

Ecole Polytechnique de l'Université de Tours

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet Recherche & Développement 2020-2021

Ordonnancement de la production de chimiothérapies avec minimisation des pertes

**POLYTECH[®]**
TOURS

Tuteurs académiques

Jean-Charles BILLAUT

Alexis ROBBES

Yannick KERGOSIEN

Étudiant

Jacob ADOR (DI5)



Liste des intervenants

Nom	Email	Qualité
Jacob ADOR	jacob.ador@etu.univ-tours.fr	Étudiant DI5
Jean-Charles BILLAUT	jean-charles.billaut@univ-tours.fr	Tuteur académique, Département Informatique
Alexis ROBBES	alexis.robbes@univ-tours.fr	Tuteur académique, Département Informatique
Yannick KERGOSIEN	yannick.kergosien@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

Ce document a été rédigé par Jacob ADOR surnommé l'auteur.

L'Ecole Polytechnique de l'Université de Tours est représentée par Jean-Charles BILLAUT, Alexis ROBBES et Yannick KERGOSIEN surnommés les tuteurs académiques.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assume l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable des tuteurs académiques et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Jacob ADOR, *Ordonnancement de la production de chimiothérapies avec minimisation des pertes*, Projet Recherche & Développement, Ecole Polytechnique de l'Université de Tours, Tours, France, 2020-2021.

```
@mastersthesis{
  author={ADOR, Jacob},
  title={Ordonnancement de la production de chimiothérapies avec minimisation des pertes:
},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université de Tours},
  address={Tours, France},
  year={2020-2021}
}
```

Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iv
1 Introduction	1
1 Acteurs, enjeux et contexte	1
2 Objectifs.....	2
3 Bases méthodologiques.....	2
2 Description générale	4
1 Environnement du projet	4
2 Caractéristiques des utilisateurs.....	4
3 Fonctionnalités du système	5
4 Structure générale du système.....	6
4.1 Processus de fabrication.....	6
4.2 Processus de traitement des matières premières	7
4.3 Les phases de fabrication	8
4.3.1 La stérilisation	8
4.3.2 La préparation.....	9
4.3.3 Le contrôle	10
4.4 Les structures de données importantes	10

3	État de l'art	12
1	Introduction	12
2	Le problème d'ordonnancement (Flow Shop hybride)	12
2.1	Présentation du problème	12
2.2	Méthodes de résolution dans la littérature	14
2.2.1	Méthodes exactes	14
	L'optimisation linéaire	14
	Programmation par contraintes	15
	Programmation dynamique.....	15
2.2.2	Méthodes approchées.....	15
	Recherche Tabou	16
	Algorithme Génétique	17
3	Le problème d'affectation des molécules (Bin Packing avec conflits)	18
3.1	Présentation du problème	18
3.2	Les heuristiques de résolution du Bin Packing avec conflits	18
3.2.1	First-Fit Decreasing.....	19
3.2.2	Best-Fit Decreasing	19
4	Optimisation multiobjectif	19
4.1	Présentation du problème multiobjectif.....	19
4.2	Les différentes approches	20
5	Rapprochements avec le problème original	21
4	Analyse et conception	22
1	Identification du problème	22
2	Analyse du problème	22
2.1	Les ressources	23
2.2	Les jobs	24
2.3	Les contraintes	25
3	Modélisation du problème	25
3.1	Modélisation mathématique des contraintes	25
3.2	Fonctions objectif	26
3.3	Le problème du Flow Shop hybride.....	27
3.4	Codage de la solution	27
3.5	Algorithmes de résolution	28
4	Génération des instances.....	28
5	Bilan et conclusion	29
1	Bilan du semestre 9	29
1.1	Faits	29
1.2	Reste à faire	29
1.3	Retards	30

Annexes	31
A Planification, gestion de projet	32
1 Evolution du projet	32
B Description des interfaces	34
1 Interfaces matérielles/logicielles	34
2 Interfaces homme/machine.....	34
3 Interfaces logiciel/logiciel	34
C Cahier de Spécification	36
1 spécifications Fonctionnelles.....	36
1.1 Fonctionnalités à développer	36
1.1.1 Fonction d'initialisation du stock (parseur)	36
1.1.2 Fonction de génération des instances	36
1.1.3 Fonction de génération d'un job	37
1.1.4 Fonction de génération d'une solution initiale	37
1.1.5 Fonction de résolution (recherche d'une solution optimale).....	38
1.1.6 Fonction de décodage d'une solution	38
1.1.7 Fonction d'écriture des résultats	39
2 Spécifications non fonctionnelles	39
2.1 Contraintes de développement et conception	39
2.2 Contraintes de fonctionnement et d'exploitation.....	39
2.2.1 Performances.....	39
2.2.2 Capacités	40
2.3 Modes de fonctionnement	40
2.3.1 Contrôlabilité	40
2.3.2 Sécurité	40
D Bibliographie	41
E Glossaire	43

Table des figures

2 Description générale

2.1	Diagramme des cas d'utilisation du système.....	5
2.2	Processus de fabrication.....	6
2.3	Processus de consommation des matières premières.....	7
2.4	Bac contenant le matériel pour une préparation	8
2.5	Paniers contenant les matières premières.....	9
2.6	Processus de préparation au sein d'un isolateur	9
2.7	Un isolateur de l'UBCO de Tours	10
2.8	Processus de contrôle des préparations.....	10
2.9	Premier diagramme de classe relativement aux structures de données.....	11

3 État de l'art

3.1	Schéma du problème du Flow Shop hybride	14
3.2	Exemple de front de Pareto pour une fonction à double objectifs.....	20

4 Analyse et conception

4.1	Poste de travail d'un opérateur à l'UBCO de Tours.....	24
-----	--	----

A Planification, gestion de projet

A.1	Le diagramme de Gantt du S9 initial	32
A.2	Le diagramme de Gantt du S9 final.....	33

1

Introduction

1 Acteurs, enjeux et contexte

Dans le cadre de la production de chimiothérapies pour le traitement de différents types de cancer, il est nécessaire de manipuler des produits dits « **Cytotoxique** » qui sont à la fois dangereux, rapidement périssables et parfois très onéreux. Le CHU Bretonneau à Tours dispose du matériel pour préparer les solutions de chimiothérapies qui seront administrées aux patients par la suite.

