81	<b>0.17</b>	<b>0.28</b>	<b>0.37</b>	470.31	569.45	670.07
30	Petit	10	3.29	5.54	8.18	11.09	13.27	15.23	<b>0.57</b>	<b>0.87</b>	<b>1.13</b>	13.07	21.58	26.74
		20	1.25	2.18	3.87	49.46	68.15	88.78	<b>0.25</b>	<b>0.54</b>	<b>0.78</b>	41.18	64.29	90.46
		30	0.61	1.25	1.84	95.31	143.49	205.63	<b>0.25</b>	<b>0.38</b>	<b>0.49</b>	122.86	139.09	158.20
	Moyen	10	2.39	3.51	5.32	52.28	82.32	131.62	<b>0.34</b>	<b>0.48</b>	<b>0.82</b>	97.54	130.02	184.62
		20	1.05	1.48	2.07	472.84	568.94	713.18	<b>0.22</b>	<b>0.31</b>	<b>0.50</b>	395.70	512.15	704.08
		30	0.91	1.79	2.98	900.46	901.15	903.30	<b>0.24</b>	<b>0.33</b>	<b>0.53</b>	774.75	1111.19	1657.39

**Tableau 20 : Stratégie S1PHG/SPPHG**

Nombre de produits	Classes de composants	Périodes	S1PHG						SPPHG					
			Gap (%)			CPU (seconde)			Gap (%)			CPU (seconde)		
			Min	Moy	Max	Min	Moy	Max	Min	Moy	Max	Min	Moy	Max
10	Petit	10	<b>0.45</b>	1.29	<b>2.38</b>	6.04	8.58	16.42	0.61	1.63	4.09	2.02	4.19	6.45
		20	0.63	1.28	1.75	21,14	35.68	57.19	<b>0.56</b>	1.00	<b>1.46</b>	24.06	33.41	54.02
		30	0.54	0.89	<b>1.33</b>	61,54	111.61	162.86	<b>0.30</b>	0.68	1.42	60.72	106.05	180.47
	Moyen	10	0.27	0.82	1.88	20,30	35.73	55.75	<b>0.25</b>	1.12	<b>1.84</b>	14.47	24.67	36.13
		20	<b>0.24</b>	0.70	1.12	101,67	208.55	319.60	0.42	0.57	<b>0.85</b>	137.57	229.57	361.27
		30	0.24	0.58	0.87	442,00	660.82	907.07	<b>0.20</b>	0.44	<b>0.54</b>	333.80	684.82	907.35
20	Petit	10	0.59	0.92	<b>1.46</b>	32,98	52.12	80.14	<b>0.37</b>	1.19	2.34	12.48	22.38	30.62
		20	0.61	1.26	1.88	72,61	107.52	207.47	<b>0.57</b>	0.81	<b>1.32</b>	50.14	86.61	126.86
		30	0.76	1.11	1.61	178,94	413.70	890.13	<b>0.35</b>	0.50	<b>0.66</b>	259.15	338.73	510.86
	Moyen	10	<b>0.35</b>	1.11	1.94	106,06	275.38	598.98	0.44	0.77	<b>1.71</b>	53.04	96.35	144.41
		20	0.47	1.08	1.60	366,80	593.79	915.38	<b>0.26</b>	0.57	<b>0.92</b>	308.83	727.67	907.99
		30	0.63	1.01	1.61	902,96	916.55	935.15	<b>0.53</b>	0.88	<b>1.31</b>	897.03	906.06	913.71
30	Petit	10	0.55	1.00	<b>1.33</b>	87,98	237.51	377.68	<b>0.43</b>	1.12	1.76	28.07	54.61	74.31
		20	0.93	1.38	2.29	233,18	327.12	524.42	<b>0.34</b>	0.64	<b>0.99</b>	232.71	297.20	351.08
		30	0.93	1.24	1.49	288,42	569.26	796.24	<b>0.32</b>	0.51	<b>0.90</b>	296.75	563.74	901.58
	Moyen	10	0.51	1.22	1.77	305,44	603.54	908.80	<b>0.15</b>	0.50	<b>0.75</b>	197.89	301.19	381.46
		20	0.73	1.02	1.20	907,62	923.97	944.88	<b>0.59</b>	0.76	<b>0.96</b>	900.19	904.23	909.29
		30	<b>0.83</b>	1.04	<b>1.31</b>	902,61	935.61	975.23	1.02	4.39	9.89	903.92	921.69	954.78

**Tableau 21 :** Algorithme génétique/ Approche hybride

Nombre de produits	Classes de composants	Périodes	AG						AH					
			Gap (%)			CPU (seconde)			Gap (%)			CPU (seconde)		
			Min	Moy	Max	Min	Moy	Max	Min	Moy	Max	Min	Moy	Max
10	Petit	10	0.95	2.18	3.66	10.86	12.78	15.84	<b>0.73</b>	<b>1.91</b>	<b>2.95</b>	2.70	3.91	5.12
		20	1.48	2.26	3.70	20.07	22.69	30.39	<b>1.01</b>	<b>1.76</b>	<b>3.47</b>	8.11	12.95	17.79
		30	1.48	2.45	3.20	26.70	32.08	37.11	<b>0.81</b>	<b>1.32</b>	<b>2.40</b>	16.45	27.78	40.60
	Moyen	10	<b>1.17</b>	<b>1.69</b>	<b>2.32</b>	47.74	84.61	117.17	1.37	2.23	3.56	9.10	16.78	22.50
		20	1.30	1.90	<b>2.78</b>	105.79	138.34	182.72	<b>0.74</b>	<b>1.53</b>	3.16	37.92	68.90	126.20
		30	1.28	1.63	<b>2.04</b>	156.76	194.07	253.47	<b>0.49</b>	<b>0.64</b>	1.16	85.12	142.20	200.60
20	Petit	10	1.67	<b>2.07</b>	<b>2.57</b>	23.34	26.96	29.91	<b>1.14</b>	2.32	3.59	6.58	13.11	16.92
		20	1.45	2.16	2.92	42.46	49.66	60.96	<b>0.74</b>	<b>1.15</b>	<b>1.64</b>	40.40	55.46	72.79
		30	1.45	1.85	2.42	69.87	74.90	81.60	<b>0.52</b>	<b>0.78</b>	<b>0.96</b>	77.47	106.48	148.68
	Moyen	10	1.11	<b>1.80</b>	2.94	140.01	167.20	203.09	<b>1.02</b>	1.86	<b>2.48</b>	43.43	56.80	81.68
		20	1.36	1.87	2.37	261.37	300.49	390.19	<b>0.45</b>	<b>0.72</b>	<b>1.12</b>	181.50	277.03	359.26
		30	1.29	1.71	1.94	343.82	395.72	450.95	<b>0.39</b>	<b>0.52</b>	<b>0.61</b>	454.20	662.17	1017.75
30	Petit	10	1.72	<b>2.43</b>	3.40	41.69	47.15	52.52	<b>1.55</b>	2.47	<b>2.96</b>	20.27	32.13	47.07
		20	1.90	2.31	2.77	77.03	83.39	92.23	<b>0.83</b>	<b>1.10</b>	<b>1.52</b>	71.28	106.01	142.22
		30	1.81	2.04	2.40	110.43	114.56	123.41	<b>0.46</b>	<b>0.81</b>	<b>0.96</b>	172.89	237.32	268.53
	Moyen	10	1.62	2.21	2.80	232.66	278.61	341.21	<b>0.40</b>	<b>1.14</b>	<b>2.07</b>	88.45	129.45	191.93
		20	1.56	2.02	2.51	374.66	459.18	510.47	<b>0.47</b>	<b>0.63</b>	<b>0.76</b>	342.86	527.96	664.51
		30	1.43	1.72	1.97	515.24	622.45	683.28	<b>0.39</b>	<b>0.50</b>	<b>0.63</b>	1024.32	1199.44	1485.92

Les tableaux 20, 21 et 22 résument les résultats obtenus par l'heuristique Fix-and-Optimize pour les différentes stratégies de décomposition. En ce qui concerne la stratégie à un seul et plusieurs produits, nous remarquons que sur le même horizon de planification (plusieurs périodes), la stratégie de plusieurs produits est toujours meilleure à celle d'un seul produit pour chaque classe d'instances. En effet, dans la stratégie de décomposition de plusieurs produits, les coûts de setup sont minimisés, d'où la minimisation du coût total. Comme nous le pouvons remarquer dans le tableau 20, plus le nombre de produits augmente, le gap diminue même si le nombre de composants et de périodes augmente. Pour le SPPHG, nous avons un gap maximal d'environ 3.11% pour la petite classe avec 10 produits et 10 périodes, contrairement à la grande classe avec 30 produits et 30 périodes, nous avons un gap maximal de 0.53%. Contrairement à S1P, nous avons un gap maximal de 6.11%. Nous pouvons constater que le coût total dépend fortement du coût de setup. Nous remarquons également qu'il y a une amélioration de 3.2% en moyenne entre les deux stratégies de décomposition. En ce qui concerne le temps de calcul, nous remarquons que les deux stratégies sont rapides, nous ne dépassons pas 1150 secondes pour toutes les instances. Le temps de calcul est très sensible par rapport à la taille de l'horizon, il croît rapidement avec l'augmentation du nombre de périodes (il est moins sensible par l'augmentation du nombre de composants et de produits).

En ce qui concerne la stratégie de décomposition sur un horizon glissant, nous avons également la stratégie de décomposition à plusieurs produits est meilleure à celle d'un seul produit. Comme la montre le tableau 21, pour le gap maximal, SPPHG n'est pas toujours meilleure par rapport à S1PHG. Nous avons un gap maximal de 2.37% pour le S1PHG, contrairement à SPPHG où nous avons un gap maximal de 4.09%. Par ailleurs, nous n'avons pas une grande différence entre les deux stratégies, nous avons en moyenne une amélioration de 0.14% pour les gaps maximaux. Comme les deux stratégies précédentes, nous n'avons pas une différence significative entre le temps de calcul de S1PHG et SPPHG, nous ne dépassons pas 905 secondes pour toutes les instances.

Dans la stratégie de décomposition d'un seul produit, S1PHG est performante par rapport à S1P. Contrairement à la stratégie de décomposition sur plusieurs produits, nous avons SPPHG qui est performante par rapport à S1PHG.

Le tableau 22 résume les résultats des tests de l'algorithme génétique avant et après l'intégration de l'heuristique Fix-and-Optimize. D'après ces résultats, nous remarquons que l'intégration de l'heuristique Fix-and-Optimize donne des meilleures solutions par rapport à l'AG (nous avons une amélioration d'environ 0.35% pour les différentes instances). Comme nous le constatons dans le tableau 22, les gaps min, moyens, et max obtenus après l'intégration de l'heuristique Fix-and-Optimize sont plus petits par rapport à ceux de l'AG. L'heuristique Fix-and-Optimize est utilisée dans chaque génération de l'AG comme un outil de diversification qui conduit à la recherche dans les régions où il existe de très bonnes solutions. C'est sur cette base que l'approche hybride permet de trouver de bonnes solutions en évitant les optimums locaux avec un nombre de générations significativement petit.

