



HAL
open science

Résolution de problèmes intégrés dans un contexte pharmaceutique : ordonnancement et tournée de véhicules

Alexis Robbes

► **To cite this version:**

Alexis Robbes. Résolution de problèmes intégrés dans un contexte pharmaceutique : ordonnancement et tournée de véhicules. Recherche opérationnelle [math.OC]. Université de Tours - LIFAT, 2021. Français. NNT: . tel-03576074

HAL Id: tel-03576074

<https://hal.science/tel-03576074v1>

Submitted on 15 Feb 2022

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

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

UNIVERSITÉ DE TOURS

École Doctorale : Mathématiques, Informatique, Physique Théorique et Ingénierie des Systèmes
(MIPTIS)

Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LIFAT, EA 6300)

Équipe Recherche Opérationnelle : Ordonnancement, Transport (ROOT, ERL CNRS 7002)

Thèse présentée par :

Alexis Robbes

soutenue le 22 octobre 2021

Discipline : Informatique

**Résolution de problèmes intégrés dans un contexte
pharmaceutique : ordonnancement et tournées de véhicules**

THÈSE DIRIGÉE PAR :

Jean-Charles Billaut Professeur, Université de Tours

CO-ENCADRÉE PAR :

Yannick Kergosien Maître de Conférences, Université de Tours

RAPPORTEURS :

Christian Artigues Directeur de Recherche, LAAS-CNRS, Toulouse

Laetitia Jourdan Professeur, Université de Lille

JURY :

Christian Artigues Directeur de Recherche, LAAS-CNRS, Toulouse

Jean-Charles Billaut Professeur, Université de Tours

Thierry Garaix Maître de Conférences, École des Mines de Saint Etienne

Laetitia Jourdan Professeur, Université de Lille

Yannick Kergosien Maître de Conférences, Université de Tours

Eric Pinson Professeur, Université Catholique d'Angers

Caroline Prodhon Maître de Conférences, HDR, Université de Technologie de Troyes

Ameur Soukhal Professeur, Université de Tours

Remerciements

Si la thèse était un voyage, ça serait une aventure. On vise une destination, mais le chemin et les embûches nous sont inconnus.

Pour oser se lancer, il faut déjà une flamme, celle-ci je la doit à mon père. Il m'a transmis sa curiosité et sa soif de connaissance. Ce fut grâce à cette ambition d'apprendre et de comprendre que j'ai pu m'élancer dans de longues études. Merci de m'avoir éduqué.

C'est mon master qui m'a si bien préparé pour ce long trajet. Si je n'avais pas eu des professeurs aussi compétents qu'André, Anthony, Evgueny, Irena, Marc et Xavier la route aurait été beaucoup plus tumultueuse. Merci pour vos enseignements.

Flavien et Hugo se sont engagés quelques mois avant moi dans la folie de la thèse. On a pu partager nos expériences respectives, se donner des conseils et se soutenir mutuellement. Merci d'être mes amis.

Tout bon périple possède des rencontres extraordinaires. Tant de collègues du LIFAT sont déjà partis, mais vous restez tous dans mes pensées. Des discussions scientifiques endiablées aux parties de babyfoot, comment les oublier ? Merci pour tout ces moments.

La thèse m'a ouvert de nombreuses opportunités, ce part mon travail, mais aussi par celui d'un grand nombre de personnes. Annie, Christelle et Sébastien vous êtes nos super-héros. Merci pour vos super-talents.

Si j'ai pu continuer d'avancer dans ma thèse malgré les épreuves, c'est aussi en grande partie dû à Khadija. Elle a su me motiver et me proposer des directions de recherche que je n'avais pas envisagée. Elle a aussi été une amie très présente aux moments les plus difficiles. Merci d'être dans ma vie.

Encadrer cette thèse ne fut certainement pas de tout repos pour Jean-Charles et Yannick, mais ils ont su tenir bon et faire preuve de compréhension lorsque ma vie me jouait des tours. Merci de votre patience.

Mais bien sur cette thèse ne peut se finir sans l'expertise de mes deux rapporteurs Christian et Laëtitia. Merci d'avoir accepté de lire ce manuscrit.

Résumé

Cette thèse porte sur l'étude et la résolution de problèmes d'optimisation dans le domaine de la production de médicaments de chimiothérapie à l'aide des outils de la Recherche Opérationnelle. Ces travaux de recherche ont été menés en collaboration avec l'hôpital de Tours et l'entreprise Eticsys et ont eu pour but de répondre à deux grands enjeux : la conservation de la qualité de service pour les patients et la réduction des coûts pour le système de santé. Ces travaux s'étendent de l'étude des flux de production d'une unité de biopharmacie à la mise en place d'une solution d'optimisation et d'aide à la décision pour cette unité.

Ce manuscrit commence par présenter le contexte hospitalier, les contraintes métiers et sanitaires à prendre en compte lors du processus de modélisation des problèmes de planification et d'optimisation au plus proche de la réalité. Le premier problème considéré est formulé comme un problème intégré de production et de livraison. La partie production est modélisée comme un problème d'ordonnancement d'atelier de type flow-shop hybride. La partie livraison s'apparente à un problème de tournées de véhicules multi-trajet. Le deuxième problème étudié est représenté par un problème intégré de production et de gestion des ressources consommables. La partie production considérée est cette fois-ci modélisée par un problème d'ordonnancement à machines parallèles avec des périodes d'indisponibilités. La partie gestion des ressources consommables est modélisée comme un problème de bin-packing avec des contraintes de conflits. Plusieurs algorithmes heuristiques sont proposés pour résoudre les problèmes de planification et leurs performances sont évaluées. Une méthode exacte basée sur une exploration arborescente est proposée pour résoudre la partie livraison d'un problème intégré et étudier la qualité des solutions trouvées par les heuristiques de construction d'itinéraires. La gestion des ressources consommables étant intrinsèquement liée à l'ordonnancement de la production, mais indépendante de la livraison, une matheuristique sur une version intégrée production et gestion de ressource consommables du problème est développée.

Les résultats montrant l'efficacité des méthodes de résolution proposées, ont encouragé l'implémentation de certains des algorithmes proposés dans un logiciel d'aide à la déci-

sion pour le service de l'hôpital de Tours en charge de la production des préparations. Les problèmes traités sont généralisables à d'autres services de même nature mais dans d'autres contextes. La recherche a abouti aussi à la publication de plusieurs articles et communications scientifiques. De nombreuses perspectives d'améliorations des algorithmes et des extensions des modèles sont évoquées afin d'approfondir l'amélioration continue des services hospitaliers.

Abstract

This thesis tackle optimization problems in a chemotherapy drug production context. Their theoretical and practical implications are studied. In collaboration with a hospital and a company, the research aims to respond to two challenges : maintaining the quality of service for patients and reducing costs for the health system. This work goes from a biopharmacy unit production flows study to the implementation of an optimization and aid decision support solution.

This manuscript begins by presenting the hospital context, and sanitary constraints. The first problem considered is formulated as an integrated production and routing problem. The production part is modeled as a hybrid flow-shop scheduling problem. The delivery part is a variant of the multi-trip vehicle routing problem. The second problem is represented by an integrated production and cutting stock problem. The production part considered is this time modeled by a parallel machine scheduling problem with unavailability periods. The cutting stock part is modeled as a bin packing problem with conflicts. Several heuristic algorithms for improving the service are proposed and evaluated. An exact method based on multiple branch-and-bound procedures is also proposed. As the cutting stock is dependent of the production scheduling, but independent of the delivery, a mat-heuristic approach for the integrated production and cutting stock problem is developed.

The efficiencies of the proposed resolution methods encouraged to implement certain of them in an aid decision software for the hospital of Tours. The problems dealt could be generalized to other contexts. The research is also relevant to the publication of several scientific articles and communications. Many perspectives for improving algorithms and extensions of models are mentioned in order to deepen the continuous improvement of hospital services.

Table des matières

1	Introduction	23
1.1	Contexte	25
1.1.1	Le cancer en France : incidence et mortalité	25
1.1.2	Les traitements par chimiothérapies	25
1.1.3	Les préparations magistrales et hospitalières à l'UBCO	26
1.2	Notions préliminaires	31
1.2.1	Complexité algorithmique et classes de complexité	31
1.2.2	Les problèmes d'ordonnancement	35
1.2.3	Les problèmes de livraison	39
1.2.4	Les problèmes intégrés	40
1.3	Étude préliminaire	41
1.3.1	L'optimisation pour les chimiothérapies	41
1.3.2	Historique de l'optimisation à l'UBCO	42
1.3.3	Les limites de la production	43
1.4	Plan du manuscrit de thèse	46
1.4.1	Problème intégré de la production et de la livraison	46
1.4.2	Optimisation de la livraison	47
1.4.3	Prise en compte de ressources consommables et périssables	48
1.4.4	Mise en place des solutions et poursuite de la recherche	49
2	Optimiser la production et la livraison	51
2.1	Présentation de la problématique	52
2.1.1	Hypothèses de modélisation	52
2.1.2	Positionnement dans l'état de l'art	56

TABLE DES MATIÈRES

2.2	Formulation mathématique	64
2.2.1	Paramètres d'entrées	64
2.2.2	Variables de décision	65
2.2.3	Contraintes	66
2.3	Méthodes de résolution heuristiques	68
2.3.1	Reference Algorithm	71
2.3.2	Multi-level heuristic	73
2.3.3	Bilevel Gradient Descent	74
2.3.4	Greedy Randomized Adaptive Search Procedure	77
2.4	Expérimentations numériques	78
2.4.1	Générateur d'instances	78
2.4.2	Temps de calcul	81
2.4.3	Écarts relatifs	82
2.4.4	Classement	88
2.4.5	Combien d'itérations pour le GRASP	88
2.5	Conclusion	92
3	Méthode exacte pour le problème de livraison avec production fixée	93
3.1	Présentation de la problématique	94
3.1.1	Hypothèses de modélisation	95
3.1.2	Positionnement dans l'état de l'art	97
3.2	Modèles de programmation linéaires en nombres entiers mixtes	98
3.2.1	Paramètres d'entrées et variables de décision	99
3.2.2	Problèmes de tournées de véhicules multi-trajet : MT-BDP et MTP	100
3.2.3	Problème du séquençement de livraison d'un trajet (TP)	102
3.3	Méthode de résolution exacte : 3BB	103
3.3.1	Optimisation de la séquence de livraison d'un trajet : TP	104
3.3.2	Optimisation de tournée d'un véhicule : MTP	109
3.3.3	Optimisation des tournées des véhicules : MT-BDP	112
3.4	Expérimentations numériques	114
3.4.1	Génération d'instances de la littérature	114
3.4.2	Comparaison du modèle MILP et du 3BB	119

TABLE DES MATIÈRES

3.4.3	Performance de la borne supérieure initiale	127
3.5	Conclusion	128
4	Optimisation de la gestion des ressources périssables	129
4.1	Présentation de la problématique	130
4.1.1	Hypothèses de modélisation	130
4.1.2	État de l’art	134
4.2	Formulation mathématique	135
4.2.1	Paramètres d’entrées	135
4.2.2	Variables de décision	135
4.2.3	Contraintes	135
4.3	Bornes inférieures	139
4.3.1	Problème de Bin-packing (BPP)	139
4.3.2	Problème de Cutting-stock (CSP)	140
4.4	Méthode de résolution matheuristiques	142
4.4.1	Ordonnancement et affectation des flacons initiale	142
4.4.2	Problème de Bin-packing avec conflits (BPC)	145
4.4.3	Permutation non-destructive et liste tabou	145
4.5	Expérimentations numériques	147
4.5.1	Génération des instances	148
4.5.2	Limites du modèle MILP et choix de la borne inférieure	151
4.5.3	Qualité de la matheuristique	151
4.5.4	Nombre de permutations non-destructives et temps de calcul	152
4.6	Conclusions	156
5	Conclusion et perspectives de recherches	157
5.1	Problème intégré de production et de livraison	158
5.2	Problème de livraison	159
5.3	Problème intégré production et de gestion des ressources consommables	160
5.4	Publications scientifiques	161

TABLE DES MATIÈRES

Liste des tableaux

1.1	Notations Landau pour les comparaisons asymptotiques les plus utilisées en algorithmique.	31
1.2	Critères d'optimisation classiques des problèmes d'ordonnement	36
2.1	Tableau résumé des études sur les problèmes de production et de distribution intégrée.	56
2.2	Paramètres d'entrées du modèle	64
2.3	Variables de décisions du modèle	65
2.4	Description des deux types d'instances : "UBCO" et "Random"	79
2.5	Temps de calcul moyen et maximum (secondes) de chaque heuristique selon le type d'instance.	81
2.6	Écart relatif (Gap) moyens des différents heuristiques pour les 24 ensembles d'instances.	84
3.1	Paramètres d'entrées des modèles	99
3.2	Variables de décisions des modèles	99
3.3	Nombre de lots et taille maximale des lots pour chaque instance.	118
4.1	Paramètres d'entrées du modèle MILP	135
4.2	Variables de décisions des modèles	136
4.3	Le nombre de partitions en nombres entiers pour $cap_i \leq 10$	141
4.4	Description des instances	149
4.5	Caractéristiques des 35 produits des instances	150
4.6	Résultats agrégés : Gap moyen par rapport à CSPLB et temps de calcul moyen, en fonction du nombre de semaines planifiées, du nombre de tâches par jour et de l'heuristique d'initialisation.	155

LISTE DES TABLEAUX

5.1 Publications liées aux différents chapitres. 161

Table des figures

1.1	Les produits nécessaires à la préparation des médicaments sont conservés dans des armoires réfrigérées.	27
1.2	Étapes pour les préparations magistrales à l'UBCO.	27
1.3	Avant la stérilisation, le matériel nécessaire à chaque préparation est installé dans un panier.	28
1.4	Trois modèles différents d'isolateurs utilisés par l'UBCO entre 2018 et 2021.	29
1.5	L'hôpital Gatien de Clocheville à gauche et l'hôpital Trousseau à droite. . .	29
1.6	A gauche, en vert, l'hôpital Bretonneau, à côté, en jaune, l'hôpital Gatien de Clocheville et en bas à droite en bleu l'hôpital Trousseau.	30
1.7	Représentation de l'évolution asymptotique de différentes complexités définies par leur notation Landau.	32
1.8	Représentation schématique d'une machine de Turing.	33
1.9	Diagramme d'Euler représentant les principales classes de complexités. . . .	34
1.10	Diagramme d'Euler représentant les environnements de production des problèmes d'ordonnancement.	36
1.11	Diagramme de Gantt d'une solution d'un problème d'ordonnancement d'atelier de type flow-shop.	37
1.12	Diagramme de Gantt d'une solution d'un problème d'ordonnance à machines parallèles avec 5 tâches.	38
1.13	Diagramme de Gantt d'une solution d'un problème d'ordonnancement d'atelier de type Flow-shop hybride à deux étages avec deux machines au premier étage, une seule machine au deuxième étage et 5 tâches.	38
1.14	Une solution d'un TSP à 5 clients.	39
1.15	Une solution à deux véhicules d'un VRP à 5 clients.	40

TABLE DES FIGURES

1.16	Règles d'affectation des pharmaciens à la préparation des plateaux ou à la préparation des injections.	44
1.17	Étude des flux des préparations magistrales de la préparations des plateaux à leur libération.	45
2.1	Exemple de solution pour $ J = 17$ tâches, $ I = 2$ isolateurs, 2 machines par isolateur et une capacité de stérilisation de 4 tâches.	55
2.2	Évolution de l'état de l'art dans le temps.	59
2.3	Représentation des environnements de production dans la littérature des problèmes intégrés de production et de livraison.	60
2.4	Les critères d'optimisation les plus couramment traités dans l'état de l'art des problèmes intégrés production et livraison.	61
2.5	Diagramme d'Euler représentant les structures des problèmes intégrés de production et de livraison de l'état de l'art.	63
2.6	Organigramme des heuristiques : "Reference Algorithm", "Multilevel Heuristic" et "Bilevel Gradient Descent"	70
2.7	Dendrogramme de l'exécution de l'Algorithme 2.3.3 sur l'exemple de la Figure 2.1.	73
2.8	Boîtes à moustaches des meilleure valeur de n_f pour les heuristiques RA et ML en fonction des types d'instances, des tailles d'instances et du nombre de livreurs $ V $	80
2.9	Boîtes à moustaches du temps de calcul en fonction de l'heuristique et pour différents types d'instances, tailles d'instances et nombres de véhicules.	83
2.10	Boîte à moustaches du gap (%) en fonction du type d'instance, de la taille d'instance et du nombre de livreurs.	86
2.11	Boîtes à moustaches du retard moyen ($\frac{T_i}{ J }$) en fonction du type d'instance, de la taille d'instance et du nombre de livreurs.	87
2.12	Histogramme du classement des heuristiques en fonction du type d'instance. En cas d'égalité, le rang minimum est retenu pour les deux.	89
2.13	Nuage de points situant les différentes heuristiques en fonction de leurs écart relatif moyen et de leurs temps de calcul moyen.	90
2.14	La convergence du GRASP en fonction du nombre d'itérations, du type d'instance et de la taille d'instance. Les tracés de gauche représentent l'évolution du gap et les tracés de droite représentent l'évolution du temps de calcul.	91

TABLE DES FIGURES

3.1	Exemple d'une solution d'une instance du MT-BDP avec 7 clients partitionnées en 4 trajets et livrés par 2 véhicules.	96
3.2	L'organigramme des trois niveaux de décisions	104
3.3	Stratégie de branchement du $B\&B_{TP}$ pour un lot avec n tâches	105
3.4	Exemple du calcul de $Lb_{TP}^a(\alpha, B)$ et $Lb_{TP}^b(\alpha, B)$: Le graphe représente un arbre couvrant de poids minimal d'un graphe composé de cinq clients et le dépôt. Le sommet 0 représente le dépôt. Le diagramme de Gantt montre les temps de trajet utilisés entre chaque livraison.	107
3.5	Stratégie de branchement du Branch and Bound $B\&B_{MTP}$: chaque losange est une solution non dominée du TP correspondant et les losanges blancs sont des solutions sans intérêt	110
3.6	Méthode de branchement du $B\&B_{MT-BDP}$ pour une instance à quatre trajets et trois véhicules	112
3.7	Taille maximale des trajets $ B_{max} $ en fonction de la répartition géographique des clients.	117
3.8	Pourcentage d'instances résolues par le modèle MILP et le 3BB dans le temps de calcul imparti(600 secondes).	120
3.9	Profil de performance pour les instances avec 25 clients	121
3.10	Profile de performance pour les instances avec 50 tâches	122
3.11	Profile de performance pour les instances avec 100 clients	123
3.12	Écart de temps de calcul sur les instances résolues plus rapidement par le modèle MILP que le 3BB.	124
3.13	Comparaison de la valeur du retard total des instances en fonction du type de date de début de livraison au plus tôt.	125
3.14	Temps de calcul du 3BB des instances résolues en fonction de la taille maximale des trajets $ B_{max} $	126
3.15	Performance de la borne supérieure initiale du $B\&B_{MT-BDP}$	127
3.16	Gap de l'heuristique par rapport à la meilleure solution trouvée	128
4.1	Périodes d'indisponibilités des machines dans la journée et dans la semaine .	131
4.2	Exemple d'une solution du problème intégré de production et de gestion des ressources consommables	133
4.3	Représentation de la matrice creuse des variables $\zeta_{j,j'}$ et les contraintes du modèle linéaire en nombres entiers.	139

TABLE DES FIGURES

4.4	L'ensemble des partitions en nombres entiers (i.e. $\Omega_l = \llbracket 1, cap_l \rrbracket$), W_l lorsque $cap_l = 4$	141
4.5	Choisir la borne inférieure en fonction des dimensions de l'instance ($ J_l , cap_l$)	142
4.6	Organigramme de la matheuristique	143
4.7	Fig	147
4.8	Répartition d'une sélection de 35 produits utilisés par l'UBCO en fonction du coût unitaire d'un flacon et de la stabilité. La couleur représente le pourcentage de la production utilisant le produit.	149
4.9	Coût moyen d'une tâche en fonction du nombre de semaines à planifier et de la méthode d'initialisation de l'ordonnancement.	152
4.10	Nombre de permutations non-destructives	153
4.11	Convergence de la matheuristique pour une instance avec 160 tâches par jour et une planification de 12 semaines en fonction de l'initialisation du planning utilisé.	153
4.12	Temps de calcul en fonction des dimensions de l'instance.	154

Liste des notations

LISTE DES NOTATIONS

Opérations mathématiques

$ A $	cardinal de l'ensemble A
$\mathcal{P}(A)$	l'ensemble des parties de l'ensemble A
$\llbracket a, a' \rrbracket$	l'ensemble des nombres entiers inclus dans $[a, a']$
$\binom{a'}{a}$	le nombre de parties de a éléments dans un ensemble de a' éléments
$O(f(a))$	notation Landau pour la fonction f de paramètre a

Index

j	une tâche / un client
v	un véhicule
B	un lot de livraison / trajet
m	une machine
i	un isolateur
l	une matière première
f	un flacon
α	une heure de départ du dépôt

Données

TABLE DES FIGURES

r_j	date de début de préparation au plus tôt de la tâche j
\tilde{r}_j	date de fin de préparation au plus tôt de la tâche j
\hat{r}_B	date de départ au plus tôt du trajet B
p_j^{Prep}	temps de préparation de la tâche j
d_j	date de livraison souhaitée de la tâche j
u_j	service d'oncologie associé à la tâche j
$tt_{j,j'}$	le temps de trajet entre les services d'oncologie des tâches j et j'
q_j	la quantité de matière première nécessaire pour préparer la tâche j
cap_l	le volume d'un flacon du produit l
$cost_l$	le coût d'un flacon du produit l
$stab_l$	le temps de stabilité du produit l
s_h	la date de début de l'intervalle de disponibilité h
e_h	la date de fin de l'intervalle de disponibilité h
p^{Ste}	le temps de stérilisation d'une charge
p^{Ctrl}	le temps de contrôle d'une tâche
Q	la capacité de charge des stérilisateurs

Ensembles et séquences

J	l'ensemble des tâches
V	l'ensemble des véhicules
I	l'ensemble des isolateurs
M_i	l'ensemble des machines associées à l'isolateur i
\mathcal{B}	la séquence des trajets.
\mathcal{B}_v	la séquence des trajets associés au véhicule v
H	l'ensemble des intervalles de disponibilité
H_i	l'ensemble des intervalles de disponibilité correspondant à l'isolateur i
L	l'ensemble des matières premières
J_l	l'ensemble des tâches utilisant le produit l
F_l	l'ensemble des flacons contenant le produit l
Ω_l	l'ensemble des quantités requises par les tâches préparées avec le produit l
W_l	l'ensemble des partitions des entiers de 1 à cap_l par les éléments de Ω_l
Ψ_j	l'ensemble des tâches en conflit avec la tâche j
Φ_j	l'ensemble des tâches compatibles avec la tâche j

Variables décisions

TABLE DES FIGURES

S_j^{Prep}	la date de début de préparation de la tâche j
C_j^{Prep}	la date de fin de préparation de la tâche j
C_j^{Ctrl}	la date de fin de contrôle de la tâche j
D_j	la date de livraison de la tâche j
T_j	le retard associé à la tâche j
$B_{v,t}$	le t -ième trajet effectué par le véhicule v
α_B	la date de départ du trajet B
ω_B	la date de retour au dépôt du trajet B

Notation de Graham des problèmes d'ordonnement

1	environnement à une machine
F	atelier de type Flow-shop
J	atelier de type Job-shop
O	atelier de type Open-shop
P	machines parallèles identiques
Q	machines parallèles uniformes
R	machines parallèles décorélées

Acronymes d'institutions

CHRU	Centre Hospitalier Régional Universitaire de Tours
LIFAT	Laboratoire d'Informatique Fondamentale et Appliquée de Tours
UBCO	Unité de Biopharmacie Clinique Oncologique

Acronymes anglais des problèmes et algorithmes

TABLE DES FIGURES

B&B	Branch and Bound
BGD	Bilevel Gradient Descent
BPC	Bin Packing Problem with conflicts
BPP	Bin Packing Problem
CSP	Cutting stock problem
GRASP	Greedy Randomized Adaptive Search Procedure
HF	Hybrid Flowshop
ILP	Integer Linear Programming
ML	Multi-level
MST	Minimum Spanning Tree
MT-BDP	Multi-Trip Batch Delivery Problem
MTP	Multi-Trip Problem
MT-VRP	Multi-trip Vehicle Routing Problem
RA	Reference Algorithm
SAT	Boolean satisfiability problem
TP	Trip Problem
TSP	Traveling Salesman Problem
TW	Time Window
VRP	Vehicle Routing Problem

Chapitre 1

Introduction

Sommaire

1.1	Contexte	25
1.1.1	Le cancer en France : incidence et mortalité	25
1.1.2	Les traitements par chimiothérapies	25
1.1.3	Les préparations magistrales et hospitalières à l'UBCO	26
1.2	Notions préliminaires	31
1.2.1	Complexité algorithmique et classes de complexité	31
1.2.2	Les problèmes d'ordonnancement	35
1.2.3	Les problèmes de livraison	39
1.2.4	Les problèmes intégrés	40
1.3	Étude préliminaire	41
1.3.1	L'optimisation pour les chimiothérapies	41
1.3.2	Historique de l'optimisation à l'UBCO	42
1.3.3	Les limites de la production	43
1.4	Plan du manuscrit de thèse	46
1.4.1	Problème intégré de la production et de la livraison	46
1.4.2	Optimisation de la livraison	47
1.4.3	Prise en compte de ressources consommables et périssables	48
1.4.4	Mise en place des solutions et poursuite de la recherche	49

Le système de santé est un service aux défis complexes et variés dont les enjeux logistiques sont importants. Son perfectionnement nécessite la résolution de nombreux problèmes d'optimisation combinatoire ayant potentiellement des répercussions sur l'efficacité de tous les services de soins proposés à la société. La crise sanitaire récente a d'ailleurs montré la nécessité de l'utilisation de méthodes sophistiquées issues de la Recherche Opérationnelle (RO) pour rendre le système plus résilient et réactif. Le milieu hospitalier est donc un champ d'application de la recherche en optimisation très exigeant et aux objectifs nombreux et originaux.

Il y a près de 15 ans, le Centre Hospitalier Régional Universitaire de Tours (CHRU de Tours) a engagé une collaboration avec le Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LIFAT) afin d'améliorer la qualité de certains de ses services. C'est grâce à l'intervention de l'entreprise Eticsys que les solutions algorithmiques proposées par le LIFAT sont aujourd'hui utilisées par l'UBCO. En effet, ils ont permis la mise en place d'un système de traçabilité et assurent la maintenance des solutions implantées, service indispensable pour un logiciel d'aide à la décision performant [Kergosien *et al.*, 2011].

Financée par la région Centre-Val de Loire, cette thèse s'intéresse aux problématiques de production et de livraison rencontrées par l'Unité de Biopharmacie Clinique Oncologique (UBCO) située dans l'hôpital de Bretonneau. Ce service prépare et distribue des injections pour des traitements de chimiothérapie à l'intention des différents services d'oncologie du CHRU de Tours. Dû à des regroupements de services, l'UBCO a une demande de plus en plus conséquente d'année en année et la livraison des préparations prend une part grandissante dans sa logistique. En 2010, la production quotidienne était autour de 80 préparations par jour. En une décennie, le débit de production a doublé. Cela a été possible grâce à l'acquisition de nouveaux isolateurs (machines à environnement stérile), au renforcement du personnel et à l'utilisation du logiciel d'aide à la décision et d'optimisation nommé Planif développé par le LIFAT et Eticsys. Ce dernier optimise la production avec pour objectif de réduire l'attente des patients. Le contexte évoluant considérablement, il est primordial d'adapter l'optimisation de la planification aux nouveaux défis.

Le premier but de ce travail de thèse est de proposer une amélioration du logiciel Planif, en prenant en compte le problème de la livraison aux différents services d'oncologie. En 2021, l'UBCO est responsable de la préparation pour une vingtaine de services d'oncologie répartis en trois hôpitaux : l'hôpital de Bretonneau, l'hôpital Gatien de Clocheville et l'hôpital Trousseau. Les temps de trajets entre les hôpitaux nécessitant des durées importantes et les livraisons se faisant dans la journée au fur et à mesure que la production avance, il est nécessaire de les considérer lors de la planification des préparations. En effet, lorsqu'une injection doit être livrée dans un hôpital externe, le temps de livraison est plus important que le temps de production.

Le respect des différentes exigences sanitaires engendre un coût financier important et nécessite des machines adaptées, du personnel qualifié, des processus stricts, un contrôle qualité, etc. Optimiser la production revient donc aussi à réduire le coût de production. Pour cet aspect, une deuxième partie de la thèse se focalise sur la problématique du coût engendré par l'utilisation des ressources périssables nécessaires à la préparation des injections. La préparation d'une injection requiert une certaine quantité d'un produit cytotoxique ayant une stabilité dans le temps. L'UBCO utilise une quarantaine de produits différents dont le prix du flacon varie entre quelques dizaines d'euros à plusieurs milliers. Le coût total de ces flacons représente plusieurs millions d'euros par an. Il est donc important de limiter les pertes en optimisant la gestion de ces ressources.

Chacune des problématiques d'optimisation rencontrées par l'UBCO et abordées dans cette thèse est généralisable à d'autres unités de Biopharmacie voire à d'autres productions de nature différente comme la restauration à domicile. C'est pour cela que cette thèse se focalise autant sur l'application concrète d'algorithmes d'optimisation au contexte de l'UBCO qu'à leurs généralisations à d'autres problèmes.

1.1 Contexte

1.1.1 Le cancer en France : incidence et mortalité

En France, la première cause de décès chez l'homme est le cancer. Cette maladie, provoquée par la prolifération de cellules anormales, apparaît sous de nombreuses formes et rassemble un ensemble de pathologies très variées. D'après [Defossez *et al.*, 2019], l'incidence (le nombre de nouveaux cas de cancer par an) en France a augmenté depuis 2010. Cependant, le taux d'incidence (l'incidence rapportée à la taille de la population) tend à diminuer chez les hommes (-1,4 %) et se stabiliser chez les femmes (+0,7%). La prévention de la cancérogénicité de certains produits ou habitudes de vie, le dépistage régulier qui permet des diagnostics plus précoces et les avancées thérapeutiques importantes ont permis une baisse du taux de mortalité en 8 ans d'environ 2% chez les hommes et 0,7% chez les femmes.

1.1.2 Les traitements par chimiothérapies

Il existe plusieurs traitements aux cancers. Souvent prescrite en complémentarité d'un autre traitement (chimiothérapies adjuvantes et néo-adjuvantes), la chimiothérapie est le deuxième traitement le plus utilisé après la chirurgie et devant la radiothérapie. Les médicaments de chimiothérapie sont composés de produits cytotoxiques qui permettent l'arrêt de la prolifération des cellules cancéreuses, mais peuvent aussi parfois détruire des cel-

lules saines. Les risques d'effets secondaires sont donc importants. Il est alors nécessaire de jauger les dosages et les produits utilisés en fonction du patient. Les médicaments de chimiothérapie sont des préparations pharmaceutiques, des médicaments préparés pour des besoins spécifiques. Les préparations sont classées en trois types d'après l'article L. 5121-1 du Code de la santé publique :

- Une préparation magistrale est une préparation sur prescription médicale dédiée à un patient particulier.
- Une préparation hospitalière est préparée sur prescription médicale par une pharmacie à usage intérieur d'un établissement de santé.
- Une préparation officinale est préparée en pharmacie d'officine pour ses patients.

L'étiquetage des préparations magistrales, hospitalières et officinales est encadré afin d'assurer au mieux la sécurité et la traçabilité des préparations (articles R. 5121-146-2 à R. 5121-146-3 du Code de la santé publique).

1.1.3 Les préparations magistrales et hospitalières à l'UBCO

La production d'une préparation pour le traitement d'un cancer par chimiothérapie suit une certaine procédure au CHRU. Après une première consultation avec un médecin oncologue, une ordonnance est envoyée à l'UBCO et une date de rendez-vous est fixée pour le traitement. Cette ordonnance indique le produit cytotoxique requis, sa quantité et son dosage. L'UBCO doit alors planifier la préparation magistrale du traitement pour être administrée le jour du rendez-vous fixé au préalable. Cependant, l'évolution de l'état de santé du patient le jour de l'administration peut remettre en question le traitement prescrit. C'est pour cela qu'il est nécessaire de valider le traitement suite à une nouvelle consultation le jour même afin d'éviter les pertes engendrées par une préparation réalisée en avance et qui ne convient plus au patient. La préparation du médicament débute une fois cette validation transmise à l'UBCO. Les produits cytotoxiques requis pour les divers médicaments préparés par l'UBCO sont stockés dans des armoires réfrigérées comme dans la Figure 1.1.

La préparation d'une injection pour un traitement de chimiothérapie suit un protocole de traçabilité de chaque opération permettant de remonter la chaîne de production. Le processus est résumé dans la figure 1.2 et se présente comme suit :

1. Tout d'abord le matériel et le produit nécessaires à la préparation du médicament sont rangés dans un panier (cf Figure 1.3).
2. Ce panier est ensuite stérilisé dans sa totalité dans une chambre de stérilisation d'un isolateur.
3. À l'intérieur de cet isolateur un pharmacien prépare, à l'aide de gants intégrés à

1.1. CONTEXTE



FIGURE 1.1 – Les produits nécessaires à la préparation des médicaments sont conservés dans des armoires réfrigérées.

- l'isolateur, la poche d'injection en respectant les quantités prescrites. Une fois la mesure effectuée, un contrôle visuel est effectué par un autre opérateur.
4. Lorsque le contrôle visuel est valide la poche d'injection est extraite de l'isolateur ainsi qu'un petit échantillon de la préparation afin d'en contrôler sa qualité.
 5. Certains produits pouvant se détériorer, les échantillons des préparations les plus à risques sont contrôlés à l'aide d'un analyseur automatique. Si le contrôle est positif, alors la préparation est dite libérée.
 6. Une fois libérée, la préparation doit être livrée au service d'oncologie correspondant.
 7. Lorsque la préparation est livrée, l'administration du traitement au patient revient alors au personnel du service d'oncologie.

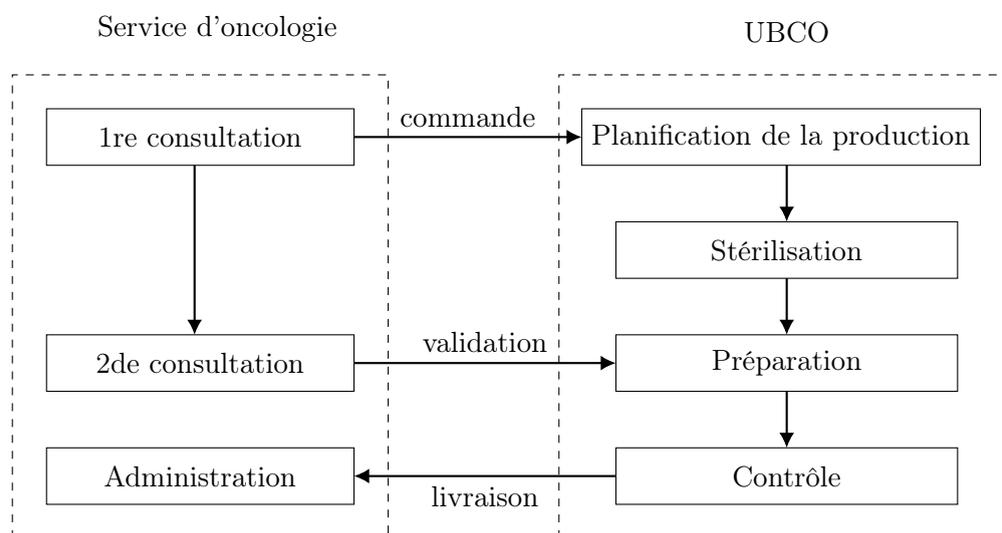


FIGURE 1.2 – Étapes pour les préparations magistrales à l'UBCO.

1.1. CONTEXTE



FIGURE 1.3 – Avant la stérilisation, le matériel nécessaire à chaque préparation est installé dans un panier.

Du point de vue de l'UBCO, la production d'un médicament de chimiothérapie peut se résumer en trois étapes importantes :

1. La **Stérilisation** par lot des paniers contenant tout le matériel nécessaire à la réalisation des préparations (un panier par médicament de chimiothérapie à réaliser).
2. La **Préparation** du médicament par un pharmacien
3. Le **Contrôle** d'un échantillon du médicament par l'analyseur automatique.

La préparation des paniers pour la stérilisation est une étape rapide, peu contrainte et ne nécessitant aucune qualification, elle peut donc être négligée. Les étapes de **Stérilisation** et de **Préparation** quant à elles doivent être effectuées dans le même isolateur et requièrent un temps important (environ 15 minutes de stérilisation et entre 5 à 15 minutes de préparation). Chaque isolateur est une unité scellée qui protège l'opérateur des produits cytotoxiques, qui sont classés comme dangereux pour la santé et potentiellement cancérigènes et protège également les médicaments d'une contamination microbologique pendant la phase de **Préparation**. La Figure 1.4 présente trois modèles utilisés par l'UBCO entre 2018 et 2021.

Chacun des modèles d'isolateurs utilisés par l'UBCO possède deux postes de travail pour les pharmaciens. La production d'un médicament peut être anticipée en stérilisant son panier avant de recevoir la validation du médecin de la deuxième consultation. Si le traitement n'est pas validé, les produits stérilisés sont alors simplement rangés dans une armoire réfrigérée, car la stérilisation ne dégrade pas le produit. La livraison des médicaments est une étape-charnière pour l'UBCO. En effet, l'UBCO cherche à livrer autant que possible les préparations magistrales dans un délai compris entre 1h00 et 1h30 après la validation. La livraison a deux types de destinations :

- Livraison interne à l'hôpital où se situe l'UBCO (hôpital Bretonneau). Pour cela, des employés sont affecté à l'UBCO pour effectuer les livraisons dès que possible.
- Livraison externe vers d'autres hôpitaux. Cette livraison est la plus contraignante, car elle se fait en collaboration avec un autre service (Logicourse) dont les horaires

1.1. CONTEXTE



FIGURE 1.4 – Trois modèles différents d’isolateurs utilisés par l’UBCO entre 2018 et 2021.

et les trajets sont négociés à l’avance.

L’UBCO produit et livre des médicaments de chimiothérapie pour environ une vingtaine de services d’oncologie répartis entre trois hôpitaux. La Figure 1.5 montre une photo des deux hôpitaux livrés par une livraison externe.



FIGURE 1.5 – L’hôpital Gatien de Clocheville à gauche et l’hôpital Trousseau à droite.

La livraison externe étant soumise aux aléas des embouteillages et comportant une durée de trajet conséquente (entre 15 minutes et 45 minutes), il est nécessaire de la prendre en compte lors de la planification des productions des médicaments puisque l’objectif d’optimisation de l’UBCO est de réduire son retard de livraison. Le coût de production des médicaments dépend principalement de trois facteurs : le coût du matériel, le coût des produits cytotoxiques et le coût de la main-d’oeuvre.

Le coût du matériel est principalement de l’investissement à long terme (achat d’isolateurs, d’analyseur automatique, ...). Sa réduction dépend majoritairement de l’évolution technologique et du prix du marché. La diminution du coût de main-d’oeuvre peut diffi-

1.1. CONTEXTE

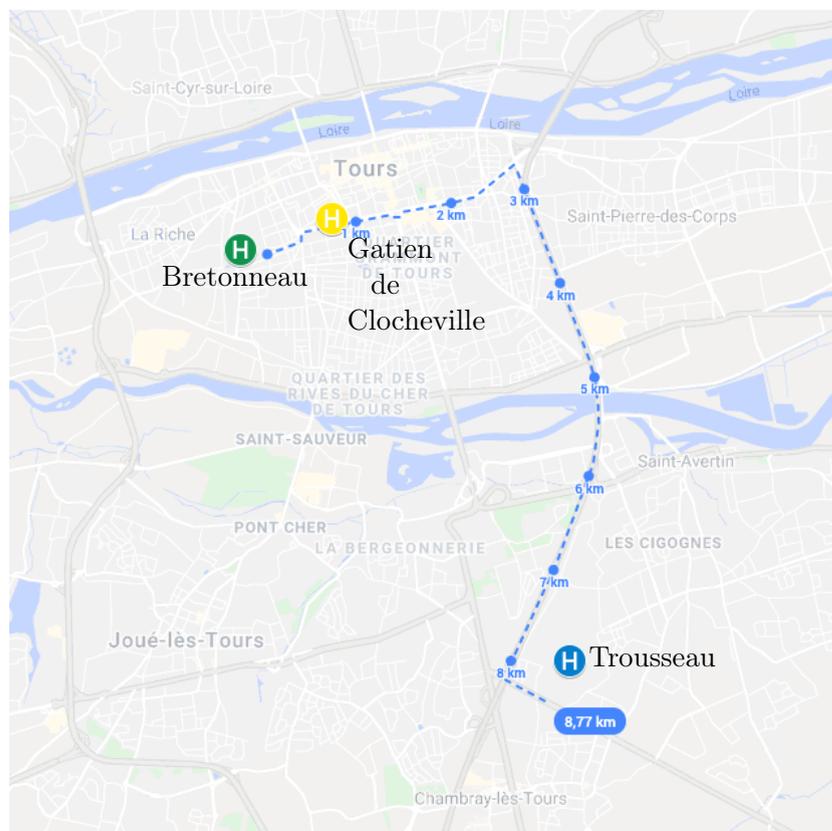


FIGURE 1.6 – A gauche, en vert, l'hôpital Bretonneau, à côté, en jaune, l'hôpital Gatiens de Clocheville et en bas à droite en bleu l'hôpital Trousseau.

cilement se faire sans détériorer la qualité de service. C'est donc sur le coût des produits cytotoxiques que l'UBCO souhaite agir en améliorant la gestion des stocks.

Chaque produit cytotoxique est fourni dans des petits flacons. Un flacon peut servir pour la préparation de plusieurs médicaments. Cependant, après ouverture, un flacon ne peut se conserver que pendant une durée limitée. Si un flacon se périmé alors qu'il n'était pas vidé, alors le reliquat est jeté et il est nécessaire d'acheter un nouveau flacon pour les préparations suivantes. L'UBCO souhaite donc réduire au maximum ses reliquats. Sachant qu'un flacon peut coûter entre quelques dizaines d'euros à plus de mille euros, une perte journalière de quelques millilitres peut se compter en centaines de milliers d'euros sur un an. Il est donc important de prendre en compte ces reliquats lors de la planification des préparations des médicaments de chimiothérapie.

1.2 Notions préliminaires

1.2.1 Complexité algorithmique et classes de complexité

La théorie de la complexité algorithmique est l'étude de la vitesse de croissance du coût en ressources (temps de calcul, espace mémoire, ...) des algorithmes en fonction de la taille des données d'entrée. Pour étudier la complexité temporelle d'un algorithme, on compte le nombre d'opérations dites élémentaires, effectuées lors de son exécution et on le compare avec la taille des données d'entrée qui est usuellement notée n . Compter avec exactitude le nombre d'opérations élémentaires peut être compliqué et souvent peu intéressant. C'est pour cela que l'on préfère appréhender la complexité temporelle par des ordres de grandeur vers l'infini. Sachant que la complexité temporelle d'un algorithme dépend non seulement de la taille des données d'entrée mais aussi des données elles mêmes. Par abus de langage, la complexité d'un algorithme représente la complexité temporelle au pire cas.

Notation Landau

En mathématiques, les comparaisons asymptotiques permettent d'ordonner et de classer les vitesses d'accroissements et de convergences de fonctions au voisinage de l'infini ou d'un point. La notation Landau, nommée d'après le mathématicien Edmund Landau, retranscrit le résultat d'une comparaison asymptotique de deux fonctions vers l'infini. Le tableau 1.1 décrit trois des notations les plus utilisées en théorie de la complexité et leurs significations.

Notation	Description informelle	$n \rightarrow +\infty$
$f(n) = O(g(n))$	f est dominée par g	$\exists k \in \mathbb{R}_+, f(n) \leq k \cdot g(n) $
$f(n) = \Omega(g(n))$	f domine g	$\exists k \in \mathbb{R}_+, f(n) \geq k \cdot g(n) $
$f(n) = \Theta(g(n))$	f est de l'ordre de g	$\exists(k, k') \in \mathbb{R}_+^2, k \cdot g(n) \leq f(n) \leq k' \cdot g(n) $

TABLE 1.1 – Notations Landau pour les comparaisons asymptotiques les plus utilisées en algorithmique.

La Figure 1.7 montre l'évolution du nombre d'opérations élémentaires d'un algorithme en fonction de sa complexité.

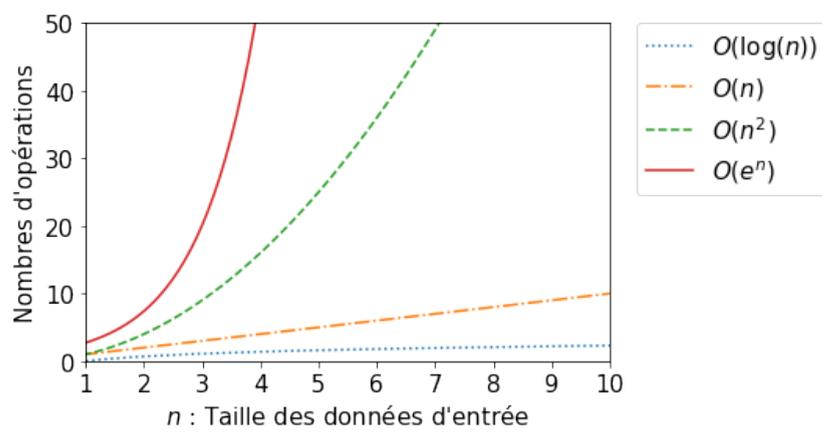


FIGURE 1.7 – Représentation de l'évolution asymptotique de différentes complexités définies par leur notation Landau.

Machine de Turing

Une machine de Turing est un concept développé par Alan Turing en 1936 qui permet de représenter de façon abstraite le fonctionnement d'une machine programmable.

La description d'une machine de Turing se présente comme suit :

- Un ruban de longueur infini composé de cases remplies de symboles appartenant à un alphabet prédéfini.
- Un lecteur pouvant lire le contenu des cases une par une.
- Un registre de table de transition. Chacune des tables de transition correspond à un état et détermine quelle action effectuer en fonction de l'état en cours du symbole lu par le lecteur.
- Une tête de réécriture permet de changer le symbole présent dans la case lue par un autre symbole du même alphabet.

La Figure 1.15 est une représentation graphique d'une machine de Turing.

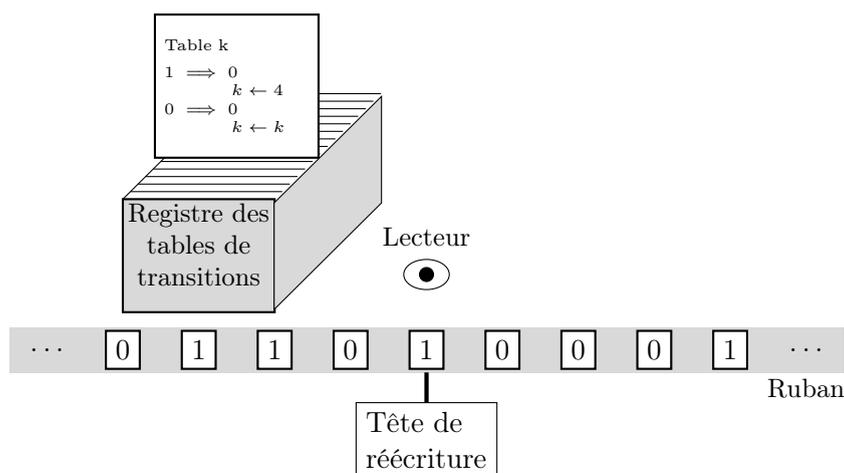


FIGURE 1.8 – Représentation schématique d'une machine de Turing.

En informatique théorique, la machine de Turing permet une étude poussée du comportement des algorithmes. Inspirée des théorèmes d'incomplétude de Kurt Gödel montrant l'indécidabilité de certaines propositions dans le formalisme arithmétique, la machine de Turing a permis de placer un formalisme de la calculabilité et de démontrer l'indécidabilité algorithmique de théories. Cependant, pour les problèmes de décision¹ décidables par une machine de Turing, le temps de décidabilité peut être variable. Il a donc été important de

1.

Definition 1.2.1 (Problème de décision). *Un problème de décision est un problèmes consistant à déterminer la véracité d'un énoncés mathématiques.*

classifier les problèmes de décision en fonction de temps de décidabilité par une machine de Turing.

Classification de complexité

Dans les années 1970, une classification des complexités des problèmes de décisions a été proposée suite aux travaux de Stephen Cook, de Richard Karp et de Leonid Levin. Les principales classes de complexité sont les classes **P**, **NP**, **NP-difficile** et **NP-complet** représentées dans la Figure 1.9 et définies ci-dessous :

- La classe **P** est l'ensemble des problèmes de décision décidables par une machine de Turing déterministe en temps polynomial par rapport à la taille des données d'entrée.
- La classe **NP**, est l'ensemble des problèmes de décision décidables par une machine de Turing non déterministe en temps polynomial par rapport à la taille des données d'entrée.
- Un problème fait partie de la classe **NP-difficile** si l'on peut réduire tous les problèmes de la classe **NP** en ce problème en temps polynomial par rapport à la taille des données d'entrée.
- La classe **NP-complet** est l'ensemble des problèmes à la fois **NP** et **NP-difficile**.

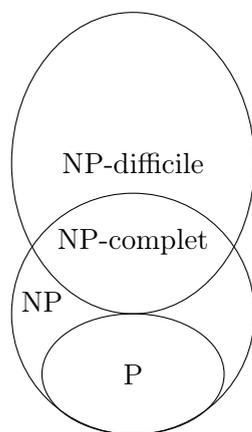


FIGURE 1.9 – Diagramme d'Euler représentant les principales classes de complexités.

Definition 1.2.2 (Problème de satisfaisabilité booléenne : SAT). *Soit une formule de logique propositionnelle, le problème SAT correspondant est le problème de décision suivant : existe-t-il une assignation des variables de la formule, telle que la formule soit vraie.*

Theorem 1.2.1 (Cook-Levin). *Le problème SAT est NP-complet.*

1.2.2 Les problèmes d'ordonnement

En optimisation, la théorie de l'ordonnement consiste à déterminer la planification optimale de la date d'exécution d'un ensemble de tâches.

Afin de faciliter la communication dans la communauté scientifique, Graham, Lawler, Lenstra et Rinnooy Kan ont proposé dans [Graham *et al.*, 1979] une standardisation de la notation des problèmes d'ordonnement en trois champs $\alpha|\beta|\gamma$: l'environnement de production α , les contraintes β et le critère d'optimisation γ .

L'environnement de production correspond à l'ensemble des ressources pouvant exécuter les tâches. Ces ressources sont plus communément appelées « machines ». Dans la littérature, il existe sept environnements classiques de production repartis en deux catégories.

Les problèmes d'ordonnement d'ateliers :

Chaque tâche est décomposée en plusieurs sous-tâches nommées opérations. La fin d'exécution d'une tâche est alors définie comme la date à laquelle toutes ses opérations ont été finies d'être exécutées.

- flow-shop (F) ou atelier à cheminement unique : pour chaque tâche, ses opérations passent sur les machines dans le même ordre.
- job-shop (J) ou atelier à cheminements multiples : pour chaque tâche, ses opérations passent sur les machines dans un ordre qui lui est propre.
- open-shop (O) ou atelier à cheminement libre : pour chaque tâche, ses opérations passent sur les machines dans un ordre non fixé.

Les problèmes d'ordonnement de tâches indépendantes :

Un ensemble de machines peut exécuter en parallèle les tâches. Ici les tâches ne sont constituées que d'une seule opération.

- une machine (1) : l'environnement de production est composé d'une seule machine.
- machines parallèles identiques (P) : le temps d'exécution des tâches est le même quelle que soit la machine.
- machines parallèles uniformes (Q) : le temps d'exécution des tâches dépend de la vitesse de la machine qui l'exécute.
- machines parallèles non reliées (R) : le temps d'exécution d'une tâches est variable selon la machine qui l'exécute.

La Figure 1.10 représente les relations entre les sept environnements classiques, d'un côté, en jaune, les différents problèmes d'ordonnement d'atelier et de l'autre, en rouge, les problèmes d'ordonnement à machines parallèles.

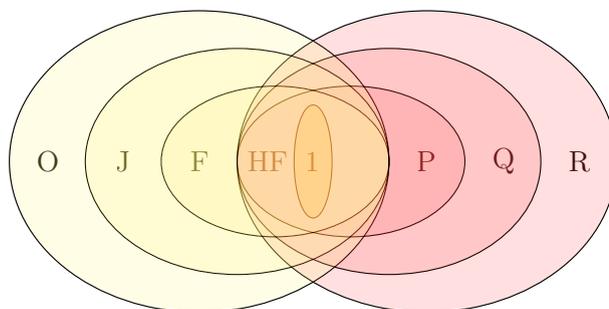


FIGURE 1.10 – Diagramme d’Euler représentant les environnements de production des problèmes d’ordonnancement.

Les contraintes temporelles pouvant être imposées à l’exécution d’une tâche j sont nombreuses, en voici une liste non exhaustive :

- Date de début au plus tôt r_j : l’exécution de la tâche j doit commencer après la date r_j .
- Temps d’exécution constant $p_j = p$: le temps d’exécution est identique pour toutes les tâches.
- Date de fin souhaitée d_j : l’exécution de la tâche j doit se terminer avant la date d_j .
- Précédence $prec$: il existe des conditions de précédences entre les tâches.

Les critères d’optimisation des problèmes d’ordonnancement se basent sur les dates de fin d’exécution des tâches. Chaque critère peut être considéré par son cas au pire (Maximum) ou sa moyenne (Total).