Actuellement, la production journalière de solutions de chimiothérapie est de 150 réparties tout au long de la journée au fil des prescriptions des médecins, ce qui est un nombre considérable.

Les acteurs qui rentrent en jeu dans cette production de chimiothérapies sont toutes les personnes sollicitées au cours de la chaîne de production d'une solution. Du médecin qui va prescrire une dose au patient et donc valider une mise en production aux opérateurs qui vont effectuer les préparations en passant par les chefs de production personnes chargées d'organiser la production via le logiciel d'ordonnancement actuel de l'**UBCO**.

Deux enjeux majeurs se dégagent de la production de chimiothérapies :

1. Tout d'abord, chaque patient doit normalement recevoir sa préparation à une date et à une heure précise, il est donc nécessaire de satisfaire la contrainte temporelle de production. Il arrive cependant qu'il y ait tout de même des retards et il convient donc d'en minimiser la durée mais aussi la fréquence.
2. A cela, s'ajoute l'enjeu économique, en effet, comme évoqué précédemment mais également dans le PFE de Thibault Drevon, les produits cytotoxiques ont un coût parfois élevé et une durée de conservation après ouverture. Le fait de jeter ne serait-ce que quelques millilitres de matière première peut se traduire par une perte de plusieurs centaines voir milliers d'euros pour certaines.

La problématique consiste donc à ordonnancer la production de chimiothérapies pour permettre de respecter les délais en minimisant la somme des retards mais aussi à minimiser les pertes des produits qui ne pourraient être utilisés avant leur date de péremption dans le but de diminuer les coûts de l'hôpital.

Actuellement, le CHU de Bretonneau dispose d'un logiciel développé par la société ETYCSYS en collaboration avec le laboratoire de l'université de Tours et notamment Monsieur Jean-Charles Billaut qui est l'encadrant de ce projet. Ce logiciel sert à ordonnancer la production en tachant de minimiser les retards. Il s'agit donc d'un logiciel que se soucie uniquement de la contrainte

temporelle sans se soucier des pertes de produit au cours de la production. Pour l'heure, les opérateurs (les personnes qui préparent les produits dans les isolateurs) choisissent eux-mêmes les reliquats qu'ils souhaitent utilisés en fonction des produits nécessaires pour les solutions à réaliser. La traçabilité des reliquats et leur gestion dans le processus de production pourrait donc permettre de réaliser des économies en évitant de jeter des reliquats qui n'auraient pas pu être utilisés à temps.

2 Objectifs

L'objectif de ce projet est de proposer une méthode permettant d'ordonnancer la production de la chimiothérapie du CHU en prenant en compte le contexte évoqué plus haut. On cherchera donc à trouver une modélisation du problème prenant en compte les différentes contraintes engendrées par la problématique que pose la production de chimiothérapies avec la prise en compte des pertes.

On va donc chercher à montrer par la modélisation puis par l'implémentation algorithmique d'une solution d'ordonnancement de la production avec prise en compte des pertes qu'il est possible de réaliser des économies sans pour autant dégrader la qualité de distribution à savoir la minimisation des retards de livraison. Il est important que la solution apportée ne soit pas défavorable aux patients même si cela permet de réaliser des économies. Il conviendra bien sûr de comparer les résultats d'exécution de méthodes prenant en compte les reliquats et de méthodes se souciant uniquement de satisfaire la contrainte temporelle. Cela nous permettra de mettre en évidence ou non l'utilité de la prise en compte des reliquats dans l'ordonnancement.

Tout comme c'était le cas pour le PFE de Thibault Drevon, nous ne nous intéresserons pas ici au problème de l'acheminement des préparations aux patients puisque cela relève d'un autre problème lui aussi complexe et qui nécessite un travail plus approfondi sur le sujet.

Ce projet de recherche consiste donc à chercher des méthodes de résolution d'un tel problème d'ordonnancement avec prise en compte de ressources autre que les machines. En effet, dans un problème d'ordonnancement, les ressources sont souvent associées aux machines de production qui sont disponibles. Ici on, ajoute des ressources qui sont épuisables, au contraire des machines qui sont des ressources inépuisables auxquelles on va attribuer des tâches. Ce type de problème qui combine ces contraintes est très complexe puisque c'est une combinaison de deux problèmes qui pourraient chacun être traités indépendamment et ce sont deux problèmes qui sont réputés NP-difficile. Cela signifie qu'on cherchera à étudier la littérature portant sur ces problèmes afin de trouver un moyen de les résoudre tous les deux. Pour cela, nous explorerons plusieurs pistes, notamment sur la façon de traiter les données et de modéliser le problème sans pour autant le rendre trop complexe et donc inexploitable. L'objectif est de parvenir à proposer un algorithme novateur pour traiter ce problème en s'appuyant sur des méthodes récentes.