Nous remarquons également que dans les grandes instances toutes les approches donnent de meilleures solutions. Elles trouvent la solution dans moins de 1200 secondes. Ainsi, le temps de calcul est assez important dans une décision tactique, donc les approches peuvent être jugées intéressantes en termes de temps d'exécution et de la qualité de solution. En conclusion, l'intégration de l'heuristique Fix-and-Optimize dans l'AG a conduit à des améliorations significatives.

## 2 Modèle de lot sizing avec ventes perdues

Contrairement au modèle développé dans la première partie de ce chapitre, nous considérons maintenant que les demandes des composants peuvent ne pas être satisfaites. Une demande qui n'est pas satisfaite est considérée comme perdue et entraîne un coût de ventes perdues. L'objectif est de déterminer la quantité à désassembler de tous les produits en fin de vie afin de minimiser la fonction de coût.

Dans cette étude, nous considérons la même structure de désassemblage à deux niveaux pour les produits ainsi que les hypothèses de la première partie avec en plus l'autorisation de ventes perdues. La fonction de coût à minimiser est composée du coût de setup, coût de stockage, coût de ventes perdues et le coût de pénalité du temps de désassemblage. Pour formuler ce modèle, nous utilisons les mêmes notations que nous avons utilisées dans le modèle sans ventes perdues en rajoutant les nouveaux paramètres et variables suivantes :

### Paramètres

$p_i$  : coût de ventes perdues de composant  $i$  à la période  $t$

### Variables de décisions

$L_{it}$  : la quantité de ventes perdues du composant  $i$  à la période  $t$

La fonction objectif peut être donnée comme suit :

$$\text{Min} \sum_{r=1}^R \sum_{t=1}^T s_{rt} \cdot Y_{rt} + \sum_{t=1}^T \sum_{i=1}^N (h_i \cdot I_{it} + p_i L_{it}) + \sum_{t=1}^T u_t \cdot O_t$$

Sous contraintes:

$$I_{it} = I_{i,t-1} + L_{it} + a_{ri} \cdot X_{r,t} - d_{i,t} \text{ pour } i = 1, \dots, N; r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.9)$$

$$X_{rt} \leq M * Y_{rt} \text{ pour } r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.10)$$

$$\sum_{r=1}^R g_r * X_{rt} - O_t \leq C_t \text{ pour } t = 1, \dots, T \quad (3.11)$$

$$I_{it}, L_{it}, X_{rt}, O_t \geq 0 \text{ } X_{rt}, I_{it} \text{ et } L_{it} \in \mathbb{N} \text{ pour } i = 1, \dots, N; r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.12)$$

$$Y_{rt} \in \{0,1\} \text{ pour } r = 1, \dots, R \text{ et } t = 1, \dots, T \quad (3.13)$$

La fonction objectif minimise la somme des coûts de setup, de ventes perdues, de stockage et le coût de dépassement du temps de désassemblage. En effet, le coût de setup correspond au désassemblage du produit, et le coût de stockage correspond au stockage de composants. La contrainte (3.9) représente la conservation des flux de stockage pour les produits en fin de vie et les de composants respectivement. Le niveau de stock d'un composant à la fin de la période  $t$  est égal à la

somme de niveau de stock à la période  $t - 1$  plus les entrées moins les sorties à la fin de la période  $t$ . Les entrées sont la quantité générée par le désassemblage, et les sorties sont les demandes. La contrainte (3.10) assure le lancement de désassemblage où la quantité produite ne doit pas dépasser le paramètre  $M$  ( $M$  est une valeur assez grande) qui peut représenter la somme maximale des demandes d'un composant sur l'horizon de planification. La contrainte (3.11) représente la contrainte de capacité. La contrainte (3.12) signifie que les variables  $X_{rt}$ ,  $O_t$ ,  $I_{it}$  et  $L_{it}$  sont discrètes non négatives. La contrainte (3.13), représente une variable binaire qui est égale à 1 si on désassemble à la période  $t$  et 0 sinon. Finalement, la contrainte (3.8) représente la capacité disponible à la période  $t$ .

Puisque le problème de lot sizing en production à deux niveaux, plusieurs produits et sans contraintes de capacité est NP-difficile, alors le problème avec ventes perdues est également NP-difficile (voir section 1.2). Il s'ensuit que pour résoudre le problème avec ventes perdues, nous utiliserons les mêmes techniques de la partie 1. Nous adopterons une heuristique Fix-and-Optimize et un algorithme génétique et une approche hybride.

## 2.1 Méthodes de résolutions

Le problème considéré dans cette étude, peut être résolu à l'aide d'un logiciel de programmation linéaire en nombres entiers. Cependant, CPLEX n'arrive pas à trouver les solutions pour les grandes tailles avec 30 périodes pour la grande classe du nombre de composants. Pour résoudre le problème dans un temps d'exécution raisonnable, les heuristiques de type Fix-and-Optimize et l'algorithme génétique sont étendus au problème pour prendre en compte les ventes perdues.

### 2.1.1 Algorithme génétique

Dans cette section, nous reprenons le même principe de l'algorithme génétique utilisé dans la première partie sans ventes perdues. Il s'agit d'apporter des modifications nécessaires prenant en compte les ventes perdues.

#### 2.1.1.1 Population initiale

Le même schéma de codage que celui utilisé dans la première partie (sans ventes perdues) est repris ici. Ici, les variables de décisions du problème sont  $X_{rt}$ ,  $I_{it}$ ,  $L_{it}$ ,  $O_t$ , et  $Y_{rt}$  où  $Y_{rt}$  est une variable binaire qui égale 0 ou 1, et les autres sont des variables de nombres réels positifs. Dans ce problème, nous ne pouvons pas coder sur la variable de setup car les autres variables ne peuvent pas être calculées à partir de  $Y_{rt}$ . Par conséquent, nous codons sur les variables  $X_{rt}$ .

Les variables  $I_{it}$  et  $L_{it}$  peuvent être calculées en se basant sur les valeurs de  $X_{rt}$ . En effet, si un setup se produit à la période  $t$ , alors  $Y_{rt} = 1$  et  $X_{rt} > 0$ , sinon, s'il y a pas de setup,  $Y_{rt} = 0$  et  $X_{rt} = 0$ . Les variables  $I_{it}$  et  $L_{it}$  sont calculées à partir des valeurs de  $X_{rt}$ . D'après l'équation de conservation des flux, le niveau de stock et la quantité de ventes perdues sont calculés comme suit :

$$I_{it} = \max(0, I_{i,t-1} + a_{ri} \cdot X_{rt} - d_{it}) \quad (3.14)$$

$$L_{it} = \max(0, d_{it} - I_{i,t-1} - a_{ri} \cdot X_{rt}) \quad (3.15)$$

Par ailleurs, nous considérons que les demandes peuvent être partiellement ou entièrement perdues. La variable de capacité de dépassement  $O_t$  peut s'exprimer en fonction de la contrainte de capacité et la quantité de désassemblage :  $O_t = \max(0, \sum_r X_{rt} - C_t)$  pour  $t = 1, \dots, T$ . Le dépassement



de la capacité est alors possible et il correspond aux heures supplémentaires qui sont pénalisées dans la fonction objectif.

Il faut donc calculer les valeurs de  $X_{rt}$  qui correspondent à  $X_{rt} > 0$ . Pour une période de désassemblage  $t$ , les valeurs correspondant à  $X_{rt} > 0$  sont générées aléatoirement selon une distribution uniforme entre  $X_{rt}^{min}$  et  $X_{rt}^{max}$  ( $X_t^{min} \leq X_{rt} \leq X_{rt}^{max}$ ).

- La valeur de  $X_{rt}^{max}$  pour une période de désassemblage  $t$ , nous considérons que la demande cumulative maximale entre la période  $t$  et la prochaine période de désassemblage  $k$  est satisfaite (pour l'ensemble des composants  $i$  où  $a_{ri} > 0$ ).

$$X_{rt}^{max} = \max_{i/a_{ri}>0} \left[ \frac{\sum_{i=1}^{t-1} d_{it} - I_{i,t-1}}{a_{ri}} \right] \quad \text{pour } r = 1, \dots, R \quad (3.16)$$

- Pour les valeurs de  $X_{rt}^{min}$ , nous supposons que la demande minimale d'un composant  $i$  à la période  $t$  est satisfaite.

$$X_{rt}^{min} = \min_{i/a_{ri}>0} \left[ \frac{d_{it}}{a_{ri}} \right] \quad \text{pour } r = 1, \dots, R \quad (3.17)$$

$$X_{rt} = \frac{\sum_{u=t}^{t+k} d_{iu}}{a_{ri}} \quad \text{pour } r = 1, \dots, R \quad (3.18)$$

Finalement, la valeur de  $X_{rt}$  serait égale  $X_{rt}^{max}$  si nous voulons satisfaire les demandes, et égale  $X_t^{min}$  si les demandes sont perdues.

En conclusion, dans le cas de plusieurs produits, la valeur de  $X_{rt} > 0$  est égale la somme de la demande d'un composant du produit entre le début de la période  $t$  et la prochaine période de désassemblage. Pour calculer la valeur de  $X_{rt}$ , il faut suivre les étapes suivantes :

**Étape 1 :** Sélectionner aléatoirement des périodes entre 0 et  $T$

**Étape 2 :** Choisir aléatoirement un composant  $l$

**Étape 3 :** Choisir aléatoirement le nombre de période entre  $t$  et  $k$  où la demande de composant est entièrement satisfaite

**Étape 4 :** Calculer  $X_t$  en utilisant la formule suivante :  $X_{rt} = \frac{\sum_{u=t}^{t+k} d_{iu}}{a_{ri}}$

**Étape 5 :** Calculer le niveau de stock et de ventes perdues à l'aide des équations (3.14) et (3.15) :

### 2.1.1.2 Sélection et évaluation

Dans cette étude, la contrainte de capacité et la fonction objectif sont utilisées pour évaluer les objectifs de chaque chromosome. Nous utilisons le codage sur la variable  $X_{rt}$ , les autres variables seront déduites à partir des valeurs de  $X_{rt}$ .

### 2.1.1.3 Croisement et mutation

Nous utilisons les mêmes méthodes de croisement et de mutation de la partie 1.

### 2.1.2 Fix-and-Optimize

Dans cette section, nous appliquons l'heuristique Fix-and-Optimize de la même manière que dans le modèle sans ventes perdues mais la solution initiale est fournie par l'algorithme génétique. En effet, dans le cas avec ventes perdues, nous avons les variables suivantes :  $X_{rt}$ ,  $I_{it}$ ,  $L_{it}$ ,  $Y_{rt}$ , et  $O_t$ , où  $Y_{rt}$  est une variable binaire. Dans ce cas, toutes ces variables sont fournies par le Solveur.