Nom	Notation	Total	Maximum
Date de fin d’exécution	C_j	$\sum_{j \in J} C_j$	$C_{\max} = \max_{j \in J} \{C_j\}$
Retard algébrique	$L_j = C_j - d_j$	$\sum_{j \in J} C_j$	$L_{\max} = \max_{j \in J} \{L_j\}$
Avance	$E_j = \max\{0, d_j - C_j\}$	$\sum_{j \in J} E_j$	$E_{\max} = \max_{j \in J} \{E_j\}$
Retard	$T_j = \max\{0, C_j - d_j\}$	$\sum_{j \in J} T_j$	$T_{\max} = \max_{j \in J} \{T_j\}$
Indicateur de retard	$U_j = 1$ si $C_j > d_j$, 0 sinon	$\sum_{j \in J} U_j$	

TABLE 1.2 – Critères d’optimisation classiques des problèmes d’ordonnancement

Le tableau 1.2 montre les principaux critères de la littérature des problèmes d’ordonnancement : la date de fin d’exécution, le retard algébrique, l’avance, le retard et l’indicateur de retard.

Un problème d’ordonnancement d’atelier de type Flow-shop à deux machines avec des dates de début au plus tôt et le retard total comme critère d’optimisation se note alors

$$F2|r_j, d_j| \sum_{j \in J} T_j.$$

Voici quelques résultats de complexité utiles pour l'appréhension des différents problèmes traités dans cette thèse :

- $1|p_j = p; r_j| \sum_{j \in J} T_j$ est polynomial [Baptiste, 2000].
- $P|p_j = p; r_j| \sum_{j \in J} C_j$ est polynomial [Simons, 1983].
- $P|p_j = p; r_j| \sum_{j \in J} T_j$ est polynomial [Brucker et Kravchenko, 2005].
- $1|| \sum_{j \in J} T_j$ est NP-difficile [Lawler, 1977, Du et Leung, 1990].
- $1|r_j| \sum_{j \in J} C_j$ est NP-difficile au sens fort [Lenstra *et al.*, 1977].
- $1|r_j|L_{\max}$ est NP-difficile au sens fort [Lenstra *et al.*, 1977].
- $F2||L_{\max}$ est NP-difficile au sens fort [Lenstra *et al.*, 1977].
- $F2|r_j| \sum_{j \in J} C_j$ est NP-difficile au sens fort [Lenstra *et al.*, 1977].

Nous nous intéresserons, dans cette thèse, au problème d'ordonnement d'atelier de type flow-shop, au problème d'ordonnement à machines parallèles et à leur hybridation nommés problèmes d'ordonnement d'atelier de type flow-shop hybride. La représentation classique d'une solution d'un problème d'ordonnement est l'utilisation d'un diagramme de Gantt. Ce diagramme présente pour chaque machine une frise chronologique des exécutions. L'exécution d'une tâche est alors représentée naturellement par un rectangle dont ses arêtes correspondent aux dates de début d'exécution, dates de fin d'exécution et la durée d'exécution.

Problème d'atelier de type flow-shop (F)

Le problème d'ordonnement d'atelier de type flow-shop est un problème d'ordonnement d'atelier que l'on retrouve dans de nombreux contextes, principalement dans l'industrie. Ce problème est défini comme suit : soit J un ensemble de tâches. Chaque $j \in J$ correspond à un même ensemble d'opérations $O = \{o_1, o_2, \dots, o_{|O|}\}$ devant être ordonnées dans un ordre prédéfini sur les machines correspondantes $m_1, m_2, \dots, m_{|O|}$. Chacune des machines correspond à un étage de production.

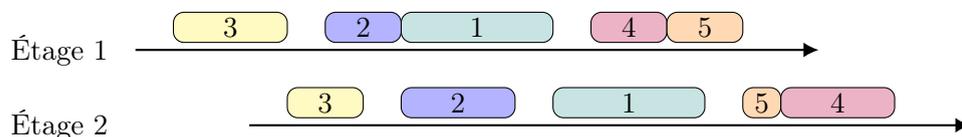


FIGURE 1.11 – Diagramme de Gantt d'une solution d'un problème d'ordonnement d'atelier de type flow-shop.

La Figure 1.11 représente la solution d'un problème d'atelier de type flow-shop avec 2 machines. On peut voir que les opérations peuvent être identifiées par leur tâche correspondante et leur machine dédiée.

Problème d'ordonnancement à machines parallèles identiques (P)

Le problème d'ordonnancement à machines parallèles est caractérisé par un ensemble de machines parallèles et identiques M qui doivent exécuter un ensemble de tâches J .

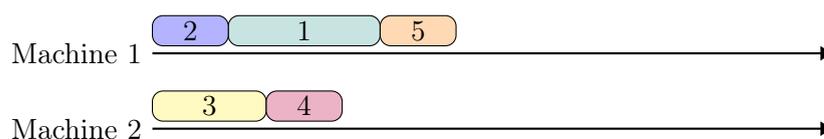


FIGURE 1.12 – Diagramme de Gantt d'une solution d'un problème d'ordonnancement à machines parallèles avec 5 tâches.

Problème d'ordonnancement d'atelier de type flow-shop hybride (HF)

Un problème d'ordonnancement d'atelier de type flow-shop hybride est une variante du problème d'ordonnancement d'atelier de type Flow-shop où chaque étage peut posséder plusieurs machines pour une exécution en parallèle. Correspondant plus aux structures actuelles des entreprises, les études sur les problèmes d'ordonnancement d'atelier de type Flow-shop hybride sont de plus en plus nombreuses depuis les années 90 et concernent de nombreux contextes [Vignier *et al.*, 1999, Ruiz et Vázquez-Rodríguez, 2010, Tosun *et al.*, 2020].

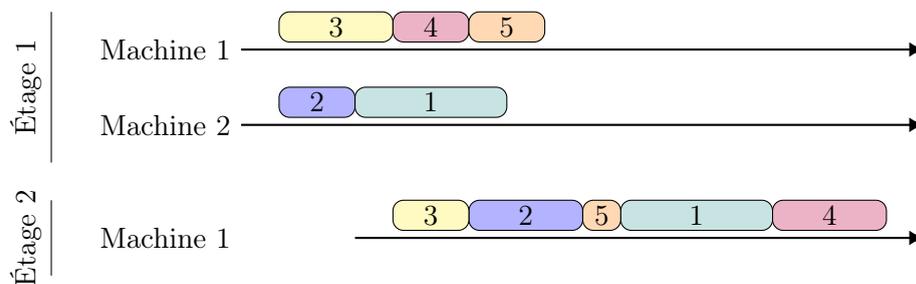


FIGURE 1.13 – Diagramme de Gantt d'une solution d'un problème d'ordonnancement d'atelier de type Flow-shop hybride à deux étages avec deux machines au premier étage, une seule machine au deuxième étage et 5 tâches.

1.2.3 Les problèmes de livraison

Depuis le problème des sept ponts de Königsberg² résolu par Euler en 1735 et le problème du cavalier et de cycle Hamiltonien³, la recherche de cycles contraints dans les graphes s'est imposée. De nombreux problèmes de livraisons ont émergé et font référence dans la communauté.

Le problème du voyageur de commerce (TSP)

Le problème du voyageur de commerce est le problème où un vendeur cherche à visiter un ensemble de clients exactement une fois en minimisant le trajet parcouru. Ce problème, aussi nommé Traveling salesman problem, revient à trouver le cycle Hamiltonien de plus courte distance dans un graphe. Ce problème fait partie des 21 problèmes prouvés NP-complet dans [Karp, 1972].

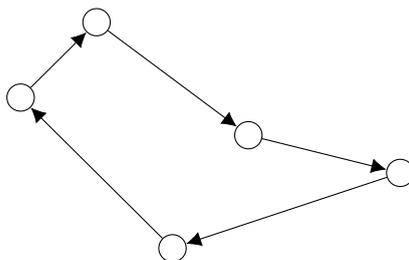


FIGURE 1.14 – Une solution d'un TSP à 5 clients.

Le problème de tournées de véhicules (VRP)

Le problème de tournées de véhicules est une version du TSP avec une infinité de livreurs à disposition. Les livreurs sont nommés "véhicules" et partent tous d'un même dépôt. L'objectif est de minimiser la distance totale parcourue par l'ensemble de la flotte de véhicules utilisés. Chaque client (sommets du graphe) doit être visité une et une seule fois par un véhicule et chaque véhicule doit revenir à son point de départ. Ce problème est NP-complet. Le VRP et ses variantes font partie des problèmes les plus étudiés dans la littérature de recherche opérationnelle [Mor et Speranza, 2020, Cattaruzza *et al.*, 2018].

2. Déterminer la faisabilité de traverser chacun des sept ponts de la ville de Königsberg une et une seule fois et revenir à son point de départ.

3. Un cycle Hamiltonien dans un graphe est un cycle passant une et une seule fois par tous les sommets du graphe.

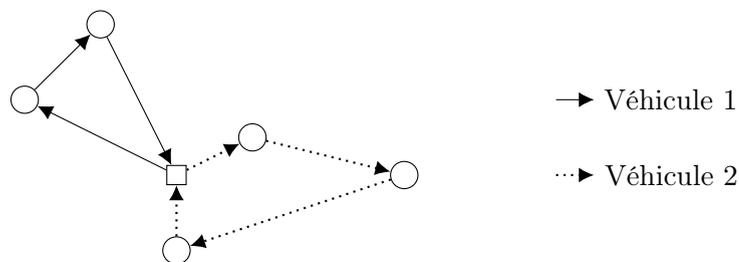


FIGURE 1.15 – Une solution à deux véhicules d'un VRP à 5 clients.

Afin de s'approcher le plus possible des problèmes réels, de nombreuses contraintes ou généralisations ont été ajoutées au VRP. Voici quelques variantes très étudiées de la littérature :

- Capacited Vehicle Routing Problem (C-VRP) : la capacité de chargement des véhicules est limitée.
- Vehicle Routing Problem with Time Windows (VRP-TW) : chaque client doit être livré dans une fenêtre de temps qui lui est propre.
- Limited Vehicle Routing Problem (limited VRP) : le nombre de véhicules disponibles est limité.
- Heterogenous Vehicle Routing Problem (Hete VRP) : la flotte de véhicules est hétérogène soit par la vitesse ou la capacité ou encore les clients accessibles.
- Multi-Trip Vehicle Routing Problem (MT-VRP) : le nombre de véhicules est limité mais chacun d'entre eux peut effectuer plusieurs trajets.
- Pickup and delivery Problem (PD) : tournées combinant le ramassage et la livraison des biens.

1.2.4 Les problèmes intégrés

Les problèmes d'ordonnement de la production et les problèmes de livraison ont été largement étudiés dans la littérature, mais ils sont souvent résolus indépendamment. Pourtant il est souvent nécessaire de considérer les contraintes de la livraison lors de l'ordonnement de la production. La forte centralisation des productions, les problématiques de stockage et les délais de livraison de plus en plus serrés, renforcent l'interdépendance entre la production et la livraison. Pour qu'un produit puisse être livré, il se doit d'être fini de produire lorsqu'un véhicule de livraison est disponible. Le gain d'optimisation possible par un traitement intégré de la chaîne logistique plutôt que séquentiellement et séparément est de plus en plus important, particulièrement lorsque le critère d'optimisation dépend des dates de livraisons.

1.3 Étude préliminaire

Afin de cerner au mieux les enjeux des problématiques rencontrées par l'UBCO, nous proposons un tour d'horizon de la littérature autour de l'optimisation pour les chimiothérapies, un historique des contributions en lien avec l'UBCO et une simulation numérique. Cette étude préliminaire sert à mettre en lumière le contexte et justifier le travail de recherche effectué durant cette thèse.

1.3.1 L'optimisation pour les chimiothérapies

Dans la littérature de l'optimisation pour les chimiothérapies, il existe deux approches différentes, soit optimiser l'efficacité du traitement ou alors optimiser la logistique. Pour chacune de ces deux approches il existe une grande diversité de problématiques : optimiser les dosages, optimiser les mélanges, optimiser la planification des rendez-vous patients, optimiser la planification de la production des préparations, ...

Dans [Shi *et al.*, 2014, Sbeity et Younes, 2015], une vue générale des études sur l'optimisation du protocole de traitement par chimiothérapie d'avant 2014 est détaillée. En effet, l'une des grandes difficultés liées au traitement par chimiothérapie est un juste équilibre. Un traitement fort et rapide peut avoir de grands effets indésirables. Un traitement doux et long à l'inverse peut rester inefficaces dû au développement progressif de résistances des cellules cancéreuses face aux médicaments. Les objectifs concrets des études rapportées se basent principalement sur la réduction de la population des cellules cancéreuses et l'augmentation du bien-être du patient (maximiser la durée de survie, minimiser la population de cellules saines détruites, minimiser la toxicité du traitement, ...).

La planification des consultations d'un point de vue logistique est une problématique plus récente. En France, le cas de l'Institut de Cancérologie de la Loire à fait l'objet de plusieurs études [Sadki *et al.*, 2010, Sadki *et al.*, 2011, Sadki *et al.*, 2013] concernant la planification des consultations. Récemment, [Garaix *et al.*, 2020] étudie l'optimisation de la planification des consultations dans une approche stochastique contrainte au respect d'un protocole de soins. Les objectifs abordés sont la minimisation du nombre d'heures supplémentaires et de la date de fin de service. Réduire l'attente des patients est l'un des objectifs principaux dans les problèmes de planification pour le service de santé. Plusieurs études se sont penchées sur cette problématique dans le cadre des traitements de chimiothérapies [Dobish, 2003, Ahmed *et al.*, 2011, Yokouchi *et al.*, 2012].

Quant à la planification de la production des médicaments pour les chimiothérapies, peu d'études sont présentes dans la littérature de la recherche opérationnelle. À notre connaissance, la première étude s'intéressant à la problématique de la chaîne logistique de production et distribution des médicaments de chimiothérapie est [Chahed *et al.*, 2009]. Les auteurs abordent un problème simplifié de production et d'administration de chimiothérapie dans un contexte de soins à domicile. Le problème est limité à une machine et à un livreur qui n'effectue qu'un seul trajet. L'objectif d'optimisation est de minimiser le temps de trajet total. Plus récemment, [Lee *et al.*, 2014a] traite d'un problème de production et d'administration de traitements médicaux nécessitant des substances radioactives d'une demi-vie de quelques minutes. Le problème d'ordonnancement est traité comme un type particulier de problème de Bin-packing. Pour la livraison, une flotte de véhicules hétérogènes est disponible.

1.3.2 Historique de l'optimisation à l'UBCO

Le LIFAT travaille depuis près de 15 ans avec l'UBCO pour optimiser sa qualité de service. En 2010, une première contribution, [Mazier *et al.*, 2010], a proposé d'optimiser le problème d'ordonnancement des préparations magistrales de chimiothérapie en modélisant la production par un problème d'ordonnancement à machines parallèles identiques où chaque machine correspond à un pharmacien préparant les injections. L'algorithme proposé a été implémenté dans un logiciel d'aide à la décision nommé Planif qui est utilisé quotidiennement par l'UBCO. Son impact a été analysé en 2011 dans [Kergosien *et al.*, 2011]. La prise en compte de la livraison a été la première fois évoquée dans [Kergosien *et al.*, 2017] où les auteurs considèrent encore un problème d'ordonnancement à machines parallèles identiques, mais avec un livreur pouvant effectuer plusieurs tournées dans la journée. Pour résoudre le problème, il est proposé une méthode de décomposition de Benders. Comme le montrera la section suivante, la modélisation de la production par un problème d'ordonnancement à machines parallèles identiques n'est pas suffisante pour représenter correctement l'environnement de production de l'UBCO. De plus, en pratique, entre trois et cinq personnes sont en charge de la livraison chaque jour. Il est donc important de généraliser les hypothèses de modélisations.

1.3.3 Les limites de la production

Afin d'optimiser la production à l'UBCO, il est nécessaire de mettre en place des capteurs à chacune des étapes de la production. L'entreprise Eticsys a mis en place la traçabilité des préparations l'UBCO à l'aide de lecteurs code-barres manuels permettant un suivi des étapes de préparation, de contrôle et de livraison. Grâce aux données collectées, il est possible de re-planifier rapidement en fonction des aléas. Il est donc possible de développer des algorithmes dit "en ligne" pouvant adapter la planification en temps réel.

Pour développer ce genre d'algorithmes, il est important de connaître les étapes les plus sensibles de la production. En effet, l'optimisation de la planification est contrainte par les limites de l'environnement de production. C'est pour cela que nous proposons une simulation numérique des flux entrants et sortants de chacune des principales étapes de production afin de déterminer les axes les plus à même d'être optimisés. La simulation considère un flux entrant de préparations magistrales illimitées et sans prise en compte de date de début au plus tôt ou de date de fin souhaitée, ceci en essayant de se rapprocher le plus possible des conditions réelles :

- L'environnement de production est constitué de deux postes de préparation des plateaux, de quatre isolateurs identiques avec chacun deux stations de travail et d'un analyseur automatique. Ce dernier n'est pas considéré dans l'affectation des pharmaciens au poste de travail, car il nécessite une présence à temps plein.
- Le nombre de pharmaciens présents pour les préparations dépend de l'heure de la journée.
- La préparation d'un plateau prend 1 minute.
- La stérilisation se fait par lots de 12 plateaux maximum et dure 15 minutes.
- La préparation a une durée moyenne de 10 minutes.
- Le contrôle prend 2 minutes par préparation.
- Une journée de travail fait 8 heures avec une pause le midi (0 correspondant à 8h et 600 à 18h).
- Un plateau n'est stérilisé que s'il est possible de le contrôler avant la fin de journée.

La règle d'affectation des pharmaciens entre les postes de préparation des plateaux et des isolateurs est décrit dans la Figure 1.16 : priorité à la préparation des plateaux s'il y a moins des plateaux qu'une demi-charge de stérilisateur (6 plateaux) en attente de stérilisation, sinon priorité à la préparation d'injection dans un isolateur si des injections sont en attente de préparation et sinon préparation de plateau tant qu'il y a moins de plateaux prêts que la quantité pour charger tous les stérilisateur en même temps (36 plateaux). Si aucune de ces conditions n'existe, alors le pharmacien est dit "libre" et il peut s'occuper des autres tâches post processus (rangement, nettoyage, etc).

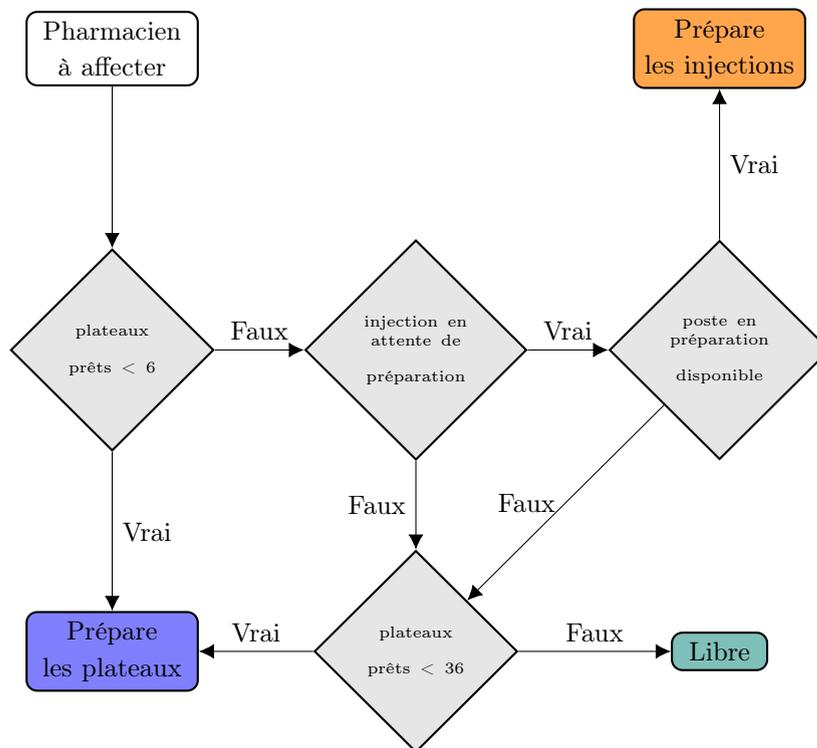


FIGURE 1.16 – Règles d’affectation des pharmaciens à la préparation des plateaux ou à la préparation des injections.

La figure 1.17 représente la quantité de préparation présente à chaque étape de la production (en plateau, en stérilisation, en préparation, en contrôle ou libéré) et l’affectation des pharmaciens aux différents postes (préparation des plateaux, préparation des injections ou libre). Les injections dites "en plateau" sont les injections dont le plateau est en attente pour la stérilisation. Les injections dites "en stérilisation" sont en cours de stérilisation. Les injections dites "en préparation" sont celles sorties d’un stérilisateur et attendant d’être préparées par un pharmacien. Les injections dites "en contrôle" sont celles sorties d’un isolateur et attendant d’être contrôlé sur l’analyseur automatique. Les injections dites "libérés" sont celles qui ont eu leur contrôle d’effectué et attendant d’être livrées.

1.3. ÉTUDE PRÉLIMINAIRE

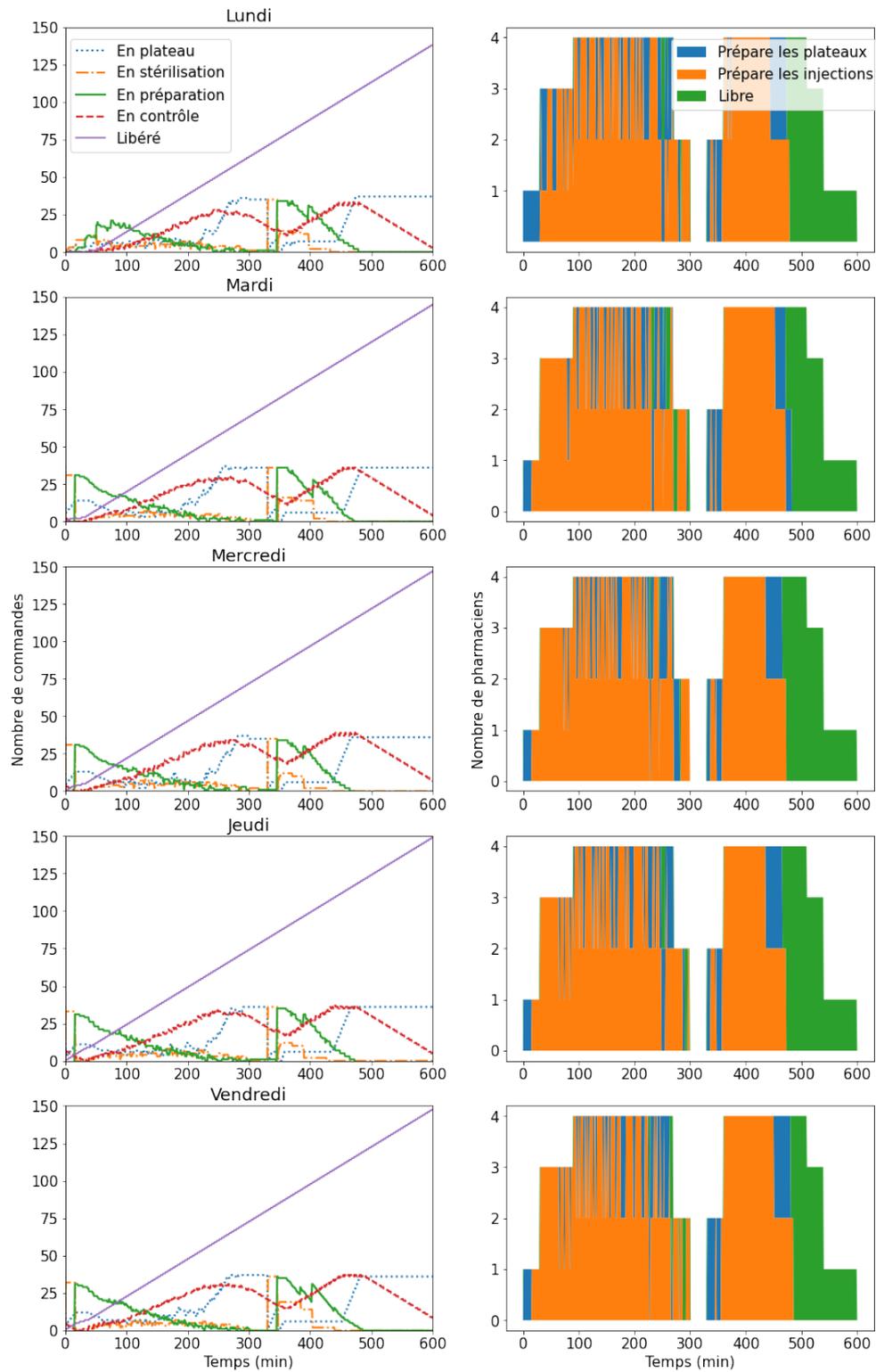


FIGURE 1.17 – Étude des flux des préparations magistrales de la préparations des plateaux à leur libération.

Tout d'abord, il faut noter que les résultats de la simulation semblent cohérents avec la réalité terrain. En effet, l'UBCO livre environ 150 préparations magistrales par jour et la quasi-totalité des préparations est faite avant 16h. Cette étude des flux nous permet de constater le goulot d'étranglement de la production qui est autour du contrôle. En effet, il n'existe pas de période d'inactivité sur l'analyseur à l'exception du début de journée le temps que les premières préparations soient prêtes. Il ne sera donc pas possible de produire et livrer à temps plus que 150 préparations magistrales sur une journée se terminant avant 18h. Ce sera donc autour des 150 préparations par jour que l'optimisation sera la plus pertinente. Un nombre d'isolateurs supérieur à 2 n'apporte pas d'augmentation du débit de production même avec plus de pharmaciens disponibles. Cependant, il accordera plus de flexibilité lorsque les dates de début au plus tôt et les dates de fin souhaitées seront à prendre en compte.

1.4 Plan du manuscrit de thèse

Suite à la présentation détaillée du contexte de l'environnement de production de l'UBCO de ce premier chapitre, cette thèse s'intéresse à trois problématiques différentes issues d'un même contexte. Chacune d'entre elles est présentée dans un chapitre dédié. Le dernier chapitre conclut ce manuscrit puis présente, les travaux en cours et les futures perspectives.

1.4.1 Problème intégré de la production et de la livraison

Tout d'abord, une considération intégrée du problème qui soit la plus proche du réel doit être faite. Le deuxième chapitre se focalise donc sur le développement de méthodes heuristiques pouvant être reprises par la société Eticsys pour une mise en œuvre à l'UBCO. Nous présentons d'abord les hypothèses choisies pour la modélisation du problème.

Suite à une modélisation formelle du problème en un problème intégré flow-shop hybride et tournées de véhicules multi-trajets, nous présentons un état de l'art des problèmes intégrés production et livraison de la littérature. Nous montrons que le problème traité est nouveau et porte sur des points souvent négligés dans la littérature. Son originalité réside sur son objectif d'optimisation (le retard Total), la contrainte de date de début au plus tôt des tâches et la combinaison de deux environnements complexes (production et livraison).

Nous détaillons une modélisation mathématique du problème. Cependant, sa grande complexité rend toute résolution exacte extrêmement difficile pour de grandes instances. Nous présenterons différentes heuristiques pouvant résoudre des instances de taille réelle en un temps de calcul plus que raisonnable. Afin de pouvoir évaluer les heuristiques, nous

modélisons les processus de décision actuellement utilisés. Pour cela, nous détaillons un algorithme de référence implémentant la plupart des règles connues et appliquées. Une première proposition d’heuristique multi-niveau permet de montrer l’importance de considérer le problème de façon intégrée. En effet, celle-ci est construite sur une base commune avec l’algorithme de référence, mais avec une modification des priorités d’ordonnement en fonction d’une livraison pré-calculée.

L’un des inconvénients de l’heuristique à multi-niveau est qu’elle impose de connaître à l’avance le nombre de trajets à effectuer dans la journée. Nous proposons donc une version avec configuration automatique, utilisant une double descente de gradient. Les temps d’exécution de cette heuristique étant suffisamment courts par rapport au temps allouable, nous avons décidé d’améliorer encore l’heuristique en ajoutant une part d’aléatoire lors de la constitution des trajets et d’effectuer plusieurs itérations jusqu’au temps imparti atteint.

Pour étudier l’efficacité de nos heuristiques et l’intérêt de chacune des améliorations, nous présentons deux types d’instances : le premier proche du contexte de l’UBCO et le second avec plus d’aléatoire. Ceci permettra de mesurer l’impact de nos méthodes à la fois en condition réelle à l’UBCO, mais aussi si elles sont appliquées à des contextes plus larges comme les chimiothérapies à domicile.

Nous étudions le temps de calcul de chacune des heuristiques, mais aussi la comparaison des écarts relatifs par rapport à la meilleure solution trouvée et un classement par performance. Comme les résultats des heuristiques peuvent dépendre de plusieurs critères, nous présentons les résultats avec plusieurs tailles d’instances et plusieurs tailles de flotte de véhicules disponibles pour la livraison.

1.4.2 Optimisation de la livraison

Le troisième chapitre se focalise sur la partie livraison. En effet, dans le chapitre précédent, l’efficacité des différentes heuristiques est étudiée uniquement comparativement l’une avec les autres. C’est pour cela que nous présenterons une méthode de résolution exacte pour la partie livraison.

Pour cela, nous posons certaines hypothèses importantes. Le problème de tournées étudié étant trop complexe pour une résolution exacte pour des tailles d’instance raisonnables, nous considérons que la constitution et l’ordre des trajets sont connus à l’avance.

Dans la littérature des problèmes de tournées de véhicules, les méthodes de résolution exactes sont peu nombreuses et utilisent principalement des propriétés liées à l’objectif d’optimisation qui est la distance totale parcourue. Or, dans notre cas l’objectif est le retard total, ce qui nous oblige à considérer d’autres directions.

Pour établir l’intérêt et l’efficacité de notre méthode exacte, nous proposons une mo-

délisation linéaire en nombres mixtes du problème complet et de deux sous-problèmes. Les solveurs actuels étant de plus en plus performants, il est nécessaire d'établir la pertinence de notre méthode dans son entièreté.

Nous détaillons alors notre méthode arborescente. Cette dernière repose sur la décomposition en trois niveaux de décision : l'affectation d'un trajet à un véhicule, sélectionner la séquence de livraison de chacun des trajets, et la construction de séquences de livraison non-dominées. Pour chacun des niveaux, nous présentons une méthode de branchement, des bornes et des propriétés de coupe.

La méthode établie dans ce chapitre peut être appliquée à des contextes très variés, c'est pour cela que nous utilisons des instances de la littérature des problèmes de tournées de véhicules que nous adaptons. Les résultats expérimentaux montrent les forces et faiblesses de la méthode et aussi la pertinence de l'heuristique de construction des séquences de livraison et d'affectation de trajet, choisis lors du chapitre précédent.

1.4.3 Prise en compte de ressources consommables et périssables

Le quatrième chapitre traite un objectif économique. Lors de la préparation des médicaments de chimiothérapie, des ressources périssables au coût important sont utilisées. Parfois, il y a des pertes dues à une gestion de ressource consommable non optimisée. C'est pour cela que nous étudions un problème intégré production et gestion de ressource consommable.

Les hypothèses que nous détaillons sont fortement basées sur le contexte de l'UBCO. Celui-ci nous entraînera à modéliser le problème comme un problème intégré d'ordonnement à machines parallèles identiques et de Bin-packing avec des contraintes de compatibilités. La littérature étant peu abondante pour ce type de problème nous nous attardons plus sur la littérature des problèmes de Bin-packing avec contraintes de compatibilités.

Suite à la modélisation mathématique du problème dans son ensemble et du sous-problème de Bin-packing avec contraintes de compatibilités, nous proposons une matheuristique. Cette méthode, allie la puissance d'une modélisation mathématique pour la résolution du problème de rangement et la rapidité d'une recherche Tabou pour l'ordonnement.

L'évaluation des performances de la matheuristique est faite sur des instances inspirées de données issues de l'UBCO. La qualité de la méthode est donnée par l'étude de sa convergence et l'impact de l'ordonnement sur le coût de la production.

1.4.4 Mise en place des solutions et poursuite de la recherche

Le manuscrit se termine par une conclusion sera faite sur les différents problèmes abordés, méthodes proposées et recherches effectuées. De plus, nous préciserons les parties qui seront mises en œuvre concrètement pour l'UBCO. Plusieurs pistes de recherches déjà entamées et des perspectives sont aussi évoquées.

1.4. PLAN DU MANUSCRIT DE THÈSE

Chapitre 2

Optimiser la production et la livraison

Sommaire

2.1	Présentation de la problématique	52
2.1.1	Hypothèses de modélisation	52
2.1.2	Positionnement dans l'état de l'art	56
2.2	Formulation mathématique	64
2.2.1	Paramètres d'entrées	64
2.2.2	Variables de décision	65
2.2.3	Contraintes	66
2.3	Méthodes de résolution heuristiques	68
2.3.1	Reference Algorithm	71
2.3.2	Multi-level heuristic	73
2.3.3	Bilevel Gradient Descent	74
2.3.4	Greedy Randomized Adaptive Search Procedure	77
2.4	Expérimentations numériques	78
2.4.1	Générateur d'instances	78
2.4.2	Temps de calcul	81
2.4.3	Écarts relatifs	82
2.4.4	Classement	88
2.4.5	Combien d'itérations pour le GRASP	88
2.5	Conclusion	92

2.1 Présentation de la problématique

Comme présenté dans le chapitre précédent, pour pouvoir réellement optimiser la production à l'UBCO, il est important de traiter le problème de manière intégrée et de développer une méthode en mesure de recalculer rapidement une bonne solution en fonction des décisions qui ont été prises et des aléas survenus. Il est régulier que la validation d'une préparation magistrale soit en retard par rapport à la date estimée. De même, les embouteillages ont un fort impact sur le temps de trajet des véhicules, ce qui a pour conséquence de retarder le départ des tournées suivantes.

Nous nous concentrons sur le développement d'heuristiques pour la résolution du problème intégré de production et de livraison. En effet, les heuristiques permettent d'obtenir rapidement des solutions de bonne qualité pour des problèmes complexes. Nous fixons à 5 minutes le délai raisonnable d'exécution des méthodes, cette durée correspondant à la granularité temporelle utilisée à l'UBCO pour l'évaluation des retards. Cependant, nous développons tout de même des méthodes capables de répondre à une exigence à la minute près.

Les environnements de production et de livraison pouvant évoluer dans le temps et la chimiothérapie à domicile se développant de plus en plus, la livraison peut se complexifier dans le futur ; et la demande de préparations magistrales pour les chimiothérapies dépend de l'incidence du cancer et du développement de nouveaux traitements. Il est donc important de prendre en considération les possibles évolutions lors de l'évaluation de nos méthodes. Nous évaluons donc nos méthodes sur un panel d'environnements proches de celui de l'UBCO et un panel d'environnements plus général.

2.1.1 Hypothèses de modélisation

Nous modélisons le problème de production et de livraison de préparation magistrale de chimiothérapie comme l'intégration d'un problème d'ordonnancement d'atelier de type flow-shop hybride (HF) à trois étages avec des dates de début au plus tôt pour le deuxième étage et d'une variante du problème de tournée de véhicules à trajets multiples (MT-VRP) avec des dates de fin souhaitées pour la livraison.

Une préparation magistrale de chimiothérapie est représentée par une tâche à exécuter et à livrer. Une fois les tâches exécutées, elles doivent être livrées et un problème de tournée de véhicule à trajets multiples doit être résolu.

Le problème traité est alors modélisé comme un ensemble de tâches J à produire et à livrer. Chaque tâche $j \in J$ est destinée à un patient situé dans un service d'oncologie u_j pour une date de fin souhaitée d_j et est divisée en trois opérations de production ordonnées

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

(Voir la Section 1.1.3 pour plus de détails) :

1. la **Stérilisation** d'une durée constante p^{Ste} ,
2. la **Préparation** (deuxième étage du HF) d'une durée spécifique p_j^{Prep} ne pouvant débuter qu'après la date de début au plus tôt r_j ,
3. et le **Contrôle** (troisième étage du HF) de durée constante p^{Ctrl} .

Les contraintes sanitaires imposent que la **Stérilisation** et la **Préparation** soient effectuées dans un même espace stérile nommé isolateur.

Le site de production est composé d'un ensemble I d'isolateurs parallèles et une machine de contrôle.

Chaque isolateur $i \in I$ est caractérisé par le nombre maximum de tâches stérilisables simultanément Q_i et un nombre de machines parallèles identiques m_i (c'est-à-dire le nombre de postes de travail pour un pharmacien) pour l'étape de **Préparation**. La **Stérilisation** est supposée faite en continu i.e. un lot de stérilisation est lancé toutes les p^{Ste} minutes. Cette hypothèse se justifie par l'absence de contrainte de début au plus tôt pour cette étape ou de temps d'attente limite avant le début de la **Préparation**. Stériliser en avance n'a donc pas d'impact négatif sur le reste de la planification.

Chaque véhicule $v \in V$ a une capacité illimitée et peut effectuer plusieurs trajets. Un trajet correspond à un lot de tâches à livrer dans un cycle partant du site de production, passant par chacun des services d'oncologie concernés et revenant au site de production. Un trajet ne peut donc commencer avant la fin de **Contrôle** de chacune des tâches qui le composent. Le temps de trajet entre le service d'oncologie u_j et le service d'oncologie $u_{j'}$ est noté $tt_{j,j'}$.

Pour chaque tâche $j \in J$, on note par :

- C_j^{Ste} sa date de fin de **Stérilisation**,
- C_j^{Prep} sa date de fin de **Préparation**,
- C_j^{Ctrl} sa date de fin de **Contrôle**,
- D_j sa date de livraison à son service d'oncologie u_j .

Chaque tâche doit donc être affectée à un isolateur pour sa **Stérilisation** (premier étage du HF), puis à l'une des machines associées à cet isolateur pour sa **Préparation**.

Elle doit ensuite subir un **Contrôle** sur la machine de contrôle. La préemption des opérations des tâches n'est pas autorisée. Une fois contrôlée, la tâche doit être affectée à un lot de livraison pour sa livraison par un des véhicules de la flotte disponible.

La fonction objectif du problème est de minimiser le retard total de livraison :

$$\text{MIN} \sum_{j \in J} T_j$$

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

où T_j est le retard de livraison de tâche j défini par $T_j = \max(0, D_j - d_j)$ et comme défini à la Section 1.2.2.

Le problème consiste donc à : affecter et ordonnancer les stérilisations et les préparations pour chaque isolateur, ordonnancer sur la machine de contrôle, constituer des lots de livraisons, les affecter aux véhicules et définir les itinéraires des trajets.

La figure 2.1 représente la solution du problème intégré de production et de livraison modélisé : le site de production est constitué de 2 isolateurs identiques avec 2 machines pour la préparation et une capacité de stérilisation de 4 tâches. Deux véhicules sont disponibles pour la livraison. Sur l'exemple, on peut voir que la **Préparation** des tâches 1, 3, 4, 7, 9, 10, 11, 13, 14 débute juste après la fin de la **Stérilisation**, que la **Préparation** des tâches 5, 6 débute juste après la date de début au plus tôt et que la **Préparation** des tâches 2, 8, 12, 15, 16, 17 débute lorsqu'une machine est disponible. De même le **Contrôle** des tâches dépend de leurs dates de fin de **Préparation** mais aussi de la disponibilité de la machine. Les tâches 1 et 4 sont destinées à un même service d'oncologie donc $u_1 = u_4$ et $tt_{1,4} = 0$. Il en est de même pour les tâches 2 et 3, les tâches 5 et 6, les tâches 11 et 12 et les tâches 15 et 16. Le véhicule 1 effectue 2 trajets : le premier commence à C_5^{Ctrl} et suit l'itinéraire $[u_{14}, u_1, u_5]$, le deuxième commence à C_2^{Ctrl} et suit l'itinéraire $[u_8, u_{16}, u_2, u_6]$. Le véhicule 2 n'effectue qu'un seul trajet commençant à C_{12}^{Ctrl} et suit l'itinéraire $[u_9, u_{11}, u_{17}]$

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

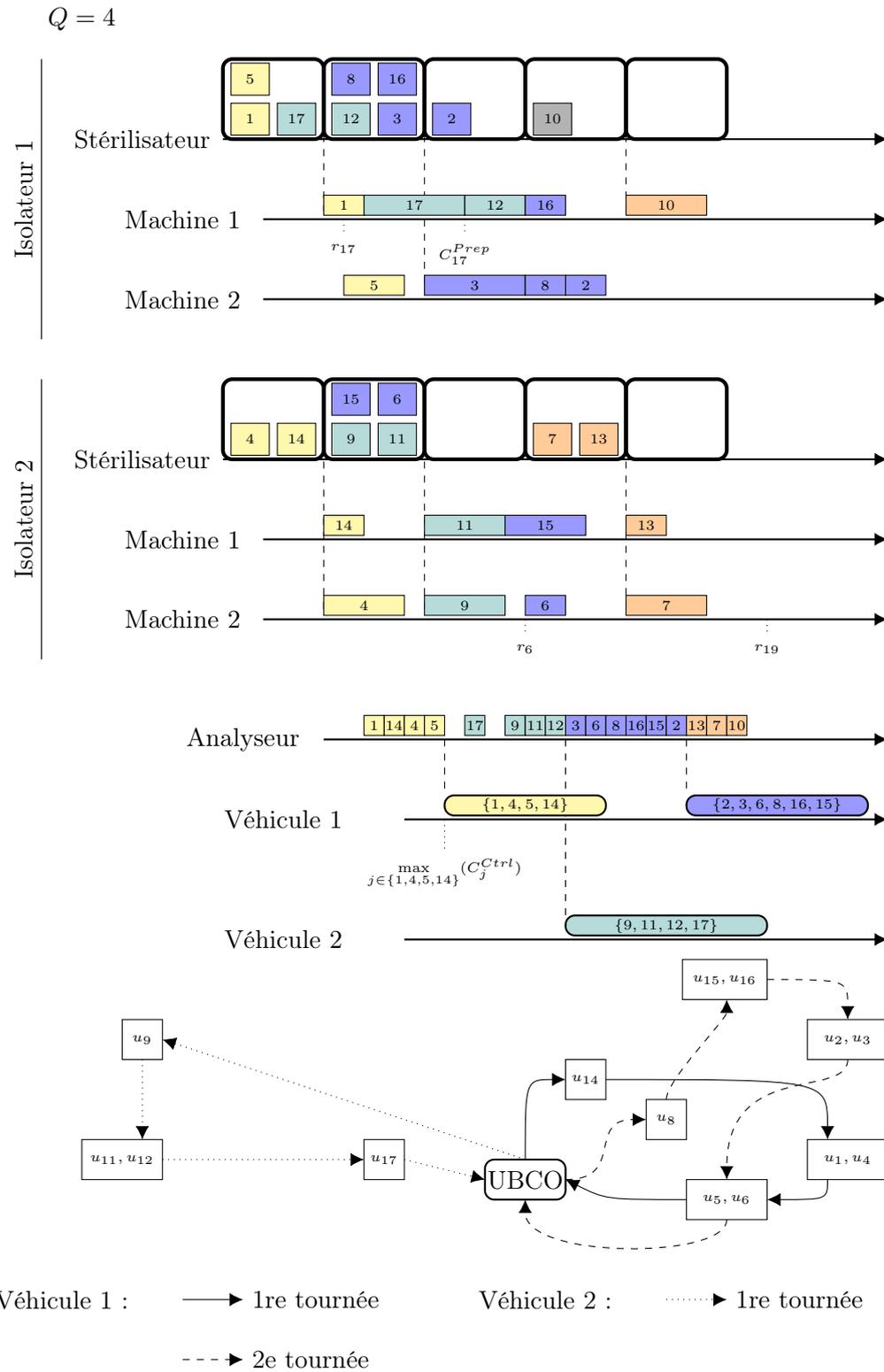


FIGURE 2.1 – Exemple de solution pour $|J| = 17$ tâches, $|I| = 2$ isolateurs, 2 machines par isolateur et une capacité de stérilisation de 4 tâches.

2.1.2 Positionnement dans l'état de l'art

Peu de temps avant le début de cette thèse, un état de l'art détaillé de la recherche sur les problèmes intégrés de production et de livraison a été fait dans [Moons *et al.*, 2017]. Les problèmes considérés sont caractérisés par une prise de décision opérationnelle au niveau de l'ordonnancement et dans l'établissement des itinéraires de livraison. En effet, de très nombreuses études traitent un problème d'ordonnancement où la livraison est soit constante, soit directe comme le montre [Chen, 2010]. [Moons *et al.*, 2017] répertorie 33 études de 1996 à 2017. Nous ajoutons 21 nouvelles études à cet état de l'art dont 18 publiées durant la durée de cette thèse. Ces études sont résumées dans le Tableau 2.1.

TABLE 2.1: Tableau résumé des études sur les problèmes de production et de distribution intégrée.

Article	Prod.	Livraison	Contrainte temporelle	Objectif	Méthodes
[Hurter et Van Buer, 1996]	1	VRP	D_{max}	D_{cost}	H
[Van Buer <i>et al.</i> , 1999]	1	VRP	D_{max}	D_{cost}	SA, TS
[Chang et Lee, 2004]	1	MT-C-TSP		D_{max}	H
[Garcia <i>et al.</i> , 2004]	1	Limited VRP	No wait	Profit	E, H
[Chen et Vairaktarakis, 2005]	1, P	VRP		$D_{mean},$ $D_{max},$ D_{cost}	E, H
[Li <i>et al.</i> , 2005]	1	MT-C-TSP		$\sum D_j$	E
[Li et Vairaktarakis, 2007]	B	C-VRP		$D_{cost}, \sum D_j$	Model, H, Approx.
[Naso <i>et al.</i> , 2007]	1	MT-C-VRP	TW	D_{cost}	Model, GA
[Geismar <i>et al.</i> , 2008]	1	MT-C-TSP	$D_j - C_j$	C_{max}	GA, MA
[Russell <i>et al.</i> , 2008]	P	Hete C-VRP	TW	$D_{cost},$ D_{max}	TS
[Chen <i>et al.</i> , 2009]	1	VRP		Profit, $\sum T_j$	H
[Chiang <i>et al.</i> , 2009]	P	Hete C-VRP	TW	D_{cost}	Model, TS, Sim.
[Park et Hong, 2009]	1	C-VRP	D_{max}	$D_{cost}, \sum T_j,$ C_{max}	Model, GA
[Scholz-Reiter <i>et al.</i> , 2011]	HF	Hete MT-C-VRP		$D_{cost}, \sum T_j$	Model
[Li et Ferrell, 2011]	1	Hete MT-VRP	$D_j - C_j$	D_{cost}	Model
[Farahani <i>et al.</i> , 2012]	P	Limited VRP	TW	$D_{cost},$ $\sum D_j - C_j$	LNS
[Amorim <i>et al.</i> , 2013]	Q	C-VRP	TW	D_{cost}	Model
[Low <i>et al.</i> , 2013]	1	Limited VRP	TW	D_{max}	Model, GA

Suite à la page suivante

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

TABLE 2.1 – suite du tableau

Article	Prod.	Livraison	Contrainte temporelle	Objectif	Méthodes
[Ullrich, 2013]	P	Hete MT-C-VRP	TW	$\sum T_j$	Model, GA
[Chang <i>et al.</i> , 2014]	R	Limited C-VRP		$D_{cost}, \sum D_j$	Model, ACO
[Lee <i>et al.</i> , 2014a]	Q	Hete MT-C-VRP	TW	D_{cost}	Model
[Low <i>et al.</i> , 2014]	1	Hete VRP	TW	$D_{cost}, \sum T_j, \sum E_j$	Model, GA
[Meinecke et Scholz-Reiter, 2014]	J	C-VRP		D_{cost}	H
[Viergutz et Knust, 2014]	1	C-TSP	TW	Demande	Model, TS, LS
[Zu <i>et al.</i> , 2014]	1	MT-PD		D_{cost}	Model
[Belo-Filho <i>et al.</i> , 2015]	Q	C-VRP	TW	D_{cost}	ALNS
[Li et Zu, 2015]	1	C-VRP		D_{cost}	H
[Cheref <i>et al.</i> , 2016b]	1	C-TSP		$\sum D_j, C_{max}$	Model, CG
[Cheref <i>et al.</i> , 2016a]	1	MT-TSP	r_j	L_{max}	Model, TS
[Jamili <i>et al.</i> , 2016]	1	VRP	r_j	D_{cost}, D_{mean}	Model, H, TS
[Li <i>et al.</i> , 2016]	1	VRP		$D_{cost}, \sum D_j$	NSGA II
[Devapriya <i>et al.</i> , 2017]	1	MT-C-VRP	Dj-Cj	D_{cost}	Model, GA, MA
[Fu <i>et al.</i> , 2017]	R	Hete C-VRP	TW	P_{cost}, D_{cost}	Model, H
[Karaođlan et Kesen, 2017]	1	MT-C-TSP		D_{max}	Model, E
[Kergosien <i>et al.</i> , 2017]	P	MT-TSP	$r_j, D_j - S_j$	T_{max}	Model, Bend.
[Low <i>et al.</i> , 2017]	1	Hete C-VRP		$D_{cost}, \sum T_j, \sum E_j$	Model, GA
[Chekoubi <i>et al.</i> , 2018]	Lot Sizing	C-PD		P_{cost}, D_{cost}	Model
[Gharaei et Jolai, 2018]	1	C-VRP		$D_{cost}, \sum T_j$	Model, MOGA
[Lacomme <i>et al.</i> , 2018]	1	C-VRP	Dj-Cj	D_{cost}	Model, GRASP, ELS
[Miranda <i>et al.</i> , 2018]	Lot Sizing	MT-C-TSP	TW	P_{cost}, D_{cost}	Model, H
[Zou <i>et al.</i> , 2018]	1	C-VRP		D_{max}	Model, H, GA
[Kesen et Bektaş, 2019]	P	Limited C-VRP		$\sum T_j, \sum E_j$	Model
[Marandi et Fatemi Ghomi, 2019]	HF	C-VRP	TW	D_{cost}, C_{max}	Model, PSO, H
[Miranda <i>et al.</i> , 2019]	Lot Sizing	MT-C-VRP	TW	P_{cost}, D_{cost}	Model

Suite à la page suivante

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

TABLE 2.1 – suite du tableau

Article	Prod.	Livraison	Contrainte temporelle	Objectif	Méthodes
[Tamannaei et Rasti-Barzoki, 2019]	1	C-VRP		$D_{cost}, \sum w_j.T_j$	Model, E, GA
[Tavares-Neto et Nagano, 2019]	P	MT-C-TSP		D_{max}	Model, H, GA
[Wang et al., 2019a]	P	MT-C-VRP	TW	$D_{cost}, \sum T_j$	Robust, MA
[Wang et al., 2019b]	1	C-VRP		D_{cost}	Model, TS
[Wang et al., 2019c]	HF	MT-C-TSP		D_{max}	Model, H, VNS
[Ganji et al., 2020]	1	Hete C-VRP		$D_{cost}, \sum T_j, \text{Preference}$	Model, MOPSO, NSGA II, MOACO
[Liu et al., 2020]	1	C-VRP		$\sum D_j$	Model, VNS
[Liu et Liu, 2020]	1	C-VRP		$\sum w_j.D_j$	Model, LNS
[Mohammadi et al., 2020]	J	C-VRP		$P_{cost}, D_{cost}, \sum T_j, \sum E_j$	Model, HPSO- ϵ const.
[Yağmur et Kesen, 2020]	F	C-TSP		$D_{cost}, \sum T_j$	Model, MA

Environnements : 1 : une machine ; B : machines groupées ; C : contrainte de capacité ; F : flow-shop ; Hete : flotte hétérogène ; HF : flow-shop hybride ; J : job-shop ; Limited : nombre de véhicules limité ; MT : multi-trajet ; P : machines parallèles identiques ; PD : pickup and delivery ; Q : machines parallèles uniformes ; R : machines parallèles non reliées ; TSP : problème du voyageur de commerce ; VRP : problème de tournées de véhicules.

Contraintes et objectifs : C_j : date de fin d'exécution ; C_{max} plus grande date de fin d'exécution ; D_{cost} : coût de livraison (coût de trajet avec ou sans coût de location/achat fixe de véhicule) ; D_j : date de livraison ; D_{max} : plus grande date de livraison ; D_{mean} : date de livraison moyenne ; E_j : avance ; L_{max} : retard algébrique maximum ; No wait : sans attente ; P_{cost} : coût de production (coût de mise en place, coût d'exécution, coût d'inventaire) ; r_j : date de début au plus tôt ; S_j : date de début d'exécution ; T_j : retard ; TW : fenêtres de temporelles ; w_j : pondération.