Nous chercherons donc à trouver une nouvelle façon d'ordonnancer la production de chimiothérapies en nous appuyant sur les travaux déjà réalisés, à la fois par Monsieur Billaut, Monsieur Robbes qui a eu l'occasion d'étudier et de proposer une première solution au problème des reliquats, mais également sur les travaux de Thibault Drevon qui a eu l'occasion d'expérimenter plusieurs approches ouvrant des portes vers des solutions envisageables.

3 Bases méthodologiques

Pour ce projet, nous utiliserons plusieurs outils de gestions :

- Gantt Project pour réaliser un diagramme de Gantt présentant les tâches et leurs échéances pour suivre le déroulement du projet

- Trello pour organiser les tâches
- Latex pour rédiger ce rapport

En ce qui concerne la gestion de projet, nous utilisons la méthodologie Agile, nous travaillons par sprint de 4 semaines durant lesquelles des objectifs sont fixés. Une réunion hebdomadaire avec l'encadrant permet de faire part des avancements, mais aussi des questionnements sur le travail effectué et sur le travail prévisionnel.

2

Description générale

1 Environnement du projet

A l'heure actuelle, c'est la société ETICSYS qui fournit la suite de logiciel qui permet à l'UBCO de pouvoir ordonnancer la production de chimiothérapies. Le logiciel Chimio est également utilisé pour permettre de créer des fiches de produits et pour faciliter le travail de production des opérateurs, cependant il ne traite pas l'ordonnancement de la production. Cela est réalisé par le logiciel Planif développé par ETICSYS et qui permet donc d'optimiser l'échéancier des tâches à réaliser pour améliorer la productivité et réduire les retards de production.

Ce logiciel est développé en collaboration avec le laboratoire informatique de l'université de Tours et notamment Monsieur Billaut ainsi que Monsieur Kergosien. Il permet d'ordonnancer les préparations émises par Chimio pour permettre aux opérateurs de connaître les priorités des productions à effectuer. Ce logiciel permet un ordonnancement qui réduit le retard des préparations en fonction de plusieurs critères. Chaque tâche, dispose d'une date de validation, date à laquelle le médecin donne son accord pour lancer la production. On a alors plusieurs informations sur la tâche comme le temps nécessaire à sa production ou encore sa date due, c'est-à-dire la date à laquelle la solution est censé parvenir au patient en vue de l'injection. Planif réalise donc, sur la base d'une programmation par contraintes, un ordonnancement sur plusieurs étages sur des machines parallèles. C'est un problème de type « **Flow Shop** hybride » qui est donc traité ici et qui se révèle être un problème complexe et coûteux en temps. Néanmoins, Planif ne tient pour le moment pas compte de la perte des produits entamés lors de sa planification des préparations. En effet, rien ne permet de prendre en compte les dates de péremption des produits dans l'ordonnancement. Cela impliquerait une modification du programme actuel pour prendre en compte les reliquats et c'est précisément ce que nous allons chercher à faire au cours de ce projet. Pour l'heure, les opérateurs essaient de veiller à ce que peu de produits se perdent, cependant les pertes restent tout de même conséquentes puisqu'elles peuvent s'élever à plusieurs millions d'euros sur une année. La production étant croissante chaque année, il devient nécessaire de traiter ce problème.

2 Caractéristiques des utilisateurs

Les utilisateurs principaux du système sont les membres du personnels chargés de réaliser la production des chimiothérapies, les opérateurs. Ils utilisent les logiciels cités précédemment et

sont habitués à l'interface que ces derniers proposent, aussi il n'est pas question d'en modifier le fonctionnement.

Ces opérateurs ont des disponibilités différentes, correspondant à leurs heures de pauses, de début et de fin de service. Ces contraintes seront à prendre en compte dans l'ordonnancement de la production. En effet, c'est aux opérateurs qu'on va affecter des tâches à effectuer. Ils seront considérés dans notre système comme les machines parallèle d'un étage de la production.

Nous étudierons des modèles mathématiques et testeront des algorithmes de traitement permettant dans le meilleur des cas d'améliorer par la suite l'ordonnancement de la production mais ces traitements seront directement intégrés aux logiciels existants s'ils venaient à se révéler efficaces et le fonctionnement global des logiciels ainsi que l'utilisation des membres du personnel n'en serait pas altéré.

Nous avons donc affaire à des utilisateurs expérimentés sur les outils qu'ils utilisent et il convient donc de ne pas en modifier le fonctionnement mais plutôt les traitements réalisés en interne.

3 Fonctionnalités du système

Le système fonctionne en traitant les prescriptions entrées par les praticiens de santé grâce au logiciel Planif qui se sert de différentes informations comme les dates de validations de prescriptions, les temps de traitements des préparations, les temps de stérilisation, les temps de contrôle des produits finis ou encore les dates dues, à savoir la date à laquelle le patient doit recevoir son traitement, pour produire un ordonnancement calculé sur la base d'algorithme d'ordonnancement à plusieurs niveaux.