## 2.2 Résultats expérimentaux

Nous générons des instances de la même manière que celles générées par (Kim et Xirouchakis 2010) en utilisant une loi uniforme discrète  $DU(a, b)$ . En effet, pour chaque nombre de produit (10, 20 et 30), nous générons aléatoirement 10 instances suivant la loi uniforme  $DU(1,10)$ ,  $DU(10,100)$  et  $DU(100,1000)$  respectivement pour petite, moyenne et grande classe. Finalement, nous considérons 3 tailles du nombre de périodes 10, 20 et 30.

Les paramètres sont générés comme suit :

- Le coût de setup est généré suivant  $DU(1000,5000)$ .
- Le rendement de désassemblage est généré suivant  $DU(1,4)$ .
- La demande est générée suivant  $DU(50,200)$ .
- Le coût de stockage est généré suivant  $DU(5,10)$ .
- Le coût de ventes perdues est généré suivant  $DU(50,100)$ .
- Le temps de désassemblage est généré suivant  $DU(1,3)$
- Le stock initial est généré suivant  $DU(20,100)$ .
- La capacité disponible pour chaque période est 600.

Pour choisir des paramètres efficaces, nous avons fait plusieurs tests. Les paramètres que nous avons sélectionné sont les suivants :

- Probabilité de croisement :  $P_c = 0.7$ .
- Probabilité de mutation :  $P_m = 1/(RT)$ .
- Taille de la population :  $P_0 = 100$ .
- Nombre de générations  $Nb_{gen} = 1000$ .

L'algorithme s'arrête si : le meilleur individu ne s'améliore pas lorsqu'on dépasse 1000 générations. L'amélioration totale des 10 dernières meilleures solutions est inférieure à 0,01 où le nombre de générations atteint est 1000.

Pour tester les performances de la méta-heuristique proposée, nous comparons avec la solution optimale trouvée par CPLEX. Le gap de chaque solution est calculé comme suit :

$$Gap = \frac{|Solution\ métaheuristique - Solution\ CPLEX|}{Solution\ CPLEX} \cdot 100$$

La solution du CPLEX est la solution obtenue après une heure de calcul. Les valeurs optimales de la solution ou des bornes inférieures ont été obtenues en résolvant le modèle avec CPLEX. En effet, les bornes inférieures ont été utilisées à la place des valeurs optimales de la solution lorsque CPLEX ne trouve pas de solution optimale à 3600 secondes. Les résultats des tests sont résumés dans le tableau 23.

**Tableau 22 : Résultats de l'AG et approche hybride**

Nombre de produits	N	T	AG						Approche hybride						CPU (seconde) CPLEX		
			Gap (%)			CPU (seconde)			Gap (%)			CPU (seconde)			Min	Moy	Max
			Min	Moy	Max	Min	Moy	Max	Min	Moy	Max	Min	Moy	Max			
10	Petit	10	0.05	0.26	0.86	0.79	1.72	3.52	0.05	0.22	0.73	0.81	1.74	4.08	12.47	137.06	1199.56
		20	0.06	0.56	1.04	1.54	3.38	6.92	0.05	0.58	1.00	1.66	3.42	8.02	20.16	159.69	1449.25
		30	0.13	0.67	1.13	1.98	4.40	8.99	0.11	0.57	0.96	2.00	4.45	10.41	21.49	155.24	1508.75
	Moyen	10	0.13	0.68	1.20	3.92	8.71	17.82	0.11	0.58	1.12	3.97	8.81	20.64	19.77	148.99	1200.63
		20	0.29	0.94	1.32	7.78	17.29	35.36	0.25	0.80	1.12	7.87	17.49	40.96	25.45	191.46	1649.12
		30	0.43	0.97	2.03	14.77	32.81	67.11	0.37	0.83	1.73	14.94	33.19	77.72	48.02	1782.99	3600.25
	Grande	10	0.38	1.04	2.31	62.32	138.43	297.02	0.32	0.89	1.97	66.14	146.91	327.95	1020.21	3254.20	-
		20	0.04	0.14	0.97	86.28	191.65	411.21	0.03	0.15	0.69	91.56	203.39	454.04	2200.21	3400.02	-
		30	0.08	0.30	1.22	154.85	343.96	738.02	0.09	0.28	1.04	164.33	365.04	814.88	2800.12	3480.14	-
20	Petit	10	0.52	1.29	2.49	1.57	3.49	7.22	0.44	1.10	1.97	1.60	3.56	8.26	1900.45	3500.11	-
		20	0.22	1.07	1.68	2.62	5.83	12.03	0.18	1.02	1.14	2.68	5.95	13.80	2011.23	-	-
		30	0.14	0.32	0.79	3.33	7.37	15.21	0.11	0.30	0.66	3.39	7.52	17.45	2410.00	-	-
	Moyen	10	0.10	0.21	2.57	10.02	22.33	46.11	0.58	1.29	2.04	10.27	22.81	52.91	2312.12	-	-
		20	0.80	1.54	2.83	18.67	41.47	85.62	0.68	1.31	2.26	19.06	42.35	98.24	3014.23	-	-
		30	0.28	0.49	0.91	24.57	54.58	112.71	0.24	0.42	0.78	25.09	55.74	129.32	3200.10	-	-
	Grande	10	0.00	<b>0.08</b>	<b>0.14</b>	128.45	285.32	612.22	0.00	<b>0.07</b>	<b>0.15</b>	136.32	302.81	675.96	-	-	-
		20	0.00	<b>0.01</b>	<b>0.11</b>	162.21	360.31	773.13	0.00	<b>0.01</b>	<b>0.13</b>	172.15	382.39	853.62	-	-	-
		30	<b>0.22</b>	<b>0.32</b>	<b>0.40</b>	264.99	588.64	1263.00	<b>0.25</b>	<b>0.36</b>	<b>0.45</b>	281.23	624.70	1394.54	-	-	-
30	Petit	10	<b>0.08</b>	<b>0.15</b>	<b>0.68</b>	2.19	4.87	10.23	<b>0.09</b>	<b>0.17</b>	<b>0.76</b>	2.26	5.02	11.54	-	-	-
		20	<b>0.03</b>	<b>0.09</b>	<b>0.29</b>	4.67	10.37	21.62	<b>0.03</b>	<b>0.10</b>	<b>0.33</b>	4.81	10.69	24.57	-	-	-
		30	<b>0.00</b>	<b>0.12</b>	<b>0.18</b>	5.43	12.06	25.14	<b>0.00</b>	<b>0.09</b>	<b>0.22</b>	5.60	12.43	28.57	-	-	-
	Moyen	10	<b>0.12</b>	<b>0.32</b>	<b>0.46</b>	16.86	37.45	78.00	<b>0.16</b>	<b>0.33</b>	<b>0.64</b>	17.38	38.61	88.72	-	-	-
		20	<b>0.02</b>	<b>0.13</b>	<b>0.44</b>	26.05	57.86	120.41	<b>0.02</b>	<b>0.15</b>	<b>0.49</b>	26.86	59.66	137.09	-	-	-
		30	0.00	<b>0.08</b>	<b>0.39</b>	36.73	81.58	170.35	0.00	<b>0.09</b>	<b>0.44</b>	37.86	84.11	193.28	-	-	-
	Grande	10	<b>0.01</b>	<b>0.12</b>	<b>0.47</b>	196.37	436.20	972.12	<b>0.01</b>	<b>0.13</b>	<b>0.55</b>	216.43	480.77	1033.39	-	-	-
		20	<b>0.03</b>	<b>0.16</b>	<b>0.55</b>	457.38	1015.99	2264.00	<b>0.03</b>	<b>0.18</b>	<b>0.58</b>	504.12	1119.82	2407.00	-	-	-
		30	<b>0.01</b>	<b>0.14</b>	<b>0.56</b>	669.15	1486.39	3585.52	<b>0.01</b>	<b>0.16</b>	<b>0.63</b>	737.53	1638.28	3591.69	-	-	-

-: CPU supérieure à 3600 s.

Les résultats des tests du problème de lot sizing en désassemblage avec ventes perdues sont présentés dans le tableau 23. Dans ce cas, les changements se font seulement au niveau de l'algorithme génétique où le codage est appliqué sur la variable  $X_{rt}$ . L'heuristique Fix-and-Optimize n'a pas été modifiée, les modifications se font sur le modèle et l'intégration du coût de ventes perdues. Comme nous pouvons le remarquer dans le tableau 23, les résultats montrent les gaps minimaux, moyens et maximaux ainsi que CPU seconde de deux approches de résolution. Les valeurs en gras signifient que la solution obtenue par les approches de résolution est meilleure à celle obtenue par CPLEX (dans une heure de calcul). Comme nous pouvons le remarquer dans le tableau 23, les deux approches donnent des solutions très proches de l'optimale pour toutes les instances. En effet, les valeurs maximales du gap moyen et maximal de l'algorithme génétique sont 1.54% et 2.83% respectivement. Contrairement à l'approche hybride, ces dernières sont 0.22% et 2.26% respectivement. Dans le cas des grandes instances où CPLEX ne trouve pas la solution dans une heure de calcul, les deux approches trouvent de très bonnes solutions par rapport à la limite inférieure obtenue par CPLEX dans 3600 secondes de calcul. Les approches trouvent de solutions dans moins d'une heure pour toutes les instances. Nous pouvons ainsi trouver de très bonnes solutions si nous augmentons la taille du problème. En effet, si nous augmentons cette dernière, CPLEX nécessitera plus de temps pour trouver la solution optimale, alors que les approches sont très rapides.

En ce qui concerne le temps de calcul, nous remarquons que ce dernier augmente rapidement que le nombre de périodes augmente alors qu'il ne semble pas augmenter rapidement avec l'augmentation du nombre de produits. Ces résultats peuvent être dus à la complexité de nos approches de résolution qui est  $O(R.N.T^2)$  avec  $R$  le nombre de produits,  $N$  le nombre de composants dans chaque produit et  $T$  le nombre de périodes.

La première remarque que nous pouvons retenir de ces résultats est que les deux approches proposées donnent de bonnes solutions pour les petites instances (nombre de produits et de périodes ne dépasse pas 10). Par ailleurs, elles sont très sensibles en fonction du nombre de composants. En effet, il y a toujours des solutions de très bonnes qualités pour des petites périodes même si le nombre de produits augmente. Malgré la tendance ascendante du temps de calcul, les approches nécessitent beaucoup de temps pour les grandes instances. Nous avons remarqué également que le temps de calcul dans le cas avec ventes perdues est élevé par rapport à celui de sans ventes perdues car les ventes perdues augmentent la taille du problème et sa complexité. Par ailleurs, les résultats expérimentaux ont permis de prouver que l'approche hybride a amélioré significativement les solutions obtenues par l'AG.