Méthodes : ACO : colonie de fourmis ; ALNS : recherche locale à large voisinage adaptatif ; Approx. : approximation ; Bend. : décomposition de Benders ; CG : génération de colonnes ; E : méthode exacte ; GA : algorithme génétique ; H : heuristique ; HPSO- ϵ const. : essais particuliers hybride avec ϵ contrainte ; LNS : recherche locale à large voisinage ; LS : recherche locale ; MA : algorithme mémétique ; MOACO : colonie de fourmis multi-objectif ; Model : modélisation mathématique ; MOGA : algorithme génétique multi-objectif ; MOPSO : essais particuliers multi-objectif ;

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

NSGA II : algorithme génétique trié par non domination ; PSO : essais particuliers ; SA : recuit simulé ; Sim. : simulation ; TS : recherche tabou ; VNS : recherche à voisinage variable

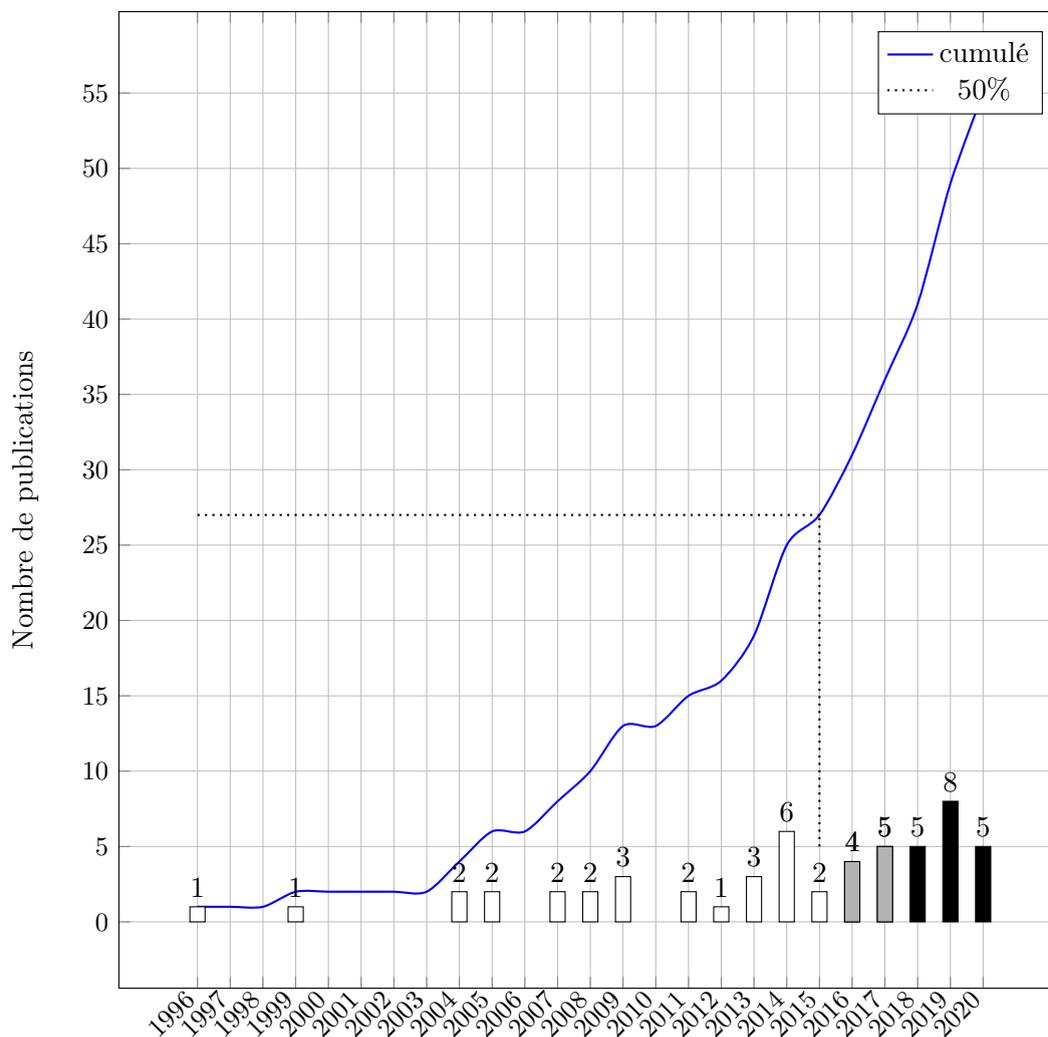


FIGURE 2.2 – Évolution de l'état de l'art dans le temps.

La Figure 2.2 montre la répartition temporelle des 54 études référencées. La courbe représente le nombre cumulé des publications, celle-ci semble suivre une évolution quadratique. La droite en pointillé partitionne en deux parties de 27 publications. On peut constater que ces cinq dernières années ont produit autant d'études que les 19 précédentes.

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

Dans la majorité de ces études, le problème de planification est majoritairement basé sur un environnement à une machine (environ 55,5 %) ou à machines parallèles (environ 26 %). La Figure 2.3 montre mieux cette répartition des environnements de production. L'ensemble "machine parallèles" fait référence aux problèmes d'ordonnancement à machines parallèles identiques, uniformes [Amorim *et al.*, 2013, Lee *et al.*, 2014b, Belo-Filho *et al.*, 2015] et non reliées [Chang *et al.*, 2014, Fu *et al.*, 2017]. Les environnements de production correspondant à l'ensemble "ateliers" sont les problèmes d'ordonnancement d'atelier de type flow-shop [Yağmur et Kesen, 2020] et de type job-shop [Meinecke et Scholz-Reiter, 2014, Mohammadi *et al.*, 2020]. Les "autres" environnements sont des problèmes d'ordonnancement à Bundling machines [Li et Vairaktarakis, 2007] ou alors des Lot sizing [Chekoubi *et al.*, 2018, Miranda *et al.*, 2018, Miranda *et al.*, 2019]. L'environnement de production qui nous intéresse dans ce chapitre est le flow-shop hybride [Scholz-Reiter *et al.*, 2011, Marandi et Fatemi Ghomi, 2019, Wang *et al.*, 2019c] qui ne représente que 5,5 % des publications.

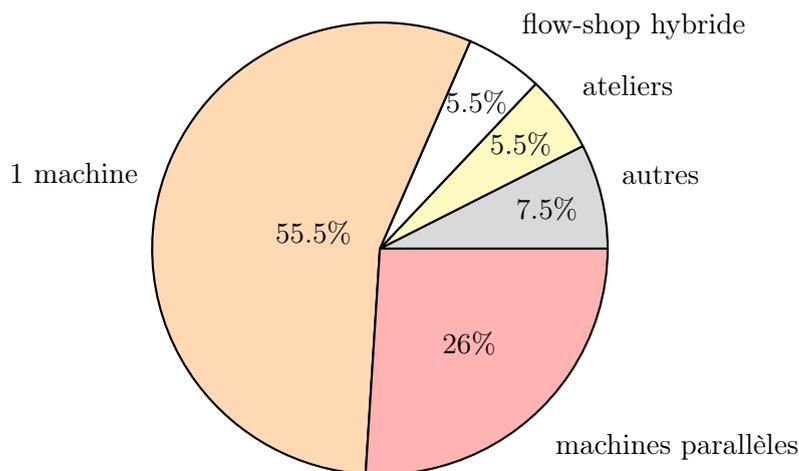


FIGURE 2.3 – Représentation des environnements de production dans la littérature des problèmes intégrés de production et de livraison.

Dans [Scholz-Reiter *et al.*, 2011], les auteurs présentent la première intégration d'un environnement de production de type flow-shop hybride. La flotte de véhicules est contrainte en capacité, hétérogène et peut effectuer plusieurs trajets. Cette étude optimise une combinaison du coût de livraison et d'une pénalité sur les retards. Cependant, aucune méthode de résolution n'est proposée à l'exception d'une modélisation mathématique.

Le problème étudié dans [Marandi et Fatemi Ghomi, 2019] est un problème intégré de production et de livraison modélisé comme un flow-shop hybride où chaque étage est constitué par un ensemble d'usines (chacune constituée d'une machine). Entre chaque étape de la production, les tâches doivent être acheminées à une autre usine pour l'opération suivante.

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

Une fois la dernière opération exécutée, une tâche est transposée à un unique dépôt pour sa livraison dans une configuration de problème de tournées de véhicules avec contrainte de capacité. Les livraisons sont contraintes par des fenêtres temporelles et l'objectif minimisé est une combinaison du coût de livraison et de la plus grande date de fin d'exécution.

Les auteurs de [Wang *et al.*, 2019c] présentent un problème intégré de production et de livraison modélisé comme un flow-shop hybride à trois étages combiné avec un problème de voyageur de commerce multi-trajet. Une contrainte de capacité est considérée sur la livraison. Afin de minimiser la date de fin de livraison maximale, les auteurs proposent une heuristique et une recherche locale à voisinage variable.

Notre objectif d'optimisation dans ce chapitre est celui du retard total. La Figure 2.4 montre les objectifs les plus traités des études recensées. Le coût de livraison est le critère le plus abordé avec plus de 60% des publications. Le retard total est second avec environ 25 % des publications. Cependant, dans la majorité des cas, le retard est considéré comme une pénalité économique.

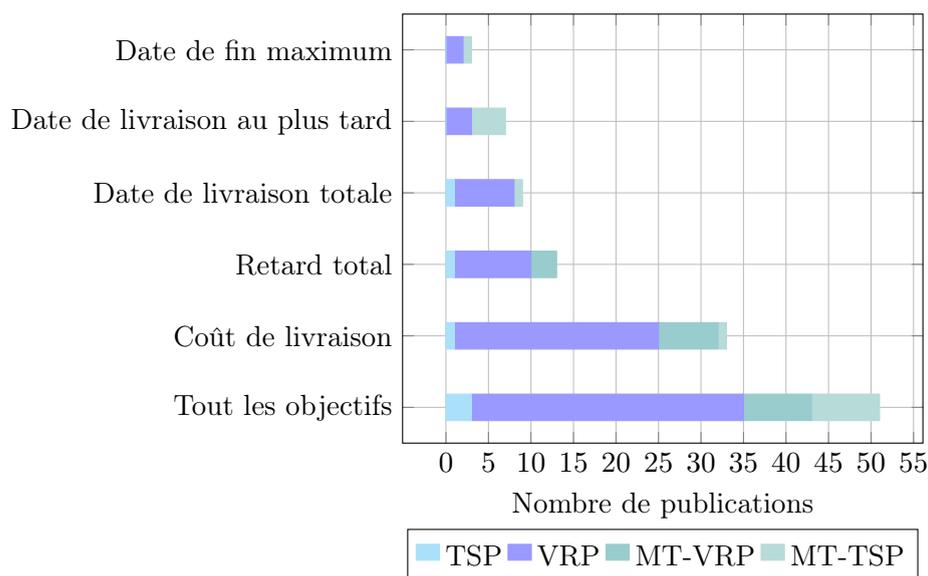


FIGURE 2.4 – Les critères d'optimisation les plus couramment traités dans l'état de l'art des problèmes intégrés production et livraison.

Parmi les articles présentés, les premiers à avoir traité la somme des retards sont [Chen *et al.*, 2009] et [Park et Hong, 2009]. Dans [Chen *et al.*, 2009] le problème intégré de production et de livraison est la combinaison d'un environnement à une machine et d'un VRP où l'objectif est de maximiser le profit. Les auteurs modélisent le profit comme une combinaison linéaire des profits directement liés à la différence entre l'offre et la demande, le coût de production et de livraison et des pénalités diverses dont la violation de fenêtres

temporelles qui s'apparente au retard. Dans un environnement intégré similaire, [Park et Hong, 2009] propose une contrainte de capacité aux véhicules et une date de fin souhaitée stricte commune aux livraisons. L'objectif traité est concentré sur la partie coût de livraison, pénalité des retards et la date de fin de production.

Seules cinq publications considèrent le retard total comme un objectif à part entière, soit comme objectif principal [Ullrich, 2013, Kesen et Bektaş, 2019] ou alors dans un traitement multi-objectifs [Gharaei et Jolai, 2018, Yağmur et Kesen, 2020, Mohammadi *et al.*, 2020].

Dans [Ullrich, 2013] un algorithme génétique est utilisé pour résoudre un problème intégrant un problème d'ordonnancement à machines parallèles identiques et un problème de tournées de véhicules hétérogènes multi-trajet avec contrainte de capacité. L'objectif optimisé est le retard total.

[Kesen et Bektaş, 2019] aborde un problème d'ordonnancement à machines parallèles identiques intégré à un problème de tournées de véhicules avec contrainte de capacité et avec deux variantes de fenêtres temporelles souples, selon que l'attente d'un véhicule est autorisée ou non. Les deux variantes sont modélisées par un modèle d'entiers mixtes, l'objectif minimisé est la combinaison de l'avance totale et du retard total. La flotte qui assure la livraison est homogène et de taille fixe limitée.

[Yağmur et Kesen, 2020] étudie un problème d'ordonnancement d'atelier de type flow-shop à permutation combiné à un problème de voyageur de commerce avec contrainte de capacité. Les auteurs développent un algorithme mémétique avec différentes stratégies d'encodage pour résoudre le problème, minimisant ainsi la combinaison du coût de livraison et du retard total.

Dans [Mohammadi *et al.*, 2020], le problème traité est l'intégration d'un problème d'ordonnancement d'atelier de type job-shop et d'un problème de tournées de véhicules hétérogènes avec contraintes. Le problème considéré est bi-objectif : minimiser le coût de la production et de la livraison, et minimiser les écarts aux fenêtres de livraisons (avance totale et retard total). Pour résoudre ce problème, les auteurs proposent un algorithme d'essai particulaire hybride avec une méthode d'épsilon contrainte.

Parmi les 13 publications prenant en compte le retard total dans le critère d'optimisation, trois études s'intéressent à un problème de tournées de véhicules multi-trajet [Scholz-Reiter *et al.*, 2011, Li et Ferrell, 2011, Wang *et al.*, 2019a].

[Wang *et al.*, 2019a] discute d'un problème de production intégrée et de tournées de véhicules multi-trajet avec fenêtres de temps et des temps de trajet incertains. L'environnement de production est modélisé comme un problème d'ordonnancement à machines parallèles identiques, et les livraisons sont effectuées par une flotte de véhicules homogène. L'objectif est de minimiser la somme pondérée du coût de livraison et la pénalité due au

2.1. PRÉSENTATION DE LA PROBLÉMATIQUE

retard total. Les auteurs proposent une approche robuste basée sur la p -robustesse élastique pour faire face aux variations de temps de trajet et un algorithme mémétique pour résoudre le problème.

La Figure 2.5 est un diagramme d'Euler représentant le nombre de publications en fonction des environnements de production et de livraison. Les environnements de livraison considérés dans l'ensemble "Multi-trajet" sont les problèmes de voyageur de commerce multi-trajet [Chang et Lee, 2004, Li *et al.*, 2005, Geismar *et al.*, 2008, Cheref *et al.*, 2016b, Karaođlan et Kesen, 2017, Miranda *et al.*, 2018, Tavares-Neto et Nagano, 2019], les problèmes de ramassages et livraison multi-trajet [Zu *et al.*, 2014] et lorsque intersecté par l'ensemble "VRP" les problèmes de tournées de véhicules multi-trajet [Naso *et al.*, 2007, Scholz-Reiter *et al.*, 2011, Li et Ferrell, 2011, Ullrich, 2013, Lee *et al.*, 2014b, Devapriya *et al.*, 2017, Miranda *et al.*, 2019, Wang *et al.*, 2019a]. L'ensemble "ateliers" correspond aux problèmes d'ordonnancement d'atelier de type flow-shop, de type job-shop et son intersection avec l'ensemble "machines parallèles" contient les problèmes de type flow-shop hybride et à une machine (en pointillé). L'ensemble en blanc aux contours gras représente la convergence des environnements de productions et de livraisons étudiés dans ce chapitre i.e. flow-shop hybride et tournées de véhicules multi-trajet. La seule étude se trouvant dans cet ensemble est [Scholz-Reiter *et al.*, 2011].

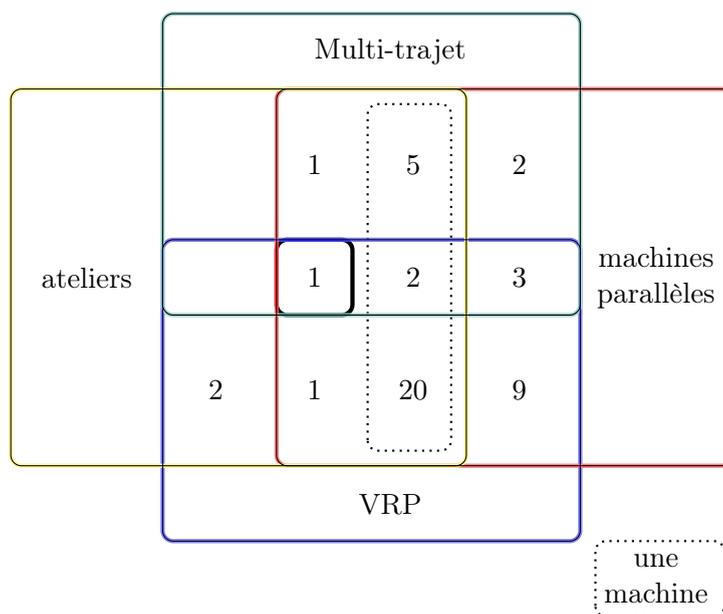


FIGURE 2.5 – Diagramme d'Euler représentant les structures des problèmes intégrés de production et de livraison de l'état de l'art.

2.2 Formulation mathématique

Pour mieux appréhender le problème, nous proposons une modélisation mathématique. Ce modèle est utilisable par un solveur commercial comme CPLEX optimizer ou Gurobi ou des solveurs libres et gratuits comme GLPK.

2.2.1 Paramètres d'entrées

Les paramètres d'entrées du modèle sont résumés dans le tableau 2.2. Les notations utilisées sont homogènes avec la description du problème 2.1. L'ensemble K représente les index des lots de stérilisation en supposant que pour chaque isolateur, le stérilisateur associé lance un lot de stérilisation dès que possible, soit toutes les p^{Ste} minutes. L'ensemble des index des trajets T est construit de sorte que $|T|$ soit une borne supérieure du nombre de trajets possibles par un même véhicule. Une borne supérieure naïve est un trajet par tâche à livrer. On suppose que la matrice de distance $(tt_{j,j'})$ satisfait l'inégalité triangulaire. Il est possible que deux tâches j et j' soient à livrer à la même position, alors $u_j = u_{j'}$ et $tt_{j,j'} = tt_{j',j} = 0$. Le modèle ne nécessite pas la symétrie des temps de trajet (i.e. on n'a pas nécessairement $\forall(j,j') \in J^2 \quad tt_{j,j'} = tt_{j',j}$).

Notation	Description
J	ensemble des tâches
I	ensemble des isolateurs
K	ensemble des index des lots de stérilisation d'un même stérilisateur
M_i	ensemble des machines correspondant à l'isolateur i
M	ensemble des machines ($M = \bigcup_{i \in I} M_i$)
V	ensemble des véhicules
T	ensemble des index des trajets d'un même véhicule
Q_i	capacité du stérilisateur de l'isolateur i
p^{Ste}	durée constante de Stérilisation
p^{Ctrl}	durée constante du Contrôle
r_j	date de début de Préparation au plus tôt de la tâche j
p_j^{Prep}	durée de Préparation de la tâche j
d_j	date de livraison souhaitée de la tâche j
u_j	point de livraison de la tâche j
$tt_{j,j'}$	temps de trajet entre u_j et $u_{j'}$

TABLE 2.2 – Paramètres d'entrées du modèle

2.2.2 Variables de décision

Les variables de décision du modèle sont résumées dans le tableau 2.3. Les notations utilisées pour ces variables sont restreintes à ce modèle à l'exception de C_j^{Prep} , C_j^{Ctrl} , D_j et T_j qui étaient déjà définies en Section 2.1.

Notation et domaine	Description
Stérilisation	
$x_{j,m,k} \in \{0, 1\}, \forall j \in J, \forall m \in M, \forall k \in K$	$x_{j,m,k} = 1$ si la tâche j est stérilisée dans le k ième lot de stérilisation associé à la machine m et est préparée par la machine m , 0 sinon.
Préparation	
$y_{j,m} \in \{0, 1\}, \forall j \in J, \forall m \in M$ $\gamma_{j,j'} \in \{0, 1\}, \forall j \in J \forall j' \in J$ $C_j^{Prep} \in \mathbb{R}_+, \forall j \in J$	$y_{j,m} = 1$ si la tâche j est préparée par la machine m , 0 sinon. $\gamma_{j,j'} = 1$ si les tâches j et j' sont préparées par la même machine, 0 sinon. date de fin de Préparation de la tâche j .
Contrôle	
$\delta_{j,j'} \in \{0, 1\}, \forall j \in J, \forall j' \in J$ $C_j^{Ctrl} \in \mathbb{R}_+, \forall j \in J$	$\delta_{j,j'} = 1$ si la tâche j est contrôlée avant la tâche j' , 0 sinon. date de fin de Contrôle de la tâche j
Livraison	
$z_{j,v,t} \in \{0, 1\}, \forall j \in J, \forall v \in V, \forall t \in T$ $\bar{z}_{v,t} \in \{0, 1\}, \forall v \in V, \forall t \in T$ $\zeta_{j,j'} \in \{0, 1\}, \forall j \in J \forall j' \in J$ $\alpha_{v,t} \in \mathbb{R}_+, \forall v \in V, \forall t \in T$ $D_j \in \mathbb{R}_+, \forall j \in J$ $T_j \in \mathbb{R}_+, \forall j \in J$	$z_{j,v,t} = 1$ si la tâche j est livrée dans le t -ième trajet du véhicule v , 0 sinon. $\bar{z}_{v,t} = 1$ le trajet t du véhicule v livre au moins une tâche, 0 sinon. $\zeta_{j,j'} = 1$ si le trajet u_j à $u_{j'}$ est emprunté par un véhicule, 0 sinon. date de départ du t -ième trajet du véhicule v . date de livraison de la tâche j . retard de la tâche j .

TABLE 2.3 – Variables de décisions du modèle

2.2.3 Contraintes

La formulation proposée ci-dessous n'est pas linéaire à cause des contraintes 2.6, 2.10, 2.11, 2.14, 2.15, 2.16, 2.17, 2.18, 2.19 et 2.20. Bien qu'il soit possible de linéariser le modèle en utilisant des contraintes de type big-M et des variables supplémentaires, nous préférons fournir une description plus lisible. La fonction objectif est de minimiser le retard total de livraison :

$$\text{MIN} \sum_{j \in J} T_j$$

La capacité du stérilisateur est respectée :

$$\sum_{m \in M_i} \sum_{j \in J} x_{j,m,k} \leq Q_i \quad \forall i \in I, \forall k \in K \quad (2.1)$$

Une tâche est stérilisée par un lot de stérilisation correspondant à la machine qui la prépare :

$$\sum_{k \in K} x_{j,m,k} = y_{j,m} \quad \forall j \in J, \forall m \in M \quad (2.2)$$

La **Préparation** d'une tâche débute après la fin de sa **Stérilisation** :

$$k \cdot p^{Ste} x_{j,m,k} + p_j^{Prep} \leq C_j^{Prep} \quad \forall j \in J, \forall m \in M, \forall k \in K \quad (2.3)$$

Une tâche est préparée une et une seule fois par une seule machine :

$$\sum_{m \in M} y_{j,m} = 1 \quad \forall j \in J \quad (2.4)$$

La **Préparation** d'une tâche débute après sa date de début au plus tôt :

$$r_j + p_j^{Prep} \leq C_j^{Prep} \quad \forall j \in J \quad (2.5)$$

La contrainte disjonctive assure que deux tâches ne peuvent être préparées en même temps par une même machine.

$$C_j^{Prep} \cdot \gamma_{j,j'} + p_{j'}^{Prep} \leq C_{j'}^{Prep} \quad \forall j \in J, \forall j' \in J, \text{ et } j \neq j' \quad (2.6)$$

Une contrainte de précédence entre deux tâches s'impose si elles sont préparées par une même machine :

$$\gamma_{j,j'} + \gamma_{j',j} \geq y_{j,m} + y_{j',m}^m - 1 \quad \forall m \in M, \forall j \in J, \forall j' \in J, \text{ and } j \neq j' \quad (2.7)$$

2.2. FORMULATION MATHÉMATIQUE

Le **Contrôle** d'une tâche débute après la fin de sa **Préparation** :

$$C_j^{Prep} + p^{Ste} \leq C_j^{Ctrl} \quad \forall j \in J \quad (2.8)$$

La notion de précédence entre deux tâches est à sens unique :

$$\delta_{j,j'} + \delta_{j',j} = 1 \quad \forall j \in J, \forall j' \in J, \text{ et } j \neq j' \quad (2.9)$$

L'analyseur ne peut contrôler qu'une tâche à la fois :

$$C_j^{Ctrl} \cdot \delta_{j,j'} + p^{Ctrl} \leq C_{j'}^{Ctrl} \quad \forall j \in J, \forall j' \in J, \text{ et } j \neq j' \quad (2.10)$$

L'heure de départ d'un trajet ne commence qu'après la fin de **Contrôle** de toutes les tâches à livrer durant ce trajet :

$$C_j^{Ctrl} \cdot z_{j,v,t} \leq \alpha_{v,t} \quad \forall j \in J, \forall v \in V, \forall t \in T \quad (2.11)$$

Une tâche est livrée une et une seule fois par un seul véhicule durant un unique trajet :

$$\sum_{v \in V} \sum_{t \in T} z_{j,v,t} = 1 \quad \forall j \in J \quad (2.12)$$

Un trajet non vide livre au moins une tâche :

$$\sum_{j \in J} z_{j,v,t} \leq |J| \cdot \bar{z}_{v,t} \quad \forall v \in V, \forall t \in T \quad (2.13)$$

Chaque trajet non vide débute à une seule heure possible

$$\sum_{j \in J} \zeta_{0,j} \cdot z_{j,v,t} = \bar{z}_{v,t} \quad \forall v \in V, \forall t \in T \quad (2.14)$$

La date de livraison d'une tâche est minorée par l'heure de départ du trajet la livrant :

$$\sum_{v \in V} \sum_{t \in T} \alpha_{v,t} \cdot z_{j,v,t} + tt_{0,j} \leq D_j \quad \forall j \in J \quad (2.15)$$

Chaque point de livraison est visité une et une seule fois (contrainte de flux entrant) :

$$\sum_{j' \in J \setminus \{j\}} \zeta_{j',j} \cdot z_{j',v,t}^{v,t} + \zeta_{0,j} \cdot z_{j,v,t} = z_{j,v,t} \quad \forall j \in J, \forall v \in V, \forall t \in T \quad (2.16)$$

Il y a un et un seul départ depuis chacun des points de livraison (contrainte de flux sortant) :

$$\sum_{j' \in J \setminus \{j\}} \zeta_{j,j'} \cdot z_{j'}^{v,t} + \zeta_{j,0} \cdot z_{j,v,t} = z_{j,v,t} \quad \forall j \in J, \forall v \in V, \forall t \in T \quad (2.17)$$

La date de livraison dépend de l'itinéraire du trajet :

$$(D_j + tt_{j,j'}) \cdot \zeta_{j,j'} \leq D_{j'} \quad \forall j \in J, \forall j' \in J, \text{ et } j \neq j' \quad (2.18)$$

Chaque trajet non vide finit son itinéraire au dépôt :

$$\sum_{j \in J} \zeta_{j,0} \cdot z_{j,v,t} = \bar{z}_{v,t} \quad \forall v \in V, \forall t \in T \quad (2.19)$$

Un trajet ne peut débuter qu'après le retour du précédent :

$$(D_j + tt_{j,0}) \cdot z_{j,v,t'} \leq \alpha_{v,t} \quad \forall j \in J, \forall v \in V, \forall t \in T, \forall t' \in \{1, \dots, t-1\} \quad (2.20)$$

Calcul du retard des tâches :

$$T_j \geq D_j - d_j \quad \forall j \in J \quad (2.21)$$

La résolution d'un tel modèle est extrêmement compliquée. En effet, les tailles d'instances résolubles sont très petites (inférieur à huit tâches, un seul véhicule et deux isolateurs). C'est pour cela que nous proposons plusieurs heuristiques.

2.3 Méthodes de résolution heuristiques

Dans cette section, nous présentons trois algorithmes heuristiques déterministes et une procédure de "Greedy Randomized Adaptive Search". Le choix de développer de telles heuristiques est motivé par le fait de disposer d'une méthode de résolution rapide et efficace à intégrer dans un outil d'aide à la décision en temps réel. Chaque heuristique déterministe contient six niveaux de décision : l'affectation des tâches aux lots de stérilisation, l'ordonnancement des préparations, l'ordonnancement du contrôle, la mise en lot pour la livraison, l'affectation des lots aux véhicules et la construction des itinéraires de livraisons.

La première heuristique, appelée "Reference Algorithm" (RA), est une simple méthode gloutonne utilisée sur le site de production de l'UBCO. Elle consiste à prendre des décisions niveau par niveau (de l'affectation des tâches à la stérilisation des lots jusqu'à la

2.3. MÉTHODES DE RÉOLUTION HEURISTIQUES

construction des itinéraires de livraison). L'idée de la seconde heuristique, appelée "Multilevel Heuristic" (ML), est d'améliorer l'heuristique précédente en prenant en compte le routage des données dans le premier niveau de décision. Les deux premières heuristiques RA et ML ont en entrée un nombre fixe de lots de livraison.

Les deux dernières méthodes de résolution sont des extensions de l'heuristique ML. Elles utilisent une procédure de descente de gradient pour déterminer automatiquement le meilleur nombre de lots de livraison à considérer. La troisième heuristique est appelée "Bilevel Gradient Descent" (BGD). La quatrième heuristique, "Greedy Randomized Adaptive Search Procedure" (GRASP), a la même structure que la "Bilevel Gradient Descent" en ajoutant une phase de diversification avec une part d'aléatoire lors de la constitution des lots de livraison.

Les trois heuristiques déterministes sont illustrées dans l'organigramme de la figure 2.6. Il présente les principales étapes de chaque algorithme selon les niveaux de décisions. On note n_f le nombre de lots de livraison finaux utilisés pour calculer l'étape *Mise en lot finale*. Il représente également le nombre de trajets effectués par les véhicules. De la même manière, n_t désigne le nombre de lots de livraison cibles ou souhaités, utilisés pour calculer l'*Ordonnancement de la production*.

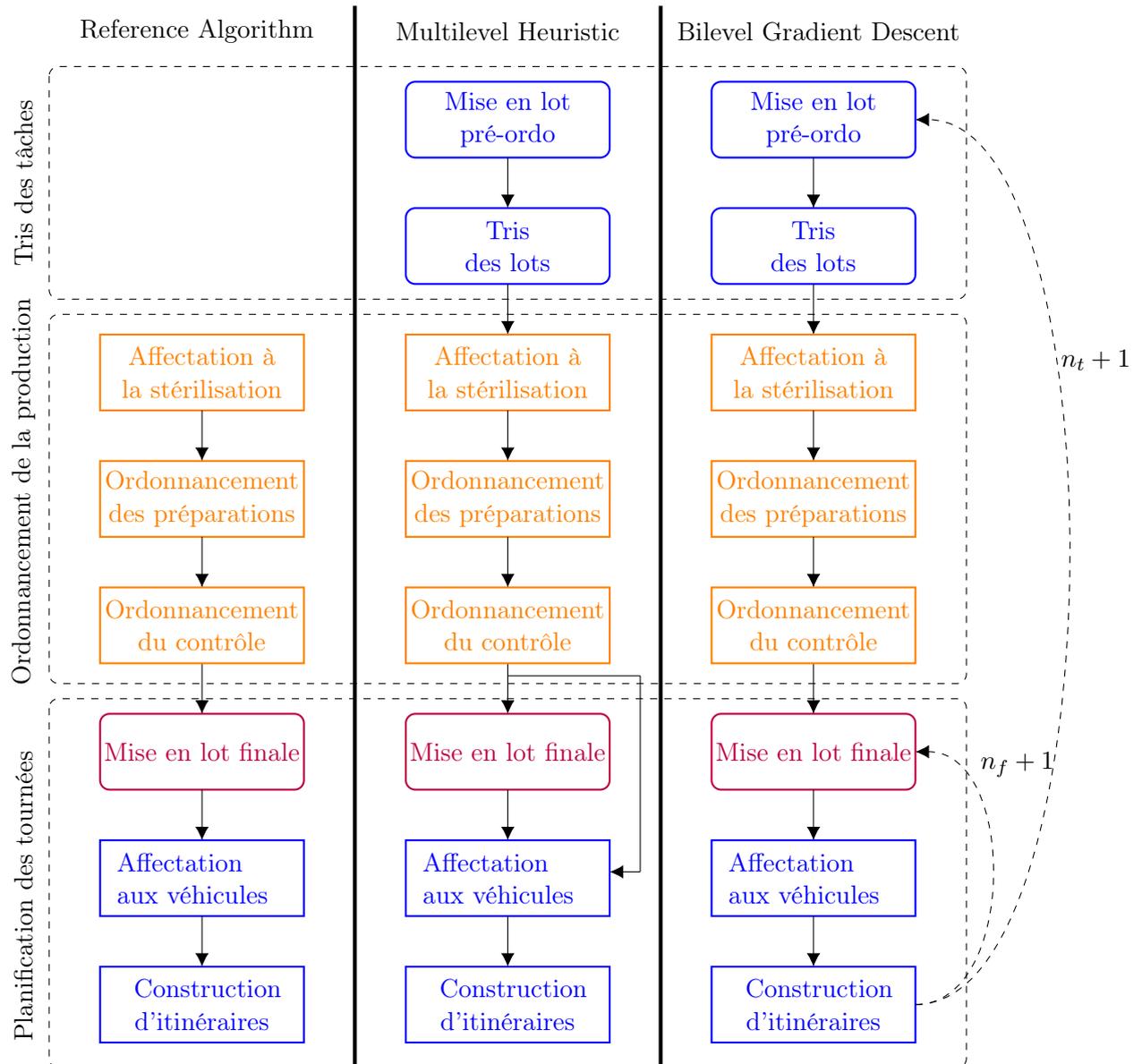


FIGURE 2.6 – Organigramme des heuristiques : "Reference Algorithm", "Multilevel Heuristic" et "Bilevel Gradient Descent"

2.3.1 Reference Algorithm

Le "Reference Algorithm" (RA) s'inspire de la méthode de processus de production et de livraison utilisée sur le site de l'UBCO. Les décisions sont prises naturellement, selon les étapes de décision suivantes. Tout d'abord, les tâches sont affectées aux lots de stérilisation d'un isolateur. Ensuite, elles sont programmées sur l'une des machines associées puis sur la machine de contrôle commune. Enfin, elles sont affectées à un trajet effectué par un véhicule.

Le temps de fin de **Stérilisation** du lot numéro k sur l'isolateur i (noté $B_{i,k}^{Ste}$) est égal à $k.p^{Ste}$. La première décision est d'affecter chaque tâche à un lot de stérilisation (*Affectation à la stérilisation*). La méthode suivante est utilisée pour trouver la composition des lots de stérilisation. Tout d'abord, les tâches sont triées par ordre croissant de date de début de **Préparation** au plus tôt. Ensuite, chaque tâche j est successivement affectée à un lot de stérilisation non plein et selon les trois règles suivantes :

- Si tous les lots de stérilisation se terminant avant r_j ne sont pas pleins, le travail j est affecté au lot de stérilisation qui terminera sa stérilisation le plus tard possible, mais avant sa date de début au plus tôt r_j .
- Sinon, la tâche est affectée au premier lot de stérilisation disponible qui termine après la date de début de **Préparation** au plus tôt.
- Puisqu'il y a plusieurs isolateurs, s'il y a plus d'un lot de stérilisation candidat, le lot sélectionné est celui avec le temps total de **Préparation** des tâches à l'intérieur du lot le plus petit (i.e. le lot $B_{i,k}^{Ste}$ minimisant $\sum_{j \in B_{i,k}^{Ste}} p_j^{Prep}$).

L'affectation aux lots de stérilisation qui en résulte est l'entrée de l'étape suivante, appelée *Ordonnancement des préparations*. Lot de stérilisation par lot de stérilisation, les tâches sont triées par date de début au plus tôt r_j croissante et sont affectées à la première machine disponible de l'isolateur correspondant.

La dernière décision d'ordonnancement est l'*Ordonnancement du contrôle*. Les tâches sont préparées puis affectées à la machine de contrôle dans l'ordre "Premier entré, premier sorti", c'est-à-dire en respectant l'ordre des dates de fin de préparations.

Enfin, le dernier niveau de décision concerne la planification des tournées. Chaque tâche est d'abord affectée à un lot de livraison par une *Mise en lot finale*, et chaque lot de livraison est affecté à un véhicule.

La *Mise en lot finale* vise à limiter les délais d'attente des tâches avant leur livraison et à réduire la durée des trajets. Elle est donc naturellement basée sur les dates de livraison souhaitées des tâches, sur les dates de fin de **Contrôle** et sur leurs lieux de livraison.

L'algorithme de partitionnement utilisé est une méthode "Agglomerative Hierarchical

2.3. MÉTHODES DE RÉOLUTION HEURISTIQUES

Clustering" [Murtagh et Contreras, 2012] décrite avec un pseudo-code dans Alg.1. Cette méthode nécessite la définition d'une fonction de distance entre deux tâches et entre deux lots. On définit la distance entre deux lots E et E' , notée $dist_{clusters}(E, E', dist_f)$ (appelé aussi "complete link" dans [Murtagh et Contreras, 2012]), comme la distance maximale entre les tâches j et j' , et notée $dist_f(j, j')$:

$$dist_{clusters}(E, E', dist_f) = \max_{j \in E, j' \in E'} (dist_f(j, j')) \quad (2.22)$$

La distance entre deux tâches j et j' est la distance euclidienne qui utilise trois dimensions dans le même ordre de grandeur (temps en minutes) : le temps de trajet entre les points de livraison $tt_{j,j'}$, les dates de livraison souhaitées d_j et $d_{j'}$ et les dates de fin de Contrôle C_j^{Ctrl} et $C_{j'}^{Ctrl}$.

$$dist_f(j, j') = \sqrt{(d_j - d_{j'})^2 + (C_j^{Ctrl} - C_{j'}^{Ctrl})^2 + tt_{j,j'}^2} \quad (2.23)$$

Le nombre souhaité de lots de livraison n_f doit être donné en entrée de l'algorithme de partitionnement. Ce paramètre important est lié à la distance maximale acceptable entre deux tâches d'un même lot de livraison. Le problème de partitionnement en lots de livraison est résolu en appelant $AH_Clustering(J, n_f, dist_f)$.

Algorithm 1 $AH_Clustering(J, n_f, dist_f)$

Paramètres : un ensemble de tâches J , un nombre de lot de livraison n_f et une distance $dist_f$

Soit \mathcal{E} une partition de J composée uniquement de singletons

```

while  $|\mathcal{E}| > n_f$  do
   $best\_dist \leftarrow +\infty$ 
  for  $E \in \mathcal{E}$  do
    for  $E' \in \mathcal{E} \setminus \{E\}$  do
      if  $dist_{clusters}(E, E', dist_f) < best\_dist$  then
         $best\_dist \leftarrow dist_{clusters}(E, E', dist_f)$ 
         $best_E \leftarrow E$ 
         $best_{E'} \leftarrow E'$ 
      end if
    end for
  end for
   $\mathcal{E} \leftarrow \mathcal{E} \setminus \{best_E, best_{E'}\}$ 
   $best_E \leftarrow best_E \cup best_{E'}$ 
   $\mathcal{E} \leftarrow \mathcal{E} \cup \{best_E\}$ 
end while
return  $\mathcal{E}$ 

```

La Figure 2.7 présente une illustration de la méthode de regroupement hiérarchique

aggloméré à l'aide d'un dendrogramme. La longueur des lignes verticales représente la distance entre deux lots. La construction d'un dendrogramme commence par le bas, où chaque lot est composé d'une seule tâche, et s'arrête lorsque la valeur de n_f est atteinte. La ligne pointillée représente la coupure du dendrogramme qui détermine les lots résultants de l'algorithme : $\{1, 4, 5, 14\}$, $\{9, 11, 12, 17\}$ et $\{2, 3, 6, 8, 16, 15\}$.

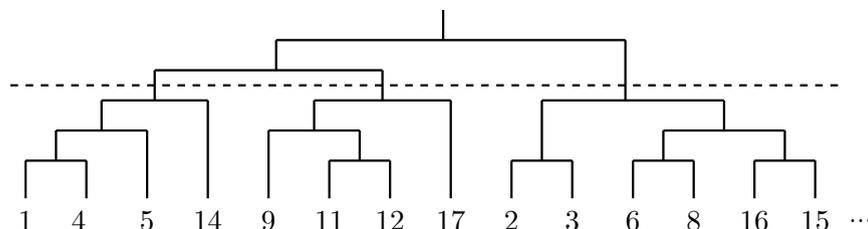


FIGURE 2.7 – Dendrogramme de l'exécution de l'Algorithme 2.3.3 sur l'exemple de la Figure 2.1.

Dans la pratique, la répartition en lots de livraison des tâches à l'UBCO est une prise de décision faite en continu par les livreurs qui n'ont qu'une vision à court terme de la journée. Nous avons décidé d'utiliser l'algorithme de regroupement hiérarchique aggloméré, car il nous semble le plus adapté pour à la fois retranscrire la prise de décision actuelle des livreurs et offrir un bon compromis entre de trop longs trajets et de trop nombreux trajets.

Les dernières étapes sont l'attribution des lots de livraison aux véhicules (*Affectation aux véhicules*) et la *Construction d'itinéraire* de chaque trajet. Les lots de livraison résultants sont attribués par ordre croissant de leurs dates de fin de production (i.e. la date de fin de **Contrôle** maximum des tâches du lot) au premier véhicule disponible. Les itinéraires de chaque véhicules doivent être déterminés de manière à minimiser le retard total associé aux tâches du lot (la fonction objectif du problème) et la durée des trajets. Ce deuxième objectif vise également à réduire le retard total sur les prochains trajets de ce même véhicule car plus tôt un véhicule revient au site de production, plus tôt il pourra entamer un nouveau trajet. Ainsi et afin de calculer les itinéraires en un temps de calcul court, nous utilisons l'heuristique du plus proche voisin [Rosenkrantz *et al.*, 1977].

2.3.2 Multi-level heuristic

Le "Multilevel Heuristic" (ML) utilise un algorithme de partitionnement et des règles de répartition avant de planifier les tâches sur les isolateurs.

Une méthode de partitionnement permet de définir de potentiels lots de livraison souhaités avant l'ordonnancement de la production. Cette procédure vise à programmer la production en prenant en compte non seulement les dates de début au plus tôt, comme dans la méthode précédente, mais aussi les dates de fin de **Contrôle** souhaitées et les lieux

de livraison des tâches.

La méthode de partitionnement utilisée pour créer les lots de livraison cibles est l'algorithme `AH_Clustering`. Nous introduisons ici la notion de date de fin de **Préparation** au plus tôt, définie par $\tilde{r}_j = r_j + p_j^{Prep}$ qui correspond aussi à la date de début de **Contrôle** au plus tôt. Le problème est résolu en appelant `AH_Clustering($J, n_t, dist_t$)` avec une nouvelle fonction de distance $dist_t(j, j')$ définie par :

$$dist_t(j, j') = \sqrt{(d_j - d_{j'})^2 + (\tilde{r}_j - \tilde{r}_{j'})^2 + tt_{j,j'}^2} \quad (2.24)$$

Le but de la création de lots de livraison souhaités est de fournir une règle de tris des tâches plus adaptée en appliquant un *Tri des lots*. Chaque lot de livraison souhaité résultant est associé à une date de fin souhaitée qui est la date de fin souhaitée minimale des tâches du lot. Les lots de livraison souhaités sont triés date de fin souhaitée croissante. Comme précédemment, le nombre de lots de livraison souhaités n_t , doit être donné en entrée de l'algorithme de partitionnement.

L'*Affectation à la stérilisation* des tâches utilise `Assign_ster_batches(σ)` où σ sont les tâches triées en premier dans l'ordre de lots de livraison souhaités. Ensuite, à l'intérieur de chaque lot, les tâches sont triées par date de fin de **Préparation** au plus tôt croissante. L'*Ordonnancement des préparations* et l'*Ordonnancement du contrôle* se font de la même manière que dans l'algorithme RA. Ensuite, deux partitionnements (i.e. l'affectation des tâches aux lots de livraison) sont calculés et le meilleur est sélectionné. Le premier partitionnement est basé sur les lots de livraison souhaités et consiste à conserver les mêmes lots calculés à l'étape *Mise en lot pré-ordo* et mettre à jour leurs dates de fin d'exécution. Le deuxième partitionnement vise à déterminer de nouveaux lots de livraison en utilisant la méthode *Mise en lot finale* et la fonction $dist_f$ comme pour l'algorithme RA. La valeur n_f pour la *Mise en lot finale* est la même valeur que n_t , ce qui implique que dans la plupart des cas, les deux partitionnements sont identiques. Chacun des partitionnements est évalué par les mêmes méthodes d'*Affectation aux véhicules* et de *Construction d'itinéraire* que dans l'algorithme RA.

2.3.3 Bilevel Gradient Descent

Le Bilevel Gradient Descent (BGD) est une amélioration de l'heuristique ML car il s'agit d'une version auto-paramétrée de l'heuristique "Multilevel Heuristic". Le "Bilevel Gradient Descent" vise à trouver le couple (n_t, n_f) qui conduit à la meilleure solution sans tester toutes les valeurs. Des expérimentations préliminaires ont montré que l'évolution des valeurs de la meilleure solution, en fonction du nombre de lots de livraison ($n_t = n_f$), est une courbe s'approchant d'une parabole avec un point de basculement. Une descente de

gradient [Snyman, 2005, Ruder, 2016] est donc particulièrement bien adaptée pour trouver le nombre de lots de livraison idéal. D'autres expériences ont aussi montré que la valeur exacte de ce point de basculement dépend de l'instance et que l'utilisation de deux valeurs différentes n_t et n_f peut améliorer les résultats de l'algorithme heuristique.

Pour résoudre ce problème d'optimisation des paramètres, nous proposons une heuristique "Bilevel Gradient Descent Heuristic". C'est une combinaison de deux descentes de gradient. Cette heuristique utilise une descente de gradient principale, BGD, pour optimiser le nombre de lots de livraison souhaités n_t . Pour chaque valeur testée de n_t , une descente de gradient secondaire, `sec_GD`, est appelé pour optimiser le nombre de lots de livraison finaux n_f .

Avant de présenter les algorithmes de descente de gradient, nous introduisons les définitions suivantes.

- Soit $PS : \mathcal{E} \mapsto \mathcal{S}$ la méthode qui renvoie un planning de production \mathcal{S} basé sur l'ensemble des lots de livraison souhaités \mathcal{E} ($|\mathcal{E}| = n_t$). Le planning de production définit les séquences et la composition des lots de stérilisation sur chaque isolateur, les séquences de production, et les heures de démarrage des travaux sur chaque opérateur et l'analyseur. La méthode PS consiste à appliquer successivement les méthodes d'*Affectation à la stérilisation*, d'*Ordonnancement des préparations* et d'*Ordonnancement du contrôle* utilisées dans les heuristiques RA et ML.
- Soit $DP : (\mathcal{S}, n_f) \mapsto \mathcal{S}'$ la méthode qui construit le planning de livraison \mathcal{S}' associé au planning de production \mathcal{S} avec un nombre de lots de livraison n_f . Le planning de livraison définit la composition de chaque lot de livraison ainsi que les séquences et la composition des tournées de chaque livreur. La méthode DP consiste à appliquer successivement les méthodes de *Mise en lot finale*, d'*Affectation aux véhicules* et de *Construction d'itinéraire* utilisées dans les heuristiques RA et ML.
- Soit $TT : \mathcal{S}' \mapsto z$ une fonction qui calcule le retard total associé au planning de livraison \mathcal{S}' .

Les pseudo-codes de BGD et de `sec_GD`'s sont respectivement présentés dans 2 and 3.

L'algorithme BGD prend quatre paramètres en entrée : l'ensemble des tâches J , la distance $dist_t$ utilisée par la méthode de partitionnement et deux taux d'apprentissage (un pour chaque descente de gradient). L'heuristique BGD est basée sur une descente de gradient classique [Snyman, 2005, Ruder, 2016]. Elle vise à trouver le meilleur nombre de lots de livraison souhaités, noté n_t , qui mène à la meilleure solution du problème. La méthode commence à $n_t = 1$ et, autant de fois que nécessaire, passe de n_t à $n_t + 1$. Une solution est alors calculée pour chaque valeur de n_t . Le critère d'arrêt dépend de la solution courante, de la solution calculée à l'itération précédente et du taux d'apprentissage α_{main} . Pour chaque valeur testée de n_t , une solution est calculée en trois étapes.

Algorithm 2 $\text{BGD}(J, dist_t, \alpha_{main}, \alpha_{sec})$

Paramètres : un ensemble de tâches J , une distance $dist_t$ et deux taux d'apprentissages α_{main} et α_{sec}

$n_t \leftarrow 1$

$\mathcal{E} \leftarrow \text{AH_clustering}(J, n_t, dist_t)$

$\mathcal{S}_{n_t} \leftarrow PS(\mathcal{E})$

$\nabla_{main} \leftarrow \text{sec_GD}(\mathcal{S}_{n_t}, \alpha_{sec})$

best_value $\leftarrow \nabla_{main}$

while $\nabla_{main} \leq 0$ **do**

$\mathcal{E} \leftarrow \text{AH_clustering}(J, n_t + 1, dist_t)$

$\mathcal{S}_{n_t+1} \leftarrow PS(\mathcal{E})$

$\nabla_{main} \leftarrow \alpha_{main} \cdot \nabla_{main} + \text{sec_GD}(\mathcal{S}_{n_t+1}, \alpha_{sec}) - \text{sec_GD}(\mathcal{S}_{n_t}, \alpha_{sec})$

best_value $\leftarrow \min(\text{best_value}, \text{sec_GD}(\mathcal{S}_{n_t+1}, \alpha_{sec}))$

$n_t \leftarrow n_t + 1$

$\mathcal{S}_{n_t} \leftarrow \mathcal{S}_{n_t+1}$

end while

return best_value

Algorithm 3 $\text{sec_GD}(\mathcal{S}, \alpha_{sec})$

Paramètres : un planning de production \mathcal{S} et un taux d'apprentissage α_{sec}

$n_f \leftarrow |V|$

$\nabla_{sec} \leftarrow -TT(DP(\mathcal{S}, n_f))$

best_value $\leftarrow TT(DP(\mathcal{S}, n_f))$

while $\nabla_{sec} \leq 0$ **do**

$\nabla_{sec} \leftarrow \alpha_{sec} \cdot \nabla_{sec} + TT(DP(\mathcal{S}, n_f + 1)) - TT(DP(\mathcal{S}, n_f))$

best_value $\leftarrow \min(\text{best_value}, TT(DP(\mathcal{S}, n_f + 1)))$

$n_f \leftarrow n_f + 1$

end while

return best_value

2.3. MÉTHODES DE RÉOLUTION HEURISTIQUES

Tout d'abord, un ensemble \mathcal{E} de n_t lots de livraison est déterminé en utilisant l'algorithme `AH_Clustering`. Ensuite, la méthode `PS` est appelée pour trouver un planning de production \mathcal{S}_{n_t} . Enfin, la solution est complétée par le calcul du planning de livraison à l'aide de la descente de gradient secondaire `sec_GD` qui recherche le meilleur nombre de lots de livraison n_f et renvoie la valeur de la fonction objectif de la meilleure solution trouvée.

L'algorithme `sec_GD` est également basé sur une descente de gradient classique qui commence à $n_f = |V|$ (nombre de véhicules disponibles) et incrémente n_f de 1 jusqu'à ce qu'un critère d'arrêt soit atteint. A partir d'un planning de production \mathcal{S} et pour chaque valeur testée de n_f , la méthode `DP` complète la solution en déterminant le plan de livraison. Ensuite, `TT` renvoie la valeur de la fonction objectif.

Notez que l'algorithme `AH_Clustering` qui construit le dendrogramme, n'est appliqué qu'une seule fois au début de l'exécution de l'heuristique BGD. Ensuite, la stratégie d'exploration s'apparente à une approche « top-down ». Elle consiste à partir du haut du dendrogramme (c'est-à-dire un seul lot de livraison), à inverser successivement chaque étape d'agglomération.

Les taux d'apprentissage α_{main} et α_{sec} appartiennent à $[0, 1]$. Un taux d'apprentissage supérieur à 0 a pour but de passer outre les minima locaux hypothétiques et l'intérêt d'être inférieur à 1 est d'éviter des pas trop importants après le passage du point de basculement.

2.3.4 Greedy Randomized Adaptive Search Procedure

Comme le montrent les expériences de calcul, la méthode proposée "Bilevel Gradient Descent" (BGD) a un temps de calcul court (environ 5 secondes) pour un programme d'une journée complète, ce qui laisse un peu de temps de calcul pour des possibilités d'amélioration de la solution trouvée par BGD. En effet, l'UBCO n'a besoin de recalculer une solution qu'une fois par minute ou qu'une fois toutes les 5 minutes. Nous développons ainsi une heuristique de type "Greedy Randomized Adaptive Search Procedure" (GRASP) [Resende et Ribeiro, 2014] basée sur la BGD, pour améliorer la qualité de la solution en utilisant au maximum le temps de calcul disponible. Les principes du GRASP sont les suivants : d'abord, générer un panel de solutions aléatoires biaisées, puis améliorer les solutions avec une recherche locale, et enfin, renvoyer la meilleure solution trouvée. Le GRASP proposé consiste à injecter un peu d'aléatoire dans l'étape *Mise en lot pré-ordo* et à utiliser la "Bilevel Gradient Descent" comme phase d'amélioration. Pour injecter de l'aléatoire dans la construction des lots, nous utilisons une nouvelle fonction de distance, où $\theta \in [0, 1]$ représente le pourcentage d'erreur autorisé dans le calcul de la distance.

$$dist_{random}(\theta, j, j') = (1 + \text{random}(-\theta, \theta)) \times dist_t \quad (2.25)$$

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

Le pseudo-code du GRASP est présenté dans Alg. 4. L'itération initiale du GRASP est la BGD (sans aléatoire), le nombre d'itérations n_{iter} représente alors le nombre de solutions aléatoires supplémentaires.