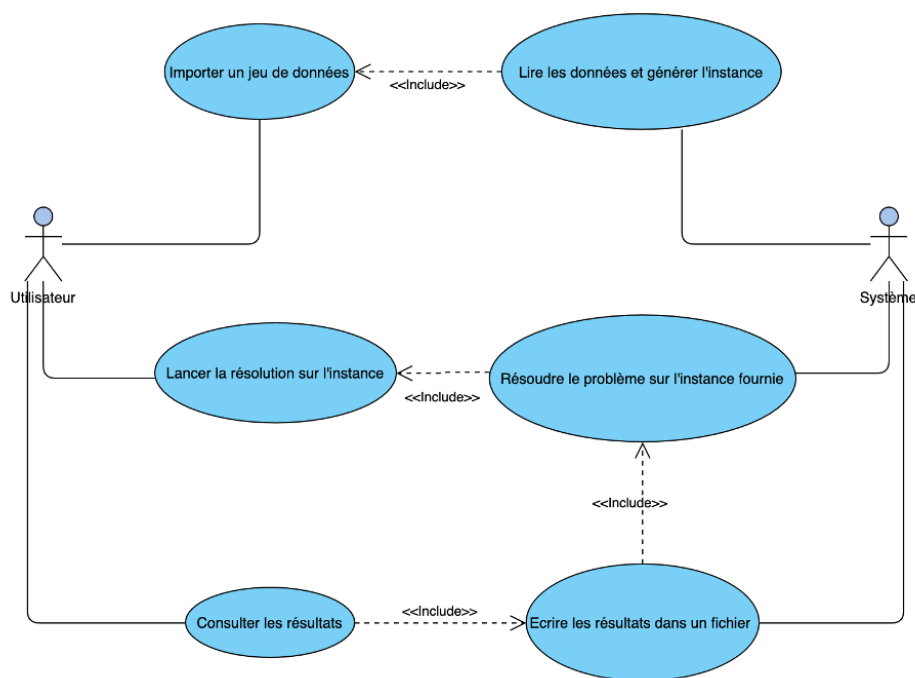


Figure 2.1 – Diagramme des cas d'utilisation du système

Nous n'aborderons pas le fonctionnement détaillé du système pour ce sujet puisque l'intérêt est de chercher à améliorer les traitements algorithmiques réalisés sur l'ordonnancement de la production. Les fonctionnalités du système ne devraient en théorie par être modifiées.

En effet, l'objectif ici n'est pas de proposer un logiciel avec une interface complète, notamment parce que cela est déjà disponible à travers le logiciel Planif. Si jamais les résultats obtenus au

terme de ce projet sont concluants, alors on pourra penser à intégrer le traitement algorithmique au logiciel.

Le système proposé devra permettre à l'utilisateur de générer des instances à partir d'un jeu de données. On utilisera ces instances pour résoudre le problème et obtenir des solutions. L'utilisateur devra dans un premier temps lire les données pour générer une instance. Il pourra ensuite exécuter le programme de résolution qui lui permettra d'obtenir les résultats générés, ce programme devra également enregistrer la solution fournie dans un fichier csv.

Le diagramme des cas d'utilisation du système est présenté sur la Figure 2.1.

4 Structure générale du système

Pour bien comprendre le fonctionnement général du système, il est nécessaire de détailler les étapes de fabrication des solutions de chimiothérapies qui seront fournies aux patients. Ces étapes demandent chacune des besoins que ce soit en termes de matières premières, de matériel de production ou encore d'intervenants humains qui sont les opérateurs.

4.1 Processus de fabrication

On détaille dans un premier temps le processus de fabrication des produits. Voici les étapes de production des solutions de chimiothérapies, incluant la phase de livraison et d'injection au patient :

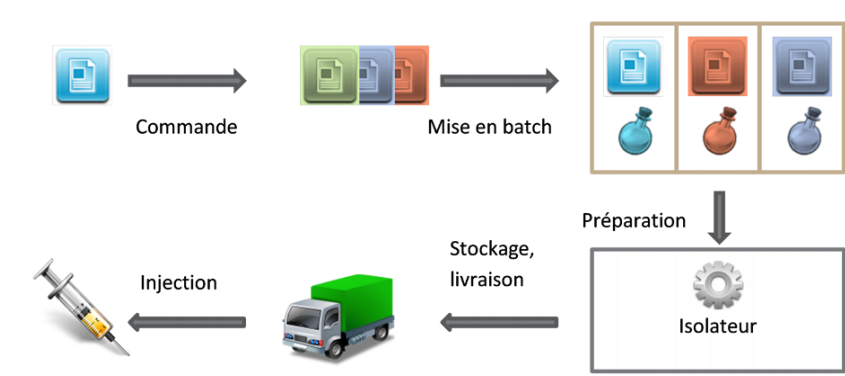


Figure 2.2 – Processus de fabrication

1. Dans un premier temps, le médecin ou tout autre intervenant qualifié dans le cadre du suivi des patients va effectuer une prescription, qui correspond à une commande à traiter par l'UBCO. Cette prescription est réalisée après avoir vu le patient et pris connaissance de son état physique et de son aptitude à recevoir le traitement. Si toutes les conditions sont réunies, le médecin transmet la prescription au service de production et le patient doit alors recevoir son traitement dans la journée. Cette commande contient toutes les informations nécessaires à la réalisation du produit comme la matière première à utiliser et dans quelle quantité, une durée de fabrication ainsi qu'une date due, c'est-à-dire la date à laquelle le patient doit recevoir son traitement qui est déterminée en fonction de la date de validation du médecin et du temps de préparation du produit. Dans la pratique, on ne peut pas satisfaire ces dates puisqu'il faudrait lancer en production les préparations immédiatement après leur validation ce qui n'est évidemment pas possible. On cherche alors à avoir le moins de retard possible grâce à un ordonnancement réalisé par Planif.
2. Le logiciel Planif se charge de préparer un ordonnancement, il donne donc un ordre de priorité pour les préparations à effectuer. Comme les préparations sont réalisées au sein

- d'isolateurs qui eux-mêmes disposent de bacs de stérilisation, elles sont toutes attribuées à des batchs de préparations qui sont repartis selon un planning déterminé par Planif.
3. Une fois les préparations assignées aux batchs et ayant des ordres de traitement, elles sont donc traitées dans les isolateurs ou elles passent par la phase de stérilisation, puis par la phase de préparation réalisée par les opérateurs au sein des isolateurs également.
 4. Lorsqu'une préparation d'un batch est réalisée, elle peut partir directement à la phase de contrôle suivant le planning sans attendre que les autres préparations du batch soient terminées.
 5. La phase de contrôle suivant la préparation du produit est indispensable car elle permet de s'assurer que la solution est conforme aux attentes et qu'elle respecte les normes fixées pour chaque produit. Le temps de contrôle pour chaque produit est fixe et il n'existe qu'une seule machine permettant de contrôler un produit à la fois.
 6. A la suite du contrôle, chaque produit est alors stocké dans un réfrigérateur dans l'optique d'être livré avec d'autres produits sur le site d'administration aux patients.