Finalement, les deux approches semblent toujours intéressantes, car pour les petites instances, nous trouvons des solutions très proches de l'optimale. Pour les grandes instances, les approches donnent de très bonnes solutions avec un temps d'exécution moins élevé (nous ne dépassons pas une heure de calcul pour toutes les instances). Quant au CPLEX ne trouve pas la solution optimale dans plus de 5 heures de calcul sur certaines instances. Par conséquent, les approches proposées dans cette étude offrent un large choix aux décideurs en termes de qualité de solution et en temps de calculs.

## Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème de lot sizing en désassemblage à plusieurs produits avec contraintes de capacité. Nous avons étudié deux problèmes différents, le premier problème est sans ventes perdues et le deuxième avec ventes perdues. Les problèmes sont formulés sous forme d'un programme linéaire en nombres entiers. Nous avons ensuite montré analogiquement que les deux problèmes sont NP-difficiles, ainsi que CPLEX ne trouve pas de solution optimale dans une heure de calcul pour les grandes instances avec 30 périodes pour la grande classe du nombre de composants. Pour résoudre le premier modèle sans ventes perdues, nous avons proposé deux approches de résolution. Nous avons proposé dans un premier temps un algorithme génétique, puis dans un deuxième temps nous avons proposé une heuristique Fix-and-Optimize en considérant quatre stratégies de décompositions (stratégie à un seul produit/plusieurs périodes, plusieurs produits/plusieurs périodes, un seul produit/horizon glissant et plusieurs produits/horizon glissant). Finalement, nous avons proposé une approche hybride intégrant l'heuristique Fix-and-Optimize dans l'AG. Afin de montrer l'efficacité des approches proposées, plusieurs tests sont effectués sur de nombreuses instances adaptées de la littérature. Les résultats des tests ont montré sur un horizon de plusieurs périodes, la stratégie à un seul produit a été meilleure par rapport à celle de plusieurs périodes. En ce qui concerne l'horizon glissant, nous avons la stratégie à plusieurs période est meilleure à celle d'un seul produit. En outre, l'approche hybride a amélioré significativement l'algorithme génétique pour toutes les instances. En ce qui concerne le temps de calcul de différentes approches de résolution, nous ne dépassons pas 1800 secondes pour toutes les instances.

Pour la résolution du deuxième modèle avec ventes perdues, nous avons utilisé les mêmes approches précédentes. Nous avons constaté que les deux approches ont été performantes pour toutes les instances où CPLEX ne trouve pas de solution (nous rappelons que le temps de calcul du CPLEX est plus 5 heures pour certaines instances). Quant à nos approches ne dépassent pas une heure de calcul pour toutes les instances. Par conséquent, les deux approches proposées donnent un large choix aux décideurs en termes de performances et du temps de résolution. Ce chapitre a fait l'objet de trois conférences internationales.

**M., Hrouga**, M. Godichaud, L. Amodeo. Efficient Metaheuristic for Multi-product Disassembly Lot Sizing Problem. 2016 IEEE International Conference on Industrial Engineering and Engineering Management, IEEE, 4-7 December 2016. Bali, Indonesia.

**M., Hrouga**, M. Godichaud, L. Amodeo. Heuristiques pour le problème de lot sizing en désassemblage. *11e Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM, 22-24 Aout 2016*. Montréal, Québec, Canada.

**M., Hrouga**, M. Godichaud, L. Amodeo. Heuristics for multi-product capacitated disassembly lot sizing with lost sales. *8th IFAC Conference on Manufacturing Modelling, Management and Control, MIM, June 28-30 2016*. Troyes, France.

M., Godichaud, L. Amodeo, **M. Hrouga** . Metaheuristic based optimization for capacitated disassembly lot sizing problem with lost sales. *International conference on industrial engineering and systems management, October 2015*. Seville Spain.

# Conclusion générale et perspectives

Dans ce manuscrit, nous avons étudié un problème de planification du désassemblage avec une structure de produits en fin de vie constituée de deux niveaux. La planification du désassemblage consiste à déterminer sur un horizon à moyen ou à court terme la quantité de produits à désassembler afin de satisfaire des demandes pour leurs composants. La problématique de planification est caractérisée par plusieurs éléments. Le contexte est décrit par la durée de planification de l'horizon, la gestion et les prévisions de la demande, la gestion et les prévisions des retours et la gestion (continue ou discrète) du temps au cours de l'horizon. Le plan établi au début de l'horizon de planification (quantité de produits à désassembler) doit répondre à des objectifs de satisfaction des clients, de minimisation des coûts et éventuellement de maximisation des profits. Les coûts qui sont pris en compte dans cette planification sont : le coût de stockage, le coût de setup et le coût de ventes perdues si les demandes ne sont pas satisfaites. Notre objectif étant de déterminer la date et la quantité de produit à désassembler tout en satisfaisant ou en perdant les demandes en vue de minimiser les coûts de stockage, de setup et de ventes perdues. Pour atteindre cet objectif, nous avons proposé dans cette étude deux nouveaux modèles de programmation linéaire en nombres entiers ainsi que des approches de résolution permettant de résoudre ces modèles. Dans cette recherche, nous avons commencé d'abord par une analyse de l'existant en vue d'identifier les problèmes traités, leurs modélisations et les méthodes d'optimisation utilisées. D'après une étude bibliographique détaillée sur le problème de la planification du désassemblage, nous nous sommes aperçu que les chercheurs travaillaient sur un horizon décomposé en périodes sur lesquelles on connaît les demandes. Dans leurs travaux, la planification consiste à déterminer sur chaque période les quantités de produits en fin de vie à désassembler afin de satisfaire toutes les demandes de composants. Pour résoudre ce problème, les chercheurs utilisent des approches MRP, ainsi que les modèles utilisés ne prennent pas en considération des incertitudes sur la quantité et la qualité des produits retournés.

Dans ce but, nous avons proposé d'organiser ce manuscrit comme suit : dans le premier chapitre, nous avons présenté une analyse bibliographique sur l'optimisation de la logistique inverse et planification du désassemblage. Dans une première partie, nous avons analysé plusieurs définitions de la logistique inverse issues de la littérature. Nous nous sommes focalisés seulement sur les définitions les plus souvent rencontrées dans la littérature, puis nous avons proposé notre propre définition en intégrant l'aspect de désassemblage. Ensuite, nous avons classé les travaux de la littérature par type de produits retournés et par type de réseau (logistique directe, logistique inverse et logistique inverse/directe). Nous avons également étudié le problème de planification en logistique inverse. Finalement, nous avons présenté les travaux menés aux niveaux de décisions stratégiques, tactiques et opérationnelles. Ces derniers correspondent respectivement à la conception de réseaux logistiques inverses, à la planification du désassemblage, à la gestion d'approvisionnement, à l'ordonnancement et à la conception des lignes de désassemblage. Dans une deuxième partie, nous avons présenté le problème de planification du désassemblage. Nous avons d'abord analysé l'évolution des travaux sur les problèmes de désassemblages tels que les modèles et les méthodes de résolution utilisées. En se basant sur cette analyse, nous avons constaté qu'au début les chercheurs se focalisent seulement sur la satisfaction de la demande en utilisant des approches MRP. Ensuite, ils ont pris en compte la minimisation des coûts de stockage. Puis ils ont intégré également le coût de setup et de stockage en traitant le problème de lot sizing en désassemblage. Finalement, Nous avons analysé les modèles et les méthodes utilisées dans ce problème ainsi que celui de lot sizing en production. L'objectif étant de



vérifier si les méthodes utilisées au problème de lot sizing en production peuvent être appliquées à celui de lot sizing en désassemblage.

Dans le deuxième chapitre de cette thèse, nous avons étudié le problème de lot sizing en désassemblage à un seul produit et deux niveaux. L'objectif étant de minimiser à la fois les coûts de setup, de stockage et de ventes perdues. Pour atteindre cet objectif, nous avons proposé deux modèles de programmation linéaire en nombres entiers.

Dans le premier modèle, nous avons cherché à satisfaire toutes les demandes des composants en minimisant les coûts de setup et de stockage. Après une analyse des modèles de la littérature, nous avons constaté qu'il y a une accumulation de stocks de surplus à la fin de l'horizon de planification pour certains composants. Ces derniers sont causés par des écarts entre les demandes et les rendements de désassemblage des composants par rapport au produit. Une analyse du modèle nous a permis de calculer les stocks de surplus pour chaque composant à la fin de l'horizon indépendamment de la solution retenue. Nous avons aussi développé trois heuristiques « Silver Meal (SM), Least Unit Cost (LUC) et Part Period Balancing (PPB) », elles sont bien connues et largement utilisées dans la résolution du problème de lot sizing en production. Ces dernières sont des extensions de la méthode EOQ (Economic Order Quantity), elles consistent à calculer la taille de lot en équilibrant entre les coûts de stockage et de setup. Nous avons ensuite testé les performances de ces heuristiques sur différents profils de demandes (demandes croissantes, demandes décroissantes et demandes variables). Finalement, nous avons testé leurs performances sur trois classes de composants (petite, moyenne et grande). Le nombre de composants est compris entre (1 et 10), (10 et 50) et (50 et 100) pour la petite, la moyenne et la grande classe respectivement. En plus, pour chaque classe, nous avons considéré trois tranches de périodes (10, 20 et 30 périodes). Les résultats ont montré que les heuristiques donnent de bonnes solutions pour les petites et les moyennes instances où le nombre de composants ne dépasse pas 50 avec un horizon de planification composé de 10 périodes. Dans ce cas, nous ne dépassons pas 4% par rapport à la solution optimale (gap maximal). Contrairement aux grandes instances où le nombre de composants est compris entre 50 et 100 avec un horizon de planification composé de 20 à 30 périodes, nous sommes presque à 27% (gap maximal) par rapport à la solution optimale. En revanche, les heuristiques ont été très rapides (la solution est trouvée dans moins de 0.5 secondes pour les grandes instances). Nous avons également constaté que l'heuristique de PPB a été meilleure dans la majorité des cas.

Dans le deuxième modèle, nous avons considéré que les demandes de composants peuvent ne pas être satisfaites. Les demandes qui ne sont pas satisfaites sont pénalisées avec un coût de ventes perdues. L'objectif étant de minimiser les coûts de setup, de stockage et de ventes perdues. Pour montrer l'originalité de notre travail, nous avons étudié l'équivalence entre le modèle que nous avons proposé et ceux de maximisation de ventes et d'approvisionnement externe. Ensuite, Nous avons défini les propriétés du problème que nous avons étudié. En se basant sur les propriétés définies, nous avons montré que le problème étudié ne peut pas être résolu directement avec la programmation dynamique utilisée dans le problème de lot sizing en production. Pour résoudre ce problème, nous avons appliqué les mêmes heuristiques utilisées dans le premier modèle. Ensuite, les heuristiques ont été testées sur les mêmes instances du premier modèle en intégrant le coût de ventes perdues. Finalement, nous avons constaté que les heuristiques proposées peuvent être utilisées également dans la résolution du problème de lot sizing en désassemblage avec et sans ventes perdues notamment pour les petites et les moyennes instances.