Algorithm 4 GRASP($J, \theta, \alpha_{main}, \alpha_{sec}, n_{iter}$)

```
Paramètres : un ensemble de tâches  $J$ , un pourcentage d'erreur  $\theta$ ,  
deux taux d'apprentissages  $\alpha_{main}$  et  $\alpha_{sec}$  et un nombre d'itérations  $n_{iter}$   
best_value  $\leftarrow$  BGD( $J, dist_t, \alpha_{main}, \alpha_{sec}$ )  
 $k \leftarrow 1$   
while  $k \leq n_{iter}$  do  
    best_value  $\leftarrow$  min(best_value, BGD( $J, dist_{random}, \alpha_{main}, \alpha_{sec}$ ))  
     $k \leftarrow k + 1$   
end while  
return best_value
```

2.4 Expérimentations numériques

2.4.1 Générateur d'instances

Nous évaluons les heuristiques proposées sur deux types d'instances : un type d'instances basé sur des données réelles de l'UBCO (appelées "UBCO instances") et un type d'instances aléatoires (appelées "Random instances") pour tester l'efficacité des méthodes dans un contexte plus général (pouvant s'étendre à d'autres unités de production de médicaments de chimiothérapie). Le processus de production de préparations magistrales de chimiothérapie est essentiellement le même quel que soit l'hôpital. Cependant, le nombre de postes de travail par isolateur, les capacités des stérilisateurs et les temps de traitement peuvent varier d'une unité à l'autre. De plus, une unité de production peut livrer plus de 3 hôpitaux, notamment dans le cas de la chimiothérapie à domicile. Le tableau 4.4 présente les paramètres utilisés pour générer les instances. Les premières lignes du tableau correspondent aux caractéristiques communes (capacités de production). Pour chaque type d'instance, nous générons plusieurs ensembles d'instances caractérisés par un certain nombre de tâches et un certain nombre de véhicules. Chaque ensemble contient 20 instances, donc un total de 480 instances sont générées, 240 instances pour chaque type. Chaque graphique et tableau de cette section est basé sur les mêmes 240 instances de type "UBCO" ou de type "Random". Les expérimentations numériques ont été réalisées sur un ordinateur portable Dell équipé d'un Intel(R) Core™ i5-7440HQ CPU (2.80 GHz), 8 GB RAM, sur Windows 10. Et les heuristiques sont codées en C++ et exécuté avec un seul thread.

Le but de la première expérience est de déterminer la valeur "idéale" du nombre de lots de livraison finaux n_f utilisés pour le *Mise en lot finale*, en fonction du nombre de livreurs

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

	UBCO instances	Random instances
Nombre de médicaments/tâches	$ J \in \{120, 140, 160, 180\}$	
Nombre de livreurs/véhicules	$ V \in \{3, 4, 5\}$	
Temps de contrôle	$p^{Ctrl} = 3$ minutes	
Durée de stérilisation	$p^{Ste} = 15$ minutes	
Horizon de temps des dates de fin souhaitées	$d_j \in [60, 600]$ minutes	
Nombre d'isolateurs	$ I = 3$	
Nombre de pharmaciens/machines par isolateur	$m_i = 2$ pharmaciens par isolateurs	
Temps de préparation	$p_j^{Prep} \in \{5, 10, 15\}$ minutes	$p_j^{Prep} \in [5, 20]$ minutes
Horizon de temps des dates de début au plus tôt	$r_j \in \{d_j - 90, d_j - 60\}$ minutes	$r_j \in [0, d_j - 60]$ minutes
Nombre d'hôpitaux	3	$ J $
Temps de trajet	l'UBCO est interne à un hôpital, distant de 15 minutes d'un autre hôpital et de 35 minutes d'un troisième hôpital. Dans chacun des hôpitaux, l'accès à un service d'oncologie peut prendre 5 minutes de plus.	Les points de livraison sont répartis dans un carré autour du site de production qui se positionne au milieu. Chaque point est à moins de 40 minutes de l'unité de production.

TABLE 2.4 – Description des deux types d'instances : "UBCO" et "Random"

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

$|V|$ et du nombre de chimiothérapies médicaments $|J|$. Pour étudier la meilleure valeur du paramètre n_f , toutes les instances de chaque type sont résolues en utilisant les heuristiques RA et ML et pour chaque valeur possible de $n_f \in \{|V|, \dots, |J|\}$ et $n_t = n_f$. Pour chaque instance, le nombre minimum de lots de livraison qui minimise le retard total, est conservé. Tous les résultats sont présentés dans la figure 2.8 avec des boîtes à moustaches et quatre graphiques différents selon le type d'instance et l'algorithme. Chaque boîte à moustache correspond, pour chaque ensemble de 20 instances, aux meilleures valeurs trouvées pour le nombre de lots de livraison n_f .

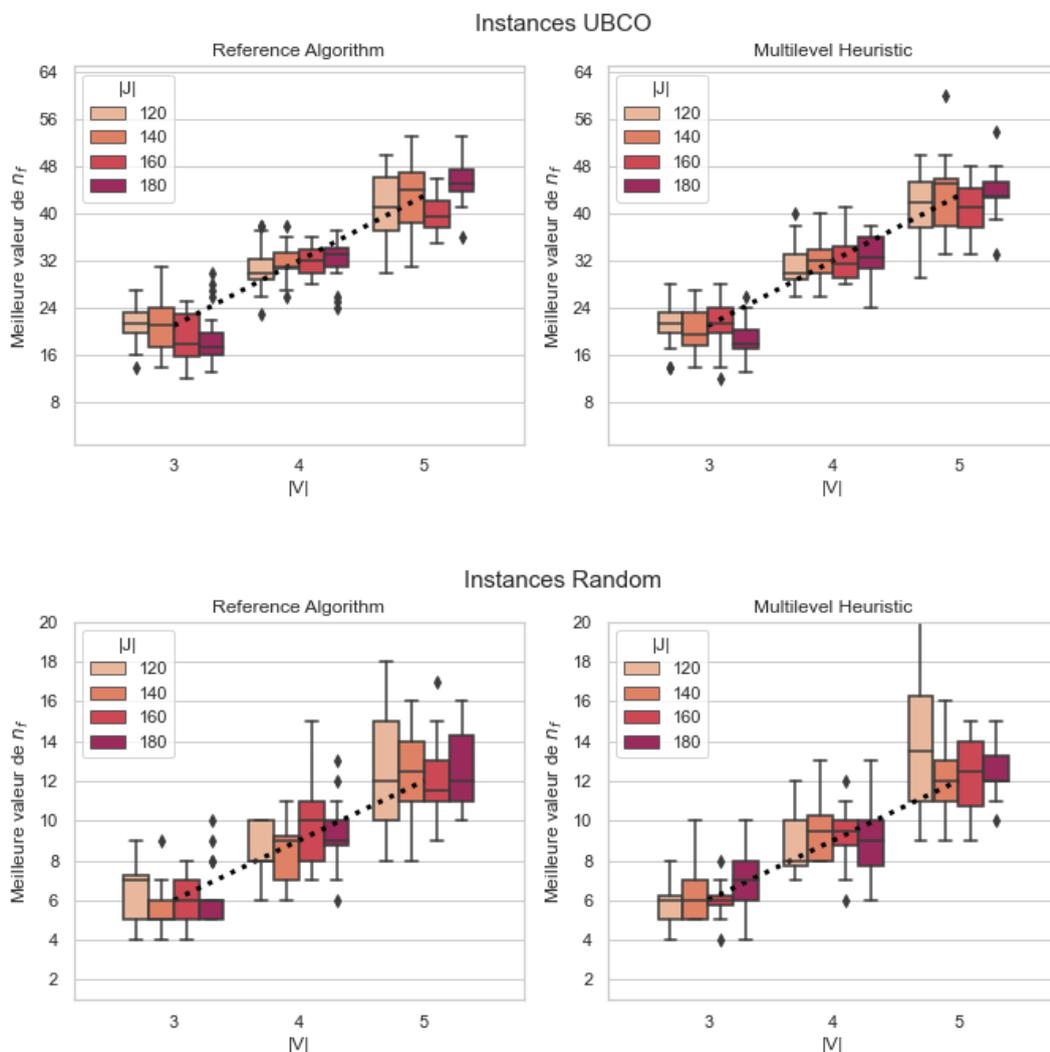


FIGURE 2.8 – Boîtes à moustaches des meilleure valeur de n_f pour les heuristiques RA et ML en fonction des types d'instances, des tailles d'instances et du nombre de livreurs $|V|$.

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

Les résultats expérimentaux présentés dans la figure 2.8 mettent en évidence l'impact du nombre de véhicules sur le meilleur nombre de lots de livraison. Cependant, la taille de l'instance (c'est-à-dire le nombre de tâches) n'a pas d'impact significatif sur le meilleur nombre de lots de livraison. La meilleure valeur médiane de n_f est également clairement liée au type d'instance ("UBCO" ou "Random"). Le meilleur nombre médian de lots de livraison est trois fois plus élevé pour les instances de type "UBCO" que pour les instances de type "Random". La raison principale est que, pour les instances de type "UBCO", l'unité de production est située dans un hôpital. Ainsi, un tiers du total des médicaments peut être livré à moins de 5 minutes du site de production, privilégiant ainsi plusieurs trajets de livraison courts.

Pour le reste des expériences numériques, le nombre de lots de livraison pour les heuristiques RA et ML est fixé comme suit :

- $n_f = 11(|V| - 1)$ pour les UBCO instances.
- $n_f = 3(|V| - 1)$ pour les Random instances,
- $n_t = n_f$ pour l'heuristique ML

Ces valeurs sont représentées par des lignes pointillées dans la Figure 2.8.

2.4.2 Temps de calcul

Le tableau 2.5 présente les temps de calcul moyens de chaque heuristique selon le type d'instance. Nous rappelons qu'un objectif est d'obtenir de bonnes solutions en un temps de calcul raisonnable pour disposer d'un outil d'aide à la décision avec un algorithme en ligne capable de calculer une nouvelle solution à chaque fois qu'un nouvel événement se produit. Pour respecter cette contrainte, nous fixons le temps de calcul maximum acceptable à une minute.

Temps de calcul					
		UBCO		Random	
Heuristique	Moyen	Maximum	Moyen	Maximum	
RA	0.18	0.36	0.18	0.37	
ML	0.34	0.63	0.35	0.60	
BGD	4.69	13.00	3.09	7.22	
GRASP 5	28.07	56.98	18.50	35.52	
GRASP 10	51.43	106.29	33.89	65.11	

TABLE 2.5 – Temps de calcul moyen et maximum (secondes) de chaque heuristique selon le type d'instance.

Toutes les heuristiques ont un temps de calcul moyen inférieur à une minute. Cependant, le temps de calcul maximum de GRASP avec 10 itérations est de près de 2 minutes. Comme prévu, la RA est l'heuristique la plus rapide et l'heuristique ML est la deuxième la plus

rapide. Les temps de calcul de l'heuristique ML est environ le double du temps de calcul de l'heuristique RA. La complexité temporelle de ces deux algorithmes étant principalement due à l'algorithme `AH_Clustering`, qui est utilisé deux fois dans l'heuristique ML et une seule fois dans le RA. L'exécution de l'algorithme BGD prend quant à lui moins de 5 secondes. Les temps de calcul de l'algorithme GRASP dépendent du nombre d'itérations. Cela prend environ 4 à 5 secondes par itération. La section suivante traite du compromis entre la qualité de la meilleure solution trouvée et le nombre d'itérations du GRASP. Les temps de calcul des méthodes BGD et GRASP sont fortement influencés par le type d'instance résolue. Cela peut s'expliquer par le fait que sur les instances de type UBCO le nombre idéal de lots de livraison est souvent plus élevé que sur les instances de type Random. L'exécution nécessite alors plus de pas de descente de gradient.

La Figure 2.9 montre que le temps de calcul des trois méthodes déterministes (RA, ML et BGD) varie peu à la différence du temps de calcul du GRASP qui est fortement lié à la taille de l'instance et aussi à moindre mesure par le nombre de véhicules disponibles. Comme évoqué précédemment, le nombre de pas de descente de gradient a un impact sur le temps de calcul, or plus de tâches à livrer avec plus de véhicules favorise un plus grand nombre de pas de descente de gradient.

2.4.3 Écarts relatifs

Pour comparer la qualité des heuristiques proposées, nous utilisons deux indicateurs : l'écart et le classement. Pour une heuristique donnée H et une instance donnée, son écart relatif $Gap(H)$ est déterminé par l'équation (2.26), où z^* correspond à la valeur de la fonction objectif de la meilleure solution trouvée par l'ensemble des heuristiques (RA, ML, BGD et GRASP avec 5 itérations et 10 itérations) et $z(H)$ est la valeur de la fonction objectif de la meilleure solution trouvée par l'heuristique H .

$$Gap(H) = 100 \times \frac{z(H) - z^*}{z^*} \quad (2.26)$$

Quelques expériences préliminaires ont été menées pour obtenir de bonnes valeurs pour les paramètres du BGD et du GRASP :

- les taux d'apprentissage pour le BGD et le GRASP sont fixés à $\alpha_{main} = \alpha_{sec} = 0.8$,
- le pourcentage d'erreur pour le GRASP est fixé à $\theta = 20\%$,
- le nombre d'itérations du GRASP est fixé à 5 et 10.

Le tableau 2.6 montre l'écart relatif (Gap) moyen pour chaque heuristique sur toutes les instances de type "UBCO" et de type "Random". On peut noter que la meilleure heuristique est clairement le GRASP à 5 ou 10 itérations. L'heuristique BGD surpasse les

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

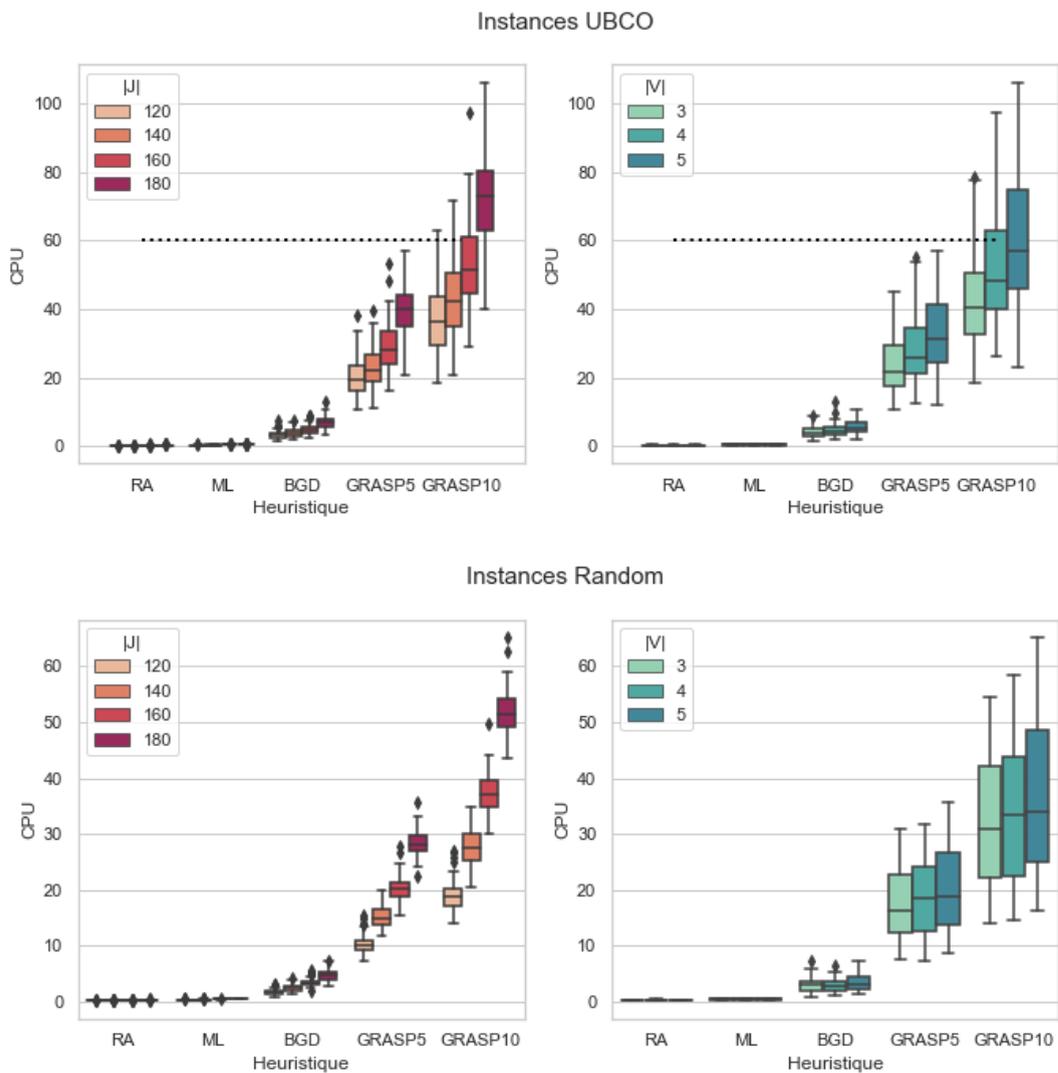


FIGURE 2.9 – Boîtes à moustaches du temps de calcul en fonction de l’heuristique et pour différents types d’instances, tailles d’instances et nombres de véhicules.

méthodes RA et ML ; cependant, sa performance reste moindre par rapport au GRASP. Enfin, le RA semble donner les pires résultats parmi toutes les heuristiques.

La Figure 2.10 présente quatre graphiques utilisant des boîtes à moustaches pour représenter les écarts relatifs (Gap) pour chaque heuristique, en fonction du type d’instance, du nombre de livreurs et de la taille de l’instance. Les variations de l’écart relatif (Gap) de chaque heuristique sont représentées par les tailles des boîtes. La variation de l’écart relatif (Gap) des heuristiques ML et RA augmentent lorsque la taille de l’instance ou le nombre des livreurs diminuent, ce qui montre que l’efficacité de ces deux heuristiques est

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

Class	Size	V	RA	ML	BGD	GRASP5	GRASP10	
UBCO	120	3	107.05	91.74	9	1.91	0	
		4	66.85	64.94	12.06	1.97	0.68	
		5	107.24	66.22	14.62	3.98	2.06	
	140	3	106.3	72.84	30.59	8.84	0.16	
		4	71.06	52.83	22.86	3.24	0	
		5	64.84	45.44	20.75	3.01	0	
	160	3	56.54	45.39	11.2	2.33	0	
		4	43.31	38.74	14.13	1.8	0	
		5	46.88	37.28	17.48	2.48	0.09	
	180	3	54.62	35.78	7.67	1.48	0	
		4	20.12	18.68	5.49	0.89	0	
		5	13.14	15.76	4.18	0.9	0.09	
		Moyenne		63.16	48.80	14.17	2.74	0.26
	Random	120	3	27.28	25.57	6.42	1.63	0
			4	34.82	34.06	9.49	2.01	0
5			38.93	35.9	7.39	1.3	0	
140		3	27.62	27.15	7.47	1.91	0	
		4	27.13	26.32	7.86	1.31	0.41	
		5	28.24	27.14	7.26	1.09	0	
160		3	23.87	27.11	7.04	1.41	0	
		4	23.01	21.83	6.04	0.92	0	
		5	21.39	19.9	4.66	0.8	0	
180		3	21.49	24.68	6.58	0.6	0	
		4	23.32	19.07	6.18	1.99	0	
		5	21.26	20.67	4.33	0.92	0	
		Moyenne		26.53	25.78	6.73	1.32	0.03

TABLE 2.6 – Écart relatif (Gap) moyens des différents heuristiques pour les 24 ensembles d'instances.

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

très variable pour les petites instances. De plus, même si l'écart moyen du ML est inférieur à l'écart moyen du RA, la différence d'écart relatif moyen entre le ML et le RA n'est pas significative par rapport à la différence entre le ML et la BGD. C'est notamment le cas pour les instances de type "Random" : la différence d'écart moyenne entre le RA et le ML est inférieure à 1%, et le RA est plus efficace que le ML pour deux ensembles d'instances (taille d'instance : {160, 180} et $|V| : 3$).

Notez que les fenêtres de temps pour produire et livrer les tâches entre les dates de début au plus tôt et les dates de livraison souhaitées de livraison sont plus longues pour les instances de type "Random". Par conséquent, cela offre plus de flexibilité pour la partie ordonnancement et donne plus d'importance à la *Construction d'itinéraire* sur la qualité de la solution.

La Figure 2.11 présente quatre graphiques utilisant des boîtes à moustaches pour représenter le retard moyen des tâches pour chaque heuristique, selon le type d'instance, le nombre de livreurs $|V|$, et le nombre de tâches (taille). Il est remarquable qu'autour de la valeur limite de production de 150 tâches calculée dans la section 1.3.3, le retard moyen est autour des 5 minutes peu importe l'heuristique utilisée pour les instances de type "UBCO". Dans la pratique, les retards sont identifiés aux 5 minutes près. Cependant, le nombre de véhicules est très important, surtout lorsque la demande est forte. Cela se confirme particulièrement sur les instances du type "Random" où comme les trajets sont en moyenne plus longs, dû à la définition des instances et que certaines dates de début au plus tôt peuvent être très proches de la date de livraison souhaitée, les grands retards sont inévitables.

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

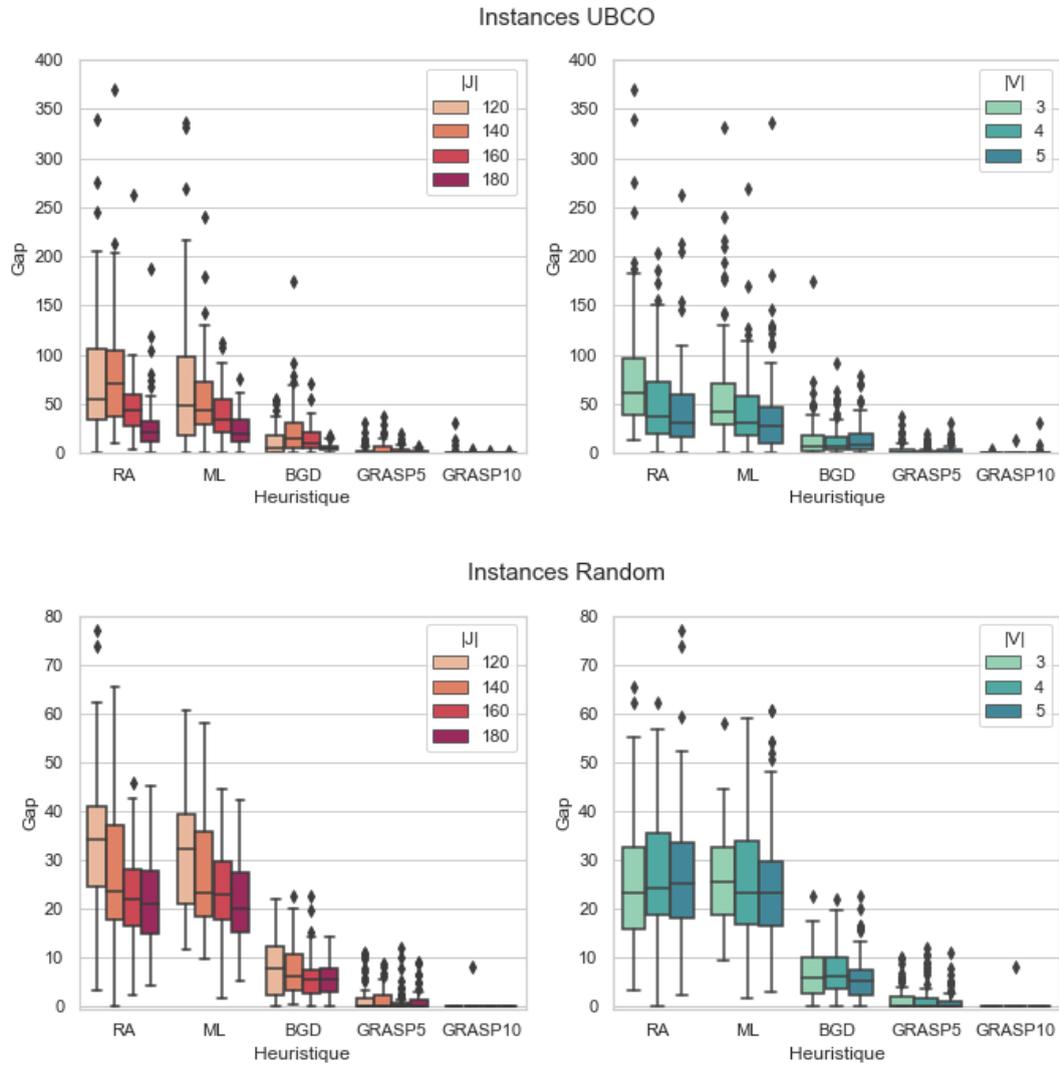


FIGURE 2.10 – Boîte à moustaches du gap (%) en fonction du type d'instance, de la taille d'instance et du nombre de livreurs.

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

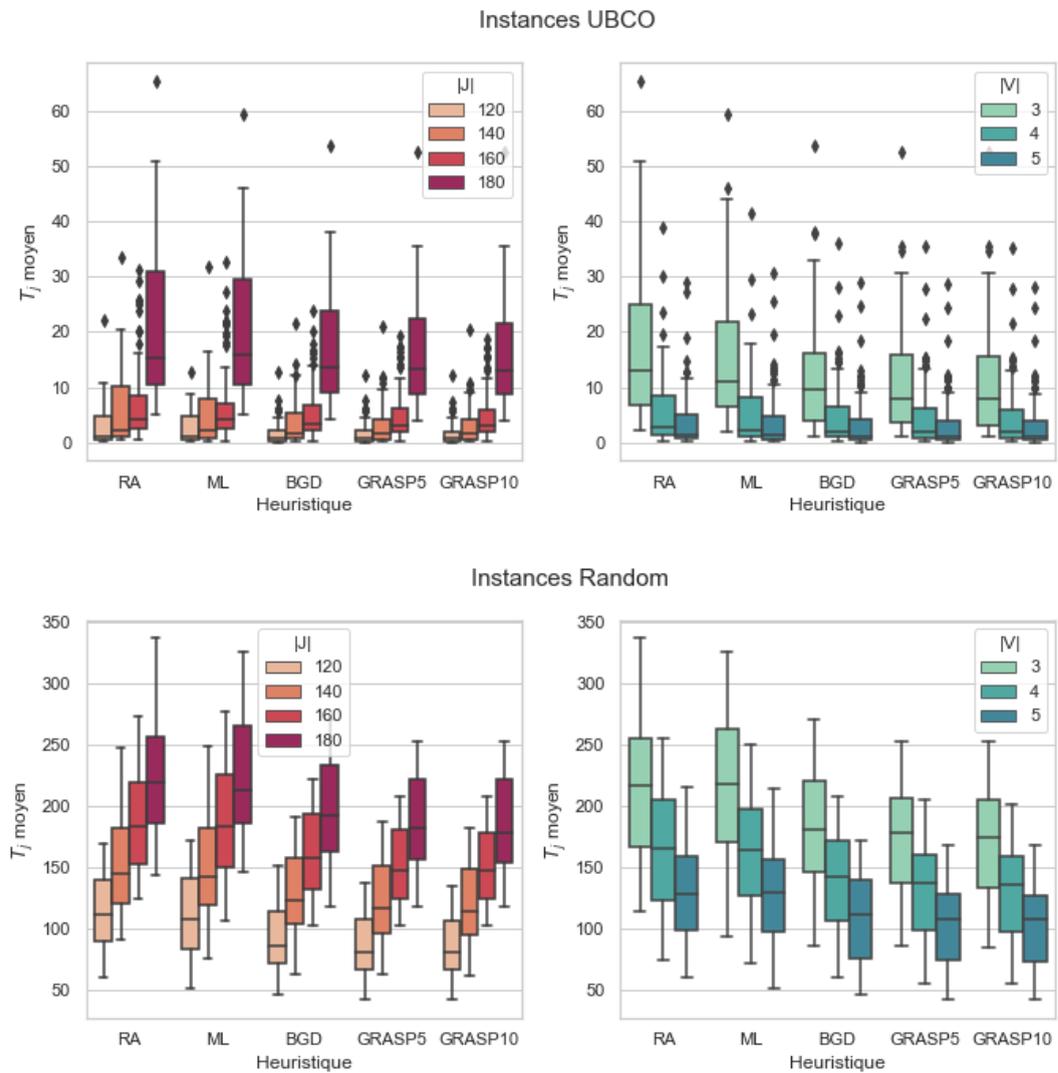


FIGURE 2.11 – Boîtes à moustaches du retard moyen ($\frac{T_j}{|J|}$) en fonction du type d'instance, de la taille d'instance et du nombre de livres.

2.4.4 Classement

L'indicateur de classement représente le rang entre les heuristiques, cela pour chacune des instances. La valeur de rang 1 est donnée à l'heuristique qui trouve la meilleure solution, et la valeur de rang 5 est donnée à l'heuristique, qui renvoie la pire solution. Pour mieux comparer les performances des méthodes, nous étudions le nombre de fois (en pourcentage) qu'une méthode donne le meilleur résultat, le deuxième meilleur résultat, etc.

La Figure 2.12 présente pour chaque méthode le pourcentage de fois qu'elle est classée pour chaque rang, cela sur l'ensemble des instances. Nous pouvons observer la même tendance que dans les résultats précédents avec l'indicateur d'écart relatif : le GRASP avec 10 itérations domine toutes les autres heuristiques, le GRASP avec 5 itérations domine l'heuristique BDG qui domine l'heuristique ML, et la pire heuristique est la RA. Comme le choix de la fonction de distance (équation (2.24)) a été défini pour être efficace pour les instances de type "UBCO", l'heuristique ML donne donc de moins bons résultats pour les instances de type "Random".

La Figure 2.13 permet de visualiser en deux dimensions les performances des différents heuristiques. En pointillé est représenté la limite de temps de calcul considéré comme raisonnable pour l'heuristique à utiliser en conditions réelles. Il paraît raisonnable donc d'utiliser le GRASP avec un nombre d'itérations entre 5 et 10 car le gain de qualité de la solution avec 5 itérations par rapport aux résultats de la BGD sont déjà significatifs.

2.4.5 Combien d'itérations pour le GRASP

Les résultats précédents ont montré que le GRASP à 5 ou 10 itérations permet d'obtenir les meilleurs résultats pour les instances de type "Random" et les instances de type "UBCO". Comme prévu, le nombre d'itérations affecte la qualité de la meilleure solution trouvée et les temps de calcul.

La Figure 2.14 montre l'évolution des écarts relatifs (Gap) et des temps de calcul en fonction du nombre d'itérations. L'évolution des temps de calcul semble proportionnelle au nombre d'itérations : environ 7 secondes par itération pour les instances avec 180 tâches et environ 3 secondes par itération pour les instances avec 120 tâches. Les instances de type "UBCO" tendent à indiquer que 5 itérations sont suffisantes, car dans la plupart des cas, l'amélioration n'est plus significative. Cependant, lors de la planification de la journée, le nombre de tâches à planifier diminue. Il est donc plus intéressant de proposer un GRASP avec une limite de temps qu'avec un nombre d'itérations limite. Il est aussi possible d'utiliser un temps de calcul beaucoup plus important (5 à 10 minutes) pour une première solution calculée la veille.

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

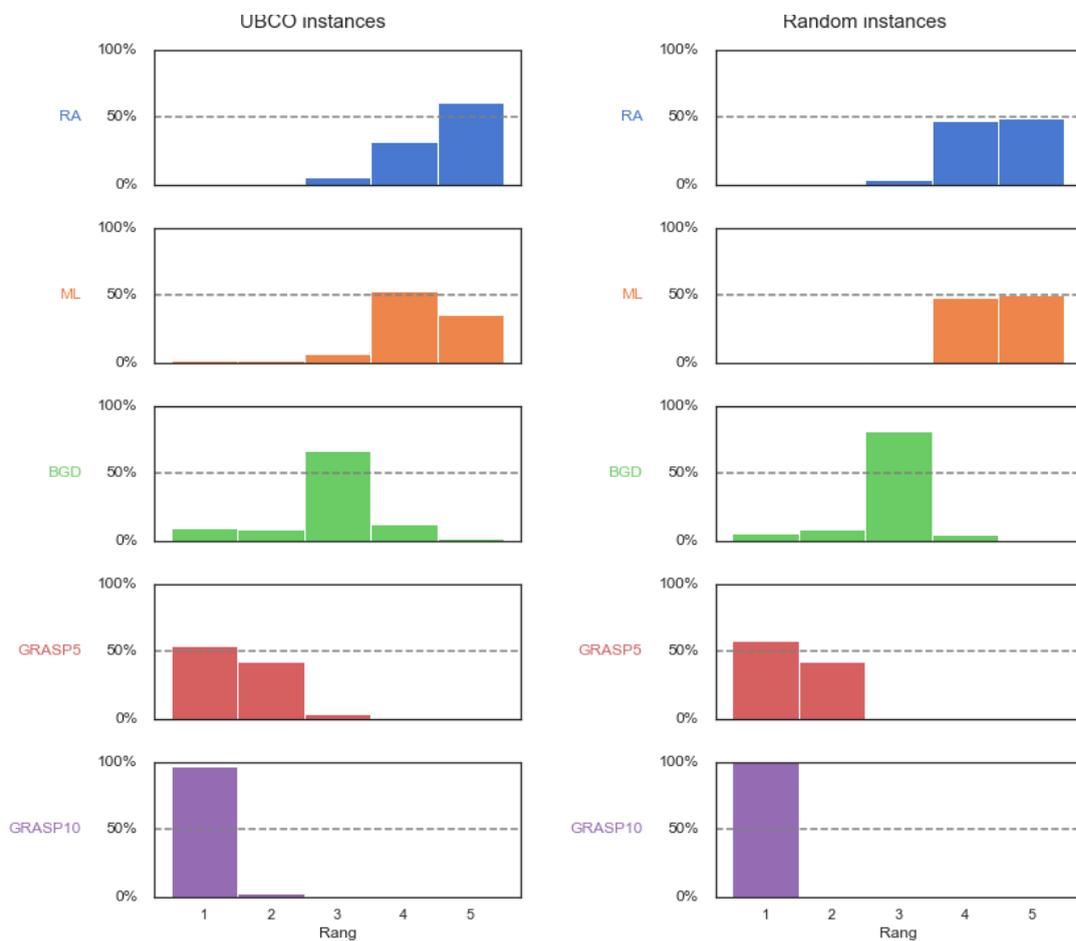


FIGURE 2.12 – Histogramme du classement des heuristiques en fonction du type d'instance. En cas d'égalité, le rang minimum est retenu pour les deux.

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

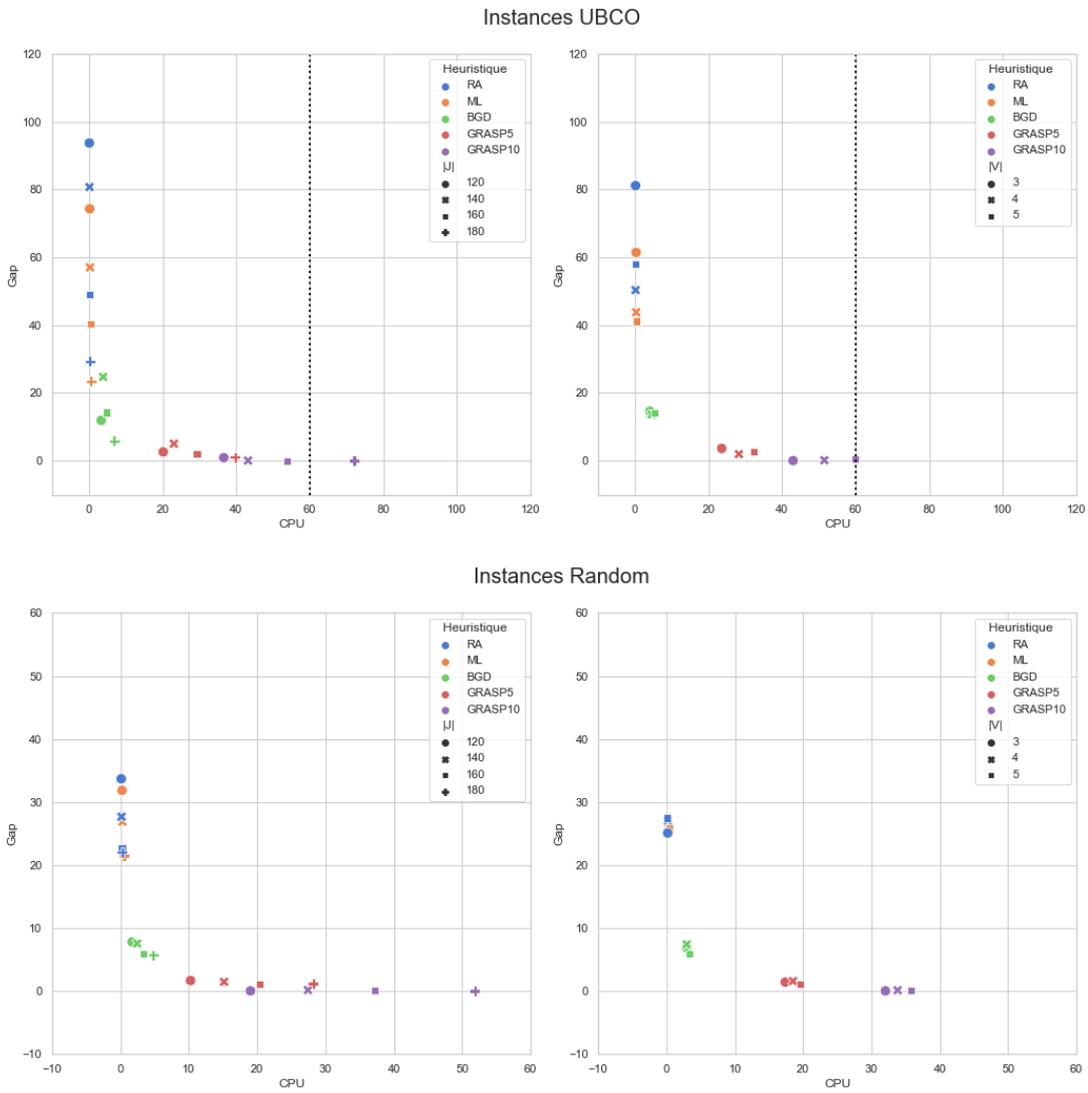


FIGURE 2.13 – Nuage de points situant les différentes heuristiques en fonction de leurs écart relatif moyen et de leurs temps de calcul moyen.

2.4. EXPÉRIMENTATIONS NUMÉRIQUES

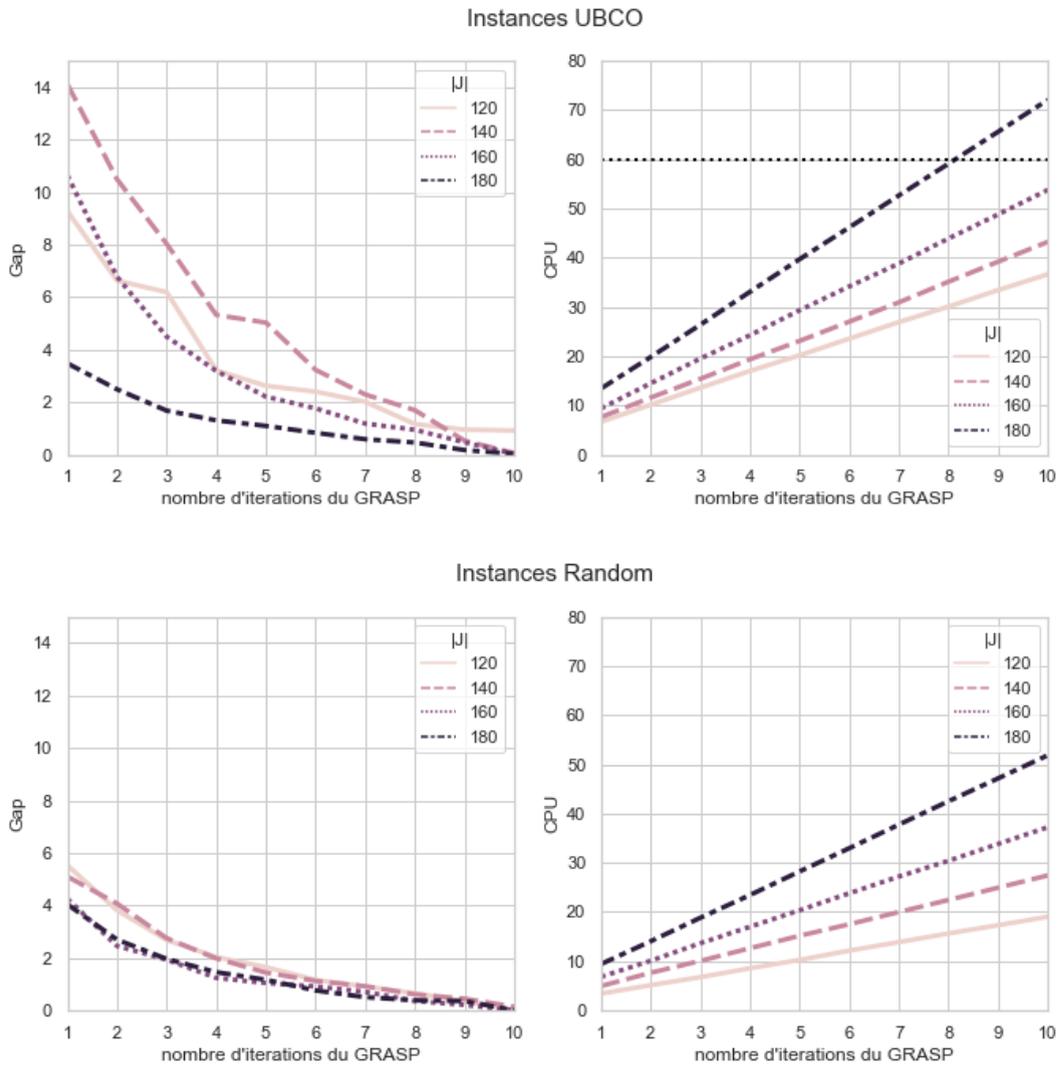


FIGURE 2.14 – La convergence du GRASP en fonction du nombre d’itérations, du type d’instance et de la taille d’instance. Les tracés de gauche représentent l’évolution du gap et les tracés de droite représentent l’évolution du temps de calcul.

2.5 Conclusion

Dans ce chapitre, nous avons modélisé le problème intégré de production et de livraison de préparation magistrale pour les chimiothérapies. Les hypothèses utilisées ont été justifiées par une étude terrain. Une modélisation mathématique a été proposée, mais sa résolution par des solveurs est beaucoup trop coûteuse en temps de calcul et en mémoire. Nous avons donc proposé des heuristiques basées sur des algorithmes de partitionnement et des règles de tris. Ces heuristiques ont été évaluées sur des instances générées artificiellement. Deux types d'instances ont été utilisés, des instances imitant les conditions de l'UBCO et des instances plus aléatoires. Suite à la comparaison des différentes heuristiques proposées, il a été mis en évidence l'importance d'une considération de livraison en amont de la production. Pour cela trois indicateurs ont été mis à l'épreuve, le temps de calcul, l'écart relatif et le classement. L'heuristique retenue pour une mise en place concrète à l'UBCO est le GRASP qui par sa diversification et sa flexibilité d'utilisation semble le plus adéquat. Cependant, l'ajout d'un protocole de rangement des échantillons avant leurs contrôles est nécessaire pour respecter l'ordonnancement du contrôle proposé. De la même manière, la mise en place d'un système de communication des directives aux livreurs est impérative.

Les travaux et les résultats menés dans ce chapitre ont donné lieu à des publications dans deux conférences internationales ([Robbes *et al.*, 2019b, Robbes *et al.*, 2019c]) et une nationale ([Robbes *et al.*, 2019a]) ainsi que dans une revue internationale [Robbes *et al.*, 2021a].

En perspectives, la méthode de construction des itinéraires de livraison pourrait être améliorée par l'utilisation d'opérateur de recherches locales tel le 2-opt. De même, proposé divers voisinages pour les différentes étapes de production permettrait un affinage des solutions.

Chapitre 3

Méthode exacte pour le problème de livraison avec production fixée

Sommaire

3.1	Présentation de la problématique	94
3.1.1	Hypothèses de modélisation	95
3.1.2	Positionnement dans l'état de l'art	97
3.2	Modèles de programmation linéaires en nombres entiers mixtes	98
3.2.1	Paramètres d'entrées et variables de décision	99
3.2.2	Problèmes de tournées de véhicules multi-trajet : MT-BDP et MTP	100
3.2.3	Problème du séquençement de livraison d'un trajet (TP)	102
3.3	Méthode de résolution exacte : 3BB	103
3.3.1	Optimisation de la séquence de livraison d'un trajet : TP	104
3.3.2	Optimisation de tournée d'un véhicule : MTP	109
3.3.3	Optimisation des tournées des véhicules : MT-BDP	112
3.4	Expérimentations numériques	114
3.4.1	Génération d'instances de la littérature	114
3.4.2	Comparaison du modèle MILP et du 3BB	119
3.4.3	Performance de la borne supérieure initiale	127
3.5	Conclusion	128

3.1 Présentation de la problématique

Dans le chapitre 2 plusieurs niveaux de décisions du problème intégré de production et de livraison ont été présentés. Les heuristiques proposées ont permis de fournir des solutions de qualité en moins de 5 minutes. Les plannings de livraisons ont été construits à l'aide d'une heuristique gloutonne. Lorsque un planning de production trouvé est construit, il est encore possible d'optimiser à l'exact la livraison afin d'améliorer la qualité des solutions.

Ce chapitre se focalise sur le développement d'une méthode exacte pour le problème de livraison avec production fixée. Il s'agit du problème d'*Affectation aux véhicules* et de la *Construction d'itinéraires*. Ce problème peut être considéré comme une variante du problème de tournées de véhicules multi-trajet (MT-VRP) où l'objectif principal est de minimiser le retard total.

Suite à une modélisation du problème à l'aide d'une décomposition en trois niveaux de décisions, nous proposons une méthode exacte arborescente utilisant plusieurs propriétés du problème. Cette méthode est testée sur des instances issues de la littérature des problèmes de tournées de véhicules.

Pour modéliser cette variante du MT-VRP, nous considérons un ensemble de clients (patients) qui doivent être livrés par une flotte de véhicules (livreurs). Chaque véhicule commence et finit ses trajets à un unique dépôt (l'UBCO). Chaque client doit être livré par un véhicule à une date de livraison souhaitée. On suppose que la composition et l'ordre des trajets (lot de livraison) sont connus. Chaque trajet est alors associé à une date de début au plus tôt correspondant à la dernière date de fin de contrôle des tâches à livrer. Le problème de livraison se concentre alors sur l'affectation des véhicules aux trajets et la définition de la séquence de livraison des patients de chacun des trajets. La fonction objectif est de minimiser le retard total de livraison. Même si les compositions des trajets sont connues, le problème reste difficile à résoudre. Le premier type de décision est une affectation des trajets aux véhicules. Le deuxième type de décision consiste à déterminer les séquences de livraison de chacun des trajets. En stratégie multi-trajet et considérant le retard total comme fonction objectif, la séquence de livraison d'un trajet doit être évaluée par deux critères conflictuels : le retard total associé au trajet et la durée du trajet. Le premier critère est corrélé à la date de départ du véhicule du dépôt qui dépend directement de la date de retour de son trajet précédent. Le deuxième critère a un impact sur l'évaluation des prochaines tournées.

Ce problème peut avoir diverses applications dans l'industrie, notamment dans la production à la commande [Stecke et Zhao, 2007] ou le commerce électronique [Boysen *et al.*, 2019]. Par exemple, dans les entreprises de fabrication et de distribution, la partie production peut être très contrainte de sorte que la production des commandes clients est

déjà planifiée sans remise en cause possible. Ainsi, les dates de fin d'exécution des commandes sont connues à l'avance. De plus, la distribution est souvent sous-traitée par des prestataires, et pour des raisons logistiques ou stratégiques, la composition des trajets est prédéterminée. En e-commerce, la préparation de commandes en entrepôt est un processus chronophage. Il consiste à regrouper les commandes en lots, à affecter ces lots aux préparateurs de commandes, à séquencer les lots affectés à chaque préparateur de commandes de manière à ce que tous les délais de commande des clients soient respectés [De Koster *et al.*, 2007]. Si l'on considère que les lots sont déjà déterminés (ou qu'un lot représente un ensemble d'articles d'un même client et doit être affecté à un seul préparateur) et que les préparateurs sont des véhicules, le problème devient similaire à la variante du MT-VRP étudiée ici.

3.1.1 Hypothèses de modélisation

Nous modélisons ce problème comme une variante du MT-VRP avec des dates de livraison souhaitées, des dates de départ au plus tôt et la composition et l'ordre des trajets fixés. Nous appelons notre problème, le problème de livraison de lots multi-trajets (MT-BDP).

Les notations utilisées sont celles présentées dans la section 2.2 : L'ensemble des tâches J à livrer est partitionné en une séquence de trajets \mathcal{B} qui est connue. Chaque trajet $B \in \mathcal{B}$ est composé d'un ensemble de clients et est caractérisé par une date de fin de production \hat{r}_B qui correspond à la date de départ au plus tôt du trajet pour le livrer. Chaque client $j \in J$ a une date de livraison souhaitée d_j sur le site u_j . On note D_j la date de livraison du client j , à déterminer. Le temps de trajet entre le site u_j et le site $u_{j'}$ est noté $t_{j,j'}$. L'unité de production, appelée dépôt, est représentée par le site 0. Le problème consiste à déterminer l'affectation des trajets aux véhicules, l'heure de départ des trajets, et les séquences de livraison de chacun des trajets, afin de minimiser le retard total $\sum_{j \in J} T_j$ où T_j représente le retard de livraison de la tâche j : $T_j = \max(0, D_j - d_j)$, $\forall j \in J$.

3.1. PRÉSENTATION DE LA PROBLÉMATIQUE

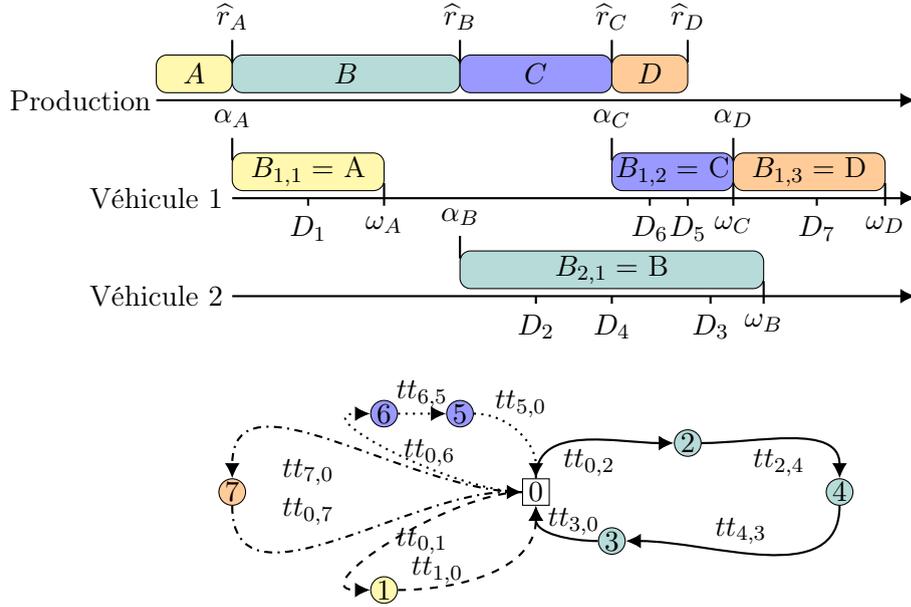


FIGURE 3.1 – Exemple d’une solution d’une instance du MT-BDP avec 7 clients partitionnées en 4 trajets et livrés par 2 véhicules.

Nous définissons les variables de décision suivantes :

- $B_{v,k}$ représente le k -ième trajet affecté au véhicule v ,
- α_t est la date de départ du dépôt du trajet t ,
- ω_t est la date d’arrivée au dépôt du trajet t .

La Figure 3.1 représente une solution d’une instance du MT-BDP. La composition et l’ordre des trajets est $A = \{1\}$, $B = \{2, 3, 4\}$, $C = \{5, 6\}$, et $D = \{7\}$. La flotte est composée de deux véhicules. Le véhicule 1 fait d’abord le trajet A , $B_{1,1} = \{1\}$, puis le trajet C , $B_{1,2} = \{5, 6\}$, et enfin le trajet D , $B_{1,3} = \{7\}$. Le trajet B est réalisé par le véhicule 2, $B_{2,1} = \{2, 3, 4\}$. Le planning de livraison des lots est représenté par un diagramme de Gantt et les itinéraires sont représentés par le graphe en dessous du diagramme de Gantt. Les trajets sont bien effectués dans l’ordre des dates de fin de production.

Même si une grande part de la complexité du MT-VRP vient de la constitution des lots, sa variante MT-BDP est tout même NP-difficile au sens fort.

Démonstration. Considérons un cas particulier du MT-BDP comme suit : chaque lot est constitué d’une seule tâche et un seul véhicule est disponible. Ce problème peut être modélisé comme un problème d’ordonnancement avec une machine et des dates de début au plus tôt où $\forall j \in J$ le temps de traitement p_j est égal à $t_{0,j} + t_{j,0}$. Ce problème d’ordonnancement est noté $1|r_j|\sum T_j$. Ce problème est montré NP-difficile au sens fort [Lenstra

et Rinnooy Kan, 1978]. □

Nous proposons une décomposition du problème MT-BDP en deux sous-problèmes :

- Le problème multi-trajet (MTP) est un cas particulier du MT-BDP où seul un véhicule est disponible pour la livraison. Si l'affectation des lots aux véhicules est donnée, résoudre le MT-BDP revient à résoudre autant de MTP qu'il y a de véhicules utilisés. Ce problème est démontré NP-difficile au sens fort par la même démonstration que pour le MT-BDP.
- Le problème du séquençement de livraison d'un trajet (TP) est le cas particulier du MT-BDP où un seul lot doit être livré. Nous introduisons cependant un objectif secondaire qui est la minimisation de la date de retour au dépôt. Cela devient donc un problème d'optimisation bi-objectif. Ce sous-problème intervient dans la résolution d'un MTP pour sélectionner des trajets de livraisons non dominés. Ce problème étant une variante du problème de voyageur de commerce avec un deuxième objectif, son problème de décision est NP-complet.

3.1.2 Positionnement dans l'état de l'art

Le problème de tournées de véhicules (VRP) est l'un des problèmes les plus étudiés dans la littérature de la recherche opérationnelle [Mor et Speranza, 2020]. Sa variante multi-trajet (MT-VRP), où chaque véhicule est autorisé à effectuer plusieurs trajets, a également une littérature abondante [Cattaruzza *et al.*, 2018]. Dans la plupart des cas, la fonction objectif est la minimisation de la distance totale parcourue par la flotte de véhicules.