Ce processus décrit la façon dont sont produits les solutions, de leur commande par le médecin ou le personnel médical qualifié, en passant par sa production et ses différentes phases jusqu'à la livraison du produit sur le lieu d'administration. Cela constitue donc le fonctionnement global des étapes de production et se concentre en particulier sur la solution finale issue du processus de fabrication. Il est à noter que le produit final d'une préparation possède une date de péremption et qu'il est donc important de veiller à ne pas produire une solution trop longtemps avant son administration au patient

Intéressons-nous maintenant à la gestion des produits qui sont entamés au cours du processus de fabrication et leur traitement au fil des préparations. En effet, dans le cadre de la fabrication des solutions de chimiothérapies, il est parfois nécessaire d'utiliser plusieurs flacons de matières premières et cela implique que, dans certains cas, il y ait des restes de matières premières non utilisées. Dans ce cas, compte tenu de la rareté et de l'onérosité de certains produits, il est totalement proscrit de perdre des flacons qui n'auraient pas été utilisés entièrement.

4.2 Processus de traitement des matières premières

Nous allons maintenant détailler le cycle de vie de ces matières premières. Voici le schéma de consommation des ressources au cours du processus de fabrication des produits :

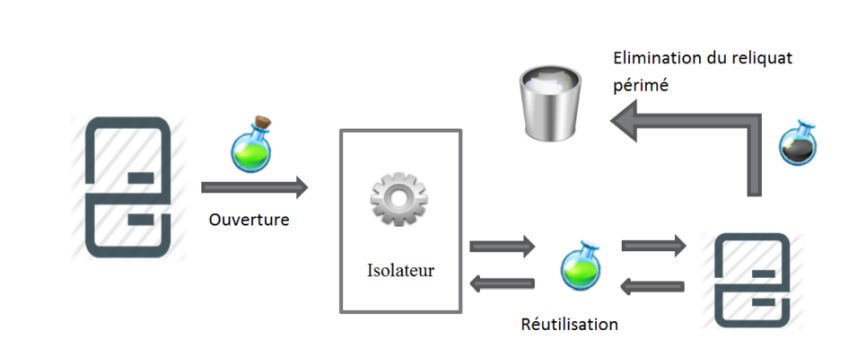


Figure 2.3 – *Processus de consommation des matières premières*

L'hôpital dispose d'un stock de matières premières qui sont utilisables pour les différentes préparations à réaliser. Ces matières premières se présentent sous la forme de flacons qui sont disposés dans de grands réfrigérateurs.

1. Tout d'abord, on récupère le ou les flacons dont on a besoin pour une préparation, puis on le dispose dans un panier contenant le reste du matériel nécessaire à l'élaboration de la solution finale.

2. Ce panier ainsi que tous ceux des préparations du batch sont envoyés dans l'isolateur où ils passent l'étape de la stérilisation avant d'être utilisés par l'opérateur pour ses préparations.
3. Enfin, les flacons vides sont éliminés tandis que ceux qui ne le sont pas rejoignent le réfrigérateur ou ils ne pourront être conservés que durant un temps limité propre à chaque matière première

Comme l'a fait remarquer Jean-Loup Combes dans PFE, il n'existe pas de relation entre la durée restante d'utilisabilité d'un reliquat et la date de péremption du produit final. En clair, une fois ouvert, un produit dispose d'une date de péremption fixe.

4.3 Les phases de fabrication

4.3.1 La stérilisation

La phase de stérilisation permet de stériliser à la fois les flacons de matières premières, mais également tout le matériel nécessaire aux préparations.

Voici une photo du matériel nécessaire pour réaliser une préparation avant qu'il soit stérilisé :

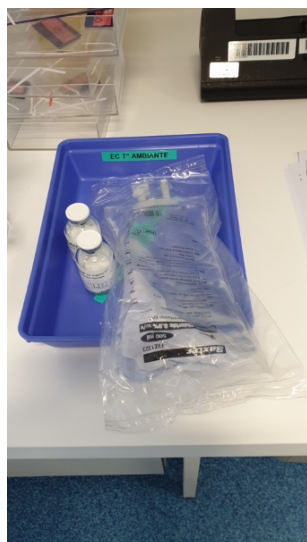


Figure 2.4 – *Bac contenant le matériel pour une préparation*

Comme expliqué précédemment, les préparations se font par batch, c'est-à-dire par groupe de produits à traiter. Tout le matériel ainsi que les flacons de matières premières sont donc placés dans des paniers que le stérilisateur va pouvoir stériliser par la suite.

Une photo de ces paniers prise à l'UBCO est visible sur la Figure 2.5.

Contrairement aux autres années, les isolateurs ont été modifiés, auparavant, nous ne pouvions avoir qu'un stérilisateur d'une capacité de 12 paniers dans chaque stérilisateur. Il y a désormais deux stérilisateurs indépendants par isolateur et comme il y a toujours deux opérateurs par isolateur, cela permet de ne pas avoir à attendre en attendant que l'ensemble des paniers soient stérilisés pour que les deux opérateurs puissent commencer à travailler.