Dans le troisième chapitre de cette thèse, nous avons étudié le problème de lot sizing en désassemblage à deux niveaux, plusieurs produits et avec contraintes de capacité. L'objectif étant de minimiser les coûts de setup, de stockage, de pénalité de dépassement du temps de désassemblage ainsi que le coût de ventes perdues. Pour atteindre cet objectif, nous avons proposé deux modèles de programmation linéaire en nombres entiers.

Dans le premier modèle, toutes les demandes de composants doivent être satisfaites en minimisant les coûts de stockage et les coûts fixes de désassemblage (setup). Nous avons d'abord montré analogiquement que le problème étudié est NP-difficile. Ensuite, plusieurs méthodes ont été développées pour résoudre ce problème :

- Un algorithme génétique permettant de résoudre le problème pour toutes les instances,
- Des heuristique Fix-and-Optimize en utilisant des stratégies de décomposition à un seul produit-plusieurs périodes, plusieurs produits-plusieurs périodes, un seul produit-horizon glissant et plusieurs produits-horizon glissant,
- Un algorithme hybride combinant les deux méthodes (Algorithme génétique et Fix-and-Optimize).

Les performances des heuristiques ont été testées sur plusieurs instances issues de la littérature caractérisées par trois éléments : le nombre de produits (10, 20 et 30), le nombre de composants par produit (compris entre (1 et 10), (10 et 100) et (100 et 1000)) et le nombre de périodes 10, 20 et 30. Les solutions trouvées ont été comparées avec la borne inférieure obtenue par une relaxation lagrangienne. Les résultats des tests ont montré que l'algorithme génétique a permis de résoudre l'ensemble des instances où CPLEX ne trouve pas de solution dans une heure de calcul. L'algorithme génétique donne de très bonnes solutions sur les petites instances notamment pour la petite classe où le nombre de composants ne dépasse pas 10 avec horizon composé de 10 périodes. Pour l'heuristique Fix-and-Optimize, nous avons remarqué que la stratégie de plusieurs produits et plusieurs périodes est meilleure par rapport à celle d'un seul produit et plusieurs périodes. Contrairement aux horizons de planification glissants, la stratégie à seul produit est meilleure par rapport à celle de plusieurs produits. Pour améliorer les performances de l'algorithme génétique, nous avons intégré une heuristique Fix-and-Optimize dans ce dernier en proposant une approche hybride. Cette approche a améliorée significativement l'algorithme pour toutes les instances. Par contre, nous avons constaté qu'il existe une petite différence au niveau du temps de calcul de l'approche hybride et l'algorithme génétique notamment pour les grandes instances, pour les petites instances, les deux approches ont presque le même temps d'exécution.

Dans le deuxième modèle, nous avons considéré que les demandes de composants peuvent ne pas être satisfaites. Les demandes qui ne sont pas satisfaites sont pénalisées avec un coût de ventes perdues. Ce problème est également résolu avec les mêmes approches utilisées dans le premier modèle (algorithme génétique et approche hybride). Nous avons testé les performances de ces approches sur les mêmes instances du premier modèle en intégrant le coût de ventes perdues. Les résultats ont montré que les deux approches donnent des solutions qui sont très proches de celles de la solution optimale dans un temps de calcul très court. Nous avons constaté que le temps de calcul augmente rapidement lorsque le nombre de produits et de nombre de composants augmentent, par contre il n'est pas très influencé par le nombre de périodes. Ces approches offrent aux décideurs un choix intéressant en termes de qualité des solutions en des temps d'exécution qui sont relativement courts.

Les travaux réalisés dans cette thèse peuvent être améliorés par rapport aux techniques de résolution utilisées. Comme premier axe de recherche, il sera intéressant de proposer d'autres techniques d'optimisation pour résoudre les modèles que nous avons proposé dans cette étude. Nous pouvons utiliser notamment des heuristiques Fix-and-Optimize et Fix-and-Relax pour résoudre le problème à un seul produit et deux niveaux sans et avec ventes perdues. Pour le problème à plusieurs produits avec contraintes de capacité, nous pouvons utiliser lesdites heuristiques ou encore des méta-heuristiques (recherche simulé, recherche tabou, algorithmes de colonies de fourmis...). Par ailleurs, les modèles deviennent compliqués en fonction de la structure de produits à désassembler ainsi que l'autorisation de ventes perdues.

Les travaux réalisés dans cette thèse peuvent être étendue en abordant de nouveaux aspects spécifiques au désassemblage. Dans ce cas, cette étude pourra être étendue à des problèmes de lot sizing en désassemblage où les ventes perdues peuvent être prises en compte dans les problèmes suivants :

- Problème de lot sizing en désassemblage à deux niveaux, plusieurs produits avec pièces communes,
- Problème de lot sizing en désassemblage à un seul produit, plusieurs niveaux avec et sans pièces communes,
- Problème de lot sizing en désassemblage à plusieurs produits, plusieurs niveaux avec et sans pièces communes.

La problématique d'accumulation de stocks de surplus peut être également considérée dans le problème à plusieurs niveaux sans et avec pièces communes. Il s'agit de déterminer la quantité totale de désassemblage à la fin de l'horizon de planification. Ensuite, déduire les niveaux de surplus à partir de cette quantité.

Dans notre problème, nous avons considéré que les composants désassemblés sont tous en état parfait alors que, ces derniers peuvent être défectueux. Dans ce cas, les rendements peuvent être incertains et le nombre de composants à désassembler ne sera pas connu à l'avance. L'autorisation de ventes perdues peut faire face à ce problème si nous ne voulons pas satisfaire la demande pour des raisons économiques par exemple.

D'autres modèles comme l'évacuation du stock et la maximisation du profit peuvent être également intéressants. Dans le modèle d'évacuation du stock, les ventes perdues ne sont pas autorisées, mais nous rajoutons une variable de décision dont le but d'évacuer une certaine quantité en stock. Dans le modèle de maximisation du profit, les ventes perdues sont autorisées, mais nous cherchons à maximiser les ventes. Dans ce cas, nous rajoutons une variable de décision correspondant au nombre de composants vendus, chaque composant vendu génère un profit appelé coût de ventes perdues.

## Bibliographie

- Absi, N. (2005). *Modélisation et résolution de problèmes de lot-sizing à capacité finie*. Thèse de doctorat, Université Paris VI.
- Absi, N., & Kedad-Sidhoum, S. (2006). Capacitated lot-sizing problem with setup times, stock and demand shortages. In *Information Control Problems in Manufacturing*, 12(1), 185-190.
- Absi, N., & Kedad-Sidhoum, S. (2007). MIP-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs. *RAIRO-Operations Research*, 41(02), 171-192.
- Absi, N., & Kedad-Sidhoum, S. (2008). The multi-item capacitated lot-sizing problem with setup times and shortage costs. *European journal of operational research*, 185(3), 1351-1374.
- Absi, N., Detienne, B., & Dauzère-Pérès, S. (2013). Heuristics for the multi-item capacitated lot-sizing problem with lost sales. *Computers & Operations Research*, 40(1), 264-272.
- Absi, N., Kedad-Sidhoum, S., & Dauzère-Pérès, S. (2011). Uncapacitated lot-sizing problem with production time windows, early productions, backlogs and lost sales. *International Journal of Production Research*, 49(9), 2551-2566.
- Achillas, C., Vlachokostas, C., Aidonis, D., Moussiopoulos, N., Iakovou, E., & Baniyas, G. (2010). Optimising reverse logistics network to support policy-making in the case of Electrical and Electronic Equipment. *Waste Management*, 30(12), 2592-2600.
- Adler, G. L., & Nanda, R. (1974). The effects of learning on optimal lot size determination—single product case. *AIIE Transactions*, 6(1), 14-20.
- Afentakis, P. (1987). A parallel heuristic algorithm for lot-sizing in multistage production systems. *IIE transactions*, 19(1), 34-42.
- Afentakis, P., & Gavish, B. (1986). Optimal lot-sizing algorithms for complex product structures. *Operations research*, 34(2), 237-249.
- Aggarwal, A., & Park, J. K. (1993). Improved algorithms for economic lot size problems. *Operations research*, 41(3), 549-571.
- Agrawal, S., Singh, R. K., & Murtaza, Q. (2015). A literature review and perspectives in reverse logistics. *Resources, Conservation and Recycling*, 97, 76-92.
- Akbalik, A., Penz, B., & Rapine, C. (2015). Multi-item uncapacitated lot sizing problem with inventory bounds. *Optimization Letters*, 9(1), 143-154.
- 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.
- Alfieri, A., Brandimarte, P., & D'orazio, S. (2002). LP-based heuristics for the capacitated lot-sizing problem: the interaction of model formulation and solution algorithm. *International Journal of Production Research*, 40(2), 441-458.
- Alumur, S., & Kara, B. Y. (2007). A new model for the hazardous waste location-routing problem. *Computers & Operations Research*, 34(5), 1406-1423.

- Arkin, E., Joneja, D., & Roundy, R. (1989). Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8(2), 61-66.
- Assavapokee, T., & Wongthatsanekorn, W. (2012). Reverse production system infrastructure design for electronic products in the state of Texas. *Computers & Industrial Engineering*, 62(1), 129-140.
- Bai, C., & Sarkis, J. (2013). Flexibility in reverse logistics: a framework and evaluation approach. *Journal of Cleaner Production*, 47, 306-318.
- Barany, I., Van Roy, T. J., & Wolsey, L. A. (1984). Strong formulations for multi-item capacitated lot sizing. *Management Science*, 30(10), 1255-1261.
- 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., & Gupta, S. M. (2008). Lot sizing in reverse MRP for scheduling disassembly. *International Journal of Production Economics*, 111(2), 741-751.
- Barker, T. J., & Zabinsky, Z. B. (2011). A multicriteria decision making model for reverse logistics using analytical hierarchy process. *Omega*, 39(5), 558-573.
- Barros, A. I., Dekker, R., & Scholten, V. (1998). A two-level network for recycling sand: a case study. *European Journal of Operational Research*, 110(2), 199-214.
- Beaulieu, M. (2000). Définir et maîtriser la complexité des réseaux de logistique à rebours. *Proceedings Les Troisièmes Rencontres Internationales de la Recherche en Logistique (Canada)*.
- Belo-Filho, M. A., Toledo, F. M., & Almada-Lobo, B. (2013). Models for capacitated lot-sizing problem with backlogging, setup carryover and crossover. *Journal of the Operational Research Society*, 65(11), 1735-1747.
- Bennekrouf, M., Sari, Z., Zara-Zaïtri, K., Boudahri, F. (2010). Conception d'une chaîne logistique inverse à un seul niveau dans une approche déterministe à produits multiples. *International Conference On Industrial Engineering and Manufacturing ICIEM*, Batna, Algeria.
- Berretta, R., & Rodrigues, L. F. (2004). A memetic algorithm for a multistage capacitated lot-sizing problem. *International Journal of Production Economics*, 87(1), 67-81.
- Bitran, G. R., & Yanasse, H. H. (1982). Computational complexity of the capacitated lot size problem. *Management Science*, 28(10), 1174-1186.
- Blackburn, J. D., & Kunreuther, H. (1974). Planning horizons for the dynamic lot size model with backlogging. *Management Science*, 21(3), 251-255.
- Bowersox, D. J., Calantone, R. J., & Rodrigues, A. M. (2003). Estimation of global logistics expenditures using neural networks. *Journal of Business Logistics*, 24(2), 21.
- Brahimi, N. (2004). *Planification de la production: modèles et algorithmes pour les problèmes de dimensionnement de lots* (Doctoral dissertation, Nantes).
- Brahimi, N., & Dauzère-Pérès, S. (2014). A Lagrangian heuristic for capacitated single item lot sizing problems. *4OR*, 13(2), 173-198.