Le problème que nous abordons a quelques points communs avec le MT-VRP avec fenêtres temporelles. Dans cette variante, chaque client doit être livré dans un intervalle de temps donné. Il est possible d'arriver chez le client plus tôt que la fenêtre horaire, mais le véhicule doit alors attendre le début de la fenêtre pour la remise du produit au client. À l'inverse, la fin des fenêtres temporelles est une date d'échéance qui doit être respectée. Dans [Hernandez *et al.*, 2014], les auteurs proposent un algorithme exact à deux phases pour résoudre le MT-VRP avec des fenêtres temporelles et une durée limitée des trajets. La première phase utilise la contrainte de durée limitée pour réduire les possibilités de mise en lots. La deuxième phase est un algorithme Branch-and-Price. Cet algorithme est amélioré dans [Hernandez *et al.*, 2016] afin de résoudre le MT-VRP avec des fenêtres temporelles sans la contrainte de durée limitée. Les dates de départ au plus tôt sont introduites dans le problème MT-VRP avec des fenêtres de temps dans [Cattaruzza *et al.*, 2016]. Les auteurs proposent un algorithme mémétique avec des individus représentés par des tours géants et scindés par une procédure de labellisation. L'algorithme est testé sur un nouvel ensemble

d’instances de référence basé sur les instances de Solomon [Solomon, 1987]. Pour s’attaquer à la version multi-dépôt du MT-VRP avec des fenêtres temporelles et des dates de départ au plus tôt, [Zhen *et al.*, 2020] proposent deux algorithmes d’optimisation : un algorithme d’essaim de particules hybride et un algorithme génétique hybride.

Dans les problèmes de tournées de véhicules, le retard total de livraison est souvent représenté comme un coût de pénalité de dépassement des fenêtres de temps. Les fenêtres temporelles souples (également appelées fenêtres temporelles flexibles, voir les premiers travaux [Sexton et Choi, 1986]) sont modélisées comme deux coûts de pénalité linéaires : un coût de pénalité d’avance sur la livraison et un coût de pénalité de retard. Une méthode exacte utilisant une génération de colonnes pour le VRP avec des fenêtres temporelles souples est présentée dans [Liberatore *et al.*, 2011]. Dans [Shelbourne *et al.*, 2017], le VRP avec des dates de livraison souhaitées et des dates de départ au plus tôt est résolu à l’aide d’un algorithme de reconnexion de chemin et d’un algorithme de recherche locale. Trois formulations et inégalités valides sont présentées dans [Archetti *et al.*, 2015] pour la version multi-période du VRP avec dates de livraison souhaitées. Ce problème est résolu par un algorithme Branch-and-Bound et un algorithme de descente à voisinage variable combiné à un modèle mathématique dans [Larrain *et al.*, 2019]. L’article récent [Karoonsontawong *et al.*, 2020] porte sur la résolution du MT-VRP avec des fenêtres temporelles souples. Le problème étudié par les auteurs est un problème de tournées de véhicules multi-trajets avec des fenêtres de temps souples et des contraintes d’heures supplémentaires, qui est une version plus générale du MT-VRP avec des fenêtres de temps souples. Les auteurs proposent une formulation hiérarchique multi-objectif et mono-objectif et deux heuristiques, l’une basée sur une construction de tournée unique et l’autre sur la construction de tournées multiples. À notre connaissance, il n’existe aucun travail dans la littérature sur le MT-VRP avec des fenêtres temporelles souples (ou dates d’échéance) et des dates de départ au plus tôt.

3.2 Modèles de programmation linéaires en nombres entiers mixtes

Dans cette section nous proposons un modèle linéaire en nombres entiers mixtes (MILP) pour le MT-BDP, le MTP et le TP. Ces MILP sont utilisés pour évaluer la qualité de notre méthode exacte.

3.2.1 Paramètres d'entrées et variables de décision

Le tableau 3.1 résume les notations des paramètres d'entrées des modèles linéaires en nombres mixtes. Ces dernières ont été introduites dans les sections 2.2 et 3.1.1. Le paramètre α est utile pour le modèle du TP et correspond à une date de départ du dépôt calculé comme la date maximale entre la date de disponibilité du véhicule et la date de fin de production du lot.

Notation	Description
J	ensemble des clients
V	ensemble des véhicules
\mathcal{B}	liste ordonnée des trajets
$B_{v,t}$	t -ième trajet effectué par le véhicule v
r_B	date de départ au plus tôt du trajet B
d_j	date de livraison souhaitée pour le client j
u_j	point de livraison du client j
$tt_{j,j'}$	temps de trajet entre u_j et $u_{j'}$
α	date de départ du dépôt.

TABLE 3.1 – Paramètres d'entrées des modèles

Pour modéliser en programmation linéaire en nombres mixtes, nous introduisons un ensemble de variables de décisions résumées dans le tableau 3.2. MTP et TP étant des cas particuliers du MT-BDP, les variables utilisées sont majoritairement les mêmes dans les trois modèles à l'exception de $y_{B,v}$ qui est une variable d'affectation nécessaire uniquement dans le modèle MILP du MT-BDP et ω qui permet de calculer l'objectif secondaire du TP.

Notation et domaine	Description
Commun aux modèles	
$x_{0,j} \in \{0, 1\}, \forall j \in J$	$x_{0,j} = 1$ si l'arête (dépôt, u_j) est empruntée par un véhicule, 0 sinon.
$x_{j,0} \in \{0, 1\}, \forall j \in J$	$x_{j,0} = 1$ si l'arête (u_j , dépôt) est empruntée par un véhicule, 0 sinon.
$x_{j,j'} \in \{0, 1\}, \forall (j, j') \in J^2$	$x_{j,j'} = 1$ si l'arête ($u_j, u_{j'}$) est empruntée par un véhicule, 0 sinon.
$D_j \in \mathbb{R}_+, \forall j \in J$	date de livraison du client j .
$T_j \in \mathbb{R}_+, \forall j \in J$	retard de livraison du client j .
Dédié au modèle du MT-BDP	
$y_{B,v} \in \{0, 1\}, \forall B \in \mathcal{B}, \forall v \in V$	$y_{B,v} = 1$ si le lot B est livré par le véhicule v , 0 sinon.
Dédié au modèle du TP	
$\omega_B \in \mathbb{R}_+$	date de retour au dépôt de la livraison du lot B .

TABLE 3.2 – Variables de décisions des modèles

3.2.2 Problèmes de tournées de véhicules multi-trajet : MT-BDP et MTP

Dans le modèle que nous présentons, deux contraintes utilisent une borne supérieure des dates de livraison que nous notons \mathbb{M} . Sélectionner la plus petite valeur possible pour \mathbb{M} permet d'améliorer l'efficacité du modèle MILP. Dans les expérimentations numériques, nous donnons une valeur à \mathbb{M} basée sur les données de l'instance.

L'objectif d'optimisation est la minimisation du retard total, soit la somme des retards des lots de livraison :

$$\text{MIN} \quad \sum_{B \in \mathcal{B}} \sum_{j \in B} T_j \quad (3.1)$$

Pour chaque trajet, il y a un et un seul départ du dépôt :

$$\forall B \in \mathcal{B} \quad \sum_{j \in B} x_{0,j} = 1 \quad (3.2)$$

Pour chaque trajet, il y a un et un seul retour au dépôt :

$$\forall B \in \mathcal{B} \quad \sum_{j \in B} x_{j,0} = 1 \quad (3.3)$$

Pour chaque trajet, il y a un unique départ depuis chacun des sites de livraison associés :

$$\forall B \in \mathcal{B}, \forall j \in B \quad \sum_{j' \in B \setminus \{j\}} x_{j,j'} + x_{j,0} = 1 \quad (3.4)$$

Pour chaque trajet, il y a une unique arrivée à chacun des sites de livraison associés :

$$\forall B \in \mathcal{B}, \forall j \in B \quad \sum_{j' \in B \setminus \{j\}} x_{j',j} + x_{0,j} = 1 \quad (3.5)$$

La date de livraison dépend de la séquence de livraison du trajet :

$$\forall B \in \mathcal{B}, \forall (j, j') \in B^2, j \neq j' \quad D_j \geq D_{j'} + t_{j',j} - \mathbb{M} \cdot (1 - x_{j',j}) \quad (3.6)$$

Les dates de livraison aux clients sont postérieures à la date de fin de production plus le temps de livraison directe :

$$\forall B \in \mathcal{B}, \forall j \in B \quad D_j \geq \hat{r}_B + t_{0,j} \quad (3.7)$$

3.2. MODÈLES DE PROGRAMMATION LINÉAIRES EN NOMBRES ENTIERS MIXTES

Calcul du retard de la livraison aux clients :

$$\forall B \in \mathcal{B}, \forall j \in B \quad T_j \geq D_j - d_j \quad (3.8)$$

Les dates de livraison des tâches d'un lot sont postérieures à la date de retour au dépôt du véhicule de son trajet précédent :

$$\forall v \in V, \forall (B, B') \in \mathcal{B}^2, \widehat{r}_B > \widehat{r}_{B'}, \forall j \in B, \forall j' \in B' \quad D_j \geq D_{j'} + t_{j',0} + t_{0,j} - \mathbb{M} \cdot (2 - y_{B,v} - y_{B',v}) \quad (3.9)$$

Chaque trajet est affecté à un unique véhicule :

$$\forall B \in \mathcal{B} \quad \sum_{v \in V} y_{B,v} = 1 \quad (3.10)$$

Le modèle du MT-BDP s'écrit donc :

$$\text{MIN} \quad \sum_{B \in \mathcal{B}} \sum_{j \in B} T_j$$

sous contraintes :

contraintes (3.2) – (3.10)

$$\begin{array}{ll} x_{0,j} \in \{0, 1\} & \forall B \in \mathcal{B}, \forall j \in B \\ x_{j,j'} \in \{0, 1\} & \forall B \in \mathcal{B}, \forall (j, j') \in B^2 \\ D_j \in \mathbb{R}_+ & \forall B \in \mathcal{B}, \forall j \in B \\ T_j \in \mathbb{R}_+ & \forall B \in \mathcal{B}, \forall j \in B \\ y_{B,v} \in \{0, 1\} & \forall B \in \mathcal{B}, \forall v \in V \\ \omega_B \in \mathbb{R}_+ & \forall B \in \mathcal{B} \end{array}$$

Étant un cas particulier du MT-BDP, le modèle du MT-BDP peut être utilisé pour résoudre le MTP. Cependant, l'utilisation de la variable $y_{B,v}$ n'a pas d'intérêt puisque la liste des trajets \mathcal{B}_v en entrée pour le MTP est affectée à un seul véhicule v . Nous proposons donc de remplacer la contrainte 3.9 par la contrainte 3.11 et de supprimer la contrainte 3.10 afin d'améliorer le modèle.

$$\forall v \in V, \forall (B, B') \in \mathcal{B}^2, \widehat{r}_B > \widehat{r}_{B'}, \forall j \in B, \forall j' \in B' \quad D_j \geq D_{j'} + t_{j',0} + t_{0,j} \quad (3.11)$$

3.2. MODÈLES DE PROGRAMMATION LINÉAIRES EN NOMBRES ENTIERS MIXTES

Le modèle du MTP s'écrit donc :

$$\text{MIN} \quad \sum_{B \in \mathcal{B}} \sum_{j \in B} T_j$$

sous contraintes :

contraintes (3.2) – (3.18), (3.11)

$$\begin{array}{ll} x_{0,j} \in \{0, 1\} & \forall B \in \mathcal{B}_v, \forall j \in B \\ x_{j,j'} \in \{0, 1\} & \forall B \in \mathcal{B}_v, \forall (j, j') \in B^2 \\ D_j \in \mathbb{R}_+ & \forall B \in \mathcal{B}_v, \forall j \in B \\ T_j \in \mathbb{R}_+ & \forall B \in \mathcal{B}_v, \forall j \in B \\ y_{B,v} \in \{0, 1\} & \forall B \in \mathcal{B}_v, \forall v \in V \\ \omega_B \in \mathbb{R}_+ & \forall B \in \mathcal{B} \end{array}$$

3.2.3 Problème du séquençement de livraison d'un trajet (TP)

Le problème du séquençement de livraison d'un trajet prend en entrée la date de départ du trajet $\alpha \in \mathbb{R}_+$. La majorité des contraintes du modèle sont issues du modèle du MT-BDP à l'exception de la contrainte 3.16 qui est la modification de la contrainte 3.7 et la contrainte 3.17 qui permet de calculer la date de retour au dépôt. Dans ce modèle on note B le trajet dont on cherche à déterminer sa séquence de livraison.

$$\min \quad \left(\sum_{j \in B} T_j, \omega_B \right) \quad (3.12)$$

$$\forall j \in B \quad \sum_{j' \in B \setminus \{j\}} x_{j,j'} + x_{j,0} = 1 \quad (3.13)$$

$$\forall j \in B \quad \sum_{j' \in B \setminus \{j\}} x_{j',j} + x_{0,j} = 1 \quad (3.14)$$

$$\forall (j, j') \in B^2, j \neq j' \quad D_j \geq D_{j'} + t_{j',j} - \mathbb{M} \cdot (1 - x_{j',j}) \quad (3.15)$$

$$\forall j \in B \quad D_j \geq \alpha + t_{0,j} \quad (3.16)$$

$$\forall j \in B \quad \omega_B \geq D_j + t_{j,0} \quad (3.17)$$

$$\forall j \in B \quad T_j \geq d_j - D_j \quad (3.18)$$

$$\forall (j, j') \in B^2 \quad x_{j,j'} \in \{0, 1\} \quad (3.19)$$

$$\forall j \in B \quad \omega_j \in \mathbb{R} \quad (3.20)$$

$$\forall j \in B \quad T_j \in \mathbb{R}_+ \quad (3.21)$$

3.3 Méthode de résolution exacte : 3BB

Pour obtenir une solution au problème MT-BDP, plusieurs décisions doivent être prises. Les trajets doivent être affectés aux véhicules et le séquençement des livraisons des trajets doivent être construits. Nous proposons une méthode de résolution exacte, nommée 3BB, basée sur un algorithme de séparation et d'évaluation à trois niveaux de décision :

1. Au plus haut niveau des décisions, les trajets doivent être affectés aux véhicules. Nous résolvons ce problème d'affectation par un algorithme de séparation et d'évaluation, appelé $B\&B_{MT-BDP}$. Cet algorithme implémente des procédures de rupture de symétrie et est décrit dans la section 3.3.3.
2. Au niveau de décision intermédiaire, les trajets sont affectés aux véhicules et une séquence de livraison doit être sélectionnée pour chaque trajet effectué par un même véhicule. À ce niveau de décision, le problème est le MTP. La résolution du MTP pour chaque véhicule est basée sur un algorithme de séparation et d'évaluation utilisant des propriétés de dominances, une mémoire externe et une approche ϵ -contrainte. Il est noté $B\&B_{MTP}$ et est décrit dans la section 3.3.2.
3. Le niveau des décisions le plus bas consiste à définir une séquence de livraison non-dominée¹ d'un unique trajet pour une date de départ du dépôt donnée. Le problème qui en résulte est le TP. Les deux fonctions objectifs sont la minimisation du retard total et la minimisation de la date de retour au dépôt. Pour résoudre le TP nous proposons un algorithme de séparation et d'évaluation lexicographique noté $B\&B_{TP}$ et décrit dans la section 3.3.1.

L'interconnexion des trois niveaux est illustrée par la figure 3.2. La résolution du problème de plus haut niveau nécessite de résoudre le problème de niveau intermédiaire (MTP) pour chaque véhicule. Résoudre le problème de décision intermédiaire nécessite de résoudre le dernier niveau de décision (TP) pour chaque trajet.

1. Une solution X d'un problème de minimisation multi-objectifs avec ses valeurs d'objectifs correspondantes x_1, \dots, x_n est dite non-dominée s'il n'existe pas de solution Y avec ses valeurs d'objectifs correspondantes y_1, \dots, y_n tels que $\forall k \in \{1, \dots, n\} y_k \leq x_k$.

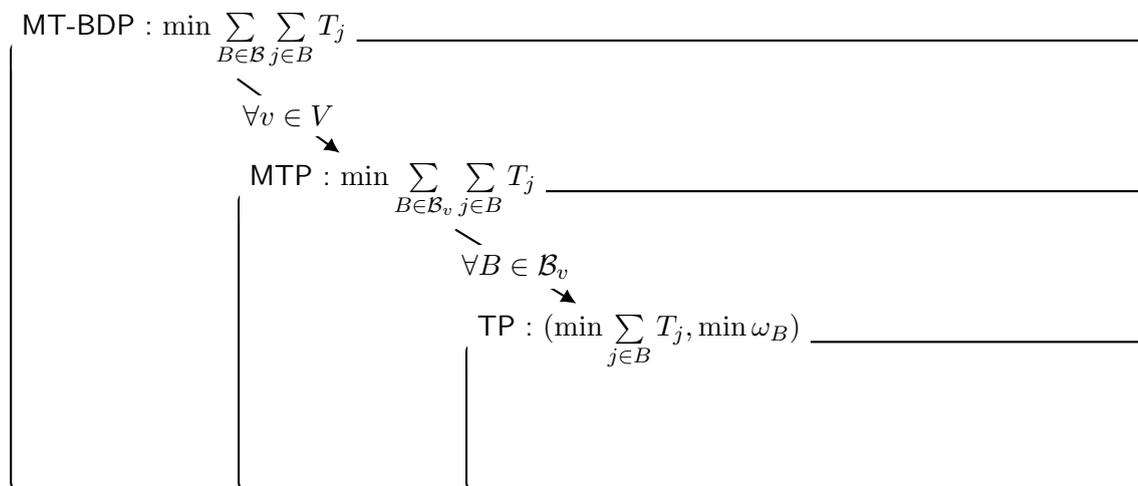


FIGURE 3.2 – L’organigramme des trois niveaux de décisions

3.3.1 Optimisation de la séquence de livraison d’un trajet : TP

Le problème du séquençage de livraison d’un trajet bi-objectif (TP) est une variante bi-objectif du problème du voyageur de commerce. En effet, les contraintes sont les mêmes, mais les objectifs sont de minimiser le retard total et de minimiser la date de retour au dépôt. La fonction objectif secondaire du TP est de minimiser la date de retour, soit une version de l’objectif original du problème du voyageur de commerce basé sur les dates et non les distances. Cela prouve directement que le TP est NP-difficile. Cependant, en raison de la fonction objectif principale, la plupart des méthodes disponibles pour le TSP, comme l’élimination de symétries (le sens de parcours d’un cycle n’a pas d’importance dans le cas du TSP), les relaxations (relaxation de la contrainte de sous-tours) et les bornes inférieures sont inutilisables. Dans la littérature, une variante du TSP avec des fenêtres temporelles a déjà été traité dans [Fachini et Armentano, 2020] à l’aide d’un algorithme de programmation dynamique pour trouver la solution optimale. Les auteurs appliquent également leur méthode au TSP avec des fenêtres temporelles souples, mais la méthode devient une heuristique pour ce problème.

Dans cette section nous supposons qu’une instance du TP est définie par une date de départ α , un trajet B et la date de retour au dépôt d’une solution S est notée par ω .

Séparation et évaluation lexicographique $B \& B_{TP}$

Nous proposons un algorithme de séparation et d’évaluation lexicographique que nous notons $B \& B_{TP}$. L’objectif principal à minimiser est le retard total $\sum_{j \in B} T_j$ induit par les

tâches du lot B , et l'objectif secondaire est de minimiser la date de retour au dépôt ω .

La stratégie de branchement du $B\&B_{TP}$ est représentée sur la figure 3.3 : une branche par tâche non séquencée jusqu'à ce que chaque tâche soit séquencée.

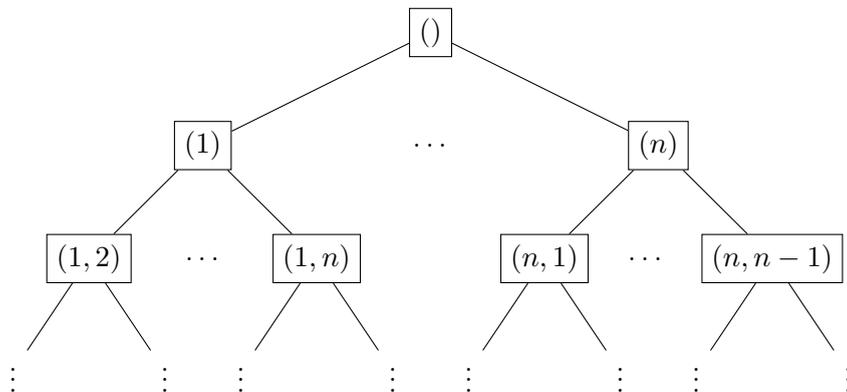


FIGURE 3.3 – Stratégie de branchement du $B\&B_{TP}$ pour un lot avec n tâches

L'efficacité d'un algorithme de séparation et d'évaluation dépend principalement de la qualité de ses bornes et aussi de sa stratégie d'exploration. Une borne supérieure et une borne inférieure est calculé pour chaque nœud. Après plusieurs expérimentations, la stratégie d'exploration retenue est d'explorer en priorité le nœud avec la meilleure borne inférieure en premier.

Borne supérieure et borne inférieure

La borne supérieure choisie est commune à chaque nœud et correspond à la meilleure solution réalisable trouvée. La borne supérieure est initialisée par la meilleure solution parmi les 6 solutions construites à l'aide des méthodes suivantes :

1. séquence de livraison par heuristique du plus proche voisin : en partant du dépôt, à chaque fois qu'un client est livré, le suivant est celui nécessitant le moins de temps de trajet pour y parvenir.
2. séquence de livraison par heuristique du plus proche voisin dans le sens inverse : lorsque la séquence de livraison par heuristique de plus proche voisin est construite, il est possible de définir la séquence inverse.
3. séquence de livraison par date de livraison souhaitée croissante : l'ordre de livraison des clients est par priorité de leurs dates de livraison souhaitées.
4. séquence de livraison par date de livraison souhaitée décroissante : cette séquence est la séquence inverse de la précédente.

5. solutions non-dominées déjà connues pour toute date de départ du dépôt. Pour chaque résolution d'un TP, les solutions non-dominées trouvées sont sauvegardées dans une mémoire externe. C'est dernières peuvent même être prouvée non-dominées avant la résolution du $B \& B_{TP}$ grâce aux propriétés de conservation de la non-dominance 3.3.1 et 3.3.2.

Proposition 3.3.1. *Soit S^* une solution optimale pour l'objectif de date de retour de dépôt d'une instance du TP pour une date de départ. Alors S^* est optimal pour l'objectif de date de retour du dépôt pour n'importe quelle date de départ.*

Proposition 3.3.2. *Si toutes les solutions non-dominées d'une instance du TP avec pour date de départ α n'ont aucune tâche livrée à l'heure, alors elles seront toujours des solutions non-dominées pour toute date de départ postérieure à α .*

Les constructions heuristiques ont une complexité en $O(|B|^2)$. Quant aux solutions non-dominées déjà connues, leur ré-évaluation pour une nouvelle date de départ est en $O(|B|)$.

À chaque nœud de l'arbre une borne inférieure bi-objectif est calculée. Nous calculons une borne inférieure bi-objectif $Lb_{TP}(\alpha, B)$ du TP pour le lot B avec une date de départ du dépôt α par la méthode suivante :

1. On pose le graphe G composé des clients du trajet B qui n'ont pas encore été livrés et du dernier client livré. Si aucun client n'a encore été livré alors le dernier client livré est remplacé par le dépôt. Les sommets sont les clients et les poids des arêtes sont les temps de trajet $tt_{j,j'}$.
2. On calcule un arbre couvrant de poids minimum MST_G du graphe G . Dans la figure 3.4, le trajet est composé de cinq clients et un arbre couvrant de poids minimum est représenté.
3. On trie les clients par date de livraison souhaitée croissante. Dans la figure 3.4, les clients sont ordonnés sur le diagramme de Gant par dates de livraison souhaitées croissantes. Alors pour tout couple de clients (j, j') si $d_j \leq d_{j'}$ alors $D_j \leq D_{j'}$.
4. On trie les d'arêtes du MST_G par temps de trajet croissant. Dans la figure 3.4, $tt_{1,2} \leq tt_{2,3} \leq tt_{0,1} \leq tt_{0,4} \leq tt_{0,5}$.
5. On calcule les dates de livraison en additionnant les arêtes triées. Dans la figure 3.4, le client 1 est livré à $D_1 = \alpha + tt_{1,2}$, le client 2 est livré à $D_2 = D_1 + tt_{2,3}$, le client 3 est livré à $D_3 = D_2 + \leq tt_{0,1}$, le client 4 est livré à $D_4 = D_3 + tt_{0,4}$ et finalement le client 5 est livré à $D_5 = D_4 + tt_{0,5}$.
6. La solution construite est non réalisable puisque les arêtes utilisées ne correspondent pas aux destinations. Cependant, il est possible d'obtenir une borne inférieure de

3.3. MÉTHODE DE RÉOLUTION EXACTE : 3BB

la somme des retards $Lb_{TP}^a(\alpha, B)$ en calculant le retard pour chaque client avec les dates de livraison calculées précédemment. Dans la figure 3.4, seuls les clients 3 et 5 ont du retard. La somme des retards est alors égale à $T_3 + T_5$.

7. On calcule une borne inférieure de la date de retour du trajet $Lb_{TP}^b(\alpha, B)$ en ajoutant le temps de trajet le plus petit pour revenir au dépôt à la dernière date de livraison. Dans la figure 3.4, $Lb_{TP}^b(\alpha, B) = D_5 + tt_{0,3}$.
8. $Lb_{TP}(\alpha, B)$ définit par le couple $(Lb_{TP}^a(\alpha, B), Lb_{TP}^b(\alpha, B))$.

La figure 3.4 illustre la construction et le calcul des deux bornes inférieures $Lb_{TP}^a(\alpha, B)$ et $Lb_{TP}^b(\alpha, B)$.

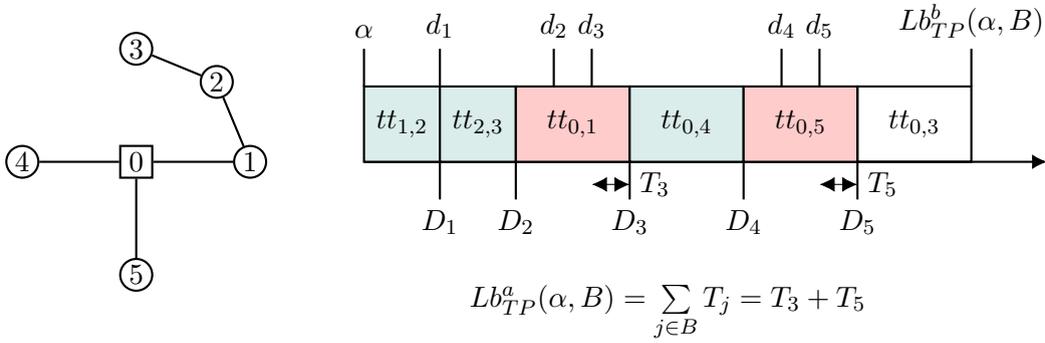


FIGURE 3.4 – Exemple du calcul de $Lb_{TP}^a(\alpha, B)$ et $Lb_{TP}^b(\alpha, B)$: Le graphe représente un arbre couvrant de poids minimal d'un graphe composé de cinq clients et le dépôt. Le sommet 0 représente le dépôt. Le diagramme de Gantt montre les temps de trajet utilisés entre chaque livraison.

La preuve que $Lb_{TP}^b(\alpha, B)$ est une borne inférieure de la date de retour au dépôt est triviale : le poids d'un arbre couvrant de poids minimum est inférieur ou égal au poids de n'importe quel chemin hamiltonien et nous ajoutons en dernier l'arête connectée au dépôt de plus petit poids.

La preuve que $Lb_{TP}^a(\alpha, B)$ est une borne inférieure de la somme des retards est plus compliquée. Le principe de cette borne inférieure s'inspire de la borne inférieure pour le problème d'ordonnancement $1|r_j|\sum T_j$ présenté dans [Chu, 1992]. L'auteur démontre le lemme 1 par récurrence afin de prouver sa borne inférieure.

Lemme 1. Soient deux listes de dates de livraison (D_1, D_2, \dots, D_n) et $(D'_1, D'_2, \dots, D'_n)$, de telle sorte que $D'_1 \leq D'_2 \leq \dots \leq D'_n$ et $\forall k, D'_k \leq D_k$. On définit la liste de dates de fin souhaitées (d_1, d_2, \dots, d_n) et sa version triée par valeurs croissantes $(d'_1, d'_2, \dots, d'_n)$. Alors

l'équation suivante est vraie :

$$\sum_{k=1}^n \max(D'_k - d'_k, 0) \leq \sum_{k=1}^n \max(D_k - d_k, 0) \quad (3.22)$$

En utilisant le lemme 1, nous pouvons démontrer de la même manière que $Lb_{TP}^a(\alpha, B)$ est bien une borne inférieure de la somme des retards du TP. Cependant, pour cela il est nécessaire de démontrer que la suite ordonnée des dates de livraison construite par notre borne inférieure $(D_1^{LB}, D_2^{LB}, \dots, D_n^{LB})$ et la suite ordonnée des dates de livraison $(D_1^{\mathcal{P}}, D_2^{\mathcal{P}}, \dots, D_n^{\mathcal{P}})$ d'un chemin quelconque \mathcal{P} respectent la propriété $\forall k, D_k^{LB} \leq D_k^{\mathcal{P}}$. Ceci peut être résumé par le théorème 1

Théorème 1. *Tout chemin \mathcal{P} de k arêtes d'un graphe pondéré G a un poids total supérieur ou égal à la somme des k plus petits poids des arêtes d'un arbre couvrant de poids minimum de G .*

Démonstration. Soit \mathcal{P} un chemin d'un graphe pondéré G et T^* un arbre couvrant de poids minimum de G .

On souhaite démontrer qu'il existe une bijection g de \mathcal{P} dans une partie T^* telle que le poids de toute arête $a \in \mathcal{P}$ est supérieur ou égal au poids de son image dans T^* par g .

Posons $E = \mathcal{P} \setminus T^*$ l'ensemble des arêtes du chemin \mathcal{P} non contenues dans T^* et notons $e_1, e_2, \dots, e_{|E|}$ ses éléments. Pour toute arête $e_h \in |E|$, le graphe $T^* \cup \{e_h\}$ contient un unique cycle C_h dont la seule arête non contenue dans T^* est e_h . D'après la propriété de cycle des arbres couvrants de poids minimum², on peut conclure que le poids de chacune des arêtes de C_h , donc en particulier de $C_h \setminus \mathcal{P}$, est inférieur ou égal au poids de l'arête e_h . De plus, comme \mathcal{P} est un chemin, il ne peut contenir de cycle et donc $C_h \setminus \mathcal{P} \neq \emptyset$.

Montrons par récurrence qu'il est possible de définir $g(e_1), g(e_2), \dots, g(e_{|E|})$ images distinctes des arêtes $e_1, e_2, \dots, e_{|E|}$, telles que pour toute arête $e_h \in E$ l'arête $g(e_h)$ appartient à $C_h \setminus \{e_h\}$. On suppose, à l'étape n , avoir construit des images distinctes $g(e_1), g(e_2), \dots, g(e_n)$ des arêtes e_1, e_2, \dots, e_n de E par g , telles que $\forall e_h \in E, g(e_h) \in C_h \setminus \{e_h\}$.

Montrons qu'à l'arête e_{n+1} on peut associer une arête $g(e_{n+1}) \in C_{n+1} \setminus \{e_{n+1}\}$, telle que $g(e_{n+1}) \notin \bigcup_{h \in [1, n]} C_h \setminus \{e_h\}$. Supposons en effet que toute les arêtes de $C_{n+1} \setminus \{e_{n+1}\}$ sont contenues dans $\bigcup_{h \in [1, n]} C_h \setminus \{e_h\}$. Soit le chemin \mathcal{P}' obtenu en remplaçant dans \mathcal{P}

2.

Propriété de cycle des arbres couvrant de poids minimum (voir [Gondran et al., 1984]). *Pour tout cycle C d'un graphe pondéré G , l'arête de plus grand poids de C n'appartient pas aux arbres couvrants de poids minimum de G .*

chaque arête $e_h \in E, h \leq n$ par le chemin $C_h \setminus \{e_h\}$. \mathcal{P}' qui est un chemin par construction contient alors le cycle C_{n+1} ce qui est contradictoire.

Posons alors $g(e_1)$ comme une des arêtes de $C_1 \setminus \mathcal{P}$. Puis à chaque arête $e_{n+1} \in E$ associons une arête $g(e_{n+1}) \notin \bigcup_{h \in [1, n]} C_h \setminus \{e_h\}$. On obtient par récurrence une injection g de E dans $T^* \setminus \mathcal{P}$, telle que chaque arête de E a un poids supérieur ou égal à celui de son image.

En prolongeant cette injection par l'identité sur $\mathcal{P} \cap T^*$ on obtient une bijection entre \mathcal{P} et une partie de T^* , vérifiant encore la propriété sur les poids. Alors la somme des poids des arêtes de \mathcal{P} est supérieure ou égale à la somme des poids des arêtes de $g(\mathcal{P})$ qui est une partie de T^* , et donc supérieure ou égale à la somme des poids des $|\mathcal{P}|$ arêtes de T^* de poids minimaux. \square

La complexité de calcul de ces bornes inférieures est de $O(|B|)^2$ car la complexité de calcul d'un arbre couvrant minimum est également de $O(|B|)^2$.

3.3.2 Optimisation de tournée d'un véhicule : MTP

Séparation et évaluation de fronts de Pareto par ϵ -contrainte : $B\&B_{MTP}$

Pour résoudre le MTP, on propose un algorithme de séparation et d'évaluation combiné à une méthode par ϵ -contrainte que l'on nomme $B\&B_{MTP}$.

Soit \mathcal{B}_v la liste des trajets à effectuer par le véhicule v . On cherche à affecter à chaque trajet $B \in \mathcal{B}_v$ une séquence de livraison non-dominée. La stratégie de branchement est représentée sur la figure 3.5 : le branchement au niveau t se fait sur les solutions non-dominées de l'instance TP définie par le t -ième trajet effectué par le véhicule v , $B_{v,t}$, avec l'heure de départ égale au maximum entre la date $\widehat{r}_{B_{v,t}}$ et la date $\omega_{B_{v,t-1}}$. Avant chaque branchement, le TP correspondant est résolu en utilisant une méthode par ϵ -contrainte présentée par l'algorithme 5 où $B\&B_{TP}(B, \epsilon)$ résout l'instance du TP pour le trajet B avec comme contrainte que la date de retour soit inférieure à ϵ et $S.\omega$ représente la date de retour au dépôt de la solution S . Si le trajet $B_{v,t}$ est le dernier effectué par le véhicule alors seule la solution qui minimise le retard total nous intéresse sinon Lb_ω est définie comme la date de départ au plus tôt $\widehat{r}_{B_{v,t+1}}$ du trajet suivant $B_{v,t+1}$.

La méthode par ϵ -contrainte retourne l'ensemble des solutions non-dominées, à l'exception des solutions dites sans intérêt (Définition 3.3.1). $B\&B_{MTP}$ branche alors sur chacune de ces solutions et produit un nœud fils.

Algorithm 5 Méthode ϵ -contrainte

 Paramètres : Trajet B et la borne inférieure de la date de retour au dépôt Lb_ω

 Ensemble des solutions non-dominés \mathcal{S}_B^*
 $\epsilon \leftarrow +\infty$
 $\mathcal{S}_B^* \leftarrow \{\}$
 $feasible \leftarrow True$
while $feasible$ and $\epsilon \geq Lb_\omega$ **do**
 $S \leftarrow B\&B_{TP}(B, \epsilon)$
 $\mathcal{S}_B^* \leftarrow \mathcal{S}_B^* \cup \{S\}$
 $\epsilon \leftarrow S.\omega$
end while
return \mathcal{S}_B^*

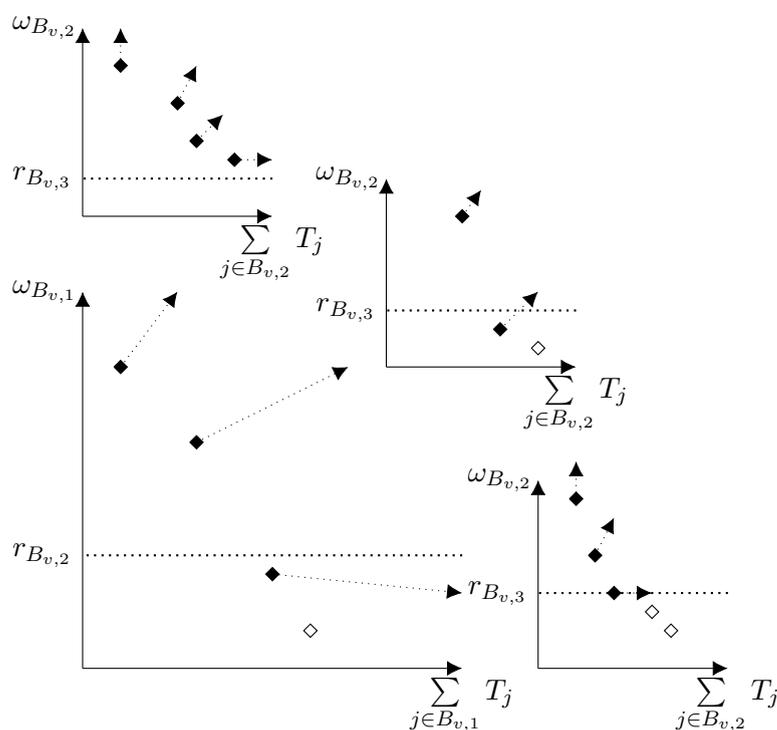


FIGURE 3.5 – Stratégie de branchement du Branch and Bound $B\&B_{MTP}$: chaque losange est une solution non dominée du TP correspondant et les losanges blancs sont des solutions sans intérêt

Definition 3.3.1. Une solution sans intérêt d'une instance du TP est une solution non-dominée qui n'est pas renvoyée par la méthode ϵ -contrainte présentée par l'algorithme 5 en raison du critère d'arrêt $\epsilon < Lb_\omega$.

Pour chaque niveau t , sauf le dernier, Lb_ω est défini comme $\hat{r}_{B_{v,t+1}}$, la date de départ au plus tôt du prochain trajet $B_{v,t+1}$.

Cette stratégie de branchement pourrait conduire à un grand nombre d'utilisations de $B\&B_{TP}$: $O(P^{|T|-1})$ où P représente une borne supérieure du nombre de solutions non-dominées par TP. Comme $B\&B_{TP}$ est un algorithme exponentiel, afin de réduire le nombre d'utilisations redondantes, nous proposons d'utiliser une mémoire externe pour enregistrer toutes les solutions non-dominées calculées des TP et les utiliser comme heuristiques d'entrées comme évoqué en section 3.3.1 pour le calcul de la borne supérieure du TP.

Remarque 1. *La méthode ϵ -contrainte n'a pas besoin d'être entièrement exécutée pour démarrer le branchement sur les solutions non dominées.*

Nous utilisons la méthode ϵ -contrainte de manière récursive pour obtenir une stratégie d'exploration en profondeur pour le $B\&B_{MTP}$.

Borne supérieure et borne inférieure

Une borne supérieure est calculée pour chaque nœud. Celle-ci est une solution réalisable construite en utilisant les séquences de livraison déjà sélectionnées puis successivement la borne supérieure du TP présentée dans la section 3.3.1 pour les trajets n'ayant pas encore de séquence de livraison de sélectionnée.

De la même manière, une borne inférieure $Lb_{MTP}(\mathcal{B}_v)$ est calculée comme la somme du retard associé aux trajets dont la séquence de livraison a déjà été sélectionnée et des bornes inférieures du TP où la date de départ de dépôt du trajet $B_{v,t}$ est fixée comme suit pour le trajet qui reste à définir :

$$\alpha_{v,k} = \begin{cases} \max\{\hat{r}_{B_{v,t}}, \omega_{B_{v,t-1}}\} & \text{si la livraison } B_{v,t-1} \text{ existe} \\ \max\{\hat{r}_{B_{v,t}}, Lb_{TP}^a(\alpha_{v,t-1}, B_{v,t-1})\} & \text{sinon} \end{cases} \quad (3.23)$$

Lors de l'exploration de l'arbre, il est possible d'obtenir un nœud dominé par un autre nœud .

Proposition 3.3.3. *Soit S_t et S'_t deux solutions partielles d'une même instance du MTP avec les solutions non-dominées des $t - 1$ premiers trajets déjà sélectionnées. Si la borne inférieure du retard total associée à S_t est inférieure ou égale à la borne inférieure du retard total associée à S'_t et la date de retour au dépôt du $t - 1$ ème trajet de S_t est inférieure à celle de S'_t où antérieure à la date de fin de production $\hat{r}_{B_{v,t}}$, alors S_t domine S'_t .*

La proposition 3.3.3 est très utile lorsque deux trajets consécutifs ont des dates de départ au plus tôt très écartées.

3.3.3 Optimisation des tournées des véhicules : MT-BDP

Nous proposons un algorithme de séparation et d'évaluation noté $B\&B_{MT-BDP}$ pour affecter chaque lot à un véhicule. La procédure de branchement proposée s'appuie sur le fait que tous les véhicules sont identiques et permet d'éliminer de nombreuses symétries.

Séparation et évaluation avec ruptures de symétries $B\&B_{MT-BDP}$

La stratégie de branchement consiste à affecter un véhicule à chaque trajet dans l'ordre des dates de départ au plus tôt. Chaque niveau de branchement t de l'arbre représente l'affectation du t ième trajet à un véhicule. Un trajet ne peut être affecté qu'aux véhicules déjà en charge (ayant des trajets d'affectés) ou au véhicule sans affectation de plus petit index.

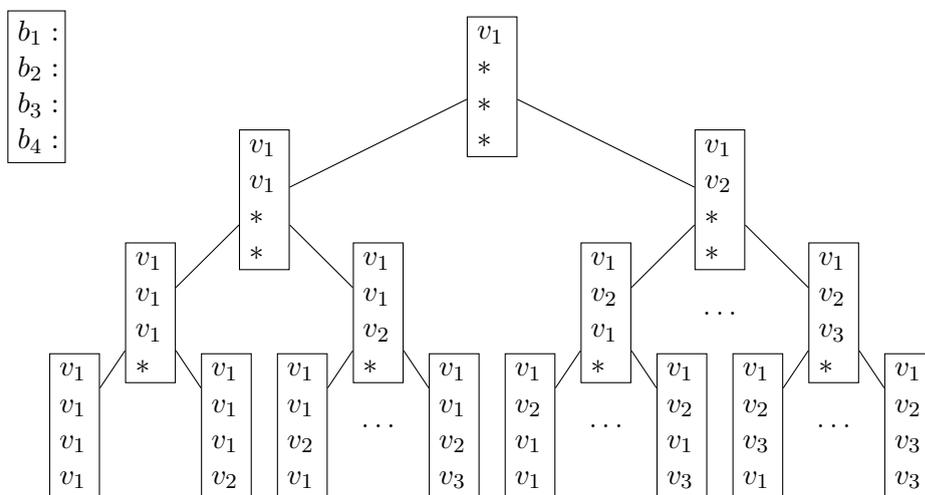


FIGURE 3.6 – Méthode de branchement du $B\&B_{MT-BDP}$ pour une instance à quatre trajets et trois véhicules

La figure 3.6 représente l'arbre pour trois véhicules et quatre trajets. Dans ce problème, les véhicules sont identiques et disponibles initialement à la même date. Pour éviter les symétries, l'affectation d'un trajet au l ème véhicule n'est possible que si au moins un des trajets précédents est affecté au $l - 1$ ème véhicule.

Cette stratégie de branchement réduit la taille de l'arbre par rapport à une stratégie classique qui consisterait à lister toutes les combinaisons possibles (trajet, véhicule).

Théorème 2. Soit \mathcal{A} l'arbre parcourut lors du $B\&B_{MT-BDP}$. Si $|\mathcal{B}| > |V| \geq 3$ alors le nombre de nœuds dans l'arbre par \mathcal{A} est de $O(|V|^{|\mathcal{B}|-2})$.

Démonstration. On note $N_{\mathcal{A}}$ et $N_{\mathcal{A}}(t)$ respectivement la taille de l'arbre \mathcal{A} et le nombre

de nœuds de profondeur t dans l'arbre \mathcal{A} .

On suppose $|V| = 3$, $\{\mathcal{N}_{\mathcal{A}}(t)\}_{t \geq 1}$ est une suite arithmético-géométrique de relation de récurrence :

$$\mathcal{N}_{\mathcal{A}}(t+1) = 3 \cdot (\mathcal{N}_{\mathcal{A}}(t) - 1) + 2 \quad (3.24)$$

La formule (3.24) provient du fait que tous les nœuds ayant au moins deux véhicules d'affectés à des lots engendrent trois nœuds fils. Les nœuds qui n'ont qu'un seul véhicule d'affecté n'engendrent que deux nœuds fils. Pour chaque profondeur t de l'arbre il existe un unique nœud avec un seul véhicule affecté. D'après la formule (3.24) de récurrence³ on peut déduire la formule suivante :

$$\mathcal{N}_{\mathcal{A}}(t) = 3^{t-1} \cdot \mathcal{N}_{\mathcal{A}}(1) + \frac{1 - 3^{t-1}}{2} \quad (3.25)$$

La taille de l'arbre est alors la somme des termes d'une suite arithmético-géométrique avec $\mathcal{N}_{\mathcal{A}}(1) = 1$ ce qui donne la formule suivante :

$$\mathcal{N}_{\mathcal{A}} = \sum_{t=1}^{|\mathcal{B}|} \mathcal{N}_{\mathcal{A}}(t) = \frac{3^{|\mathcal{B}|} + 2 \cdot |\mathcal{B}| - 1}{4} \quad (3.26)$$

On suppose maintenant $|V| > 3$. La formule (3.25) devient une majoration du nombre de nœud au niveau t puisque à chaque niveau t il y a toujours un unique nœud qui engendre deux nœuds fils et tous les autres nœuds engendrent au plus $|V|$ nœuds fils. Nous obtenons donc :

$$\mathcal{N}_{\mathcal{A}}(t) \leq |V|^{t-1} \cdot (\mathcal{N}_{\mathcal{A}}(1) - 1) + 2 \quad (3.27)$$

$$\begin{aligned} \mathcal{N}_{\mathcal{A}} &\leq \left(\mathcal{N}_{\mathcal{A}}(1) - \frac{2 - |V|}{1 - |V|} \right) \cdot \frac{1 - |V|^{|\mathcal{B}|}}{1 - |V|} + |\mathcal{B}| \cdot \frac{2 - |V|}{1 - |V|} \\ \mathcal{N}_{\mathcal{A}} &\leq \frac{|V|^{|\mathcal{B}|} + |\mathcal{B}| \cdot |V|^2 - 1 + |\mathcal{B}| \cdot (|V| - 2) \cdot (1 - |V|)}{(1 - |V|)^2} \\ \mathcal{N}_{\mathcal{A}} &\leq \frac{|V|^{|\mathcal{B}|} + |\mathcal{B}| \cdot |V|^2 - 3 \cdot |\mathcal{B}| \cdot |V| + 2 \cdot |\mathcal{B}| - 1}{(1 - |V|)^2} \end{aligned} \quad (3.28)$$

On en conclu que si $|V| \geq 3$ alors $\mathcal{N}_{\mathcal{A}} = O(|V|^{|\mathcal{B}|-2})$. □

3. On pose la suite géométrique translaté $w_t = \mathcal{N}_{\mathcal{A}}(t-1) - \frac{1}{2}$ de raison 3 alors $\mathcal{N}_{\mathcal{A}}(t-1) = 3^{t-1} \cdot w_0 + \frac{1}{2}$. Voir [Ramis et Warusfel,]

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

Si le nombre de véhicules $|V|$ est supérieur ou égal au nombre de lots $|\mathcal{B}|$, l'affectation optimale est triviale : affecter chaque lot à un véhicule différent.

L'évaluation exacte d'un nœud nécessite la résolution de plusieurs instances du MTP : une pour chaque véhicule avec au moins un lot affecté. Même si chaque évaluation utilise la même mémoire externe, il serait trop coûteux en temps de calcul de l'appliquer pour chaque nœud. Nous suggérons d'évaluer uniquement les nœuds feuille. Il est aussi envisageable dans la situation où les dates de fin de productions de deux lots consécutifs sont très écartés d'évaluer un nœud intermédiaire afin de réduire des dominations entre des nœuds. Cependant nous ne l'avons pas encore implémenté car cela nécessite de mettre en place des critères pour déterminer quand l'écart est suffisamment important pour que la résolution de MTP partiels soit rentable. La stratégie d'exploration priorise le nœud avec la meilleure borne supérieure en premier. En cas d'égalité, le nœud avec la plus petite borne inférieure est exploré.

Borne supérieure et borne inférieure

La borne supérieure d'un nœud est évaluée à l'aide de l'heuristique suivante :

1. Pour chaque trajet affecté à un véhicule la borne supérieure du MTP est utilisée.
2. Pour les trajets encore non affectés, ils sont impérativement affectés au véhicule disponible au plus tôt et sa séquence de livraison est construite grâce à la borne supérieure du TP.

De la même manière une borne inférieure est calculée pour chaque nœud :

1. Pour chaque véhicule la borne inférieure du MTP est utilisée.
2. Pour les trajets encore non affectés, on utilise la borne inférieure du TP avec comme heure de départ sa date de départ au plus tôt.

Comme les bornes supérieures du MT-BDP et du MTP utilisent celles du TP qui sélectionne la meilleure solution entre des constructions heuristiques et des solutions non-dominées pour différentes dates de départ gardées en mémoire, les qualités de celles-ci s'améliorent grandement au fur et à mesure du déroulement de la méthode exacte.

3.4 Expérimentations numériques

3.4.1 Génération d'instances de la littérature

Pour évaluer notre méthode exacte nous avons décidé d'adapter des instances de la littérature. [Solomon, 1987] a proposé un benchmark d'instances pour le problème de tournées

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

de véhicules avec fenêtres temporelles. Les instances sont caractérisées par deux critères : la répartition géographique et la largeur des fenêtres temporelles. Trois types de répartitions géographiques des clients ont été proposés :

1. Une répartition "Groupé" (C) où les clients sont concentrés autour de quelques centres névralgiques.

Cette répartition est celle qui représente le mieux le contexte de l'UBCO car chaque hôpital correspond à un groupe d'unité d'oncologies.

2. Une répartition "Aléatoire" (R) où les clients sont aléatoirement répartis autour du dépôt.

3. Une répartition "Mixte" (RC) où deux parties sont considérées : une première partie où les clients sont regroupés et une deuxième où les clients sont répartis aléatoirement autour du dépôt.

Cette répartition est celle qui représente le mieux les cas réels de livraison à des particuliers car une partie des clients est regroupée dans les grandes villes.

Les fenêtres temporelles sont soit :

1. Des fenêtres de temps serrées (type 1).
2. Des fenêtres de temps larges (type 2).

Dans [Azi *et al.*, 2010], les auteurs ont adapté les instances de Solomon avec fenêtres temporelles larges (type 2) de 25 et 40 clients pour le problème de tournées de véhicules multi-trajet avec fenêtres temporelles (MT-VRP-TW) en définissant un nombre de véhicules égal à 2. Les instances à fenêtres temporelles serrées ont été écartées car elles nécessitent un large nombre de véhicules et de très courtes tournées pour être réalisables. [Hernandez *et al.*, 2014] et [Macedo *et al.*, 2011] réutilisent ces mêmes instances pour évaluer leurs méthodes exactes.

En 2016, [Cattaruzza *et al.*, 2016] reprennent les instances de Solomon mais ne considèrent plus la date de fin des fenêtres temporelles comme des dates d'échéances et y ajoutent des dates de début de livraison au plus tôt. Pour générer ces dernières, les auteurs les identifient à l'arrivée de camions à une plateforme de logistique. Ils suggèrent alors une génération par lots des dates de début de livraison au plus tôt. Les instances étudiées sont toujours uniquement celles avec des fenêtres temporelles larges mais les nombres de clients considérés sont de 25, 50 et 100. Le nombre de véhicules utilisé est alors de 2 véhicules pour 25 clients pour rester cohérent avec les instances utilisées dans [Azi *et al.*, 2010, Macedo *et al.*, 2011, Hernandez *et al.*, 2014].

Nous avons décidé de reprendre les instances de [Cattaruzza *et al.*, 2016] en assimilant les dates d'échéances à des dates de livraison souhaitées d_j et de générer nos propres dates

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

de début de livraison au plus tôt \widehat{r}_j car les instances de [Cattaruzza *et al.*, 2016] ont pour but d'être réalisables sans retard. Nous proposons deux générations différentes des dates de début de livraison au plus tôt :

1. "cst" où $\forall j \in J, d_j - \widehat{r}_j = \lceil \max_{j' \in J} \{tt_{0,j'}\} \rceil$

Cet écart constant, est égal à la plus grande $tt_{0,j}$ et permet d'assurer qu'avec un nombre suffisant de véhicules, il est possible de livrer sans retard. Cette méthode de génération est celle qui correspond le mieux au cas de l'UBCO où les dates de livraison souhaitées sont soit 1h soit 1h30 après la date de début de production au plus tôt.

2. "rnd" où l'écart entre les dates de fin de production et les dates de livraison souhaitées est aléatoire : $\forall j \in J, \widehat{r}_j \geq [0, d_j]$.

Pour la constitution des trajets nous avons décidé de regrouper les tâches en fonction des dates de début de livraison aux plus tôt. L'algorithme de partitionnement utilisé est l'Agglomerative Hierarchical Clustering (Algo 1) avec une fonction de distance basée sur l'écart des dates de livraison au plus tôt.

$$dist_{release}(j, j') = |\widehat{r}_j - \widehat{r}_{j'}| \quad (3.29)$$

Pour définir le nombre de trajets à constituer nous ajoutons trois contraintes :

— Deux trajets B et B' sont dit compatibles à un regroupement uniquement si

$$\max_{j \in B \cup B'} \{\widehat{r}_j\} \leq \min_{j \in B \cup B'} \{d_j\}.$$

— Le plus grand trajet, noté B_{max} , est limité à 18 clients.