Cependant, chaque stérilisateur ne peut donc stériliser qu'un batch à la fois et d'une contenance de 6 paniers chacun ce qui laisse une capacité de 12 paniers par isolateur. Cela apporte des modifications à l'ordonnancement puisque les batch à ordonnancer sont de plus petite taille. Nous verrons dans la partie analyse comment nous gérons ce problème.

On note également qu'une fois un batch stérilisé, il est forcément traité dans l'isolateur en cours.



Figure 2.5 – Paniers contenant les matières premières

4.3.2 La préparation

La préparation de la solution est réalisée au sein des isolateurs par les deux opérateurs de part et d'autre. Le fait de travailler dans un isolateur permet de réaliser la solution dans un environnement parfaitement stérile. Voici le schéma de fonctionnement au sein des isolateurs :

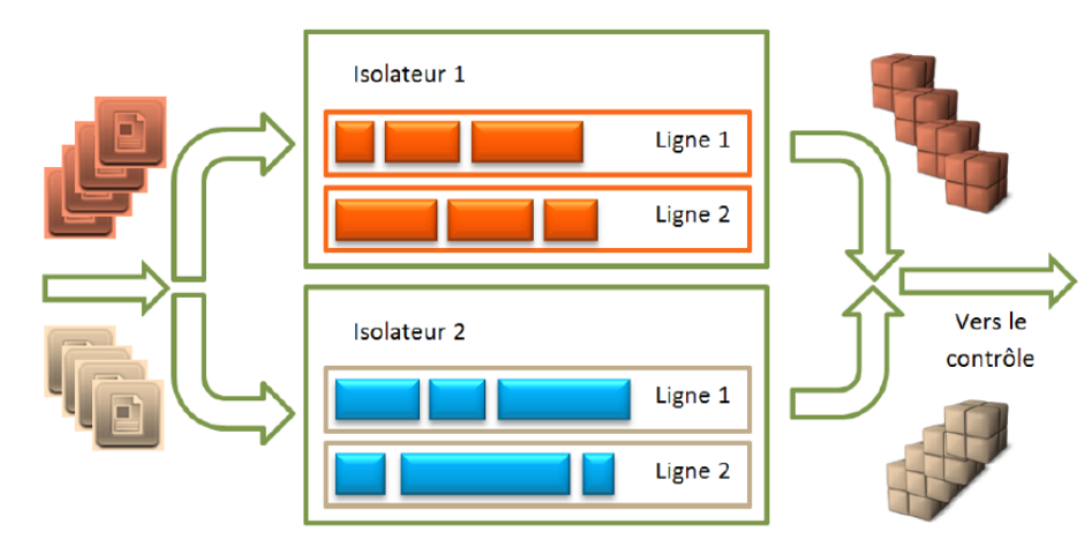


Figure 2.6 – Processus de préparation au sein d'un isolateur

Une photo d'un isolateur de l'UBCO du CHU de Tours est disponible sur la Figure 2.7.

On peut notamment y voir les deux chambres de stérilisation évoquées dans le point précédent à l'extrémité de l'isolateur.

À l'UBCO de Tours, on compte au total 4 isolateurs disposant chacun de deux lignes de production comme on peut le voir sur le schéma ci-dessus. 3 d'entre eux sont équipés des stérilisateurs indépendants et permettent de produire des préparations, le 4ème qui ne dispose que d'un stérilisateur d'une capacité de 12 paniers, est réservé à des tests cliniques.



Figure 2.7 – *Un isolateur de l'UBCO de Tours*

Une fois que la stérilisation est terminée, les batchs de préparations sont alors acheminés aux opérateurs. Il est possible que les opérateurs disposent de plus de préparations à réaliser que la taille d'un batch, en effet chaque isolateur peut accueillir plus de 12 paniers.

Il n'est pas possible de lancer un batch dans le stérilisateur tant que les paniers du batch précédent n'ont pas été retirés. On peut toutefois optimiser les temps de traitements en préparant à l'avance les batch suivant avant de les stériliser, ou même lancer la stérilisation si on sait qu'entre temps les préparations en cours seront terminées.

4.3.3 Le contrôle

La phase de contrôle permet de s'assurer que tous les produits sont bien conformes aux attentes. Ce contrôle s'effectue sur une machine dédiée et les produits ne peuvent être contrôlés qu'un par un. Le temps de contrôle est constant et on ne considère pas de temps d'attente entre deux contrôles. Voici un schéma illustrant le déroulement des contrôles :

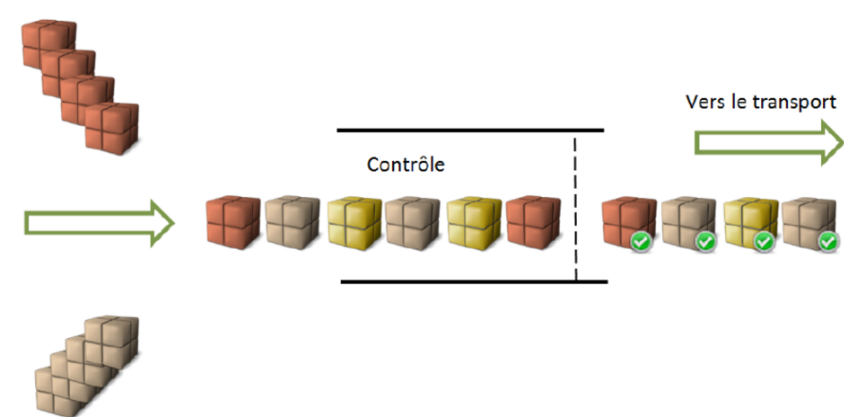


Figure 2.8 – *Processus de contrôle des préparations*

4.4 Les structures de données importantes

En étudiant le processus de fabrication des produits, on peut mettre en évidence plusieurs structures de données qui nous permettront de produire un ordonnancement par la suite :