- Brahimi, N., Dauzère-Pérès, S., & Wolsey, L. A. (2010). Polyhedral and Lagrangian approaches for lot sizing with production time windows and setup times. *Computers & Operations Research*, 37(1), 182-188.
- Brahimi, N., Dauzère-Pérès, S., Najid, N. M., & Nordli, A. (2006). Single item lot sizing problems. *European Journal of Operational Research*, 168(1), 1-16.
- Brandimarte, P. (2006). Multi-item capacitated lot-sizing with demand uncertainty. *International Journal of Production Research*, 44(15), 2997-3022.
- Brennan, L., Gupta, S. M., & Taleb, K. N. (1994). Operations planning issues in an assembly/disassembly environment. *International Journal of Operations & Production Management*, 14(9), 57-67.
- Buschkühl, L., Sahling, F., Helber, S., & Tempelmeier, H. (2010). Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. *Or Spectrum*, 32(2), 231-261.
- Cárdenas-Barrón, L. E., González-Velarde, J. L., & Treviño-Garza, G. (2015). A new approach to solve the multi-product multi-period inventory lot sizing with supplier selection problem. *Computers & Operations Research*, 64, 225-232.
- 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), 85-102.
- Cattrysse, D., Maes, J., & Van Wassenhove, L. N. (1990). Set partitioning and column generation heuristics for capacitated dynamic lotsizing. *European Journal of Operational Research*, 46(1), 38-47.
- Chen, H. (2015). Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, 56, 25-36.
- Chen, H. D., Hearn, D. W., & Lee, C. Y. (1994). A new dynamic programming algorithm for the single item capacitated dynamic lot size model. *Journal of Global Optimization*, 4(3), 285-300.
- Chen, H. K., Chou, H. W., & Chiu, Y. C. (2007). On the modeling and solution algorithm for the reverse logistics recycling flow equilibrium problem. *Transportation Research Part C: Emerging Technologies*, 15(4), 218-234.
- Chen, W. H., & Thizy, J. M. (1990). Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research*, 26(1), 29-72.
- Chouinard, M. (2007). Modélisation et conception de boucles d'approvisionnement: contexte multiproduit, multi état et multi-alternative de traitement—Application à un service dans le domaine de la santé. *Doctorat, Université Laval, Québec, Canada*.
- Chu, C., Chu, F., Zhong, J., & Yang, S. (2013). A polynomial algorithm for a lot-sizing problem with backlogging, outsourcing and limited inventory. *Computers & Industrial Engineering*, 64(1), 200-210.
- Chu, F., & Chu, C. (2007). Polynomial algorithms for single-item lot-sizing models with bounded inventory and backlogging or outsourcing. *Automation Science and Engineering, IEEE Transactions on*, 4(2), 233-251.
- Coleman, B. J., & McKnew, M. A. (1991). An improved heuristic for multilevel lot sizing in material requirements planning. *Decision Sciences*, 22(1), 136-156.

- Das, K., & Chowdhury, A. H. (2012). Designing a reverse logistics network for optimal collection, recovery and quality-based product-mix planning. *International Journal of Production Economics*, 135(1), 209-221.
- 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.
- Dauzere-Péres, S., Brahimi, N., Najid, N. M., & Nordli, A. (2005). *Uncapacitated lot-sizing problems with time windows*. Technical report, Ecole des Mines de Saint-Etienne.
- De Brito, M. P., & Dekker, R. (2002). A framework for reverse logistics. *Economic Institute Report (2002) EI 38*: pp. 1-19.
- De Reyck, B. (2015). Period Decompositions for the Capacitated Lot Sizing Problem with Setup Times. *Inform's Journal on Computing*.
- Dellaert, N., Jeunet, J., & Jonard, N. (2000). A genetic algorithm to solve the general multi-level lot-sizing problem with time-varying costs. *International Journal of Production Economics*, 68(3), 241-257.
- Dillenberger, C., Escudero, L. F., Wollensak, A., & Zhang, W. (1994). On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research*, 75(2), 275-286.
- Dowlatshahi, S. (2000). Developing a theory of reverse logistics. *Interfaces*, 30(3), 143-155.
- Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling—survey and extensions. *European Journal of Operational Research*, 99(2), 221-235.
- Duhaime, R., Riopel, D., & Langevin, A. (2001). Value analysis and optimization of reusable containers at Canada Post. *Interfaces*, 31(3), 3-15.
- Duta, L. (2006). *Contribution à l'étude de la conduite des systèmes de désassemblage* (Doctoral dissertation, Université de Franche-Comté; Université de Bucarest).
- Duta, L., Filip, F. G., Henrioud, J. M., & Popescu, C. (2008). Disassembly line scheduling with genetic algorithms. *International Journal of Computers, Communications & Control*, 3(3), 230-240.
- El-Sayed, M., Afia, N., & El-Kharbotly, A. (2010). A stochastic model for forward–reverse logistics network design under risk. *Computers & Industrial Engineering*, 58(3), 423-431.
- Federgruen, A., & Tzur, M. (1991). A simple forward algorithm to solve general dynamic lot sizing models with  $n$  periods in  $O(n \log n)$  or  $O(n)$  time. *Management Science*, 37(8), 909-925.
- Federgruen, A., & Tzur, M. (1993). The Dynamic Lot-Sizing Model with Backlogging: A Simple  $O(n \log n)$  Algorithm and Minimal Forecast Horizon Procedure. *Naval research*, 40, 459-478.
- Fleischmann, M., Bloemhof-Ruwaard, J. M., Dekker, R., Van der Laan, E., Van Nunen, J. A., & Van Wassenhove, L. N. (1997). Quantitative models for reverse logistics: A review. *European journal of operational research*, 103(1), 1-17.
- Fleishmann, M. (2000). Quantitative models for reverse logistics. *Rotterdam. Springer Science & Business Media*.
- Florian, M., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1980). Deterministic production planning: Algorithms and complexity. *Management science*, 26(7), 669-679.

- Gao, N., & Chen, W. (2008, October). A genetic algorithm for disassembly scheduling with assembly product structure. In *Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on* (Vol. 2, pp. 2238-2243).
- Gicquel, C., & Minoux, M. (2015). Multi-product valid inequalities for the discrete lot-sizing and scheduling problem. *Computers & Operations Research*, 54, 12-20.
- Glock, C. H., Grosse, E. H., & Ries, J. M. (2014). The lot sizing problem: A tertiary study. *International Journal of Production Economics*, 155, 39-51.
- Go, T. F., Wahab, D. A., Rahman, M. A., Ramli, R., & Hussain, A. (2012). Genetically optimised disassembly sequence for automotive component reuse. *Expert Systems with Applications*, 39(5), 5409-5417.
- Godichaud, M. (2009). Outils d'aide à la décision pour la sélection des filières de valorisation des produits de la déconstruction des systèmes en fin de vie: application au domaine aéronautique. *Thèse: Institut National Polytechnique de Toulouse*, Toulouse.
- González-Ramírez, R. G., Smith, N. R., & Askin, R. G. (2011). A heuristic approach for a multi-product capacitated lot-sizing problem with pricing. *International Journal of Production Research*, 49(4), 1173-1196.
- Gopalakrishnan, M., Ding, K., Bourjolly, J. M., & Mohan, S. (2001). A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Management Science*, 47(6), 851-863.
- Goren, H. G., Tunali, S., & Jans, R. (2010). A review of applications of genetic algorithms in lot sizing. *Journal of Intelligent Manufacturing*, 21(4), 575-590.
- Goren, H. G., Tunali, S., & Jans, R. (2012). A hybrid approach for the capacitated lot sizing problem with setup carryover. *International Journal of Production Research*, 50(6), 1582-1597.
- Graves, S. C., & Orlin, J. B. (1985). A minimum concave-cost dynamic network flow problem with an application to lot-sizing. *Networks*, 15(1), 59-71.
- Grellier, É. (2008). Optimisation de tournées de véhicules dans le cadre de la logistique inverse: modélisation et résolution par des méthodes hybrides *Doctoral dissertation, Université de Nantes*.
- Guide Jr, V. D. R., & van Wassenhove, L. N. (2003). Full cycle supply chains. *Carnegie Mellon Press, USA*.
- Gungor, A., & Gupta S. M., (1999) «Disassembly line balancing.» *Proceedings of the 1999 Annual Meeting of the Northeast Decision Sciences Institute, Newport, Rhode Island*, 24(6)193-195, 24-26.
- Gupta, S. M., & McLean, C. R. (1996). Disassembly of products. *Computers & industrial engineering*, 31(1), 225-228.
- Gupta, S. M., & Taleb, K. N. (1994). Scheduling disassembly. *The International Journal of Production Research*, 32(8), 1857-1866.
- Harrison, T. P., & Lewis, H. S. (1996). Lot sizing in serial assembly systems with multiple constrained resources. *Management Science*, 42(1), 19-36.