— Le nombre minimum de trajets est de $\lfloor \frac{|J|}{5} \rfloor$.

Pour évaluer les performances de notre méthode exacte dans les cas limites nous avons décidé de résoudre chacune des instances résultantes avec le nombre de véhicules défini par [Cattaruzza *et al.*, 2016] et la moitié de ce nombre.

Les caractéristiques principales sont résumées dans le tableau 3.3 : la répartition géographique, le nombre de clients $|J|$, l'horizon de temps des dates de livraison souhaitées, le type de génération des dates de fin de production, le nombre de trajets $|\mathcal{B}|$ et la taille maximale des trajets $|B_{max}|$.

Pour un visuel plus clair, la figure 3.7 montre des boîtes à moustaches représentant la distribution du nombre de trajets et de la taille maximale des trajets des différents sous-ensembles d'instances.

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

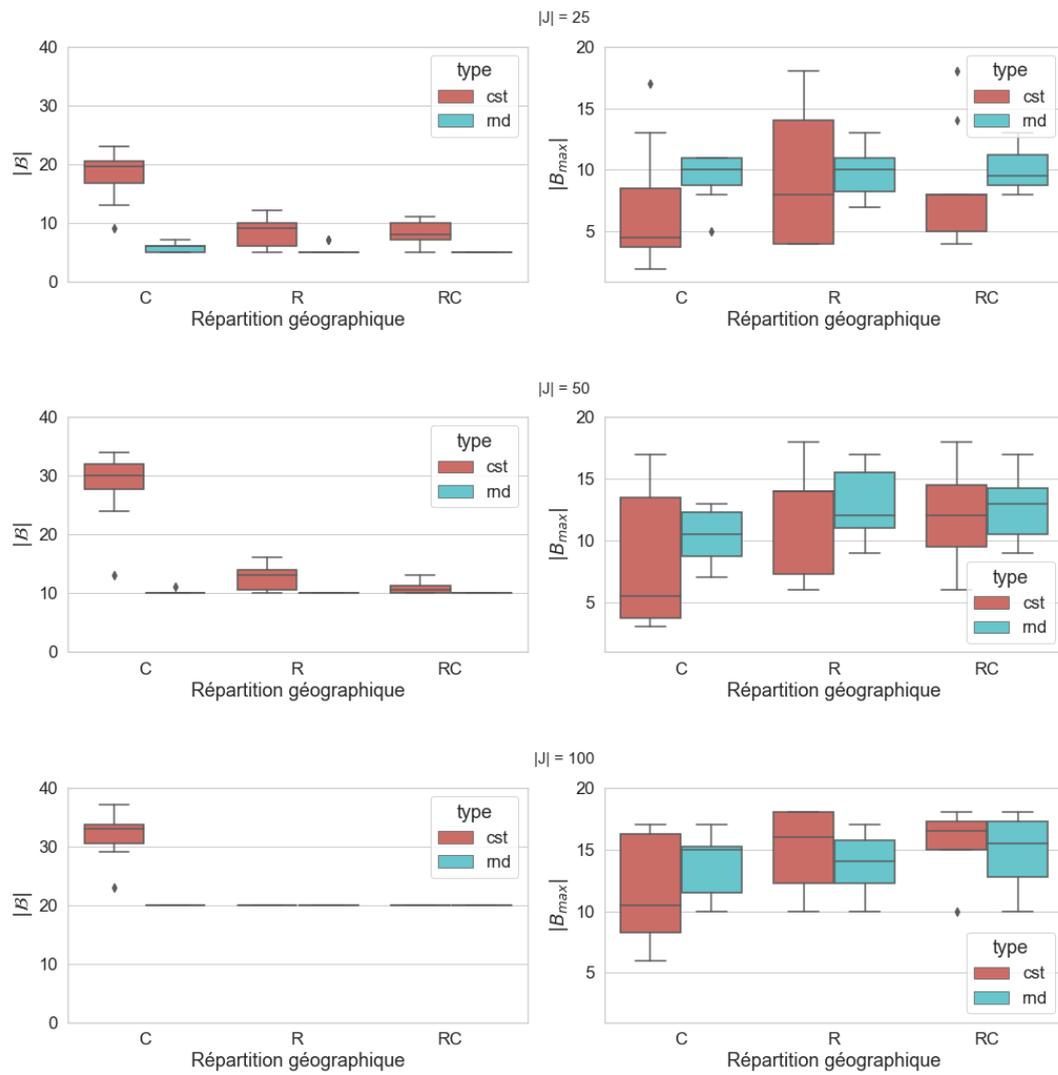


FIGURE 3.7 – Taille maximale des trajets $|B_{max}|$ en fonction de la répartition géographique des clients.

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

Répartition Géographique	Horizon de temps	Nom de l'instance	type $d_j - \hat{r}_j$	25		50		100	
				$ \mathcal{B} $	$ B_{max} $	$ \mathcal{B} $	$ B_{max} $	$ \mathcal{B} $	$ B_{max} $
Groupé (C)	3390	C201	cst	23	3	32	3	33	6
			rnd	6	10	11	7	20	16
		C202	cst	18	7	29	13	33	16
			rnd	6	10	10	9	20	17
		C203	cst	13	13	24	15	33	17
			rnd	5	11	10	11	20	15
		C204	cst	9	17	13	17	23	17
			rnd	5	11	10	8	20	10
C205	cst	22	4	31	4	31	9		
	rnd	6	9	10	10	20	10		
C206	cst	20	2	32	3	36	6		
	rnd	5	8	10	12	20	15		
C207	cst	19	4	34	5	37	9		
	rnd	7	5	10	13	20	12		
C208	cst	20	5	29	6	29	12		
	rnd	6	11	10	13	20	15		
Aléatoire (R)	1000	R201	cst	12	4	16	6	20	10
			rnd	5	12	10	12	20	14
		R202	cst	10	8	14	14	20	17
			rnd	7	7	10	16	20	16
		R203	cst	6	14	13	14	20	18
			rnd	5	8	10	17	20	14
		R204	cst	5	18	10	18	20	18
			rnd	5	11	10	10	20	12
		R205	cst	11	4	15	8	20	15
			rnd	5	11	10	16	20	17
		R206	cst	9	8	14	14	20	18
rnd	7		9	10	14	20	13		
R207	cst	6	14	12	14	20	18		
	rnd	5	10	10	12	20	11		
R208	cst	5	18	10	18	20	16		
	rnd	5	9	10	9	20	10		
R209	cst	10	4	13	7	20	12		
	rnd	5	7	10	12	20	13		
R210	cst	9	5	12	7	20	13		
	rnd	5	10	10	11	20	16		
R211	cst	10	6	10	12	20	11		
	rnd	5	13	10	11	20	15		
Mixte (RC)	960	RC201	cst	11	4	13	6	20	15
			rnd	5	9	10	15	20	17
		RC202	cst	9	8	11	14	20	17
			rnd	5	11	10	9	20	14
		RC203	cst	6	14	10	16	20	18
			rnd	5	8	10	17	20	10
		RC204	cst	5	18	10	18	20	18
			rnd	5	10	10	9	20	12
RC205	cst	10	5	11	10	20	17		
	rnd	5	8	10	12	20	17		
RC206	cst	10	4	12	8	20	15		
	rnd	5	12	10	11	20	18		
RC207	cst	7	6	10	12	20	16		
	rnd	5	13	10	14	20	18		
RC208	cst	8	8	10	12	20	10		
	rnd	5	9	10	14	20	13		

TABLE 3.3 – Nombre de lots et taille maximale des lots pour chaque instance.

On peut apercevoir que le nombre de trajets pour les instances "rnd" est très inférieur à celui des instances "cst". Cela est dû à la contrainte de comptabilité de regroupement. La contrainte de taille maximale pour les trajets et celle d'un nombre minimum de trajets évitent de trop forts regroupements pour les instances "rnd". Le nombre de lots pour les instances "rnd" avec 50 ou 100 tâches est quasiment systématiquement égal à $\frac{|J|}{5}$.

Pour la résolution des instances nous avons décidé d'utiliser deux nombres de véhicules :

- $2 \cdot \frac{|J|}{25}$: deux véhicules pour 25 clients. Cela correspond au nombre utilisé par [Cattaruzza *et al.*, 2016].
- $1 \cdot \frac{|J|}{25}$: un véhicule pour 25 clients. Cela correspond à deux fois moins de véhicules que dans [Cattaruzza *et al.*, 2016].

3.4.2 Comparaison du modèle MILP et du 3BB

Afin de déterminer l'efficacité de notre méthode exacte 3BB nous la comparons à la résolution du modèle MILP présenté en section 3.2. Nous utilisons l'horizon de temps défini des instances comme valeur pour M dans les contraintes 3.6 et 3.9 du modèle MILP. Le modèle MILP est lancé avec comme première solution réalisable celle obtenue par le calcul de la borne supérieure du premier nœud du $B\&B_{MT-BDP}$.

Les expérimentations numériques ont été réalisées sur un ordinateur portable Dell équipé d'un Intel(R) Core™ i5-7440HQ CPU (2.80 GHz), 8 GB RAM, sur Windows 10. Et les heuristiques sont codées en C++ et exécutées avec un seul thread.

La figure 3.8 montre le pourcentage d'instances résolues en moins de 600 secondes par chacune des deux méthodes. Pour ce temps limite notre méthode montre sa supériorité dans la majorité des cas à l'exception des instances de type "rnd" avec 100 clients pour lesquelles le modèle MILP a de meilleurs résultats. En ce qui concerne les instances de type "cst" avec 100 clients, aucune n'a été résolue par le modèle MILP. La répartition géographique la plus facile à traiter est la répartition "Groupé" (C). En effet peu importe le nombre de clients, le 3BB résout quasiment toutes les instances dans le temps imparti. En revanche, la répartition "Mixte" (RC) est un défi pour le 3BB comme pour le modèle MILP.

Les résultats globaux sont très prometteurs. La figure 3.9 montre une disparité de la qualité de notre méthode en fonction du nombre de véhicules $|V|$. Malgré un modèle simplifié du modèle MILP pour un seul véhicule, le 3BB est largement supérieur. Le classement devient tout autre avec un nombre de véhicules plus important. Le modèle MILP résout une instance de plus que le 3BB avant le temps limite lorsque $|V| = 2$. Toujours avec $|V| = 2$, on peut remarquer que les instances résolues par le modèle MILP sont majoritairement résolues en moins de 200 secondes alors que la distribution des temps de calcul pour le

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

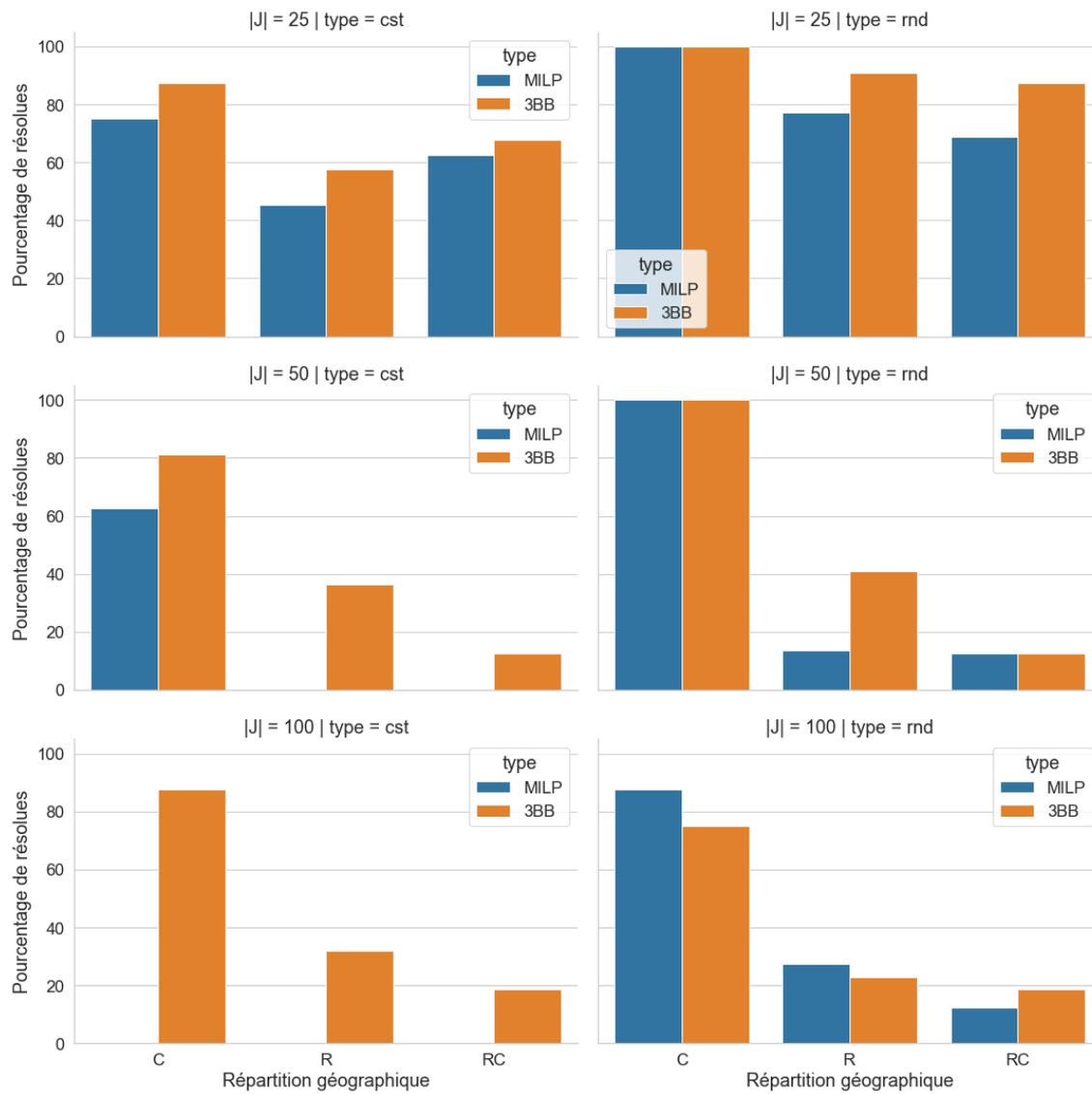


FIGURE 3.8 – Pourcentage d’instances résolues par le modèle MILP et le 3BB dans le temps de calcul imparti(600 secondes).

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

3BB est plus étendue, en particulier pour les instances de type "cst". Cela peut laisser supposer qu'avec un temps limite légèrement supérieur le 3BB pourrait résoudre quelques instances supplémentaires.

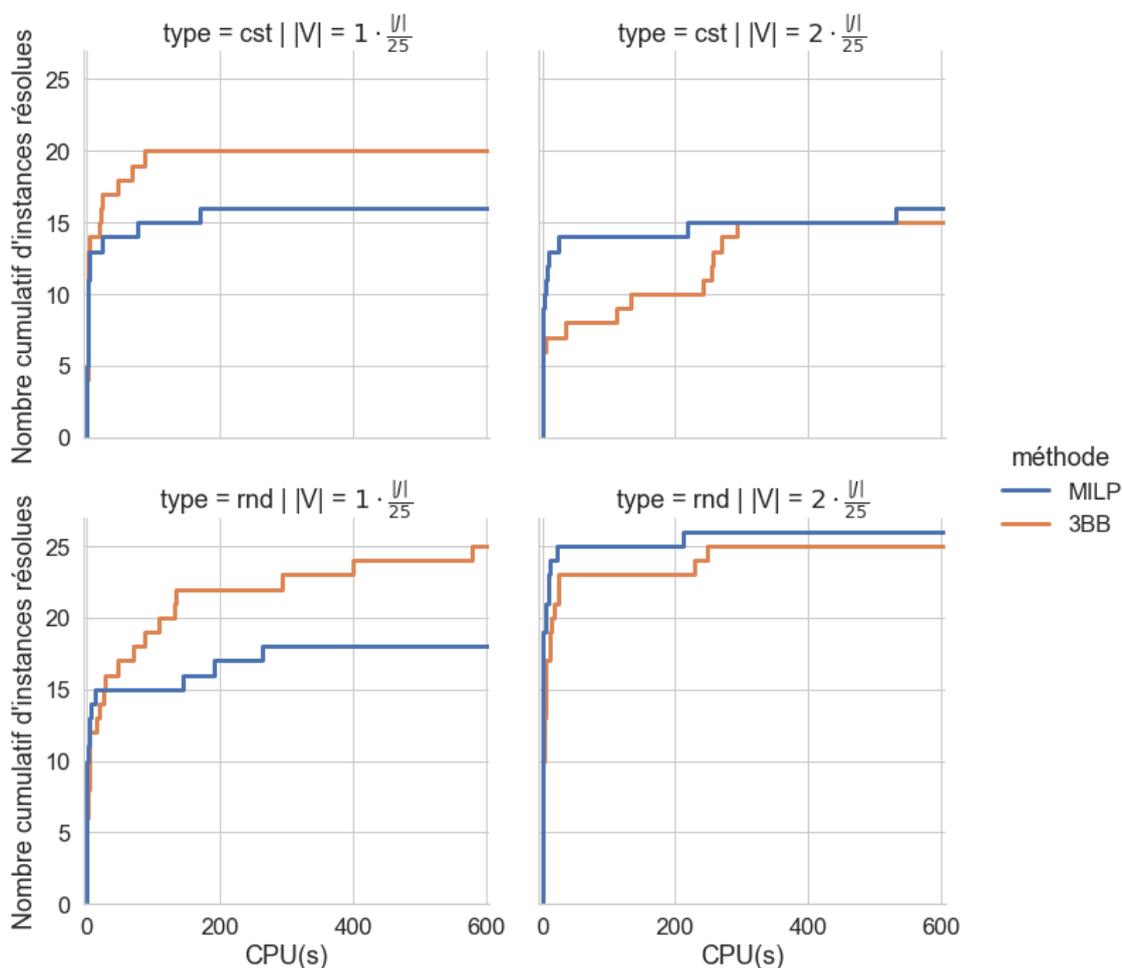


FIGURE 3.9 – Profil de performance pour les instances avec 25 clients

La figure 3.10 permet de confirmer que lorsque le nombre de véhicules disponibles est trop faible, la résolution se complexifie. Cela peut s'expliquer par un aspect plus conflictuel des deux objectifs du TP. Lorsque les quatre véhicules sont disponibles, le 3BB résout en moins de 200 secondes 17 instances de type "cst" et 16 de type "rnd" (environ 60%). Du côté du modèle MILP la différence entre les instances de type "cst" et "rnd" est beaucoup plus importante. Le nombre de trajets étant plus faible pour les instances "rnd" le nombre de variables et le nombre de contraintes du modèle s'en trouvent fortement réduits.

C'est avec la figure 3.11 que le 3BB montre à la fois sa force et sa faiblesse. En effet, les instances de type "cst" semblent ne pas poser de problème pour le 3BB mais sont

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

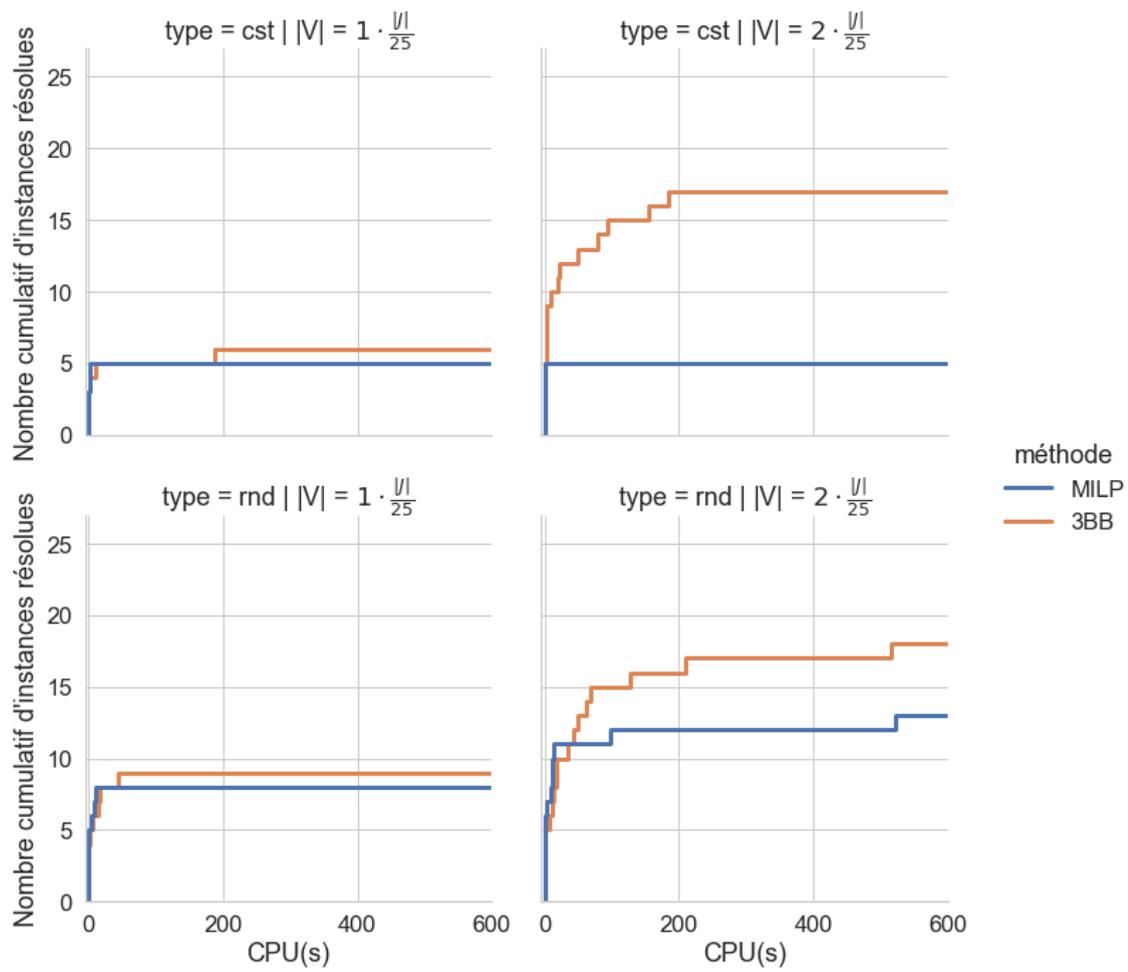


FIGURE 3.10 – Profile de performance pour les instances avec 50 tâches

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

impossible à résoudre pour le modèle MILP alors que le modèle MILP montre une légère supériorité sur les instances "rnd".

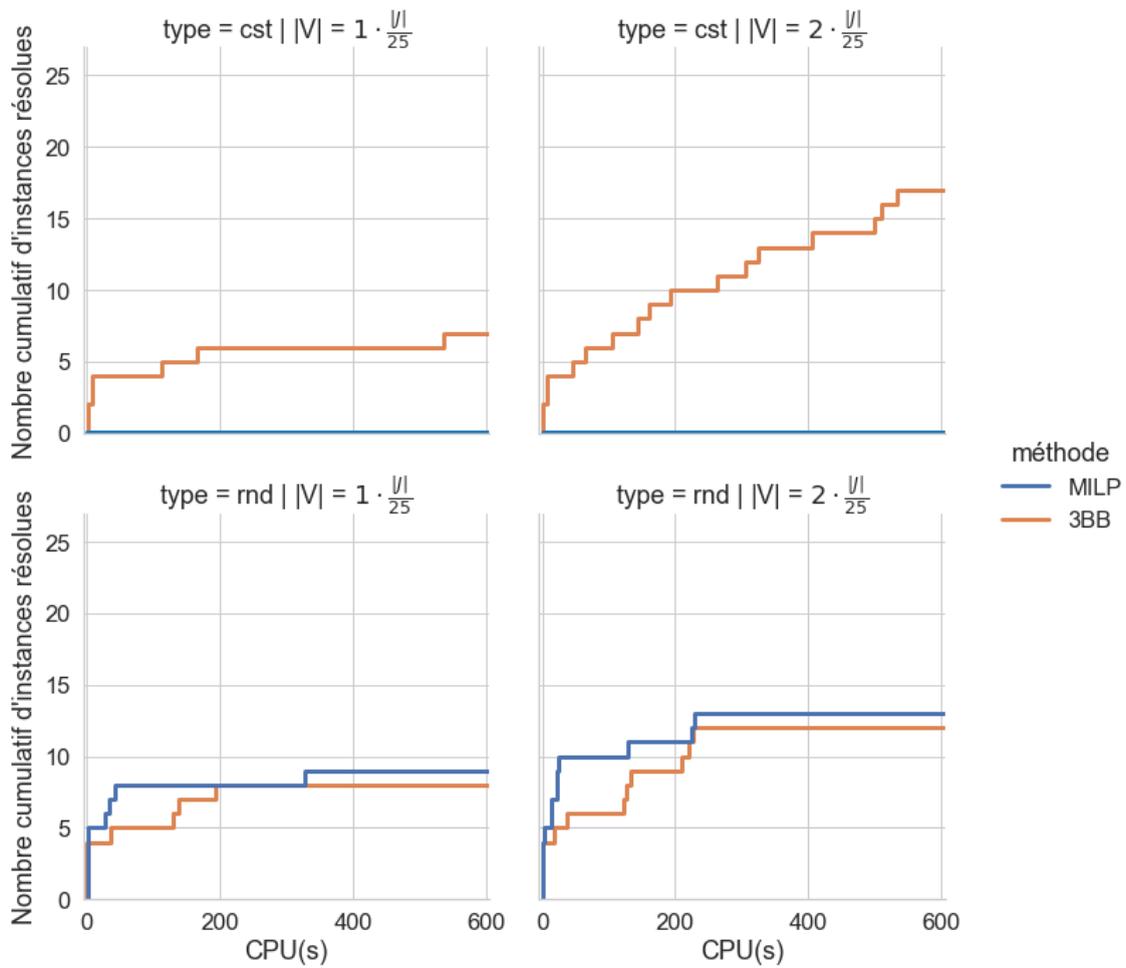


FIGURE 3.11 – Profile de performance pour les instances avec 100 clients

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

Lorsque le modèle MILP obtient de meilleurs résultats que le 3BB, seule une instance est résolue en plus. Il est alors logique de se demander si l'écart de temps de calculs sur les instances résolues plus rapidement par le modèle MILP que par le 3BB est important. La figure 3.12 représente l'écart de temps de calcul qu'il existe sur les instances résolues par les deux méthodes et plus rapidement par le modèle MILP que le 3BB. On constate qu'à l'exception des instances de type "cst" avec 25 clients et deux véhicules et les instances de type "rnd" avec 100 clients, le gain moyen de temps d'utiliser le modèle MILP par rapport au 3BB est inférieur à 30 secondes soit 5% du temps alloué.

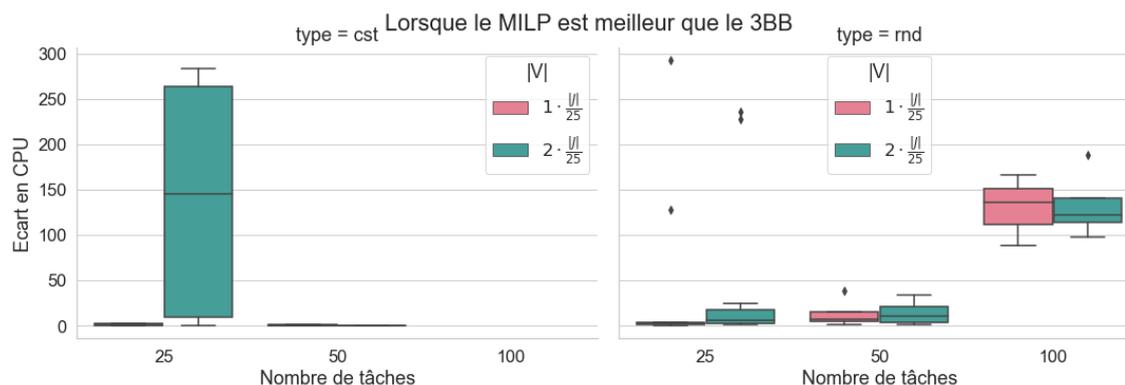


FIGURE 3.12 – Écart de temps de calcul sur les instances résolues plus rapidement par le modèle MILP que le 3BB.

Pour mieux comprendre pourquoi les différences de résultats sur les instances de type "cst" et "rnd" et en fonction du nombre de véhicules, la figure 3.13 représente le retard moyen de la meilleure solution trouvée de chaque instance. La première constatation est que les instances de type "rnd" ont des solutions avec nettement moins de retard que leurs homologues "cst". Cela se justifie par la contrainte de compatibilité de regroupement. Les instances de type "cst" ayant plus de trajets que celles de type "rnd", les véhicules effectuent beaucoup plus de retours au dépôt et la fenêtre de temps entre le départ et la date de livraison souhaitée est très souvent serrée. La supériorité occasionnelle du modèle MILP est alors sûrement liée au faible retard des solutions qui engendreraient beaucoup de symétries mieux détectées par le solveur que par le 3BB.

Grâce à la figure 3.14 qui montre le temps de calcul des instances résolues par le 3BB en fonction de la taille maximale des trajets, on peut noter que le temps de calcul est corrélé exponentiellement à la taille maximale des trajets.

Le 3BB a montré son efficacité et sa supériorité au MILP sur la plupart des instances et en particulier celles qui se rapprochent du cas de l'UBCO. En effet, les unités à livrer par l'UBCO sont "Groupées" géographiquement, l'écart entre la date de livraison au plus-tôt et la date de livraison souhaitée est serré et quasi constant (une heure ou une heure et

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

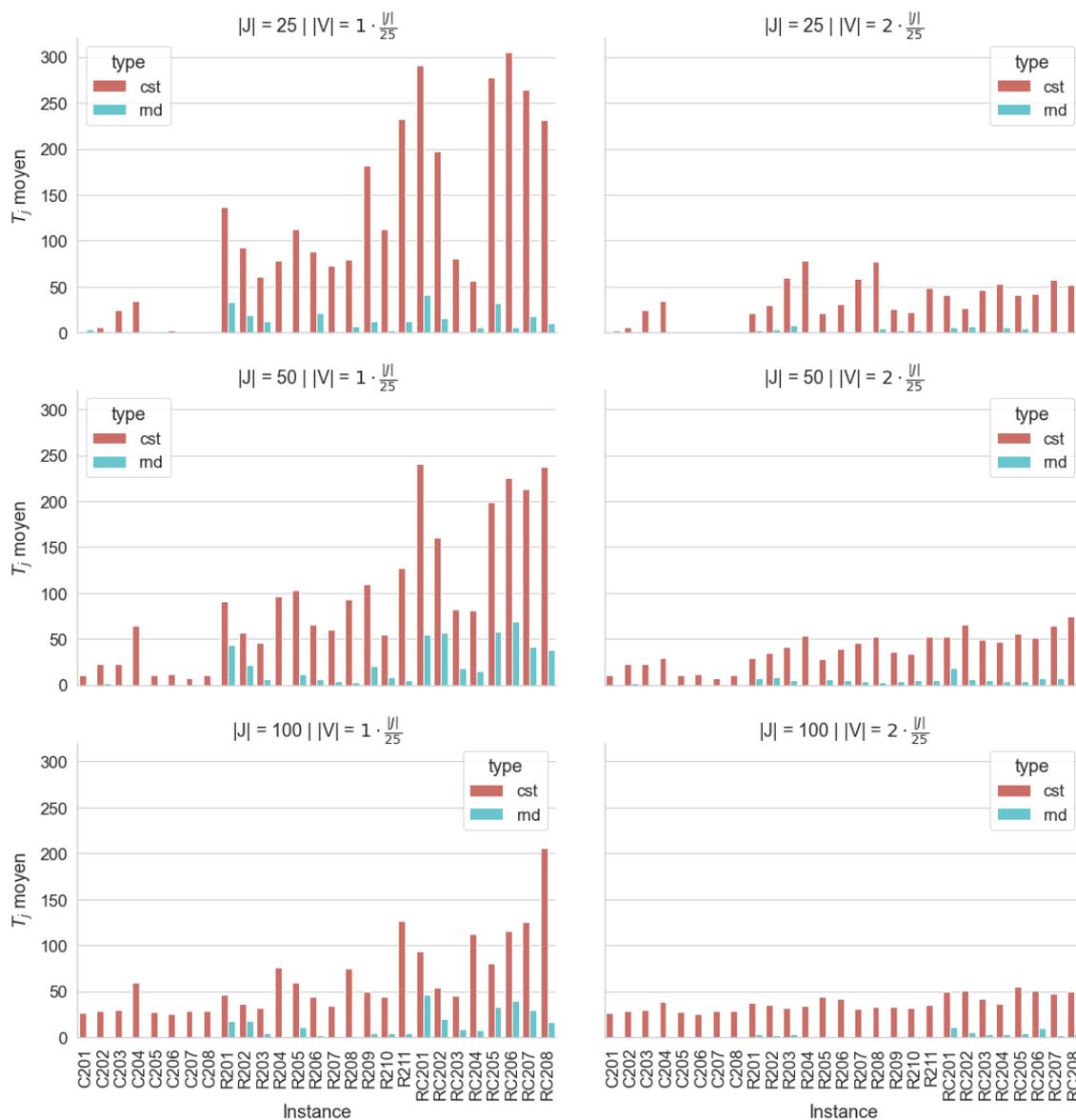


FIGURE 3.13 – Comparaison de la valeur du retard total des instances en fonction du type de date de début de livraison au plus tôt.

dem), le nombre de clients est important (environ 150) et le nombre de véhicules est limité (entre 3 et 5). Cependant, il n'est pas toujours possible d'allouer 10 minutes de calcul et il est important d'obtenir rapidement une première solution de qualité. C'est pour cela qu'il est important d'évaluer la qualité de la borne supérieure initiale du $B\&B_{MT-BDP}$.

3.4. EXPÉRIMENTATIONS NUMÉRIQUES

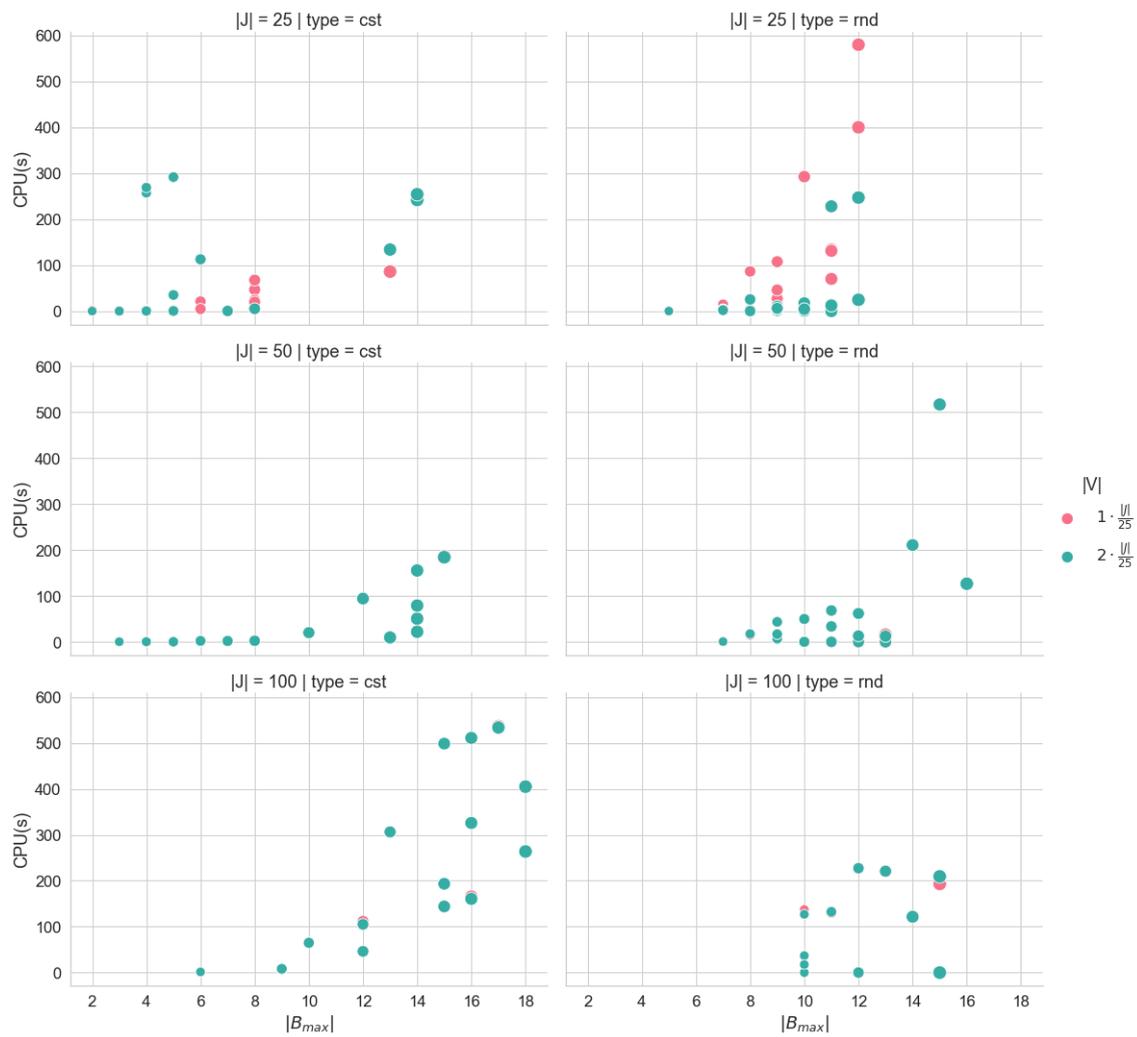


FIGURE 3.14 – Temps de calcul du 3BB des instances résolues en fonction de la taille maximale des trajets $|B_{max}|$.

3.4.3 Performance de la borne supérieure initiale

L'heuristique évaluée ici est celle utilisée pour calculer la borne supérieure à un nœud du $B\&B_{MT-BDP}$.

La figure 3.15 nous fournit le nombre de fois que l'heuristique d'entrée a été prouvée optimale et le nombre de fois qu'elle est la meilleure solution trouvée. On remarque que l'heuristique est la meilleure solution trouvée principalement lorsque les instances sont "faciles" à résoudre. Très souvent, lorsque l'heuristique est la meilleure solution trouvée, celle-ci est majoritairement prouvée optimale.

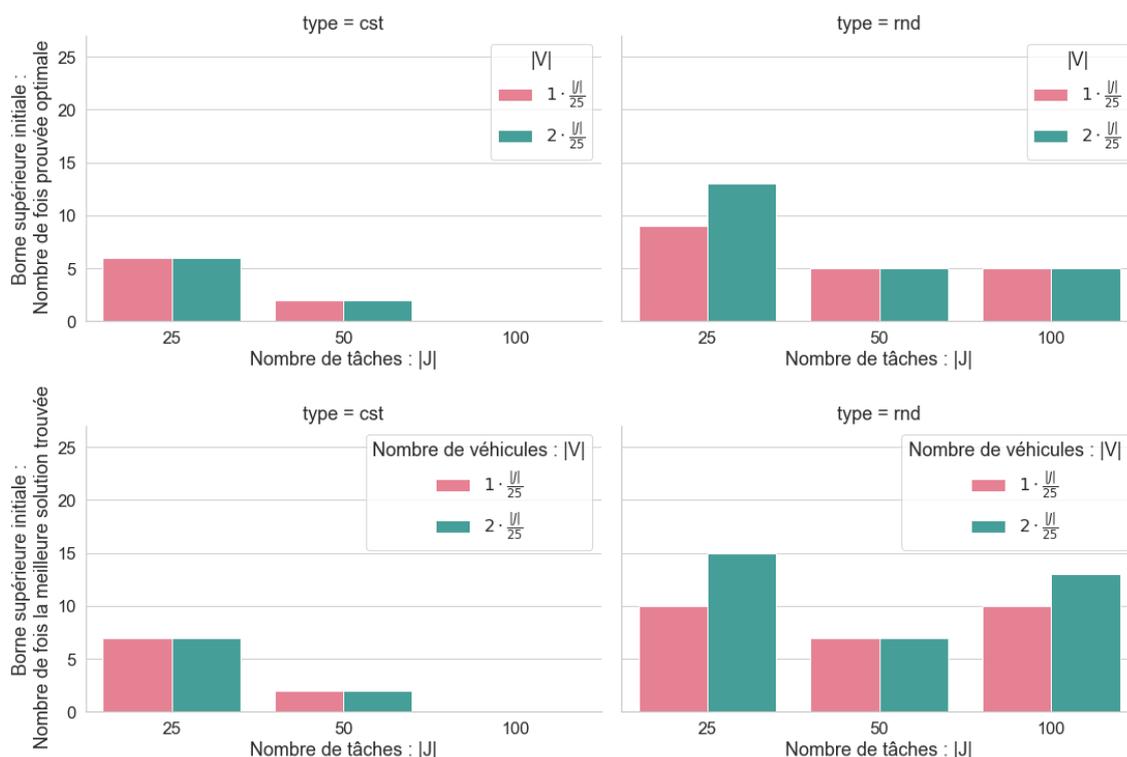


FIGURE 3.15 – Performance de la borne supérieure initiale du $B\&B_{MT-BDP}$

Même si l'heuristique n'est pas la meilleure solution trouvée, il est important d'évaluer son Gap relatif avec la meilleure solution trouvée. La figure 3.16 permet de voir que même si l'heuristique était beaucoup plus souvent la meilleure solution trouvée pour les instances de type "rd" que pour les instances de type "cst", celle-ci a un Gap relatif moyen beaucoup moins important sur ces dernières.

3.5. CONCLUSION

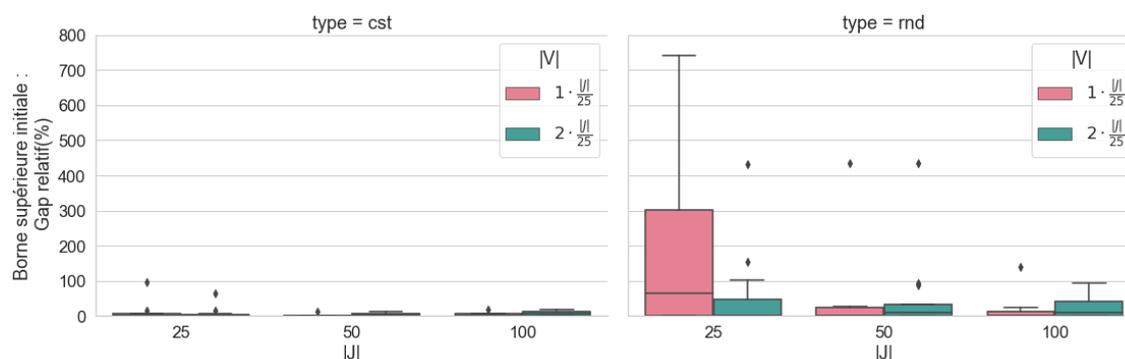


FIGURE 3.16 – Gap de l’heuristique par rapport à la meilleure solution trouvée

3.5 Conclusion

Dans ce chapitre, nous avons modélisé un problème de tournées de véhicules multi-trajets dont la fonction objectif est la somme des retards. Ce dernier représente le sous-problème de livraison du problème intégré de production et de livraison étudié dans le chapitre précédent. Nous avons proposé une décomposition en trois niveaux de décisions : affectation des trajets aux véhicules, sélection de séquences de livraison non-dominées et la détermination de l’ensemble des séquences de livraison non-dominées pour les trajets. Nous avons proposé un modèle linéaire en nombres mixtes pour les trois sous-problèmes et une méthode exacte par séparation et évaluation. La méthode exacte proposée a été évaluée sur des instances adaptées de la littérature. Les résultats ont montré que notre méthode était plus efficace dans la majorité des cas et plus résiliente face à l’augmentation du nombre de clients. Cette méthode exacte pourra être utilisée comme complément à l’heuristique proposée dans le chapitre précédent afin d’améliorer la solution retournée.

Les travaux et les résultats menés dans ce chapitre ont donné lieu à des publications dans deux conférences internationales ([Robbes *et al.*, 2021b], [Robbes *et al.*, 2019c]) et une conférence nationale ([Robbes *et al.*, 2019a]).

En perspectives, le temps de calcul de la méthode exacte peut être réduit en posant de nouvelles propriétés de conservation de la non-dominance des séquences de livraison en fonction des dates de départ. Le développement d’une méthode exacte où la composition des trajets n’est pas connue en avance serait possible en combinant la méthode exacte proposée avec une méthode de génération de colonnes. Il serait aussi intéressant de comparer notre méthode à une adaptation d’une méthode sur la base de génération d’un tour géant et la décomposition en plusieurs trajets.

Chapitre 4

Optimisation de la gestion des ressources périssables

Sommaire

4.1	Présentation de la problématique	130
4.1.1	Hypothèses de modélisation	130
4.1.2	État de l’art	134
4.2	Formulation mathématique	135
4.2.1	Paramètres d’entrées	135
4.2.2	Variables de décision	135
4.2.3	Contraintes	135
4.3	Bornes inférieures	139
4.3.1	Problème de Bin-packing (BPP)	139
4.3.2	Problème de Cutting-stock (CSP)	140
4.4	Méthode de résolution heuristiques	142
4.4.1	Ordonnancement et affectation des flacons initiale	142
4.4.2	Problème de Bin-packing avec conflits (BPC)	145
4.4.3	Permutation non-destructive et liste tabou	145
4.5	Expérimentations numériques	147
4.5.1	Génération des instances	148
4.5.2	Limites du modèle MILP et choix de la borne inférieure	151
4.5.3	Qualité de la heuristique	151
4.5.4	Nombre de permutations non-destructives et temps de calcul	152
4.6	Conclusions	156

4.1 Présentation de la problématique

Bien que la qualité de service soit la priorité, le coût de production des médicaments de chimiothérapie reste un critère important à optimiser. Ce dernier est directement lié au coût des matières premières.

Ce chapitre se focalise sur l'optimisation du coût des produits utilisés pour la préparation des médicaments. Pour préparer une injection, le pharmacien en charge prélève une certaine quantité d'un produit cytotoxique défini par l'ordonnance correspondant au patient. Ces produits sont achetés à divers laboratoires pharmaceutiques à des prix négociés (parfois encadrés par la législation). En fonction du produit utilisé, le prix d'un flacon varie de quelques euros à plus de 1000 euros pour un coût annuel total de plusieurs millions d'euros. La réduction de cette dépense a donc potentiellement un impact considérable pour la sécurité sociale. En effet, il est possible de perdre une quantité de certains produits pour diverses raisons. Lorsqu'un flacon est entamé pour la préparation d'une injection, si son contenu n'est pas entièrement utilisé, le reste, appelé reliquat, attend d'être utilisé pour une future préparation. Cependant, cette utilisation peut être compromise lorsque la durée de vie du produit est courte et que peu de préparations nécessitent ce produit. Réciproquement, certains produits sont beaucoup utilisés, ce qui oblige l'ouverture simultanée de plusieurs flacons et génère un afflux de reliquats important. La recherche opérationnelle semble alors adaptée pour répondre à cette problématique en proposant une gestion des reliquats optimisée pour éviter les pertes inutiles et réduire le coût moyen du médicament par la même occasion.

4.1.1 Hypothèses de modélisation

Pour modéliser le problème de gestion des reliquats, nous considérons tout d'abord L l'ensemble des produits cytotoxiques. Chaque produit $l \in L$ est caractérisé par le coût unitaire $cost_l$ d'un flacon, le volume cap_l d'un flacon et sa stabilité $stab_l$ correspondant à la durée de vie du reliquat après ouverture d'un flacon. On ne considère ici qu'un seul type de flacon possible pour chaque produit. S'agissant de produits cytotoxiques, le respect des temps de stabilité est extrêmement important, car ces produits se détériorent avec le temps et deviennent moins efficace voire dangereux.

L'ensemble des médicaments à préparer est représenté par l'ensemble de tâches J . Chaque tâche j utilise un unique produit et est définie par la quantité requise q_j de ce produit et son temps de Préparation p_j . On peut alors définir pour chaque produit l le sous-ensemble de tâches correspondantes J_l . Dans ce chapitre, nous ne considérons pas les dates de début au plus tôt ni les dates de fin souhaitées. Cette première étude vise prioritairement à optimiser la gestion des reliquats.

4.1. PRÉSENTATION DE LA PROBLÉMATIQUE

Ces tâches doivent être ordonnancées sur des machines et plus précisément sur des intervalles de disponibilités de celles-ci. Pour l'environnement de production, l'utilisation des produits cytotoxiques se fait durant la phase de **Préparation**. Le **Contrôle** et la livraison n'ont aucun impact sur les reliquats. La production est alors représentée par un ensemble d'isolateurs I , où chaque isolateur i est composé d'un ensemble de machines M_i .

L'optimisation de la gestion des reliquats doit s'opérer sur une échelle de temps plus grande que celles des deux chapitres précédents. En effet, ces derniers se focalisaient sur la planification d'une journée, alors que la planification considérée ici est de plusieurs semaines voire plusieurs mois. Il est donc important de prendre en compte la nouvelle contrainte des intervalles de disponibilités des machines. La figure 4.1 montre en rouge les périodes d'indisponibilités des machines sur deux échelles de temps : la journée et la semaine. La **Préparation** des tâches par la machine 1 n'est possible que dans l'intervalle de disponibilité $[8h,13h]$ ou l'intervalle de disponibilité $[13h30,18h]$ du lundi au vendredi. Aucune machine n'est disponible le week-end. Les périodes d'indisponibilité des machines sont contraignantes, en particulier celles du week-end, car elles sont décisives pour les produits à courte stabilité. On définit alors un ensemble d'intervalles de disponibilité H . Chaque intervalle de disponibilité h est représenté par sa date de début de disponibilité s_h et sa date de fin de disponibilité e_h . Un intervalle de disponibilité est dédié à une machine et donc de fait à un isolateur. On définit alors pour l'isolateur i , le sous-ensemble des intervalles de disponibilité correspondant H_i .

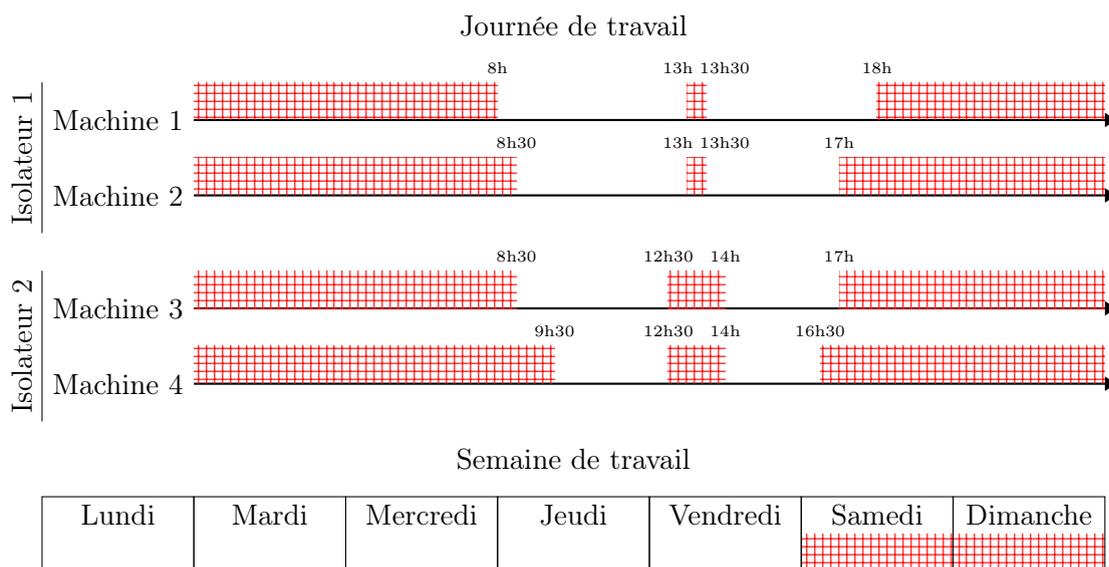


FIGURE 4.1 – Périodes d'indisponibilités des machines dans la journée et dans la semaine

Ordonnancer la tâche j revient à définir sa date de début de **Préparation** S_j , sa date

de fin de **Préparation** C_j et la machine à laquelle elle est affectée. Une fois les tâches ordonnancées, il faut leur affecter des flacons. Nous posons comme hypothèse que chaque tâche nécessite au plus un unique flacon. Cette hypothèse forte n'est pas posée comme une contrainte dure par l'UBCO, mais sans celle-ci les solutions proposées pourraient être difficilement applicables et l'utilisation de plusieurs flacons impacterait aussi fortement le temps de **Préparation**. L'affectation des flacons aux tâches est soumise à des contraintes de compatibilités entre les tâches :

1. **Conflit de chevauchement** : deux tâches qui se chevauchent ne peuvent pas utiliser un même flacon. Deux tâches j et j' sont incompatibles si :

$$\max(S_j, S_{j'}) - \min(C_j, C_{j'}) < 0 \quad (4.1)$$

2. **Conflit de stérilisation** : un flacon ne peut être utilisé par deux tâches préparées dans deux isolateurs différents s'il n'y a pas le temps de le stériliser. Deux tâches j et j' sont incompatibles si :

$$\max(S_j, S_{j'}) - \min(C_j, C_{j'}) < p^{Ste} \quad (4.2)$$

3. **Conflit de stabilité** : deux tâches ne peuvent pas utiliser un même flacon si le temps d'attente entre les préparations dépasse la durée de stabilité du produit. Deux tâches j et j' sont incompatibles si :

$$\max(C_j, C_{j'}) - \min(S_j, S_{j'}) > stab_l \quad (4.3)$$

Ces différents conflits permettent de construire un **graphe de conflits**. Le problème d'affectation des flacons aux tâches revient alors à résoudre un problème de **Bin-packing avec conflits** (BPC). Le problème de BPC est une variante du problème de **Bin-packing** (BPP), où un graphe de conflits définit les paires d'objets ne pouvant être rangés ensemble dans un même bin. Le graphe complémentaire du graphe de conflits est nommé graphe de compatibilités.