- Les jobs : Les jobs sont les tâches à effectuer, ce sont les prescriptions émanants du médecin. Ils contiennent toutes les informations nécessaires à la production de la solution comme la date de validation du médecin, le type et la quantité de molécule à utiliser ou encore le temps de production de la solution.
- Les molécules : Ce sont les matières premières indispensables à la préparation des solutions. On doit pouvoir retrouver le type de molécule ainsi que la durée de vie de celle-ci, ce qui permettra de connaître sa date de péremption une fois ouverte.
- Les flacons : Les flacons contiennent un volume de molécule exprimé en ml. C'est aux flacons qu'on va affecter des tâches en fonction des besoins de ceux-ci. On pourra être amené à utiliser un flacon de molécule pour préparer plusieurs tâches.
- Les opérateurs : Ce sont les machines du premier étage de notre ordonnancement. On va devoir affecter les tâches à ces opérateurs, il est donc nécessaire d'avoir une structure permettant de lister les jobs affectés à chaque opérateur. On aura aussi besoin des disponibilités de chaque opérateur (heures de pause, début et fin de journée)
- Le stock : Cette classe de données nous permettra de connaître l'état du stock en permanence. On aura accès à la liste des molécules présentes en stock ainsi qu'à la quantité restante de chacune d'entre-elles.

On définit également une structure qui correspondra aux batches de stérilisation.

Voici un diagramme de classe présentant ces différentes structures de données et leurs liens :

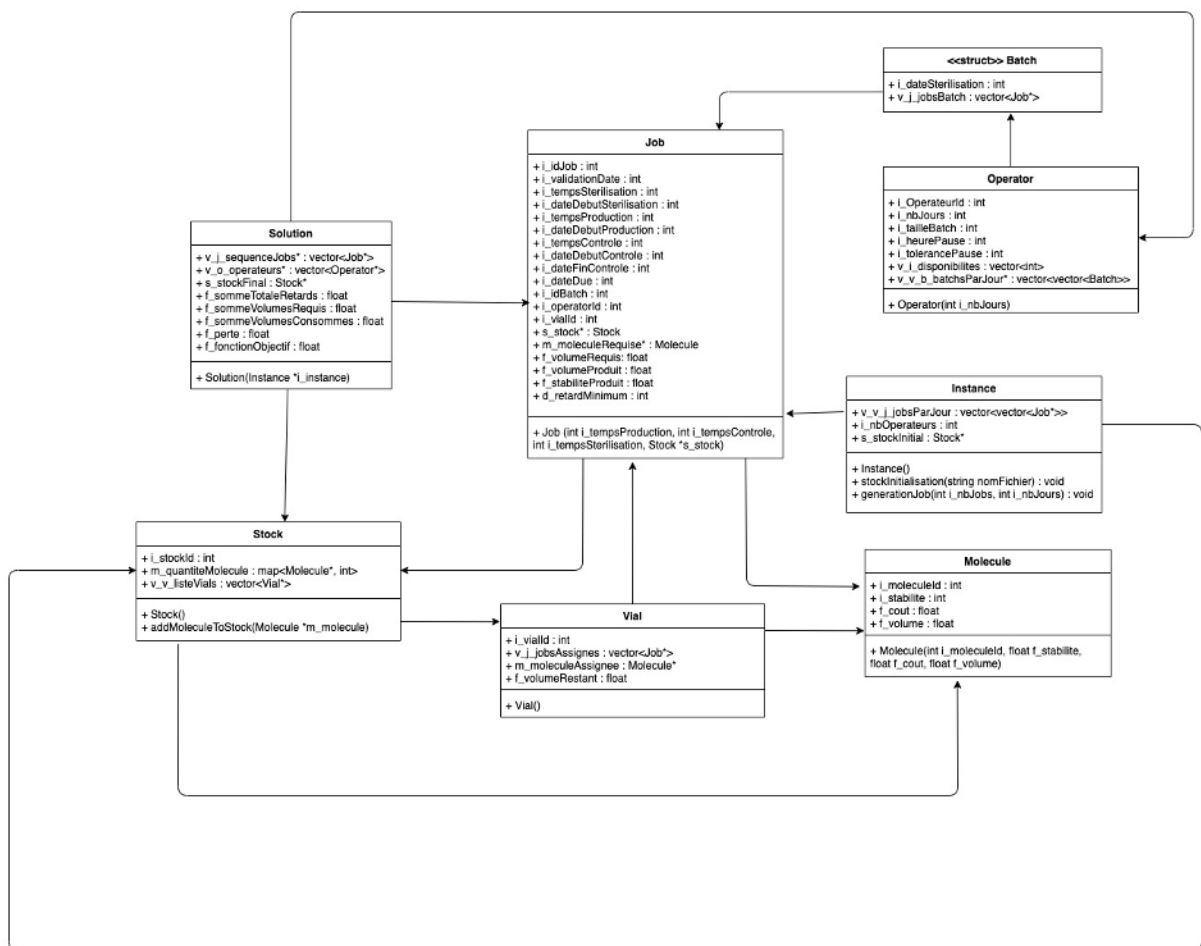


Figure 2.9 – Premier diagramme de classe relativement aux structures de données

3

État de l'art

1 Introduction

Le problème que nous devons résoudre est composé de deux parties bien distinctes que l'on doit d'abord considérer indépendamment l'une de l'autre. En effet, d'une part, nous avons besoin de connaître les méthodes de résolution de problèmes d'ordonnancement dans le but de parvenir à réduire au maximum le retard de livraison des traitements aux patients. D'autre part, nous devons résoudre le problème d'affectation des flacons aux jobs dans l'ordonnancement en minimisant le nombre de flacons ouverts. On cherche alors à résoudre un problème de type « Bin Packing ».