- Helber, S., & Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2), 247-256.
- Hindi, K. S. (1995). Computationally efficient solution of the multi-item, capacitated lot-sizing problem. *Computers & industrial engineering*, 28(4), 709-719.
- Hrouga, M., Godichaud, M., & Amodeo, L. (2014). Disassembly lot-sizing problem: Detection and quantification of surplus. In *Proceeding of the 5th conference on Information Systems Logistics and Supply Chain (ILS)*. Breda.
- Hu, T. L., Sheu, J. B., & Huang, K. H. (2002). A reverse logistics cost minimization model for the treatment of hazardous wastes. *Transportation Research Part E: Logistics and Transportation Review*, 38(6), 457-473.
- Ilgin, M. A., & Gupta, S. M. (2009). The effect of disassembly process on spare parts inventory management in post product life cycle. *Mechanical and Industrial Engineering Faculty Publications*, 4, 561-566.
- Ilgin, M. A., & Gupta, S. M. (2010). Environmentally conscious manufacturing and product recovery (ECMPRO): a review of the state of the art. *Journal of environmental management*, 91(3), 563-591.
- Imtavanich, P., & Gupta, S. M. (2006). Calculating disassembly yields in a multi-criteria decision making environment for a disassembly-to-order system. *Applications of Management Science: In Productivity, Finance, and Operations*. Elsevier Ltd, 109-125.
- Inderfurth, K., & Langella, I. M. (2006). Heuristics for solving disassemble-to-order problems with stochastic yields. *OR Spectrum*, 28(1), 73-99.
- Jans, R., & Degraeve, Z. (2004). Improved lower bounds for the capacitated lot sizing problem with setup times. *Operations Research Letters*, 32(2), 185-195.
- Jans, R., & Degraeve, Z. (2007). Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*, 177(3), 1855-1875.
- Jans, R., & Degraeve, Z. (2008). Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46(6), 1619-1643.
- Jayaraman, V., Patterson, R. A., & Rolland, E. (2003). The design of reverse distribution networks: models and solution procedures. *European journal of operational research*, 150(1), 128-149.
- Jeunet, J., & Jonard, N. (2000). Measuring the performance of lot-sizing techniques in uncertain environments. *International Journal of Production Economics*, 64(1), 197-208.
- Ji, X., Zhang, Z., Huang, S., & Li, L. (2015). Capacitated disassembly scheduling with parts commonality and start-up cost and its industrial application. *International Journal of Production Research*, 1-19.
- Kang, K. W., Doh, H. H., Park, J. H., & Lee, D. H. (2012). Disassembly leveling and lot sizing for multiple product types: a basic model and its extension. *The International Journal of Advanced Manufacturing Technology*, 1-11.
- Kannan, G., Pokharel, S., & Kumar, P. S. (2009). A hybrid approach using ISM and fuzzy TOPSIS for the selection of reverse logistics provider. *Resources, conservation and recycling*, 54(1), 28-36.
- Karimi, B., Ghomi, S. F., & Wilson, J. M. (2003). The capacitated lot sizing problem: a review of models and algorithms. *Omega*, 31(5), 365-378.

- Karimi-Nasab, M., & Modarres, M. (2015). Lot sizing and job shop scheduling with compressible process times: A cut and branch approach. *Computers & Industrial Engineering*, 85, 196-205.
- Hwa-Joong, Kim., 2005. *Disassembly Scheduling: Models and Algorithms*. Thèse de doctorat, École Polytechnique Fédérale De Lausanne.
- Kim, D. H., & Lee, D. H. (2011). A heuristic for multi-period disassembly leveling and scheduling. In *System Integration (SII), 2011 IEEE/SICE International Symposium on* (pp. 762-767). IEEE. Korea.
- Kim, H. J., & Xirouchakis, P. (2010). Capacitated disassembly scheduling with random demand. *International Journal of Production Research*, 48(23), 7177-7194.
- Kim, H. J., Lee, D. H., & Xirouchakis, P. (2006a). A polynomial exact algorithm for two-level disassembly scheduling. *Technical Report, Institute of Production and Robotics, Swiss Federal Institute of Technology, Lausanne (EPFL)*.
- Kim, H. J., Lee, D. H., & Xirouchakis, P. (2006b). Two-phase heuristic for disassembly scheduling with multiple product types and parts commonality. *International Journal of Production Research*, 44(1), 195-212.
- Kim, H. J., Lee, D. H., & Xirouchakis, P. (2006c). A Lagrangean heuristic algorithm for disassembly scheduling with capacity constraints. *Journal of the Operational Research Society*, 57(10), 1231-1240.
- Kim, H. J., Lee, D. H., & Xirouchakis, P. (2007). Disassembly scheduling: literature review and future research directions. *International Journal of Production Research*, 45(18-19), 4465-4484.
- Kim, H. J., Lee, D. H., & Xirouchakis, P. (2008). An exact algorithm for two-level disassembly scheduling. *Journal of the Korean Institute of Industrial Engineers*, 34(4), 414-424.
- Kim, H. J., Lee, D. H., Xirouchakis, P., & Kwon, O. K. (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., Lee, D. H., Xirouchakis, P., & Züst, R. (2003). Disassembly scheduling with multiple product types. *CIRP Annals-Manufacturing Technology*, 52(1), 403-406.
- Kim, J. G., Jeon, H. B., Kim, H. J., Lee, D. H., & Xirouchakis, P. (2005). Capacitated disassembly scheduling: minimizing the number of products disassembled. *Computational Science and Its Applications– ICCSA 2005* (pp. 538-547). Springer Berlin Heidelberg.
- Kim, J. G., Jeon, H. B., Kim, H. J., Lee, D. H., & Xirouchakis, P. (2006d). Disassembly scheduling with capacity constraints: minimizing the number of products disassembled. *Proceedings of the Institution of Mechanical Engineers. Part B. Journal of engineering manufacture*, 220(9), 1473-1481.
- Kim, J. G., Sun, D., He, X. J., & Hayya, J. C. (2004). The (s, Q) inventory model with Erlang lead time and deterministic demand. *Naval Research Logistics (NRL)*, 51(6), 906-923.
- Kim, K., Song, I., Kim, J., & Jeong, B. (2006). Supply planning model for remanufacturing system in reverse logistics environment. *Computers & Industrial Engineering*, 51(2), 279-287.
- Kongar, E., & Gupta, S. M. (2002). A multi-criteria decision making approach for disassembly-to-order systems. *Journal of Electronics Manufacturing*, 11(02), 171-183.

- Krarup, J. O. Bilde. 1977. Plant location, set covering and economic lot sizes: An O (mn) algorithm for structured problems. L. Collatz et al. eds. *Optimierung bei Graphentheoretischen and Ganzzahligen Probleme*. L. COLLATZ et al. (eds), Birkhauser Verlag, Basel, 155-180.
- Krikke, H. R. (1998). Recovery strategies and reverse logistic network design. *Thèse : University of Twente, Enschede, The Netherlands*.
- Krikke, H. R., van Harten, A., & Schuur, P. C. (1999). Business case Océ: reverse logistic network re-design for copiers. *OR-Spektrum*, 21(3), 381-409.
- Kropp, D. H., Carlson, R. C., & Jucker, J. V. (1983). Heuristic lot-sizing approaches for dealing with MRP system nervousness. *Decision Sciences*, 14(2), 156-169.
- Lambert, A. J. (2003). Disassembly sequencing: a survey. *International Journal of Production Research*, 41(16), 3721-3759.
- Lambert, D. M., & Stock, J. R. (1982). *Strategic physical distribution management*. RD Irwin.
- Lambert, S., Riopel, D., & Abdul-Kader, W. (2011). A reverse logistics decisions conceptual framework. *Computers & Industrial Engineering*, 61(3), 561-581.
- Lang, J. C., & Shen, Z. J. M. (2011). Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. *European Journal of Operational Research*, 214(3), 595-605.
- Langella, I. M. (2007). Heuristics for demand-driven disassembly planning. *Computers & Operations Research*, 34(2), 552-577.
- 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.
- Lee, D. H., Bian, W., & Dong, M. (2007). Multiobjective model and solution method for integrated forward and reverse logistics network design for third-party logistics providers. *Transportation Research Record: Journal of the Transportation Research Board*, (2032), 43-52.
- Lee, D. H., Kim, H. J., 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., & Zust, R. (2002). Disassembly scheduling with capacity constraints. *CIRP Annals-Manufacturing Technology*, 51(1), 387-390.
- Lee, J. E., Gen, M., & Rhee, K. G. (2009). Network model and optimization of reverse logistics by hybrid genetic algorithm. *Computers & Industrial Engineering*, 56(3), 951-964.
- Levi, R., & Shi, C. (2013). Approximation algorithms for the stochastic lot-sizing problem with order lead times. *Operations Research*, 61(3), 593-602.
- Liu, X., & Tu, Y. (2008). Production planning with limited inventory capacity and allowed stockout. *International Journal of Production Economics*, 111(1), 180-191.
- Loparic, M., Pochet, Y., & Wolsey, L. A. (2001). The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming*, 89(3), 487-504.

- Louwens, D., Kip, B. J., Peters, E., Souren, F., & Flapper, S. D. P. (1999). A facility location allocation model for reusing carpet materials. *Computers & Industrial Engineering*, 36(4), 855-869.
- Lu, Z., & Bostel, N. (2007). A facility location model for logistics systems including reverse flows: The case of remanufacturing activities. *Computers & Operations Research*, 34(2), 299-323.
- Maes, J., & Van Wassenhove, L. (1988). Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review. *Journal of the Operational Research Society*, 991-1004.
- 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.
- Manne, A. S. (1958). Programming of economic lot sizes. *Management science*, 4(2), 115-135.
- Martel, A., & Gascon, A. (1998). Dynamic lot-sizing with price changes and price-dependent holding costs. *European Journal of Operational Research*, 111(1), 114-128.
- Mazdeh, M. M., Emadikhiav, M., & Parsa, I. (2015). A heuristic to solve the dynamic lot sizing problem with supplier selection and quantity discounts. *Computers & Industrial Engineering*, 85, 33-43.
- McGovern, S. M., & Gupta, S. M. (2004, March). Metaheuristic technique for the disassembly line balancing problem. In *Proceedings of the 2004 Northeast Decision Sciences Institute Conference* (pp. 223-225).
- McGovern, S. M., & Gupta, S. M. (2004a, February). 2-opt heuristic for the disassembly line balancing problem. In *Photonics Technologies for Robotics, Automation, and Manufacturing* (pp. 71-84). International Society for Optics and Photonics.
- McGovern, S. M., & Gupta, S. M. (2007). A balancing method and genetic algorithm for disassembly line balancing. *European Journal of Operational Research*, 179(3), 692-708.
- Meng, X. (2008, September). Network design on reverse logistics of electronic wastes recycling. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on* (pp. 2110-2115). IEEE.
- Merzouk, S. E. (2007). *Problème de dimensionnement de lots et de livraisons: application au cas de la chaîne logistique* (Doctoral dissertation, Besançon).
- Moyer, L. K., & Gupta, S. M. (1997). Environmental concerns and recycling/disassembly efforts in the electronics industry. *Journal of Electronics Manufacturing*, 7(01), 1-22.
- Nakashima, K., & Gupta, S. M. (2012). A study on the risk management of multi Kanban system in a closed loop supply chain. *International Journal of Production Economics*, 139(1), 65-68.
- Neuendorf, K. P., Lee, D. H., Kiritsis, D., & Xirouchakis, P. (2001). Disassembly scheduling with parts commonality using Petri nets with time stamps. *Fundamenta Informaticae*, 47(3-4), 295-306.
- Özdamar, L., & Barbarosoglu, G. (2000). An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. *International Journal of production economics*, 68(3), 319-331.
- Pappis, C. P., Stavros, E. D., & Tsoulfas, G. (2004). Reverse logistics and environmental management. *Quantitative Approaches to Reverse Logistics*.