Le problème intégré de production et de gestion des ressources consommables consiste alors à affecter et ordonnancer les préparations des tâches sur les intervalles de disponibilités et aux flacons de manière à réduire le coût total des flacons. C'est un problème intégré de production et de gestion des ressources consommables. La figure 4.2 représente à la fois le diagramme de Gantt et l'affectation des flacons aux tâches pour la planification de 27 tâches utilisant 4 produits sur 4 machines réparties en 2 isolateurs. Pour pouvoir préparer les 27 tâches, la solution utilise 2 flacons du produit a , 4 flacons du produit b , 4 flacons du produit c et 3 flacons du produit d .

4.1. PRÉSENTATION DE LA PROBLÉMATIQUE

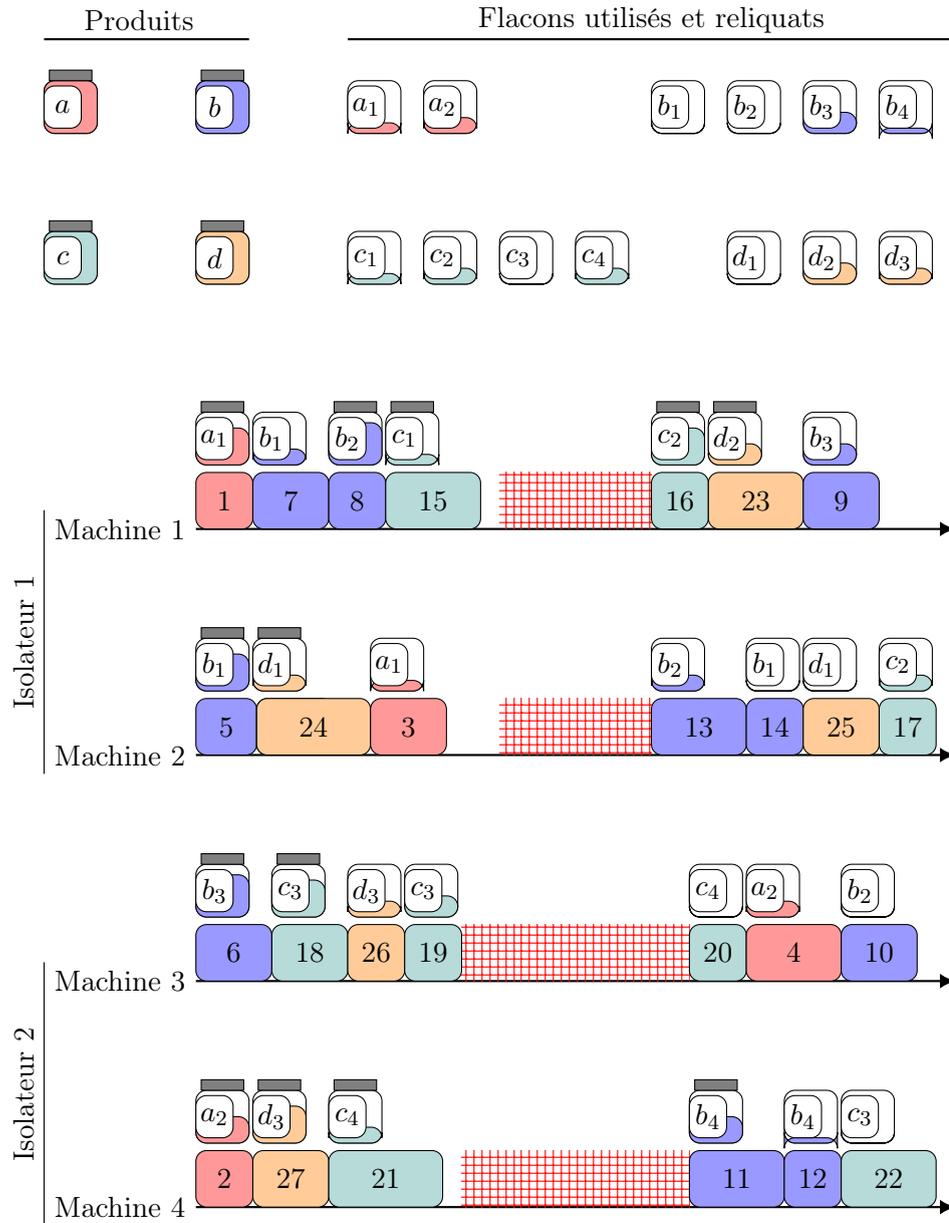


FIGURE 4.2 – Exemple d’une solution du problème intégré de production et de gestion des ressources consommables

Ce problème est NP-difficile au sens fort puisque si l'on suppose une version du problème avec un seul intervalle de disponibilité et un seul produit dont le coût est de 1 et de stabilité supérieur à la durée de l'intervalle de disponibilité, alors le problème est un simple Bin-packing.

4.1.2 État de l'art

La littérature sur les problèmes intégrés de production et de découpe est éparse et de différentes natures. Dans [Li, 1996] un problème intégré de découpe à deux dimensions contraint par des dates de début au plus tôt et des dates de fin souhaitées est introduit. Depuis plusieurs études se sont intéressées aux variantes de ce problème comme dans [Reinertsen et Vossen, 2010, Arbib et Marinelli, 2014, Braga *et al.*, 2015, Braga *et al.*, 2016, Pitombeira-Neto et Prata, 2020]. L'étude des problèmes intégrés de lot-sizing et de cutting-stock font l'œuvre d'un intérêt croissant depuis 2006 d'après [Melega *et al.*, 2018].

Dans [Billaut *et al.*, 2015], les auteurs traitent un problème d'ordonnancement à une machine avec des contraintes de rangement. L'exécution de chaque tâche est contrainte par l'utilisation d'une certaine quantité d'une même ressource consommable prélevée dans un flacon qui doit être mis à disposition pendant une durée définie. Chaque tâche a une date de fin souhaitée et la somme des retards est contrainte par une borne supérieure. Le critère d'optimisation est la minimisation du nombre de flacons utilisés. Pour résoudre le problème, une méthode de Recovering Beam Search algorithm et une matheuristique sont proposées.

Cependant, à notre connaissance étude traite de l'intégration d'un problème de Bin-packing avec conflit à un problème d'ordonnancement à machines parallèles. La version unidimensionnelle du BPC a été introduite dans [Jansen et Öhring, 1997]. Depuis la littérature, s'est enrichie, mais à un rythme beaucoup plus lent que celle du BPP. Quelques papiers se sont intéressés au développement de bornes inférieures tel que [Gendreau *et al.*, 2004, Khanafer *et al.*, 2010]. Des méthodes exactes ont été aussi développées en particulier des algorithmes de branch and price comme dans [Elhedhli *et al.*, 2011, Sadykov et Vanderbeck, 2013]. Des algorithmes d'approximation ont également été développés dans [Jansen, 1999, Bujtás *et al.*, 2011]. [Khanafer, 2010] présente un ensemble d'heuristiques de la littérature, certaines adaptées du problème de Bin-packing et d'autres adaptées du problème de coloriage de graphes. Dans [Khanafer *et al.*, 2012] une nouvelle heuristique décomposition en arbre a été développée pour le BPC à deux dimensions.

4.2 Formulation mathématique

Nous proposons une modélisation mathématique du problème intégré de production de chimiothérapie et de gestion des reliquats à l'aide d'un modèle de programmation linéaire en nombres entiers mixtes (MILP).

4.2.1 Paramètres d'entrées

Le tableau 4.1 résume les paramètres d'entrées nécessaires au MILP du problème de gestion des reliquats.

Notation et domaine	Description
L	ensemble des produits utilisés
I	ensemble des isolateurs
H	ensemble des intervalles de disponibilités
$H_i, \forall i \in I$	ensemble des intervalles de disponibilités de l'isolateur i
$s_h, \forall h \in H$	date de début de l'intervalle de disponibilité h
$e_h, \forall h \in H$	date de fin de l'intervalle de disponibilité h
F	ensemble des flacons
$F_l, \forall l \in L$	ensemble des flacons contenant le produit l
$stab_l, \forall l \in L$	durée de stabilité du produit l
$cost_l, \forall l \in L$	coût d'un flacon du produit l
$cap_l, \forall l \in L$	capacité d'un flacon du produit l
J	ensemble des tâches
$J_l, \forall l \in L$	ensemble des tâches nécessitant le produit l
p^{Ste}	temps de Stérilisation
$p_j^{Prep}, \forall j \in J$	temps de Préparation de la tâche j
$q_j, \forall j \in J$	quantité de produit nécessaire pour préparer la tâche j

TABLE 4.1 – Paramètres d'entrées du modèle MILP

4.2.2 Variables de décision

Le modèle MILP utilise cinq familles variables binaires, quatre pour représenter les diverses affectations (x, γ, δ, z) et une (y) pour modéliser la précedence entre les tâches et une variable continue pour les dates de fin de **Préparation**.

4.2.3 Contraintes

Notation et domaine	Description
$x_{j,h} \in \{0,1\}, \forall (j,h) \in J \times H$	$x_{j,h} = 1$ si la tâche j est préparée dans l'intervalle de disponibilité h .
$\gamma_{j,j'} \in \{0,1\}, \forall (j,j') \in J^2$	$\gamma_{j,j'} = 1$ si les tâches j et j' sont préparées dans deux isolateurs différents
$y_{j,j'} \in \{0,1\}, \forall (j,j') \in J^2$	$y_{j,j'} = 1$ si la tâche j est préparée avant la tâche j'
$C_j \in \mathbb{R}_+$	la date de fin de Préparation de la tâche j .
$\delta_f \in \{0,1\}, \forall f \in F$	$\delta_f = 1$ si le flacon f est utilisé.
$z_{f,j} \in \{0,1\}, \forall (f,j) \in F \times J$	$z_{f,j} = 1$ si la tâche j utilise le flacon f

TABLE 4.2 – Variables de décisions des modèles

4.2. FORMULATION MATHÉMATIQUE

La fonction objectif est la minimisation du coût total, soit la somme des coûts associés à chaque flacon utilisé :

$$\text{MIN} \quad \sum_{l \in L} (\text{cost}_l \cdot \sum_{f \in F_l} \delta_f)$$

Chaque tâche doit être attribuée à un unique intervalle de disponibilité :

$$\forall j \in J \quad \sum_{h \in H} x_{j,h} = 1 \quad (4.4)$$

Il existe une relation de précédence entre deux tâches :

$$\forall (j, j') \in J^2, j' > j \quad y_{j,j'} + y_{j',j} = 1 \quad (4.5)$$

La **Préparation** d'une tâche est postérieure à la date de début de disponibilité de l'intervalle auquel elle est assignée :

$$\forall j \in J \quad C_j^{\text{Prep}} - p_j^{\text{Prep}} \geq \sum_{h \in H} s_h \cdot x_{j,h} \quad (4.6)$$

La date de début de **Préparation** de la tâche j' est postérieure à la date de fin de **Préparation** de la tâche j si les tâches j et j' sont préparées dans le même intervalle de disponibilité et si j précède j' :

$$\forall (j, j', h) \in J^2 \times H, j \neq j' \quad C_{j'}^{\text{Prep}} - p_{j'}^{\text{Prep}} \geq C_j^{\text{Prep}} - \mathbb{M} \cdot (2 - x_{j,h} - x_{j',h} + y_{j',j}) \quad (4.7)$$

La **Préparation** d'une tâche est antérieure à la date de fin de disponibilité de l'intervalle auquel elle est assignée :

$$\forall j \in J \quad C_j^{\text{Prep}} \leq \sum_{h \in H} e_h \cdot x_{j,h} \quad (4.8)$$

Si la tâche j est préparée dans l'intervalle de disponibilité associé à l'isolateur i et la tâche j' est préparée dans un intervalle de disponibilité associé à un autre isolateur que i , alors les deux tâches sont préparées dans deux isolateurs différents :

$$\forall i \in I, \forall (j, j', h) \in J^2 \times H_i, j' \neq j \quad \gamma_{j,j'} \geq x_{j,h} + \sum_{i' \in I \setminus \{i\}} \sum_{h' \in H_{i'}} x_{j',h'} - 1 \quad (4.9)$$

4.2. FORMULATION MATHÉMATIQUE

Chaque tâche doit être affectée à exactement un flacon :

$$\forall l \in L, \forall j \in J_l, \quad \sum_{f \in F_l} z_{f,j} = 1 \quad (4.10)$$

Le nombre de flacons utilisés doit pouvoir subvenir à la demande. Cette contrainte est une inégalité valide servant à accélérer le modèle :

$$\forall l \in L \quad \sum_{f \in F_l} cap_l \cdot \delta_f \geq \sum_{l \in J_l} q_j \quad (4.11)$$

La quantité prélevée à un flacon est limitée par le volume du flacon :

$$\forall l \in L, \forall f \in F_l \quad \sum_{j \in J_l} q_j \cdot z_{f,j} \leq cap_l \cdot \delta_f \quad (4.12)$$

Si le temps entre la date de début de **Préparation** de la tâche j et la date de fin de **Préparation** de la tâche j' est plus grand que la stabilité du produit alors soit la tâche j' précède la tâche j soit les tâches n'utilisent pas le même flacon :

$$\forall l \in L, \forall (j, j', f) \in J_l^2 \times F_l, j \neq j' \quad C_{j'}^{Prep} - C_j^{Prep} + p_j^{Prep} \leq stab_l + \mathbb{M} \cdot (2 - z_{f,j} - z_{f,j'} + y_{j',j}) \quad (4.13)$$

Si le temps entre la date de fin de **Préparation** de la tâche j et la date de début de **Préparation** de la tâche j' est plus petit que le temps de stérilisation alors soit la tâche j' précède la tâche j soit les tâches n'utilisent pas le même flacon soit les deux tâches sont préparées dans le même isolateur. Si ce même temps est strictement négatif alors soit les deux tâches n'utilisent pas le même flacon soit la tâche j' précède la tâche j :

$$\forall l \in L, \forall (j, j') \in J_l^2, j \neq j' \quad C_{j'}^{Prep} - p_{j'}^{Prep} - C_j^{Prep} \geq p^{Ste} \cdot \gamma_{j,j'} - \mathbb{M} \cdot (2 - z_{f,j} - z_{f,j'} + y_{j',j}) \quad (4.14)$$

La valeur de \mathbb{M} dans ce modèle représente une borne supérieure de la date de fin de **Préparation**, il est donc possible de l'assigner à la date de fin de disponibilité au plus tard $\max_{h \in H} \{e_h\}$.

4.3 Bornes inférieures

Afin d'évaluer nos algorithmes, nous proposons trois bornes inférieures. La première borne inférieure est calculée par la formule 4.15 qui représente les nombres minimums nécessaire de flacons pour répondre aux demandes des produits, pondérés par les coût unitaires des flacons.

$$\sum_{l \in L} cost_l \cdot \left\lceil \frac{\sum_{j \in J_l} q_j}{cap_l} \right\rceil \quad (4.15)$$

Cependant, cette borne inférieure est l'adaptation d'une borne inférieure faible pour le problème de Bin-packing. Donc, comme dans le problème intégré de production et de gestion des ressources consommables, il y a des conflits entre les tâches, nous proposons deux autres bornes inférieures basées sur la résolution de deux problèmes NP-difficiles : Bin-Packing et Cutting-Stock.

4.3.1 Problème de Bin-packing (BPP)

Tout d'abord la borne inférieure la plus évidente est de résoudre une instance du problème de Bin-packing pour chaque produit l . Cela revient tout simplement à relâcher les contraintes de conflits entre les tâches. Pour modéliser le problème de Bin-packing nous utilisons une modélisation plus performante proposée dans [Hadj Salem et Kieffer, 2020] que la modélisation classique. Cette modélisation n'utilise qu'une seule famille de variables binaires $\zeta_{j,j'}$ tel que $\zeta_{j,j'} = 1$ si la tâche j est la tâche de plus petit index de son bin et si j' est dans ce même bin. L'avantage de cette modélisation est qu'elle réduit par deux le nombre de variables par rapport à la modélisation classique ($\frac{|J_l|^2 + |J_l|}{2}$ variables au lieu de $|J_l|^2$) et casse les symétries entre les bins.

$$Obj : \sum_{j=1}^{|J_l|} \zeta_{j,j} \quad \left(\begin{array}{cccc} \zeta_{j,1} = 1 & \sum_{j=1}^2 \zeta_{j,2} = 1 & \dots & \sum_{j=1}^{|J_l|} \zeta_{j,|J_l|} = 1 \\ \sum_{j'=1}^{|J_l|} q_{j'} \cdot \zeta_{1,j'} \leq cap_l & \zeta_{1,1} & \zeta_{1,2} & \dots & \zeta_{1,|J_l|} \\ \sum_{j'=2}^{|J_l|} q_{j'} \cdot \zeta_{2,j'} \leq cap_l & & \zeta_{2,2} & \vdots & \zeta_{2,|J_l|} \\ \vdots & & & \ddots & \vdots \\ q_{|J_l|} \cdot \zeta_{|J_l|,|J_l|} \leq cap_l & & & & \zeta_{|J_l|,|J_l|} \end{array} \right)$$

FIGURE 4.3 – Représentation de la matrice creuse des variables $\zeta_{j,j'}$ et les contraintes du modèle linéaire en nombres entiers.

4.3. BORNES INFÉRIEURES

La figure 4.3 représente la matrice creuse des variables $\zeta_{j,j'}$ et leurs implications dans les contraintes du modèle linéaire en nombres entiers. La somme de la diagonale donne le nombre de bins. Il y a exactement un 1 par colonne et la somme des variables d'une même ligne pondérées par la quantité requise est inférieure à la capacité des bins.

La formulation mathématique correspondant à cette modélisation est ci-dessous :

$$\begin{aligned} \text{MIN} \quad & \sum_{j \in J_l} \zeta_{j,j} \\ \forall j' \in J_l, \quad & \sum_{j \in J_l, j \leq j'} \zeta_{j,j'} = 1 \end{aligned} \quad (4.16)$$

$$\forall j \in J_l \quad \sum_{j' \in J_l, j \leq j'} q_{j'} \cdot \zeta_{j,j'} \leq \text{cap}_l \cdot \zeta_{j,j} \quad (4.17)$$

$$\forall (j', j'') \in J_l^2, j' \leq j'' \quad \zeta_{j',j''} + \sum_{j \in J_l, j < j'} \zeta_{j,j'} \leq 1 \quad (4.18)$$

La fonction objectif 4.3.1 est de minimiser le nombre de bins. La contrainte 4.16 assure que chaque tâche est affectée à un unique flacon. La contrainte 4.17 est la contrainte de capacité. La contrainte 4.18 est une inégalité valide qui contraint les indexations des bins par leurs affectations.

Si l'on fait la somme pondérée par cost_l des valeurs optimales des problèmes de Bin-packing correspondant à chaque produit alors on obtient une borne inférieure du problème.

4.3.2 Problème de Cutting-stock (CSP)

Le problème de la borne inférieure calculée à l'aide de $|L|$ instances du problème de Bin-packing est que le temps de calcul nécessaire est très sensible aux nombres de tâches $|J_l|$. C'est pour cela que nous proposons une troisième borne inférieure basée sur la résolution de $|L|$ problèmes Cutting-stock. La différence majeure de la modélisation en problème de Cutting-stock par rapport à la modélisation en problème de Bin-packing est d'indifférencier les tâches en les agrégeant par quantité q_j commune. On pose $\Omega_l = \{q_j\}_{j \in J_l}$ l'ensemble des différentes quantités existantes pour le produit l . On définit pour chaque quantité $q \in \Omega_l$ sa demande $\rho_{l,q}$ qui est définie par l'équation 4.19.

$$\rho_{l,q} = |j \in J_l, q_j = q| \quad (4.19)$$

Pour un produit l il est possible de générer W_l l'ensemble des partitions par les éléments de Ω_l des nombres entiers compris dans $\llbracket 1, \text{cap}_l \rrbracket$. La figure 4.4 représente les différents

4.3. BORNES INFÉRIEURES

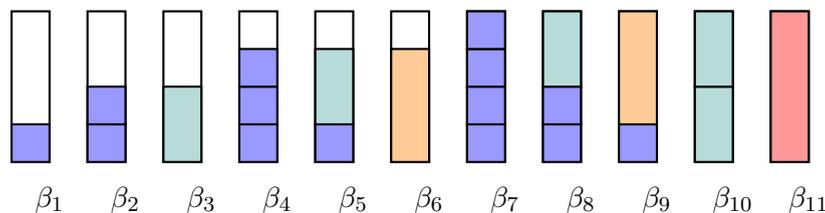


FIGURE 4.4 – L'ensemble des partitions en nombres entiers (i.e. $\Omega_l = \llbracket 1, cap_l \rrbracket$), W_l lorsque $cap_l = 4$

cap_l	1	2	3	4	5	6	7	8	9	10
$ W_l $	1	3	6	11	18	29	44	66	96	138

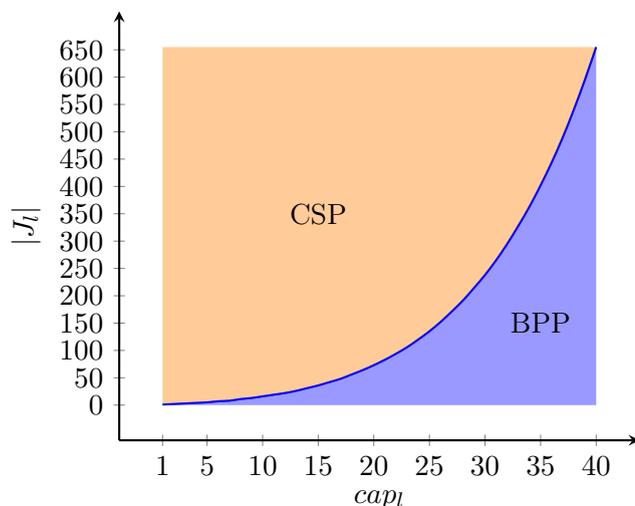
TABLE 4.3 – Le nombre de partitions en nombres entiers pour $cap_l \leq 10$.

partitions pour $cap_l = 4$ lorsque $\Omega_l = \llbracket 1, cap_l \rrbracket$.

Une fois l'ensemble des partitions W_l défini, il est possible de construire le paramètre $a_{w,q}$ représentant le nombre de partitions de taille q dans le partitionnement w et la variable $\beta_w \in \mathbb{N}$ qui définit le nombre de fois que le partitionnement w est utilisé par la solution. La fonction objectif du problème de Cutting-Stock est alors de minimiser le nombre des partitions utilisés. Le modèle linéaire en nombres entiers est défini ci-dessous :

$$\begin{aligned} \min \quad & \sum_{w \in W_l} \beta_w \\ \forall q \in \Omega \quad & \sum_{w \in W_l} a_{w,q} \cdot \beta_w \geq \rho_{l,q} \end{aligned} \tag{4.20}$$

Une fois chaque problème de Cutting-stock résolu, la borne inférieure est définie par la somme pondérée $cost_l$ des valeurs optimales. L'intérêt de cette borne inférieure est qu'elle donne la même valeur que celle basée sur le problème de Bin-packing, mais elle peut résoudre plus rapidement si le nombre de partitions est faible. Le tableau 4.3 donne le nombre de variables du modèle pour $cap_l \leq 10$. La figure 4.5 permet de déterminer quel modèle entre celui du Bin-packing (BPP) et celui du Cutting-stock (CSP) a le moins de variables en fonction du nombre de tâches et de cap_l (en supposant que $\Omega_l = \llbracket 1, cap_l \rrbracket$). On remarque que le nombre de variables du modèle CSP est très largement inférieur à celui du modèle BPP lorsque $|J_l|$ est grand et que cap_l est petit.

FIGURE 4.5 – Choisir la borne inférieure en fonction des dimensions de l'instance $(|J_l|, cap_l)$

4.4 Méthode de résolution matheuristiques

Nous proposons une méthode matheuristique pour résoudre le problème intégré de production et de gestion des ressources consommables. L'organigramme de la matheuristique est présenté dans la figure 4.4. Tout d'abord, un planning de production initiale est construit à l'aide d'un algorithme heuristique (section 4.4.1). Une fois l'affectation des tâches aux machines et leurs dates de **Préparation** définies, un graphe de conflits est généré pour chaque produit grâce aux règles : 4.1, 4.2 et 4.3. Une affectation heuristique des flacons aux tâches est alors construite (section 4.4.1). Une première solution complète du problème intégré est alors générée. Indépendamment, chacun des graphes de conflits est partitionné en composantes connexes. Pour chaque composante connexe, un modèle linéaire en nombre entier du problème de Bin-packing avec conflits (BPC) (section 4.4.2), initialisé avec l'affectation heuristique, est résolu. À la fin de cette opération une seconde solution complète du problème intégré est alors générée. Celle-ci est de qualité égale ou supérieure à la première. Pour finir, une boucle permet une recherche avec liste tabou de permutations dites "non-destructives" (section 4.4.3) permettant d'améliorer la solution.

4.4.1 Ordonnement et affectation des flacons initiale

Nous proposons d'étudier deux algorithmes d'initialisation du planning de production. Le premier nommé "random" est un planning construit aléatoirement avec pour règle de n'avoir aucun temps d'attente durant un même intervalle de disponibilité.

La deuxième heuristique nommée "greedy" est décrite par l'algorithme 6. Elle consiste

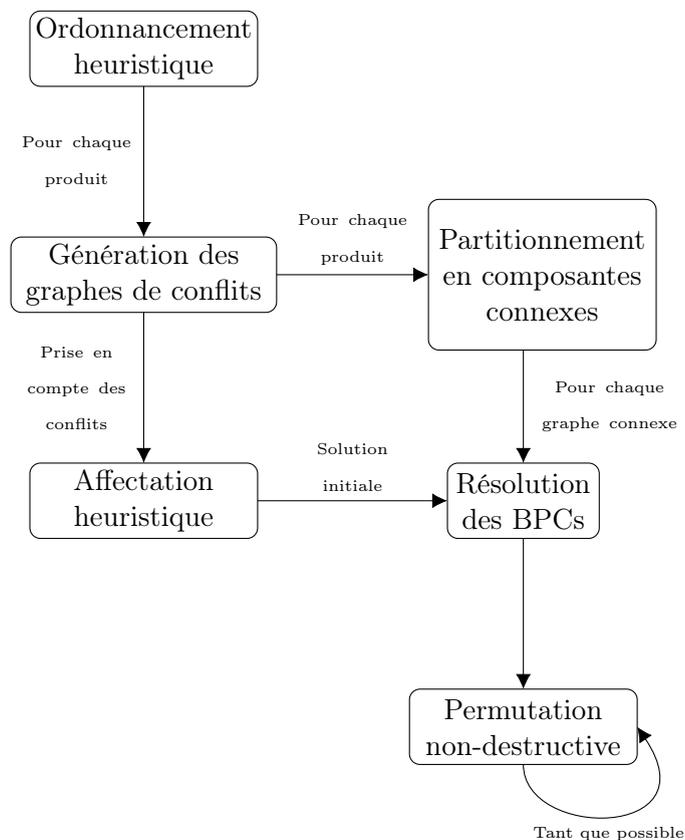


FIGURE 4.6 – Organigramme de la matheuristique

à trier les tâches par $cost_l$ décroissant et de successivement les planifier tant que possible sur la même machine au plus tôt. Ce processus qui consiste à planifier les tâches nécessitant le même produit sur le moins de machines différentes possibles, vise à limiter les conflits de chevauchement (équation 4.1) et les conflits de stérilisation (équation 4.2). Quant à la règle de planification, au plus tôt, elle diminue les risques de conflits de stabilité (équation 4.3).

Une fois un planning de production défini, il est possible de construire les graphes de conflits et par la même occasion de définir une affectation heuristique des flacons. Pour cela, nous proposons l'adaptation de l'heuristique "worst-fit" du Bin-packing pour le Bin-packing avec conflits. Cette heuristique décrite par l'algorithme 7 se base sur l'affectation prioritaire aux tâches nécessitant de grandes quantités de produits des flacons les moins utilisés, ceci en respectant les compatibilités entre les tâches. Une première solution heuristique réalisable du problème intégré de production et gestion des ressources consommables est alors construite.

Algorithm 6 Ordonnancement initial : greedy

Paramètres : Ensemble des machines M , l'ensemble des produits L et l'ensemble des tâches J

Soit \mathcal{L} la liste des produits trié par $cost_l$ décroissant

$m \leftarrow$ première machine

for $l \in \mathcal{L}$ **do**

for $j \in \mathcal{J}_l$ **do**

if La machine m peut préparer la tâche j **then**

 Préparer la tâche j au plus tôt sur la machine m

else

$m \leftarrow m + 1$

end if

end for

end for

return Planning de Préparation $Prep$

Algorithm 7 Heuristique : worst-fit

Paramètres : Planning de Préparation, l'ensemble des produits L et l'ensemble des tâches J

$F \leftarrow \emptyset$

for $l \in L$ **do**

 Soit \mathcal{J}_l la liste des tâches utilisant le produit l trié par q_j décroissant

$\mathcal{F}_l \leftarrow \emptyset$

for $j \in \mathcal{J}_l$ **do**

$assign \leftarrow False$

for $f \in \mathcal{F}_l$ **do**

if La tâche j peut prélever sa quantité depuis le flacon f **then**

 Affecter la tâche j au flacon f

$assign \leftarrow True$

end if

if $assign$ **then**

break

end if

end for

if not $assign$ **then**

 Affecter la tâche j à un nouveau flacon f

$\mathcal{F}_l \leftarrow \mathcal{F}_l \cup \{f\}$

end if

 Trier \mathcal{F}_l par capacité restante décroissante

end for

$F \leftarrow F \cup \mathcal{F}_l$

end for

return F

4.4.2 Problème de Bin-packing avec conflits (BPC)

Une fois les graphes de conflits construits, on peut définir les ensembles suivants :

- Ψ_j est l'ensemble des tâches en conflit avec la tâche j .
- Φ_j est l'ensemble des tâches compatibles (sans conflit) avec la tâche j .

Le problème de Bin-packing avec conflits peut alors être modélisé en adaptant les contraintes du modèle du problème de Bin-packing et en ajoutant la contrainte 4.24.

$$\text{MIN} \quad \sum_{j \in J_l} \zeta_{j,j}$$

$$\forall j' \in J_l, \quad \sum_{j \in \Phi_{j'}, j \leq j'} \zeta_{j,j'} = 1 \quad (4.21)$$

$$\forall j \in J_l \quad \sum_{j' \in \Phi_j, j \leq j'} q_{j'} \cdot \zeta_{j,j'} \leq \text{cap}_l \cdot \zeta_{j,j} \quad (4.22)$$

$$\forall j' \in J_l, \forall j'' \in \Phi_{j'}, j' \leq j'' \quad \zeta_{j',j''} + \sum_{j \in \Phi_{j'}, j < j'} \zeta_{j,j'} \leq 1 \quad (4.23)$$

$$\forall j \in J_l, \forall j' \in \Phi_j, j < j', \forall j'' \in \Phi_j \cap \Psi_{j'}, j' < j'' \quad \zeta_{j,j'} + \zeta_{j,j''} \leq 1 \quad (4.24)$$

Afin d'obtenir des instances du problème de Bin-packing avec conflit plus petites, il est possible de partitionner les différents graphes de compatibilités (graphe complémentaire du graphe de conflits) en composantes connexes et de résoudre indépendamment une instance du BPC pour chacune d'entre elle. En effet, la résolution indépendante des composantes connexes du graphe de compatibilité est possible sans perte de l'optimalité de la solution car les contraintes 4.21 et 4.22 ne permettent pas l'affectation d'un flacon défini par sa tâche de plus petit index à une tâche non compatible à celle-ci. L'avantage de la résolution sur des graphes de compatibilité connexes est de limiter le nombre de contraintes 4.24.

4.4.3 Permutation non-destructive et liste tabou

La résolution des différents BPC est une opération coûteuse en temps de calcul. Il est alors préférable de ne le faire qu'une seule fois durant toute l'exécution de la matheuristique. Nous proposons une recherche locale avec liste tabou pour trouver un planning de production offrant une affectation de flacon moins coûteuse. Comme opérateur de voisinage nous proposons une méthode de permutation de tâches dans le planning de production qui

permet de réduire le nombre de flacons à utiliser sans avoir à recalculer de BPC, en ne changeant le flacon affecté que d'une seule tâche à la fois.

Cette permutation échange la position dans l'ordonnancement de deux tâches j et j' et assigne un nouveau flacon f à la tâche j en respectant les six conditions suivantes :

1. Le flacon f contient le produit requis pour la tâche j .
2. Le flacon affecté à j n'est pas totalement utilisé. Si un flacon n'a aucun reliquat, il est préférable de ne pas le modifier.
3. Le flacon f peut être affecté à la tâche j après sa permutation dans l'ordonnancement avec j' .
4. Les tâches j et j' n'utilisent pas le même produit. Il est même préférable que le produit utilisé par j est un coût supérieur au produit utilisé par j' .
5. Les tâches j et j' ont le même temps de préparation ($p_j = p_{j'}$). Cela évite d'impacter le reste de l'ordonnancement en décalant les dates des tâches suivantes et donc de remettre en cause de très nombreuses affectations de flacons, même pour les produits non concernés par la permutation.
6. la permutation dans l'ordonnancement des deux tâches ne remet pas en cause le flacon affecté à j' .

Les trois premières conditions permettent d'assurer qu'il y a une possibilité de diminuer le nombre de flacons utilisés pour le produit utilisé par la tâche j . La condition 4 limite le blocage prématuré d'amélioration de l'affectation des flacons d'un même produit. La cinquième et la sixième conditions sont là pour éviter de compromettre les affectations de flacons aux autres tâches. Une permutation respectant ces critères est nommée permutation non-destructive.

La figure 4.7 montre le fonctionnement d'une permutation non-destructive sur un exemple avec 12 tâches réparties en deux produits. Les graphes de conflits des deux produits sont représentés avec : en noir les conflits inchangés par la permutation, en pointillés rouges les conflits supprimés par la permutation et en tirets verts les conflits créés par la permutation. Le diagramme de Gantt, l'affectation des flacons et les deux graphes de conflits sont représentés pour mieux comprendre les impacts d'une telle permutation. La tâche 2 est permutée avec la tâche 12 et transférée dans le flacon 2. La permutation est bien non-destructive, car les temps de Préparation p_2^{Prep} et p_{12}^{Prep} sont égaux, la quantité q_2 peut être prélevée au flacon 2 et la permutation des deux tâches n'a pas compromis l'affectation de la tâche 12.

L'utilisation successive de permutations non-destructives permet de réduire le coût total de la solution par modification de l'ordonnancement. Pour maximiser l'efficacité des permutations non-destructives, celles-ci sont choisies en priorité pour l'amélioration de

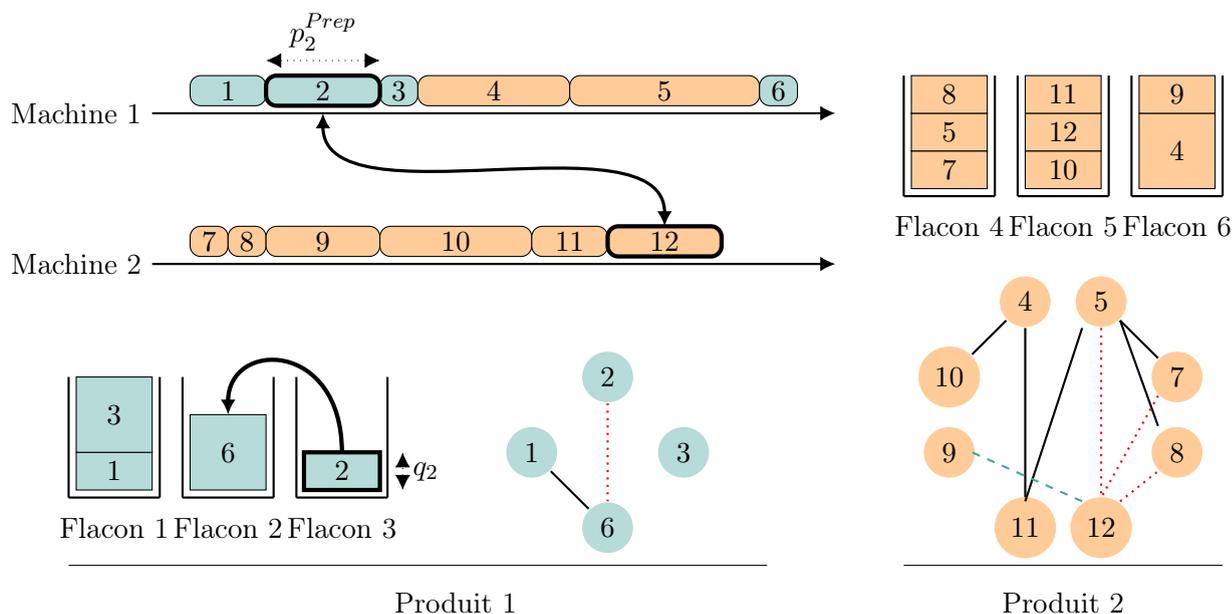


FIGURE 4.7 – Exemple d’une permutation non-destructive entre deux tâches.

l’affectation des flacons des produits les plus coûteux. La recherche locale est une recherche locale dite à première amélioration, i.e. la première permutation non-destructive possible trouvée est appliquée. De plus, la mise en place d’une liste tabou est nécessaire pour éviter de tomber dans une boucle infinie, car toute permutation non-destructive n’induit pas forcément une diminution du coût total. Chaque tâche peut voir son flacon d’affecté changé qu’une seule fois. Le critère d’arrêt de la recherche locale est atteint lorsque plus aucune permutation non-destructive possible n’est trouvée en respectant la liste tabou.

4.5 Expérimentations numériques

Nous présentons dans cette section des expérimentations numériques de notre méthode matheuristique et de notre borne inférieure basée sur le calcul de CSP. La matheuristique a été codée en Python 3.7, la résolution des différents modèles linéaires est réalisée par le solveur Gurobi et les expérimentations ont été réalisées sur un MacBook Air (M1, 2020) 16 Go. Pour la résolution des BPCs, des limites de temps de calcul ont été fixées. Soit UB représente la valeur de la fonction objectif de la solution construite par l’heuristique "worst-fit" et UB_l la part de la valeur de la fonction objectif concernant le produit l . De la même manière, on pose $CSPLB$ la valeur de la borne inférieure calculée par CSPs et $CSPLB_l$. On définit CPU le temps de calcul total accordé à l’ensemble des instances

de BPCs. Alors le temps de calcul limite CPU_l pour l'ensemble des instances de BPCs concernant le produit l (ensembles des composantes connexes du graphe de compatibilité des tâches utilisant le produit l) :

$$CPU_l = CPU \cdot \frac{UB_l - CSPLB_l}{UB - CSPLB} \quad (4.25)$$

Cette répartition offre plus de temps de résolution là où la marge d'amélioration est la plus importante. De la même manière si G_l représente le graphe des compatibilités entre des tâches utilisant le produits l , le temps limite CPU_l est réparti entre les différentes composantes connexes de G_l en fonction du cardinal de ces derniers. Le temps limite cpu_g accordé à la composante connexe $g \subset G_l$ est alors défini de la façon suivante :

$$cpu_g = CPU_l \cdot \frac{|g|}{|G_l|} \quad (4.26)$$

La valeur CPU est fixée en fonction du nombre de semaines à planifier, 1 minute par semaine à planifier.

4.5.1 Génération des instances

Pour l'évaluation de notre matheuristique, nous avons généré 10 instances à partir des données correspondantes à l'UBCO. Chacune de ces instances est résolue avec différents nombres de tâches par jours $\{120, 160\}$ et différents nombres de semaines de planification $\{1, 2, 4, 8, 12\}$. Les différentes caractéristiques sont résumées dans le tableau 4.4. Au total, 10 versions de chacune des 10 instances sont résolues, soit un total de 100 instances. Les 35 produits présentés dans le tableau 4.5 et leurs caractéristiques ont été définis en rassemblant les données depuis différents documents (2011 et 2019). Les prix et les proportions peuvent alors être légèrement différents du cas réel en 2021.

Pour mieux se rendre compte de la répartition des 35 produits sélectionnés pour la génération d'instances, la figure 4.8 utilise une représentation en nuage de points regroupés en six ensembles avec leurs proportions d'utilisation respectives (en abscisse le temps de stabilité, en ordonnée le coût d'un flacon).

4.5. EXPÉRIMENTATIONS NUMÉRIQUES

Nombre de tâches par jour	{120, 160}		
Durée de stérilisation	$p^{Ste} = 15$ minutes		
Nombre d'isolateurs	$ I = 2$		
Nombre de machines par isolateur	$m_i = 2$		
Intervalles de disponibilités journaliers	machine	matinée	après-midi
	1	8h-13h	13h30-18h
	2	8h30-13h	13h30-17h
	3	8h-12h30	14h-17h
4	9h30-12h30	14h-16h30	
Nombre de jours de travail par semaine	5		
Nombre de semaines	{1, 2, 4, 8, 12} semaines		
Temps de préparation	$p_j^{Prep} \in \{5, 10, 15\}$ minutes		
Volume des flacons	$cap_l = 10$		
Quantité de produit requise par une tâche	$q_j \in \llbracket 1, 10 \rrbracket$		

TABLE 4.4 – Description des instances

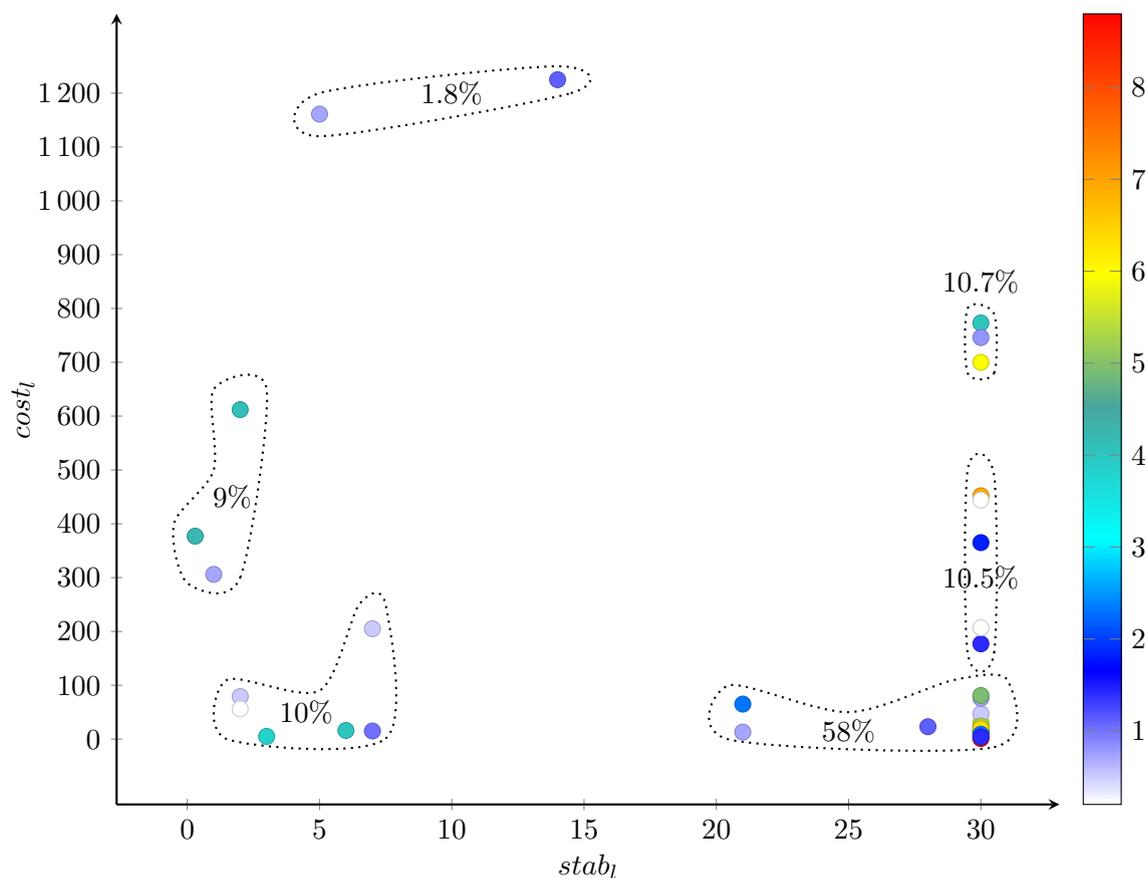


FIGURE 4.8 – Répartition d'une sélection de 35 produits utilisés par l'UBCO en fonction du coût unitaire d'un flacon et de la stabilité. La couleur représente le pourcentage de la production utilisant le produit.

4.5. EXPÉRIMENTATIONS NUMÉRIQUES

Produit l	Coût unitaire $cost_l$	Stabilité $stab_l$	Proportion $\frac{ J_l }{ J }\%$
1	377	8 heures	4.2
2	700	30 jours	5.9
3	612	2 jours	4.1
4	1161	5 jours	0.7
5	773	30 jours	4.0
6	452	30 jours	6.9
7	746	30 jours	0.8
8	306	24 heures	0.7
9	365	30 jours	1.8
10	1225	14 jours	1.1
11	177	30 jours	1.4
12	444	30 jours	0.2
13	23	30 jours	5.7
14	65	21 jours	2.3
15	31	30 jours	5.7
16	16	6 jours	4.0
17	47	30 jours	0.5
18	205	7 jours	0.5
19	23	28 jours	1.1
20	207	30 jours	0.2
21	76	30 jours	0.7
22	14	30 jours	5.2
23	81	30 jours	4.9
24	8	30 jours	5.4
25	79	2 jours	0.5
26	56	2 jours	0.2
27	13	21 jours	0.7
28	24	30 jours	5.3
29	15	30 jours	1.7
30	19	30 jours	6.3
31	9	30 jours	2.3
32	5	3 jours	3.8
33	1	30 jours	8.8
34	15	7 jours	1.0
35	4	30 jours	1.4

TABLE 4.5 – Caractéristiques des 35 produits des instances

4.5.2 Limites du modèle MILP et choix de la borne inférieure

Le MILP présenté en section 4.2 a été testé sur des petites instances générées comme présenté précédemment, mais où la planification était de seulement quelques jours. Avec un temps limite de calcul fixé à 10 minutes, le modèle MILP n'arrive pas à résoudre à l'optimal des instances de plus de 300 tâches (planification de 2 jours avec 120 tâches par jour).

Comme présenté dans la section 4.3, pour les dimensions sélectionnées ($|J_l|$, cap_l), la borne inférieure calculée par la résolution de CSPs est meilleure que celle calculée à partir de la demande et beaucoup plus rapide à calculer que celle utilisant la résolution de BPPs. Nous utiliserons donc la borne inférieure CSPLB qui est calculée par la résolution de CSPs.

4.5.3 Qualité de la matheuristique

Chacune des 100 instances a été résolue par la matheuristique deux fois, une fois en initialisant l'ordonnancement par l'heuristique "greedy" et une autre fois par l'heuristique "random". On pose comme coût moyen d'une tâche, le coût total divisé par le nombre de tâches de l'instance. La figure 4.9 représente le coût moyen d'une tâche induit par la solution heuristique produite par "worst-fit", celui induit par la solution produite à la fin des résolutions des différents BPCs, celui correspondant à la solution retournée à la fin d'exécution de la matheuristique et celui calculé par la borne inférieure basée sur la résolution de CSPs.

On définit le coût moyen d'une tâche comme le coût total (valeur de la fonction objectif) divisé par le nombre de tâches de l'instance. La raison de représenter le coût moyen plutôt que le coût total est de pouvoir avoir sur la même échelle les planifications d'une à douze semaines et d'évaluer la qualité de la borne inférieure en fonction du nombre de semaines planifiées. On peut alors s'apercevoir que la borne inférieure du coût moyen est très fortement impactée par la durée de planification. Cela peut s'expliquer par deux facteurs, tout d'abord car l'augmentation du nombre de tâches dilue la perte inévitable, mais aussi car le rapport perte inévitable pour quantité demandée se réduit par l'augmentation des combinaisons possibles. Il est tout de suite remarquable que le choix de l'heuristique d'initialisation de l'ordonnancement est très important pour la qualité de la solution issue des résolutions des BPCs. En effet, les résultats sont beaucoup plus proches de la borne inférieure CSPLB lorsque l'heuristique "greedy" est utilisée. On peut en conclure que l'ordonnancement a un impact très important sur le coût de production. Grâce aux permutations non-destructives, la matheuristique arrive à obtenir des résultats de qualité proches de la borne inférieure CSPLB avec un planning initial aléatoire. Cependant, on peut constater les limites d'efficacité des permutations non-destructives en particulier sur les instances avec peu de tâches par jour (120). De même, l'apport des permutations non-

4.5. EXPÉRIMENTATIONS NUMÉRIQUES

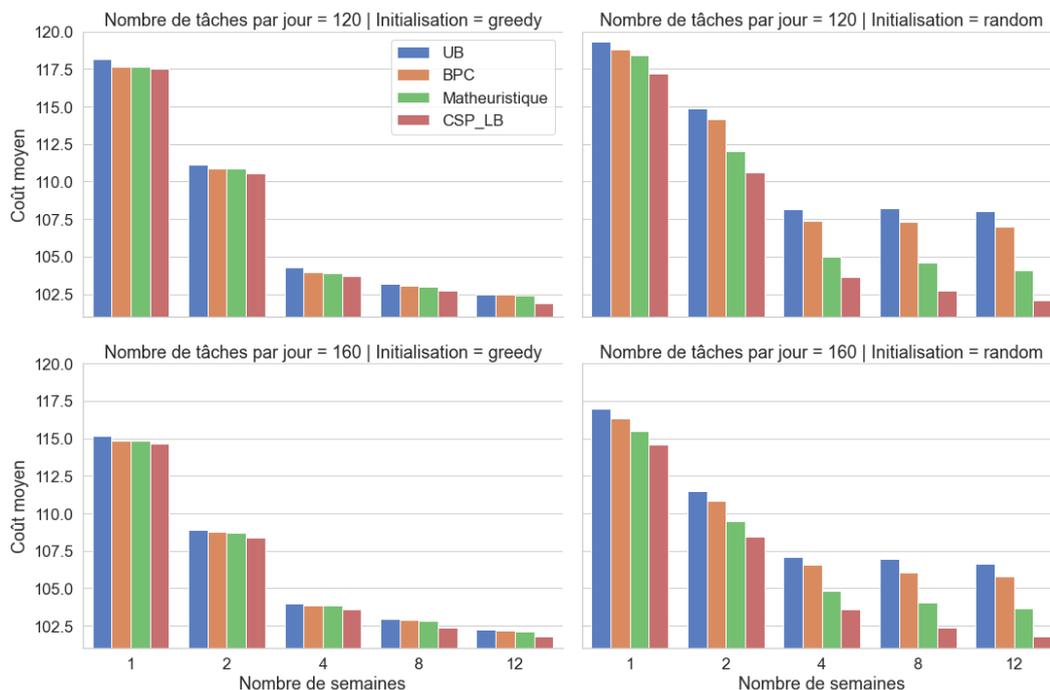


FIGURE 4.9 – Coût moyen d'une tâche en fonction du nombre de semaines à planifier et de la méthode d'initialisation de l'ordonnancement.

destructives sur un ordonnancement construit par le "greedy" est très faible principalement car la solution à la sortie des résolutions des BPCs est presque optimale.

4.5.4 Nombre de permutations non-destructives et temps de calcul

Pour mieux évaluer l'impact des permutations non-destructives, la figure 4.10 montre des boîtes à moustaches du nombre de permutations non-destructives effectuées par la matheuristique en fonction du nombre de semaines planifiées, du nombre de tâches par jour et de la méthode d'initialisation du planning. Le nombre de permutations non-destructives effectuées est beaucoup plus important (plus du triple) lorsque le planning de production initial est aléatoire que lorsque l'heuristique "greedy" est utilisée. Ceci démontre encore que le planning de production impacte fortement le coût de production.

Afin d'avoir un meilleur aperçu, la figure 4.11 représente la convergence de la matheuristique sur l'instance ayant requis le plus de permutations non-destructives. On peut voir que malgré les améliorations apportées par les permutations non-destructives, la solution construite par les deux heuristiques "greedy" et "worst-fit" reste meilleure que la solution dont l'initialisation du planning de production est aléatoire.

Le temps de calcul dépend de la difficulté des BPCs à résoudre et du nombre de permu-

4.5. EXPÉRIMENTATIONS NUMÉRIQUES

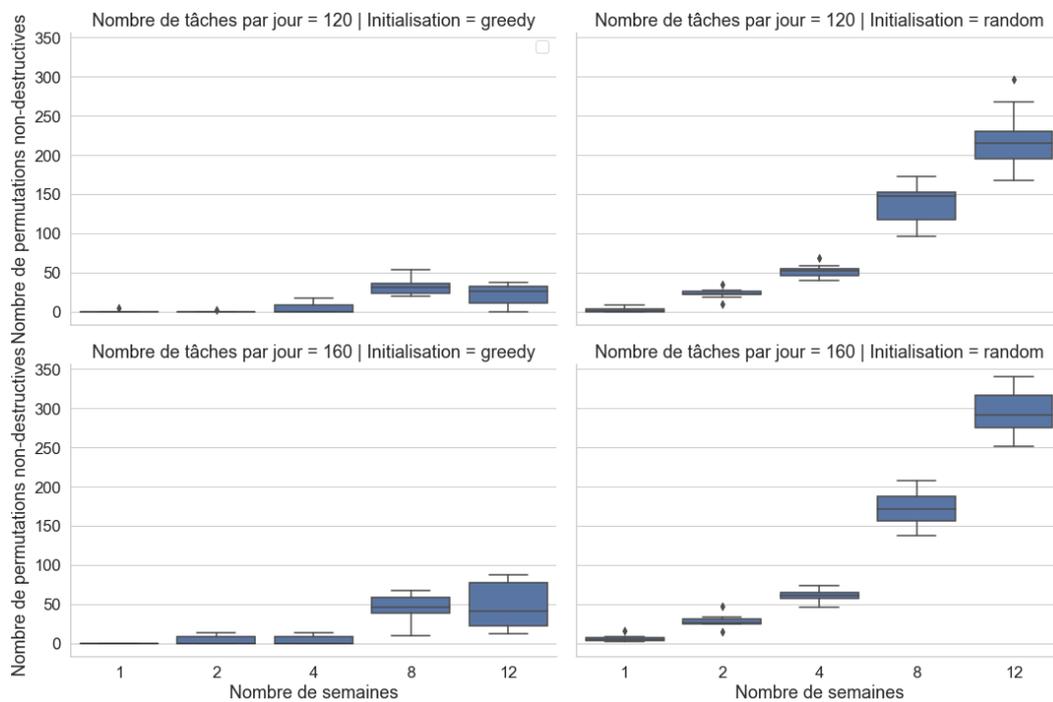


FIGURE 4.10 – Nombre de permutations non-destructives

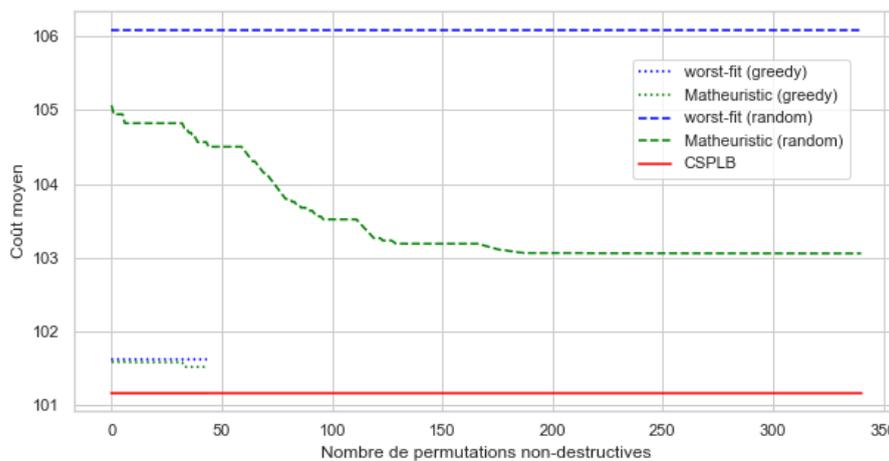


FIGURE 4.11 – Convergence de la matheuristique pour une instance avec 160 tâches par jour et une planification de 12 semaines en fonction de l'initialisation du planning utilisé.

tations non-destructives. La figure 4.12 montre des boîtes à moustaches du temps de calcul de la matheuristique en fonction du nombre de tâches par jour et du nombre de semaines planifiées. Malgré le plus grand nombre de permutations non-destructives, lorsque l'initia-

4.5. EXPÉRIMENTATIONS NUMÉRIQUES

lisation du planning de production est aléatoire, le temps de calcul est légèrement plus important lorsque l'heuristique "greedy" est utilisée. La raison est que les permutations non-destructives requièrent un très faible temps de calcul et que la majorité du temps de calcul de la matheuristique réside dans la résolution des BPCs. Il est alors possible que les graphes de conflits produits par un planning "greedy" soient plus difficiles à traiter. La méthode permet de planifier 3 mois en environ 5 minutes de calcul avec un écart à la borne inférieure de moins de 1%.

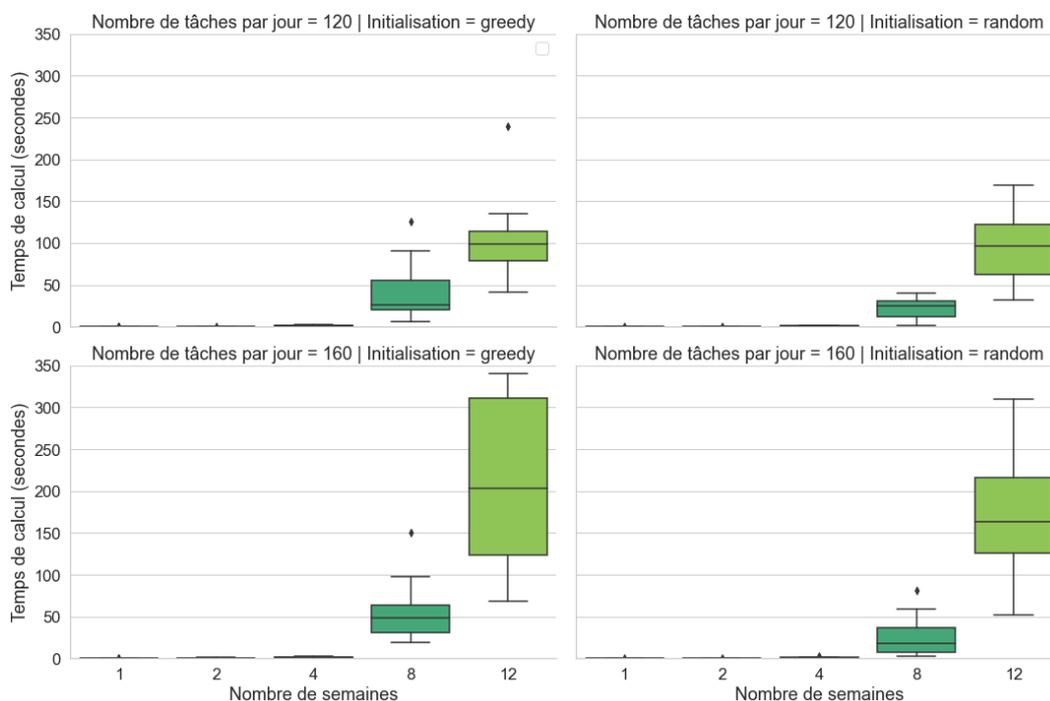


FIGURE 4.12 – Temps de calcul en fonction des dimensions de l'instance.

Le tableau 4.6 résume les Gaps moyen et temps de calculs moyen de la matheuristique avant la recherche locale et à la fin de la recherche locale en fonction de l'heuristique d'initialisation utilisé. On constate que malgré une nette différence de qualité entre l'initialisation greedy et l'initialisation random avant la recherche locale, cette dernière divise par 2 le Gap pour l'initialisation random. Cela montre que malgré les restrictions fortes pour les permutations non-destructives, celles-ci permettent une large optimisation. Cependant, les résultats de la matheuristique complète avec l'initialisation random reste nettement moins bon que la résolution des BPCs avec l'initialisation greedy. En terme du temps de calcul, le calcul des instances de BPCs représente en moyenne 73% du temps global de la matheuristique.

4.5. EXPÉRIMENTATIONS NUMÉRIQUES

Nombre de semaines	Nombre de tâches par jour	Initialisation	BPC		Matheuristique		
			Gap	CPU	Nombre de permutations	Gap	CPU
1	120	greedy	0.33	0.02	0.5	0.26	0.03
		random	1.38	0.04	2.6	0.92	0.05
	160	greedy	0.29	0.04	0	0.25	0.04
		random	1.55	0.05	6	0.77	0.07
2	120	greedy	0.27	0.08	0.2	0.23	0.08
		random	3.27	0.12	23.1	1.27	0.19
	160	greedy	0.28	0.26	4	0.23	0.35
		random	2.20	0.12	28.1	0.93	0.20
4	120	greedy	0.27	0.94	4.2	0.24	1.10
		random	3.56	0.62	51.3	1.30	0.82
	160	greedy	0.27	1.06	4	0.26	1.22
		random	2.92	0.82	60.8	1.20	1.09
8	120	greedy	0.30	10.49	32.5	0.27	42.44
		random	4.46	21.37	137.4	1.83	22.41
	160	greedy	0.45	17.97	43.5	0.41	57.73
		random	3.60	24.51	171.8	1.61	26.16
12	120	greedy	0.39	37.54	21.8	0.38	106.63
		random	4.83	90.65	219.5	2.01	93.97
	160	greedy	0.39	56.37	46.9	0.35	244.19
		random	3.97	167.16	295	1.85	173.70

TABLE 4.6 – Résultats agrégés : Gap moyen par rapport à CSPLB et temps de calcul moyen, en fonction du nombre de semaines planifiées, du nombre de tâches par jour et de l’heuristique d’initialisation.

4.6 Conclusions

Dans ce chapitre, nous avons proposé un modèle linéaire en nombres entiers mixtes et une matheuristique pour minimiser le coût de production des médicaments de chimiothérapie induit par le gaspillage de ressources périssables. Nous avons modélisé la production comme un problème d’ordonnancement à machines parallèles regroupées en isolateurs avec des périodes d’indisponibilités. La gestion des ressources consommables est quant à elle présentée comme un problème de Bin-packing avec conflits. Plusieurs bornes inférieures ont été développées, l’une basée sur la satisfaction de la demande, une seconde sur la résolution d’instances du problème de Bin-packing et une dernière sur la résolution d’instances du problème de Cutting-stock. Une étude complémentaire propose un moyen de choisir quelle borne inférieure utiliser en fonction du nombre de tâches et des capacités des flacons. La matheuristique proposée est la combinaison d’une heuristique d’initialisation, de la résolution d’instances du problème de Bin-packing avec conflits et d’une recherche locale avec liste tabou. L’efficacité de l’heuristique a été évaluée sur des instances générées grâce au regroupement d’informations fournies par l’UBCO. La taille des instances varie d’une à 12 semaines. La matheuristique a montré des résultats très concluants (écart à la borne inférieur à 1%) pour la planification à long terme de la production et de la gestion des reliquats avec un temps de calcul raisonnable par rapport à la grande taille des instances utilisées.

Les travaux et les résultats menés dans ce chapitre ont donné lieu à des publications dans deux conférences internationales [Robbes *et al.*, , Robbes *et al.*, 2020b] et une nationale [Robbes *et al.*, 2020a].

Les instances étudiées étaient composées de nombreux produits. Cette grande diversité a permis de résoudre des instances du problème de Bin-packing avec conflits de taille raisonnable (moins de 1000 tâches). Il serait difficile de résoudre des instances avec peu de produits sur un horizon de planification aussi long. L’une des perspectives envisagées est donc dans un premier temps d’améliorer la borne inférieure du problème de Bin-packing avec conflits présentée par [Khanafar *et al.*, 2010] pour déterminer plus précisément les résolutions inutiles. Dans un second temps le développement d’un algorithme par séparation et coupe comme présenté par [Sadykov et Vanderbeck, 2013] permettrait alors d’envisager des instances avec plusieurs milliers de tâches. Un autre point d’amélioration est dans la génération du planning de production. En effet, actuellement, aucune solution avec un temps d’attente n’est testée par la matheuristique. Or, lorsque le nombre de tâches par jour est suffisamment faible par rapport à la capacité de production, ces différents temps d’attente permettraient sûrement d’atteindre la borne inférieure plus régulièrement.

Chapitre 5

Conclusion et perspectives de recherches

Sommaire

5.1	Problème intégré de production et de livraison	158
5.2	Problème de livraison	159
5.3	Problème intégré production et de gestion des ressources consommables	160
5.4	Publications scientifiques	161

Les travaux que nous avons présentés dans ce manuscrit traitent des différentes problématiques rencontrées par une unité de bio-pharmacie (UBCO) au CHU de Tours pour la production et la livraison de médicaments de chimiothérapie. Chacun des problèmes étudiés a été modélisé et généralisé de sorte à proposer des solutions qui répondent aux exigences de l'UBCO et qui peuvent aussi s'appliquer à d'autres contextes. Nous présentons successivement les résultats de recherches abouties de chacune des contributions, suivi de travaux préliminaires ainsi qu'une liste de nombreuses perspectives.

Grâce à la collaboration avec l'UBCO, nous avons pu lister les contraintes métiers liées à la production de médicaments de chimiothérapie et les enjeux sanitaires et économiques rencontrés. La priorité de l'UBCO est l'amélioration de la qualité de service. L'objectif principal est alors de livrer avec le moins de retards possibles les médicaments aux différents services d'oncologie. Cependant, pour l'UBCO, l'augmentation importante de la demande en médicaments depuis ces dix dernières années et les difficultés dues à la livraison à trois hôpitaux rend la tâche plus complexe. Ainsi, nous avons développé une simulation des flux de production qui nous a permis d'identifier les goulets d'étranglement de la production et les étapes sensibles.

5.1 Problème intégré de production et de livraison

Afin de pouvoir optimiser la qualité de service, nous avons posé comme objectif la minimisation du retard total. Nous avons modélisé le problème de production et de livraison de médicaments de chimiothérapie comme un problème intégré d'ordonnancement et de tournées de véhicules. La partie production correspond à un problème d'ordonnancement d'ateliers de type flow-shop hybride. La partie livraison quand à elle correspond à un problème de tournées de véhicules multi-trajets. La complexité d'un tel problème intégré nous a mené au développement de trois heuristiques qui ont été évaluées sur des instances basées sur des données de l'UBCO ainsi que des instances générées aléatoirement. Les résultats obtenus nous ont permis de sélectionner l'heuristique la plus adaptée pour répondre aux enjeux de l'UBCO et de proposer son implémentation à l'entreprise ETICSYS pour l'UBCO.

Le travail de ce premier chapitre nous a conduits à introduire un deuxième mode de livraison basé sur des horaires négociés avec un service externe. Cette nouvelle variante du problème intégré de production et de livraison considère deux modes de livraison et a fait l'objet d'un stage de master. Durant ce stage, l'étudiante a développé une matheuristique basée sur la génération heuristique d'un grand nombre de tournées, l'utilisation d'un modèle mathématique pour la sélection de tournées et l'affectation des médicaments à ces dernières et la modification du planning de production à l'aide d'un opérateur de voisinage. Les

premiers résultats ont été présentés dans [Castan *et al.*, 2021].

Une autre perspective se porte sur l'amélioration de différentes heuristiques proposées dans le chapitre 2 par l'ajout d'opérateur de voisinage comme le 2-Opt pour la construction des itinéraires de livraison. De même, des opérateurs de voisinage pour chacune des étapes de production pourraient sensiblement améliorer la qualité des heuristiques.

5.2 Problème de livraison

Afin de pouvoir évaluer avec précision la qualité d'un planning de production des médicaments de chimiothérapie, nous avons développé une méthode exacte pour résoudre la partie livraison du problème. Le problème de livraison considéré est une variante du problème de tournées de véhicules multi-trajets avec des dates de départ au plus tôt, et comme fonction objectif la minimisation du retard total tout en considérant l'hypothèse que la constitution des trajets est connue et l'ordre des trajets est défini par les dates de fin de productions. Nous avons décidé de décomposer le problème en trois niveaux de décisions et de modéliser chaque niveau comme un sous-problème : le premier niveau de décision considéré est l'affectation des trajets aux véhicules, le deuxième niveau est la sélection des séquences de livraison pour les trajets et le dernier niveau est la génération de l'ensemble des séquences de livraison non-dominées pour chacun des trajets. Pour résoudre ce problème, la méthode exacte que nous avons proposée est basée sur la combinaison de trois méthodes par séparation et évaluation. Chacune d'entre elles a été enrichie par des bornes inférieures, des bornes supérieures et des méthodes de d'élimination de symétrie ou des propriétés de dominance. Cette méthode exacte a été évaluée et comparée à un modèle mathématique sur des instances de la littérature des problèmes de tournées de véhicules. Les résultats obtenus ont montré l'efficacité de notre méthode exacte en particulier sur les instances de grandes tailles et se rapprochant du contexte de l'UBCO.

Les perspectives de cette contribution résident principalement dans la réduction du temps de calcul de la méthode exacte proposée et du développement d'une version adaptée à des variantes du problème. Pour réduire le temps de calcul de notre méthode exacte, nous devons étudier les possibles propriétés de conservation de la non-dominance des séquences de livraison en fonction des dates de départ. L'hypothèse de la connaissance des compositions des trajets est une hypothèse forte. Il serait donc intéressant d'adapter notre méthode sur un problème plus général où les compositions des trajets sont à déterminer. Pour cela, une méthode de type génération de colonnes combinée à notre méthode exacte pourrait être considérée.

5.3 Problème intégré production et de gestion des ressources consommables

L'une des problématiques secondaires importantes de l'UBCO est de réduire son coût de production. Nous avons alors posé comme nouveau critère d'optimisation la minimisation du coût engendré par les produits utilisés pour la préparation des médicaments de chimiothérapie. En effet, pour préparer un médicament de chimiothérapie, des flacons de produits cytotoxiques sont utilisés par les pharmaciens. Ces flacons ont un coût pouvant être important, jusqu'à 1200 euros le flacons de quelques millilitres. Or, les produits contenus par ces flacons ont des durées de vie de quelques jours après ouverture. L'UBCO souhaite réduire les pertes potentielles des produits par une mauvaise gestion des fonds de flacons, dénommés "reliquats".

Nous avons modélisé le problème de gestion des reliquats comme un problème intégré de production et de découpe où la partie production est représentée par un problème d'ordonnancement à machines parallèles et la partie découpe est un problème de Bin-packing avec des contraintes de conflits. Pour résoudre ce problème, nous avons développé une matheuristique et des bornes inférieures pour en évaluer sa qualité. Au total, nous avons proposé trois bornes inférieures avec des temps de calculs différents en fonction des dimensions des instances. La matheuristique proposée est construite sur la génération d'un planning de production heuristique initial, la résolution du problème de Bin-packing avec conflits pour chaque produit et une recherche locale avec liste tabou dont l'opérateur de voisinage sur le planning de production permet l'amélioration de la découpe sans recalcul de cette dernière. Les instances utilisées pour l'évaluation de notre méthode matheuristique ont été construites sur des données fournies par l'UBCO : stabilités, coût et proportion d'utilisation de 35 produits. Les résultats obtenus ont montré la qualité de la matheuristique proposée et l'impact du planning de production sur le coût de production.

Une première extension de ce travail a été explorée par la modélisation d'un problème bi-niveau où deux objectifs (minimiser la somme des retards et minimiser le coût de production) sont en compétition. Le problème est représenté par deux agents, le premier définit le planning de production avec pour objectif la minimisation de la somme des retards, le second doit définir les dates de fin de production souhaitées afin de minimiser le coût de production induit par le planning construit par le premier agent. Des premiers résultats ont été présentés dans [Robbes *et al.*, 2020b].

Diverses perspectives s'ouvrent à partir de ce dernier travail de recherche. La méthode matheuristique proposée peut résoudre des instances de 3 mois mais cela est proche de ses limites de performance. De même, sur des instances avec moins de produits, la méthode pourrait être limitée à la planification sur 1 mois. En effet, la résolution des problèmes de

Bin-packing avec conflits est très coûteuse en temps de calcul lorsque le nombre de tâches à affecter à des flacons augmente. L'utilisation d'une méthode de branch-and-price pour la résolution des problèmes de Bin-packing avec conflits permettrait de résoudre des instances plus grandes. Avec l'accélération de calcul possible par ce type de méthode, de nouveaux opérateurs de voisinage moins restrictifs pourraient être testés en autorisant le recalcul de l'affectation des flacons.

La matheuristique proposée fixe un premier planning de production pour avoir une affectation des flacons et enfin une modification du planning. Il serait intéressant de développer une méthode dans le sens inverse où une affectation idéale serait construite à partir de la résolution de problèmes de Cutting-stock défini comme pour le calcul de la borne inférieure suivie par l'utilisation de la programmation par contrainte pour la planification des tâches.

5.4 Publications scientifiques

Les différentes publications liées aux chapitres 2, 3 et 4 sont résumées dans le tableau 5.1 :

Type de publication	Chapitre 2	Chapitre 3	Chapitre 4
Conférence nationale	[Castan <i>et al.</i> , 2021]	[Robbes <i>et al.</i> , 2019a]	[Robbes <i>et al.</i> , 2020a]
Conférence internationale	[Robbes <i>et al.</i> , 2019b], [Robbes <i>et al.</i> , 2019c]	[Robbes <i>et al.</i> , 2020b]	[Robbes <i>et al.</i> , 2020b]
Revue internationale	[Robbes <i>et al.</i> , 2021a]		

TABLE 5.1 – Publications liées aux différents chapitres.

Bibliographie

- [Ahmed *et al.*, 2011] AHMED, Z., ELMEKKAWY, T. et BATES, S. (2011). Developing an efficient scheduling template of a chemotherapy treatment unit : A case study. *The Australasian medical journal*, 4(10):575.
- [Amorim *et al.*, 2013] AMORIM, P., BELO-FILHO, M., TOLEDO, F., ALMEDER, C. et ALMADA-LOBO, B. (2013). Lot sizing versus batching in the production and distribution planning of perishable goods. *International Journal of Production Economics*, 146(1):208–218.
- [Arbib et Marinelli, 2014] ARBIB, C. et MARINELLI, F. (2014). On cutting stock with due dates. *Omega*, 46:11–20.
- [Archetti *et al.*, 2015] ARCHETTI, C., JABALI, O. et SPERANZA, M. G. (2015). Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122–134.
- [Azi *et al.*, 2010] AZI, N., GENDREAU, M. et POTVIN, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3):756–763.
- [Baptiste, 2000] BAPTISTE, P. (2000). Scheduling equal-length jobs on identical parallel machines. *Discrete Applied Mathematics*, 103(1-3):21–32.
- [Belo-Filho *et al.*, 2015] BELO-FILHO, M., AMORIM, P. et ALMADA-LOBO, B. (2015). An adaptive large neighbourhood search for the operational integrated production and distribution problem of perishable products. *International Journal of Production Research*, 53(20):6040–6058.
- [Billaut *et al.*, 2015] BILLAUT, J.-C., DELLA CROCE, F. et GROSSO, A. (2015). A single machine scheduling problem with two-dimensional vector packing constraints. *European Journal of Operational Research*, 243(1):75–81.
- [Boysen *et al.*, 2019] BOYSEN, N., DE KOSTER, R. et WEIDINGER, F. (2019). Warehousing in the e-commerce era : A survey. *European Journal of Operational Research*, 277(2):396–411.

- [Braga *et al.*, 2016] BRAGA, N., ALVES, C., MACEDO, R. et CARVALHO, J. V. D. (2016). Combined cutting stock and scheduling : a matheuristic approach. *International Journal of Innovative Computing and Applications*, 7(3):135–146.
- [Braga *et al.*, 2015] BRAGA, N., ALVES, C., MACEDO, R. et de CARVALHO, J. V. (2015). A model-based heuristic for the combined cutting stock and scheduling problem. *In International Conference on Computational Science and Its Applications*, pages 490–505. Springer.
- [Brucker et Kravchenko, 2005] BRUCKER, P. et KRAVCHENKO, S. A. (2005). *Scheduling jobs with release times on parallel machines to minimize total tardiness*. Universität Osnabrück. Fachbereich Mathematik/Informatik.
- [Bujtás *et al.*, 2011] BUJTÁS, C., DÓSA, G., IMREH, C., NAGY-GYÖRGY, J. et TUZA, Z. (2011). The graph-bin packing problem. *International Journal of Foundations of Computer Science*, 22(08):1971–1993.
- [Castan *et al.*, 2021] CASTAN, J., ROBBES, A., KERGOSIEN, Y. et JEAN-CHARLES, B. (2021). Un problème intégré d’ordonnancement et de distribution avec deux modes de livraison. *In Recherche Opérationnelle et d’Aide à la Décision (ROADEF 2021)*.
- [Cattaruzza *et al.*, 2016] CATTARUZZA, D., ABSI, N. et FEILLET, D. (2016). The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, 50(2):676–693.
- [Cattaruzza *et al.*, 2018] CATTARUZZA, D., ABSI, N. et FEILLET, D. (2018). Vehicle routing problems with multiple trips. *Annals of Operations Research*, 271(1):127–159.
- [Chahed *et al.*, 2009] CHAHED, S., MARCON, E., SAHIN, E., FEILLET, D. et DALLERY, Y. (2009). Exploring new operational research opportunities within the home care context : the chemotherapy at home. *Health Care Management Science*, 12(2):179–191.
- [Chang et Lee, 2004] CHANG, Y.-C. et LEE, C.-Y. (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research*, 158(2):470–487.
- [Chang *et al.*, 2014] CHANG, Y.-C., LI, V. C. et CHIANG, C.-J. (2014). An ant colony optimization heuristic for an integrated production and distribution scheduling problem. *Engineering Optimization*, 46(4):503–520.
- [Chekoubi *et al.*, 2018] CHEKOUBI, Z., TRABELSI, W. et SAUER, N. (2018). The integrated production-inventory-routing problem in the context of reverse logistics : The case of collecting and remanufacturing of eol products. *In 2018 4th International Conference on Optimization and Applications (ICOA)*, pages 1–6. IEEE.
- [Chen *et al.*, 2009] CHEN, H.-K., HSUEH, C.-F. et CHANG, M.-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & operations research*, 36(7):2311–2319.

- [Chen, 2010] CHEN, Z.-L. (2010). Integrated production and outbound distribution scheduling : review and extensions. *Operations research*, 58(1):130–148.
- [Chen et Vairaktarakis, 2005] CHEN, Z.-L. et VAIRAKTARAKIS, G. L. (2005). Integrated scheduling of production and distribution operations. *Management Science*, 51(4):614–628.
- [Cheref *et al.*, 2016a] CHEREF, A., ARTIGUES, C. et BILLAUT, J.-C. (2016a). Online recoverable robustness based on groups of permutable jobs for integrated production scheduling and delivery routing.
- [Cheref *et al.*, 2016b] CHEREF, A., ARTIGUES, C., BILLAUT, J.-C. et NGUEVEU, S. U. (2016b). Integrated production scheduling and delivery routing : complexity results and column generation. *In International Symposium on Combinatorial Optimization*, pages 439–450. Springer.
- [Chiang *et al.*, 2009] CHIANG, W.-C., RUSSELL, R., XU, X. et ZEPEDA, D. (2009). A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. *International Journal of Production Economics*, 121(2):752–767.
- [Chu, 1992] CHU, C. (1992). A branch-and-bound algorithm to minimize total tardiness with different release dates. *Naval Research Logistics (NRL)*, 39(2):265–283.
- [De Koster *et al.*, 2007] DE KOSTER, R., LE-DUC, T. et ROODBERGEN, K. J. (2007). Design and control of warehouse order picking : A literature review. *European journal of operational research*, 182(2):481–501.
- [Defossez *et al.*, 2019] DEFOSSEZ, G., LE GUYADER-PEYROU, S., UHRY, Z., GROS-CLAUDE, P., COLONNA, M., DANTONY, E., DELAFOSSE, P., MOLINIÉ, F., WORONOFF, A.-S., BOUVIER, A.-M. *et al.* (2019). Estimations nationales de l’incidence et de la mortalité par cancer en France métropolitaine entre 1990 et 2018. *Etude à partir des registres des cancers du réseau Francim. Résultats préliminaires. Synthèse. Saint-Maurice (Fra) : Santé Publique France*, page 19.
- [Devapriya *et al.*, 2017] DEVAPRIYA, P., FERRELL, W. et GEISMAR, N. (2017). Integrated production and distribution scheduling with a perishable product. *European Journal of Operational Research*, 259(3):906–916.
- [Dobish, 2003] DOBISH, R. (2003). Next-day chemotherapy scheduling : a multidisciplinary approach to solving workload issues in a tertiary oncology center. *Journal of Oncology Pharmacy Practice*, 9(1):37–42.
- [Du et Leung, 1990] DU, J. et LEUNG, J. Y.-T. (1990). Minimizing total tardiness on one machine is np-hard. *Mathematics of operations research*, 15(3):483–495.

- [Elhedhli *et al.*, 2011] ELHEDHLI, S., LI, L., GZARA, M. et NAOUM-SAWAYA, J. (2011). A branch-and-price algorithm for the bin packing problem with conflicts. *INFORMS Journal on Computing*, 23(3):404–415.
- [Fachini et Armentano, 2020] FACHINI, R. F. et ARMENTANO, V. A. (2020). Exact and heuristic dynamic programming algorithms for the traveling salesman problem with flexible time windows. *Optimization Letters*, 14(3):579–609.
- [Farahani *et al.*, 2012] FARAHANI, P., GRUNOW, M. et GÜNTHER, H.-O. (2012). Integrated production and distribution planning for perishable food products. *Flexible services and manufacturing journal*, 24(1):28–51.
- [Fu *et al.*, 2017] FU, L.-L., ALOULOU, M. A. et TRIKI, C. (2017). Integrated production scheduling and vehicle routing problem with job splitting and delivery time windows. *International Journal of Production Research*, 55(20):5942–5957.
- [Ganji *et al.*, 2020] GANJI, M., KAZEMIPOOR, H., MOLANA, S. M. H. et SAJADI, S. M. (2020). A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. *Journal of Cleaner Production*, page 120824.
- [Garaix *et al.*, 2020] GARAIX, T., ROSTAMI, S. et XIE, X. (2020). Daily outpatient chemotherapy appointment scheduling with random deferrals. *Flexible Services and Manufacturing Journal*, 32(1):129–153.
- [Garcia *et al.*, 2004] GARCIA, J., LOZANO, S. et CANCA, D. (2004). Coordinated scheduling of production and delivery from multiple plants. *Robotics and Computer-Integrated Manufacturing*, 20(3):191–198.
- [Geismar *et al.*, 2008] GEISMAR, H. N., LAPORTE, G., LEI, L. et SRISKANDARAJAH, C. (2008). The integrated production and transportation scheduling problem for a product with a short lifespan. *INFORMS Journal on Computing*, 20(1):21–33.
- [Gendreau *et al.*, 2004] GENDREAU, M., LAPORTE, G. et SEMET, F. (2004). Heuristics and lower bounds for the bin packing problem with conflicts. *Computers & Operations Research*, 31(3):347–358.
- [Gharaei et Jolai, 2018] GHARAEI, A. et JOLAI, F. (2018). A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain. *Applied Soft Computing*, 65:577–589.
- [Gondran *et al.*, 1984] GONDRAN, M., MINOUX, M. et VAJDA, S. (1984). *Graphs and algorithms*. John Wiley & Sons, Inc.
- [Graham *et al.*, 1979] GRAHAM, R. L., LAWLER, E. L., LENSTRA, J. K. et KAN, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling : a survey. In *Annals of discrete mathematics*, volume 5, pages 287–326. Elsevier.

- [Hadj Salem et Kieffer, 2020] HADJ SALEM, K. et KIEFFER, Y. (2020). New symmetry-less ilp formulation for the classical one dimensional bin-packing problem. *In International Computing and Combinatorics Conference*, pages 423–434. Springer.
- [Hernandez et al., 2014] HERNANDEZ, F., FEILLET, D., GIROUDEAU, R. et NAUD, O. (2014). A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4or*, 12(3):235–259.
- [Hernandez et al., 2016] HERNANDEZ, F., FEILLET, D., GIROUDEAU, R. et NAUD, O. (2016). Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 249(2):551–559.
- [Hurter et Van Buer, 1996] HURTER, A. P. et VAN BUER, M. G. (1996). The newspaper production/distribution problem. *Journal of Business Logistics*, 17(1):85.
- [Jamili et al., 2016] JAMILI, N., RANJBAR, M. et SALARI, M. (2016). A bi-objective model for integrated scheduling of production and distribution in a supply chain with order release date restrictions. *Journal of Manufacturing Systems*, 40:105–118.
- [Jansen, 1999] JANSEN, K. (1999). An approximation scheme for bin packing with conflicts. *Journal of combinatorial optimization*, 3(4):363–377.
- [Jansen et Öhring, 1997] JANSEN, K. et ÖHRING, S. (1997). Approximation algorithms for time constrained scheduling. *Information and computation*, 132(2):85–108.
- [Karaođlan et Kesen, 2017] KARAOĐLAN, İ. et KESEN, S. E. (2017). The coordinated production and transportation scheduling problem with a time-sensitive product : a branch-and-cut algorithm. *International Journal of Production Research*, 55(2):536–557.
- [Karonsoontawong et al., 2020] KAROONSOONTAWONG, A., PUNYIM, P., NUEANGNITNARAPORN, W. et RATANAVARAHA, V. (2020). Multi-trip time-dependent vehicle routing problem with soft time windows and overtime constraints. *Networks and Spatial Economics*, pages 1–50.
- [Karp, 1972] KARP, R. M. (1972). Reducibility among combinatorial problems. *In Complexity of computer computations*, pages 85–103. Springer.
- [Kergosien et al., 2017] KERGO SIEN, Y., GENDREAU, M. et BILLAUT, J.-C. (2017). A benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints. *European Journal of Operational Research*, 262(1):287–298.
- [Kergosien et al., 2011] KERGO SIEN, Y., TOURNAMILLE, J.-F., LAURENCE, B. et BILLAUT, J.-C. (2011). Planning and tracking chemotherapy production for cancer treatment : A performing and integrated solution. *International journal of medical informatics*, 80(9):655–662.

- [Kesen et Bektaş, 2019] KESEN, S. E. et BEKTAŞ, T. (2019). Integrated production scheduling and distribution planning with time windows. *In Lean and Green Supply Chain Management*, pages 231–252. Springer.
- [Khanafar, 2010] KHANAFER, A. (2010). *Algorithmes pour des problèmes de bin-packing mono et multi-objectif*. Thèse de doctorat.
- [Khanafar et al., 2010] KHANAFER, A., CLAUTIAUX, F. et TALBI, E.-G. (2010). New lower bounds for bin packing problems with conflicts. *European journal of operational research*, 206(2):281–288.
- [Khanafar et al., 2012] KHANAFER, A., CLAUTIAUX, F. et TALBI, E.-G. (2012). Tree-decomposition based heuristics for the two-dimensional bin packing problem with conflicts. *Computers & Operations Research*, 39(1):54–63.
- [Lacomme et al., 2018] LACOMME, P., MOUKRIM, A., QUILLIOT, A. et VINOT, M. (2018). Supply chain optimisation with both production and transportation integration : multiple vehicles for a single perishable product. *International Journal of Production Research*, 56(12):4313–4336.
- [Larrain et al., 2019] LARRAIN, H., COELHO, L. C., ARCHETTI, C. et SPERANZA, M. G. (2019). Exact solution methods for the multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 110:148–158.
- [Lawler, 1977] LAWLER, E. L. (1977). A “pseudopolynomial” algorithm for sequencing jobs to minimize total tardiness. *In Annals of discrete Mathematics*, volume 1, pages 331–342. Elsevier.
- [Lee et al., 2014a] LEE, J., KIM, B.-I., JOHNSON, A. L. et LEE, K. (2014a). The nuclear medicine production and delivery problem. *European Journal of Operational Research*, 236(2):461–472.
- [Lee et al., 2014b] LEE, J., KIM, B.-I., JOHNSON, A. L. et LEE, K. (2014b). The nuclear medicine production and delivery problem. *European Journal of Operational Research*, 236(2):461–472.
- [Lenstra et al., 1977] LENSTRA, J. K., KAN, A. R. et BRUCKER, P. (1977). Complexity of machine scheduling problems. *In Annals of discrete mathematics*, volume 1, pages 343–362. Elsevier.
- [Lenstra et Rinnooy Kan, 1978] LENSTRA, J. K. et RINNOOY KAN, A. (1978). Complexity of scheduling under precedence constraints. *Operations Research*, 26(1):22–35.
- [Li et Vairaktarakis, 2007] LI, C.-L. et VAIRAKTARAKIS, G. (2007). Coordinating production and distribution of jobs with bundling operations. *IIE transactions*, 39(2):203–215.

- [Li *et al.*, 2005] LI, C.-L., VAIRAKTARAKIS, G. et LEE, C.-Y. (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research*, 164(1):39–51.
- [Li *et al.*, 2016] LI, K., ZHOU, C., LEUNG, J. Y. et MA, Y. (2016). Integrated production and delivery with single machine and multiple vehicles. *Expert Systems with Applications*, 57:12–20.
- [Li, 1996] LI, S. (1996). Multi-job cutting stock problem with due dates and release dates. *Journal of the Operational Research Society*, 47(4):490–510.
- [Li et Ferrell, 2011] LI, W. et FERRELL, W. G. (2011). Integrated production and distribution schedule problem with a perishable product. In *IIE Annual Conference. Proceedings*, page 1. Institute of Industrial and Systems Engineers (IISE).
- [Li et Zu, 2015] LI, W. et ZU, L. (2015). A heuristic method for the integrated production distribution problem. In *IIE Annual Conference. Proceedings*, page 3027. Institute of Industrial and Systems Engineers (IISE).
- [Liberatore *et al.*, 2011] LIBERATORE, F., RIGHINI, G. et SALANI, M. (2011). A column generation algorithm for the vehicle routing problem with soft time windows. *4OR*, 9(1):49–82.
- [Liu *et al.*, 2020] LIU, L., LI, W., LI, K. et ZOU, X. (2020). A coordinated production and transportation scheduling problem with minimum sum of order delivery times. *Journal of Heuristics*, 26(1):33–58.
- [Liu et Liu, 2020] LIU, L. et LIU, S. (2020). Integrated production and distribution problem of perishable products with a minimum total order weighted delivery time. *Mathematics*, 8(2):146.
- [Low *et al.*, 2017] LOW, C., CHANG, C.-M. et GAO, B.-Y. (2017). Integration of production scheduling and delivery in two echelon supply chain. *International Journal of Systems Science : Operations & Logistics*, 4(2):122–134.
- [Low *et al.*, 2014] LOW, C., CHANG, C.-M., LI, R.-K. et HUANG, C.-L. (2014). Coordination of production scheduling and delivery problems with heterogeneous fleet. *International Journal of Production Economics*, 153:139–148.
- [Low *et al.*, 2013] LOW, C., LI, R.-K. et CHANG, C.-M. (2013). Integrated scheduling of production and delivery with time windows. *International Journal of Production Research*, 51(3):897–909.
- [Macedo *et al.*, 2011] MACEDO, R., ALVES, C., de CARVALHO, J. V., CLAUTIAUX, F. et HANAFI, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, 214(3):536–545.

- [Marandi et Fatemi Ghomi, 2019] MARANDI, F. et FATEMI GHOMI, S. (2019). Integrated multi-factory production and distribution scheduling applying vehicle routing approach. *International Journal of Production Research*, 57(3):722–748.
- [Mazier et al., 2010] MAZIER, A., BILLAUT, J.-C. et TOURNAMILLE, J.-F. (2010). Scheduling preparation of doses for a chemotherapy service. *Annals of Operations Research*, 178(1):145–154.
- [Meinecke et Scholz-Reiter, 2014] MEINECKE, C. et SCHOLZ-REITER, B. (2014). A heuristic for the integrated production and distribution scheduling problem. *International Science Index*, 8(2):290–297.
- [Melega et al., 2018] MELEGA, G. M., de ARAUJO, S. A. et JANS, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research*, 271(1):1–19.
- [Miranda et al., 2018] MIRANDA, P. L., MORABITO, R. et FERREIRA, D. (2018). Optimization model for a production, inventory, distribution and routing problem in small furniture companies. *Top*, 26(1):30–67.
- [Miranda et al., 2019] MIRANDA, P. L., MORABITO, R. et FERREIRA, D. (2019). Mixed integer formulations for a coupled lot-scheduling and vehicle routing problem in furniture settings. *INFOR : Information Systems and Operational Research*, 57(4):563–596.
- [Mohammadi et al., 2020] MOHAMMADI, S., Al-e HASHEM, S. M. et REKIK, Y. (2020). An integrated production scheduling and delivery route planning with multi-purpose machines : A case study from a furniture manufacturing company. *International Journal of Production Economics*, 219:347–359.
- [Moons et al., 2017] MOONS, S., RAMAEKERS, K., CARIS, A. et ARDA, Y. (2017). Integrating production scheduling and vehicle routing decisions at the operational decision level : a review and discussion. *Computers & Industrial Engineering*, 104:224–245.
- [Mor et Speranza, 2020] MOR, A. et SPERANZA, M. G. (2020). Vehicle routing problems over time : a survey. *4OR*, pages 1–21.
- [Murtagh et Contreras, 2012] MURTAGH, F. et CONTRERAS, P. (2012). Algorithms for hierarchical clustering : an overview. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 2(1):86–97.
- [Naso et al., 2007] NASO, D., SURICO, M., TURCHIANO, B. et KAYMAK, U. (2007). Genetic algorithms for supply-chain scheduling : A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research*, 177(3):2069–2099.
- [Park et Hong, 2009] PARK, Y.-B. et HONG, S.-C. (2009). Integrated production and distribution planning for single-period inventory products. *International Journal of Computer Integrated Manufacturing*, 22(5):443–457.

- [Pitombeira-Neto et Prata, 2020] PITOMBEIRA-NETO, A. R. et PRATA, B. d. A. (2020). A matheuristic algorithm for the one-dimensional cutting stock and scheduling problem with heterogeneous orders. *Top*, 28(1):178–192.
- [Ramis et Warusfel,] RAMIS, J.-P. et WARUSFEL, A. *Mathématiques : Tout-en-un pour la Licence – niveau 1*. Dunod, coll. « Sciences Sup ».
- [Reinertsen et Vossen, 2010] REINERTSEN, H. et VOSSEN, T. W. (2010). The one-dimensional cutting stock problem with due dates. *European Journal of Operational Research*, 201(3):701–711.
- [Resende et Ribeiro, 2014] RESENDE, M. G. et RIBEIRO, C. C. (2014). Grasp : Greedy randomized adaptive search procedures. *In Search methodologies*, pages 287–312. Springer.
- [Robbes et al., 2021a] ROBBES, A., KERGOSIEN, Y., ANDRÉ, V. et BILLAUT, J.-C. (2021a). Efficient heuristics to minimize the total tardiness of chemotherapy drug production and delivery. *Flexible Services and Manufacturing Journal*, pages 1–36.
- [Robbes et al., 2019a] ROBBES, A., KERGOSIEN, Y. et BILLAUT, J.-C. (2019a). Méthode exacte pour un problème de livraison de chimiothérapie avec séquence de lots fixée. *In Recherche Opérationnelle et d’Aide à la Décision (ROADEF’19)*.
- [Robbes et al., 2019b] ROBBES, A., KERGOSIEN, Y. et BILLAUT, J.-C. (2019b). Multi-level heuristic to optimize the chemotherapy production and delivery. *In International Conference on Human-Centred Software Engineering*, pages 263–273. Springer.
- [Robbes et al., 2021b] ROBBES, A., KERGOSIEN, Y. et BILLAUT, J.-C. (2021b). An exact method for the multi-trip batch delivery problem. *In EURO 2021*.
- [Robbes et al., 2020a] ROBBES, A., KERGOSIEN, Y., BILLAUT, J.-C. et ANDRÉ, V. (2020a). Réduire le coût de production des médicaments de chimiothérapie par une gestion des reliquats. *In ROADEF 2020*.
- [Robbes et al.,] ROBBES, A., KERGOSIEN, Y. et JEAN-CHARLES, B. Minimizing the costs induced by perishable resource waste in a chemotherapy production unit. *In 17th International Workshop on Project Management and Scheduling*.
- [Robbes et al., 2019c] ROBBES, A., KERGOSIEN, Y. et JEAN-CHARLES, B. (2019c). Minimize the total tardiness of chemotherapy drugs production and delivery. *In Conference of the European Working Group on Operations Research Applied to Healthcare services (ORAHS 2019)*.
- [Robbes et al., 2020b] ROBBES, A., KERGOSIEN, Y. et JEAN-CHARLES, B. (2020b). Chemotherapy production bilevel problem : Cost versus delay. *In 46th International Conference on Operational Research Applied to Health Services*.

- [Rosenkrantz *et al.*, 1977] ROSENKRANTZ, D. J., STEARNS, R. E. et LEWIS, II, P. M. (1977). An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, 6(3):563–581.
- [Ruder, 2016] RUDER, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv :1609.04747*.
- [Ruiz et Vázquez-Rodríguez, 2010] RUIZ, R. et VÁZQUEZ-RODRÍGUEZ, J. A. (2010). The hybrid flow shop scheduling problem. *European journal of operational research*, 205(1):1–18.
- [Russell *et al.*, 2008] RUSSELL, R., CHIANG, W.-C. et ZEPEDA, D. (2008). Integrating multi-product production and distribution in newspaper logistics. *Computers & Operations Research*, 35(5):1576–1588.
- [Sadki *et al.*, 2010] SADKI, A., XIE, X. et CHAUVIN, F. (2010). Patients assignment for an oncology outpatient unit. In *2010 IEEE International Conference on Automation Science and Engineering*, pages 891–896. IEEE.
- [Sadki *et al.*, 2011] SADKI, A., XIE, X. et CHAUVIN, F. (2011). Appointment scheduling of oncology outpatients. In *2011 IEEE International Conference on Automation Science and Engineering*, pages 513–518. IEEE.
- [Sadki *et al.*, 2013] SADKI, A., XIE, X. et CHAUVIN, F. (2013). Planning oncologists of ambulatory care units. *Decision Support Systems*, 55(2):640–649.
- [Sadykov et Vanderbeck, 2013] SADYKOV, R. et VANDERBECK, F. (2013). Bin packing with conflicts : a generic branch-and-price algorithm. *INFORMS Journal on Computing*, 25(2):244–255.
- [Sbeity et Younes, 2015] SBEITY, H. et YOUNES, R. (2015). Review of optimization methods for cancer chemotherapy treatment planning. *Journal of Computer Science & Systems Biology*, 8(2):74.
- [Scholz-Reiter *et al.*, 2011] SCHOLZ-REITER, B., MAKUSCHEWITZ, T., NOVAES, A. G., FRAZZON, E. M. et LIMA JR, O. F. (2011). An approach for the sustainable integration of production and transportation scheduling. *International Journal of Logistics Systems and Management*, 10(2):158–179.
- [Sexton et Choi, 1986] SEXTON, T. R. et CHOI, Y.-M. (1986). Pickup and delivery of partial loads with “soft” time windows. *American Journal of Mathematical and Management Sciences*, 6(3-4):369–398.
- [Shelbourne *et al.*, 2017] SHELBOURNE, B. C., BATTARRA, M. et POTTS, C. N. (2017). The vehicle routing problem with release and due dates. *INFORMS Journal on Computing*, 29(4):705–723.

- [Shi *et al.*, 2014] SHI, J., ALAGOZ, O., ERENAY, F. S. et SU, Q. (2014). A survey of optimization models on cancer chemotherapy treatment planning. *Annals of Operations Research*, 221(1):331–356.
- [Simons, 1983] SIMONS, B. (1983). Multiprocessor scheduling of unit-time jobs with arbitrary release times and deadlines. *SIAM Journal on Computing*, 12(2):294–299.
- [Snyman, 2005] SNYMAN, J. A. (2005). *Practical mathematical optimization*. Springer.
- [Solomon, 1987] SOLOMON, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265.
- [Stecke et Zhao, 2007] STECKE, K. E. et ZHAO, X. (2007). Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. *Manufacturing & Service Operations Management*, 9(2):206–224.
- [Tamannaie et Rasti-Barzoki, 2019] TAMANNAEI, M. et RASTI-BARZOKI, M. (2019). Mathematical programming and solution approaches for minimizing tardiness and transportation costs in the supply chain scheduling problem. *Computers & Industrial Engineering*, 127:643–656.
- [Tavares-Neto et Nagano, 2019] TAVARES-NETO, R. F. et NAGANO, M. S. (2019). An iterated greedy approach to integrate production by multiple parallel machines and distribution by a single capacitated vehicle. *Swarm and evolutionary computation*, 44:612–621.
- [Tosun *et al.*, 2020] TOSUN, Ö., MARICHELVAM, M. et TOSUN, N. (2020). A literature review on hybrid flow shop scheduling. *International Journal of Advanced Operations Management*, 12(2):156–194.
- [Ullrich, 2013] ULLRICH, C. A. (2013). Integrated machine scheduling and vehicle routing with time windows. *European Journal of Operational Research*, 227(1):152–165.
- [Van Buer *et al.*, 1999] VAN BUER, M. G., WOODRUFF, D. L. et OLSON, R. T. (1999). Solving the medium newspaper production/distribution problem. *European Journal of Operational Research*, 115(2):237–253.
- [Viergutz et Knust, 2014] VIERGUTZ, C. et KNUST, S. (2014). Integrated production and distribution scheduling with lifespan constraints. *Annals of Operations Research*, 213(1):293–318.
- [Vignier *et al.*, 1999] VIGNIER, A., BILLAUT, J.-C. et PROUST, C. (1999). Les problèmes d’ordonnancement de type «flow-shop» hybride : état de l’art. *RAIRO-Operations Research-Recherche Operationnelle*, 33(2):117–183.
- [Wang *et al.*, 2019a] WANG, D., ZHU, J., WEI, X., CHENG, T., YIN, Y. et WANG, Y. (2019a). Integrated production and multiple trips vehicle routing with time windows and uncertain travel times. *Computers & Operations Research*, 103:1–12.

- [Wang *et al.*, 2019b] WANG, J., YAO, S., SHENG, J. et YANG, H. (2019b). Minimizing total carbon emissions in an integrated machine scheduling and vehicle routing problem. *Journal of Cleaner Production*, 229:1004–1017.
- [Wang *et al.*, 2019c] WANG, S., WU, R., CHU, F. et YU, J. (2019c). Variable neighborhood search-based methods for integrated hybrid flow shop scheduling with distribution. *Soft Computing*, pages 1–20.
- [Yağmur et Kesen, 2020] YAĞMUR, E. et KESEN, S. E. (2020). A memetic algorithm for joint production and distribution scheduling with due dates. *Computers & Industrial Engineering*, 142:106342.
- [Yokouchi *et al.*, 2012] YOKOUCHI, M., AOKI, S., SANG, H., ZHAO, R. et TAKAKUWA, S. (2012). Operations analysis and appointment scheduling for an outpatient chemotherapy department. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–12. IEEE.
- [Zhen *et al.*, 2020] ZHEN, L., MA, C., WANG, K., XIAO, L. et ZHANG, W. (2020). Multi-depot multi-trip vehicle routing problem with time windows and release dates. *Transportation Research Part E : Logistics and Transportation Review*, 135:101866.
- [Zou *et al.*, 2018] ZOU, X., LIU, L., LI, K. et LI, W. (2018). A coordinated algorithm for integrated production scheduling and vehicle routing problem. *International Journal of Production Research*, 56(15):5005–5024.
- [Zu *et al.*, 2014] ZU, L., LI, W. et KURZ, M. E. (2014). Integrated production and distribution problem with pickup and delivery and multiple trips. In *IIE Annual Conference. Proceedings*, page 1587. Institute of Industrial and Systems Engineers (IISE).

Résumé

Cette thèse porte sur l'étude et la résolution de problèmes d'optimisation dans le domaine de la production de médicaments de chimiothérapie à l'aide des outils de la Recherche Opérationnelle. Ces travaux de recherche ont été menés en collaboration avec l'hôpital de Tours et l'entreprise Eticsys et ont eu pour but de répondre à deux grands enjeux : la conservation de la qualité de service pour les patients et la réduction des coûts pour le système de santé. Ces travaux s'étendent de l'étude des flux de production d'une unité de biopharmacie à la mise en place d'une solution d'optimisation et d'aide à la décision pour cette unité.

Ce manuscrit commence par présenter le contexte hospitalier, les contraintes métiers et sanitaires à prendre en compte lors du processus de modélisation des problèmes de planification et d'optimisation au plus proche de la réalité. Le premier problème considéré est formulé comme un problème intégré de production et de livraison. La partie production est modélisée comme un problème d'ordonnancement d'atelier de type flow-shop hybride. La partie livraison s'apparente à un problème de tournées de véhicules multi-trajet. Le deuxième problème étudié est représenté par un problème intégré de production et de gestion des ressources consommables. La partie production considérée est cette fois-ci modélisée par un problème d'ordonnancement à machines parallèles avec des périodes d'indisponibilités. La partie gestion des ressources consommables est modélisée comme un problème de bin-packing avec des contraintes de conflits. Plusieurs algorithmes heuristiques sont proposés pour résoudre les problèmes de planification et leurs performances sont évaluées. Une méthode exacte basée sur une exploration arborescente est proposée pour résoudre la partie livraison d'un problème intégré et étudier la qualité des solutions trouvées par les heuristiques de construction d'itinéraires. La gestion des ressources consommables étant intrinsèquement liée à l'ordonnancement de la production, mais indépendante de la livraison, une matheuristique sur une version intégrée production et gestion de ressource consommables du problème est développée.

Les résultats montrant l'efficacité des méthodes de résolution proposées, ont encouragé l'implémentation de certains des algorithmes proposés dans un logiciel d'aide à la décision pour le service de l'hôpital de Tours en charge de la production des préparations. Les problèmes traités sont généralisables à d'autres services de même nature mais dans d'autres contextes. La recherche a abouti aussi à la publication de plusieurs articles et communications scientifiques. De nombreuses perspectives d'améliorations des algorithmes et des extensions des modèles sont évoquées afin d'approfondir l'amélioration continue des services hospitaliers.

Mots-clefs : ordonnancement, livraison, découpe, heuristique, exacte

Abstract

This thesis tackle optimization problems in a chemotherapy drug production context. Their theoretical and practical implications are studied. In collaboration with a hospital and a company, the research aims to respond to two challenges : maintaining the quality of service for patients and reducing costs for the health system. This work goes from a biopharmacy unit production flows study to the implementation of an optimization and aid decision support solution.

This manuscript begins by presenting the hospital context, and sanitary constraints. The first problem considered is formulated as an integrated production and routing problem. The production part is modeled as a hybrid flow-shop scheduling problem. The delivery part is a variant of the multi-trip vehicle routing problem. The second problem is represented by an integrated production and cutting stock problem. The production part considered is this time modeled by a parallel machine scheduling problem with unavailability periods. The cutting stock part is modeled as a bin packing problem with conflicts. Several heuristic algorithms for improving the service are proposed and evaluated. An exact method based on multiple branch-and-bound procedures is also proposed. As the cutting stock is dependent of the production scheduling, but independent of the delivery, a matheuristic approach for the integrated production and cutting stock problem is developed.

The efficiencies of the proposed resolution methods encouraged to implement certain of them in an aid decision software for the hospital of Tours. The problems dealt could be generalized to other contexts. The research is also relevant to the publication of several scientific articles and communications. Many perspectives for improving algorithms and extensions of models are mentioned in order to deepen the continuous improvement of hospital services.

Keywords : scheduling, routing, Bin-packing, heuristic, Branch-and-bound