- Pirard, F., Iassinovski, S., & Riane, F. (2006). Une démarche hybride d'aide à la décision pour la reconfiguration et la planification stratégique des réseaux logistiques des entreprises multi-sites. In *6ème Conférence Francophone de Modélisation et Simulation—MOSIM'06*.
- Pishvaei, M. S., Jolai, F., & Razmi, J. (2009). A stochastic optimization model for integrated forward/reverse logistics network design. *Journal of Manufacturing Systems*, 28(4), 107-114.
- Pochet, Y., & Wolsey, L. A. (1992). Solving multi-item lot-sizing problems using strong cutting planes. *Management Science*, 37(1), 53-67.
- Pohlen, T. L., & Theodore Farris, M. (1992). Reverse logistics in plastics recycling. *International Journal of Physical Distribution & Logistics Management*, 22(7), 35-47.
- Prakash, P. K. S., Ceglarek, D., & Tiwari, M. K. (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.
- Ramezani, M., Bashiri, M., & Tavakkoli-Moghaddam, R. (2013). A new multi-objective stochastic model for a forward/reverse logistic network design with responsiveness and quality level. *Applied Mathematical Modelling*, 37(1), 328-344.
- Robinson, P., Narayanan, A., & Sahin, F. (2009). Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, 37(1), 3-15.
- Rogers, D. S., & Tibben-Lembke, R. S. (1998). *Going backwards: reverse logistics trends and practices* (Vol. 2). Pittsburgh, PA: Reverse Logistics Executive Council.
- Sahling, F., Buschkühl, L., Tempelmeier, H., & Helber, S. (2009). Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, 36(9), 2546-2553.
- Salema, M. I. G., Barbosa-Povoa, A. P., & Novais, A. Q. (2007). An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty. *European Journal of Operational Research*, 179(3), 1063-1077.
- Sambasivan, M., & Yahya, S. (2005). A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers & Operations Research*, 32(3), 537-555.
- Sandbothe, R. A., & Thompson, G. L. (1990). A forward algorithm for the capacitated lot size model with stockouts. *Operations Research*, 38(3), 474-486.
- Sarkis, J. (1995). Manufacturing strategy and environmental consciousness. *Technovation*, 15(2), 79-97.
- Schrady, D. A. (1967). A deterministic inventory model for repairable items. *Naval Research Logistics Quarterly*, 14(3), 391-398.
- Schulz, T. (2011). A new Silver–Meal based heuristic for the single-item dynamic lot sizing problem with returns and remanufacturing. *International Journal of Production Research*, 49(9), 2519-2533.  
setup times and shortage costs. *RAIRO-Operations Research*, 41(02), 171-192.
- Shaw, D. X., & Wagelmans, A. P. (1998). An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. *Management Science*, 44(6), 831-838.

- Shih, L. H. (2001). Reverse logistics system planning for recycling electrical appliances and computers in Taiwan. *Resources, conservation and recycling*, 32(1), 55-72.
- Shih, L. H., & Lee, S. C. (2007). Optimizing disassembly and recycling process for EOL LCD-type products: A heuristic method. *Electronics Packaging Manufacturing, IEEE Transactions on*, 30(3), 213-220.
- Sodhi, M. S., & Reimer, B. (2001). Models for recycling electronics end-of-life products. *OR-Spektrum*, 23(1), 97-115.
- Sox, C. R. (1997). Dynamic lot sizing with random demand and non-stationary costs. *Operations Research Letters*, 20(4), 155-164.
- Spengler, T., Ploog, M., & Schröter, M. (2003). Integrated planning of acquisition, disassembly and bulk recycling: a case study on electronic scrap recovery. *OR spectrum*, 25(3), 413-442.
- Suerie, C., & Stadtler, H. (2003). The capacitated lot-sizing problem with linked lot sizes. *Management Science*, 49(8), 1039-1054.
- Sung, J., & Jeong, B. (2014). A heuristic for disassembly planning in remanufacturing system. *The Scientific World Journal*, 2014.
- Taleb, K. N., & Gupta, S. M. (1997). Disassembly of multiple product structures. *Computers & Industrial Engineering*, 32(4), 949-961.
- Taleb, K. N., Gupta, S. M., & Brennan, L. (1997). Disassembly of complex product structures with parts and materials commonality. *Production Planning & Control*, 8(3), 255-269.
- Tang, O. (2004). Simulated annealing in lot sizing problems. *International Journal of Production Economics*, 88(2), 173-181.
- Tempelmeier, H., & Derstroff, M. (1996). A Lagrangean-based heuristic for dynamic multilevel multi-item constrained lot sizing with setup times. *Management Science*, 42(5), 738-757.
- Tempelmeier, H., & Hilger, T. (2015). Linear programming models for a stochastic dynamic capacitated lot sizing problem. *Computers & Operations Research*, 59, 119-125.
- Teunter, R. H., Bayindir, Z. P., & Den Heuvel, W. V. (2006). Dynamic lot sizing with product returns and remanufacturing. *International Journal of Production Research*, 44(20), 4377-4400.
- Thierry, M., Salomon, M., Van Nunen, J., & Van Wassenhove, L. (1995). Strategic issues in product recovery management. *California management review*, 37(2), 114-135.
- Trigeiro, W. W., Thomas, L. J., & McClain, J. O. (1989). Capacitated lot sizing with setup times. *Management science*, 35(3), 353-366.
- Toy, A. Ö., & Berk, E. (2013). Dynamic lot sizing for a warm/cold process: heuristics and insights. *International Journal of Production Economics*, 145(1), 53-66.
- Ullrich, C. (2013). *Advanced Disassembly Planning: Flexible, Price-quantity Dependent, and Multi-period Planning Approaches*. Springer Science & Business Media.

- Vahdani, B., Tavakkoli-Moghaddam, R., Modarres, M., & Baboli, A. (2012). Reliable design of a forward/reverse logistics network under uncertainty: a robust-M/M/c queuing model. *Transportation Research Part E: Logistics and Transportation Review*, 48(6), 1152-1168.
- Van Vyve, M., Wolsey, L. A., & Yaman, H. (2014). Relaxations for two-level multi-item lot-sizing problems. *Mathematical Programming*, 146(1-2), 495-523.
- Verma, M., & Sharma, R. R. K. (2015). Lagrangian based approach to solve a two level capacitated lot sizing problem. *Cogent Engineering*, 2(1), 1008861.
- Wagelmans, A., Van Hoesel, S., & Kolen, A. (1992). Economic lot sizing: an  $O(n \log n)$  algorithm that runs in linear time in the Wagner-Whitin case. *Operations Research*, 40(1-supplement-1), S145-S156.
- Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, 5(1), 89-96.
- Wolosewicz, C. (2008). *Approche intégrée en planification et ordonnancement de la production* (Doctoral dissertation, Ecole Nationale Supérieure des Mines de Saint-Etienne).
- Wolosewicz, C., Dauzère-Pérès, S., & Aggoune, R. (2015). A Lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem. *European Journal of Operational Research*, 244(1), 3-12.
- Wolsey, L. A. (1995). Progress with single-item lot-sizing. *European Journal of Operational Research*, 86(3), 395-401.
- Wu, T., Shi, L., Geunes, J., & Akartunalı, K. (2011). An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging. *European Journal of Operational Research*, 214(2), 428-441.
- Xanthopoulos, A., & Iakovou, E. (2009). On the optimal design of the disassembly and recovery processes. *Waste Management*, 29(5), 1702-1711.
- Xiao, J., Yang, H., Zhang, C., Zheng, L., & Gupta, J. N. (2015). A Hybrid Lagrangian-Simulated Annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. *Computers & Operations Research*.
- Xie, J., & Dong, J. (2002). Heuristic genetic algorithms for general capacitated lot-sizing problems. *Computers & Mathematics with applications*, 44(1), 263-276.
- Yano, C. A., & Lee, H. L. (1995). Lot sizing with random yields: A review. *Operations Research*, 43(2), 311-334.
- Zhou, X., & Zhou, Y. (2015). Designing a multi-echelon reverse logistics operation and network: A case study of office paper in Beijing. *Resources, Conservation and Recycling*, 100, 58-69.

# Mustapha HROUGA

## Doctorat : Optimisation et Sûreté des Systèmes

### Année 2016

#### Optimisation de la logistique inverse et planification du désassemblage

Dans cette thèse, nous traitons essentiellement des problèmes de lot sizing en désassemblage avec une structure de produits à désassembler à deux niveaux sans composants communs. Nous traitons deux problèmes différents. Dans le premier problème, nous considérons un seul produit et la contribution porte sur le développement de deux modèles de programmation en nombres entiers. Le premier modèle est considéré sans ventes perdues où toutes les demandes doivent être satisfaites, et le deuxième est considéré avec ventes perdues où les demandes peuvent ne pas être satisfaites. Pour la résolution de ce problème, nous développons d'abord une approche analytique permettant de calculer les stocks de surplus (avant la résolution du problème) à la fin de l'horizon de planification. Ensuite, nous adaptons trois heuristiques connues pour leurs performances et largement utilisées dans le problème lot sizing en production « Silver Meal, Part Period Balancing et Least Unit Cost ». Dans le deuxième problème, nous considérons plusieurs produits avec contrainte de capacité et la contribution porte sur l'extension des deux modèles précédents. Le premier est également considéré sans ventes perdues et le deuxième avec ventes perdues. En ce qui concerne la résolution de ce problème et compte tenu de sa complexité, un algorithme génétique est d'abord proposé. Ensuite, afin d'améliorer cet algorithme, nous intégrons une heuristique Fix-and-Optimize dans ce dernier tout en proposant une approche hybride. Finalement, des tests sont effectués sur de nombreuses instances de la littérature afin de montrer l'efficacité et les limites de chaque approche de résolution.

Mots clés : distribution logistique - planification - optimisation mathématique - recherche opérationnelle - programmation heuristique – programmation linéaire.

#### Optimization of Reverse Logistics and Disassembly Planning

In this thesis, we mainly deal with lot sizing problems by disassembling with a structure of products to disassemble with two levels and without commonality components. We treat two different problems. In the first problem, we consider a single product whose contribution focuses on developing the two programming models integers. The first model is considered without lost sales where all demands must be satisfied, and the second one is considered with lost sales where demands may not be met. To solve this problem, we first develop an analytical approach to calculate the surplus stocks (before solving the problem) at the end of the planning horizon. Then we adapt three heuristics known for their performance and widely used in the lot sizing problem of production "Silver Meal, Part Period Balancing and Least Unit Cost". In the second problem, we consider a number of products with capacity constraint, and the contribution relates to the extension of the two previous models. The first is considered without lost sales and the second with lost sales. Regarding the resolution of this problem and given its complexity, a genetic algorithm is first proposed. Then, to improve this algorithm, we integrate a Fix-and-Optimize heuristic in the latter while offering a hybrid approach. Finally, various tests are performed on different literature instances to demonstrate the effectiveness and limitations of each solving approach.

Keywords: logistic distribution - mathematical optimization - planning - operations research - heuristic programming – linear programming.