Nous allons étudier dans un premier temps ces deux problèmes séparément avant de faire le point sur les informations récoltées vis-à-vis de notre problème.

2 Le problème d'ordonnancement (Flow Shop hybride)

2.1 Présentation du problème

Dans la plupart des entreprises où le processus de fabrication suit un ordre particulier et où le produit doit passer sur plusieurs machines successivement, on retrouve souvent un besoin logiciel afin de planifier un ordonnancement permettant d'augmenter la productivité. Cela peut passer par la réduction de la somme des retards de livraison ou encore la minimisation de la somme des temps de production afin libérer plus de ressources. On considère donc un problème d'ordonnancement comme une succession de produit à fabriquer. Ces produits sont matérialisés par des tâches à réaliser que l'on appelle des « jobs » et chaque job doit passer par plusieurs machines réparties sur divers étages d'un atelier de production.

Il existe plusieurs types de problèmes d'ordonnancement et tous ne nous intéresseront pas car ils ne correspondent pas à notre problème. On peut notamment citer les problèmes de type job shop, open shop ou encore flow shop.

L'ordonnancement de la production de chimiothérapies au CHU de Tours, tel qu'il est décrit actuellement, relève des problèmes de type « flow shop hybride ». Afin de mieux comprendre de quoi il en ressort, détaillons dans un premier temps à quoi correspond un problème de type « flow shop » classique :

Un problème de type flow shop classique est un problème d'ordonnancement où on dispose d'un nombre de machines qu'on note m , et d'un nombre de jobs à effectuer que l'on note n . Chaque job doit passer sur les m machines successivement. L'objectif de l'ordonnancement est de fixer des dates de passage sur les différentes machines afin de pouvoir fournir un échéancier des dates de passage de chaque job sur chaque machine. Ce problème qui est un classique de l'ordonnancement devient NP-difficile au sens fort au-delà de deux machines dans le système. Il n'est donc pas possible de trouver la solution optimale à de tels problèmes en un temps polynomial, cependant, les recherches sur le problème et les nombreuses heuristiques proposées permettent de trouver de bonnes solutions. Il est à noter que le problème du flow shop à deux machines est résolu en temps polynomial par [3, Johnson, 1954] à l'aide d'un algorithme toujours utilisé aujourd'hui, notamment en association à d'autres algorithmes dans des problèmes plus vastes.

Le problème que nous traitons dans ce projet est un problème de type « flow shop hybride ». Cela signifie que l'on a plus une machine par étage mais plusieurs qui travaillent en parallèle. Dans le cas du CHU, les machines parallèles sont les opérateurs qui réalisent les préparations dans les isolateurs. Il s'agit d'une variante plus complexe qui est donc aussi réputée NP-difficile. On aura donc à faire à un système à trois étages :

- Le premier étage correspond aux machines de stérilisation. Nous verrons plus tard que cet étage peut être associé au deuxième étage durant la phase d'ordonnancement puisqu'un stérilisateur est associé à un opérateur. Les jobs qui sortent d'un stérilisateur sont traités par un même opérateur.
- Le deuxième étage correspond aux machines de production, ce sont les opérateurs qui travaillent en parallèles sur les isolateurs pour préparer les solutions.
- Le troisième et dernier étage correspond à la machine de contrôle des produits. Il n'y en qu'une et les produits passent les uns après les autres pour subir la phase de contrôle.

Nous verrons au cours de l'analyse que nous représenterons ce système par un flow shop hybride à deux étages. Cela est notamment dû au fait qu'il est possible de fusionner les deux premiers étages en regroupant les phases de stérilisation et de production. Nous étudierons en détails les raisons qui nous poussent à faire ce choix dans l'analyse.

Il existe une notation pour les problèmes d'ordonnancement de type flow shop hybride définie par [1, Graham, 1979]. Cette notation qui permet de caractériser les flow shop hybride est la suivante : $\alpha|\beta|\gamma$:

- α se décompose en trois sous-entités qui sont α_1 , qui désigne ici le type d'atelier, α_2 , qui désigne le nombre d'étage du système et enfin $(\alpha_3\alpha_4)$ qui est présenté sous la forme d'un couple désignant pour α_3 le type de machines utilisées (parallèles identiques, parallèles proportionnelles) et pour α_4 le nombre de machines. Le couple $(\alpha_3\alpha_4)$ est répété autant de fois qu'il y a d'étages dans le système. Dans un problème d'ordonnancement, il existe plusieurs types d'ateliers, le flow shop en est un mais il en existe plusieurs autres comme le job shop par exemple.
- β correspond aux contraintes du problème concernant les consommations de ressources (les machines) mais également les processus de fabrication. On note usuellement les contraintes du problème en utilisant des indices et le paramètre β est alors l'agrégation de ces contraintes qu'on va énumérer. On aura alors $\beta = \beta_1\beta_2\beta_3 \dots \beta_n$.
- Enfin, le paramètre γ correspond à la fonction objectif, ou en langage naturel au critère qu'on cherche à optimiser comme la somme des travaux en retard ou la durée d'exécution des jobs par exemple.

Voici un schéma illustrant une solution minimaliste à titre d'exemple pour notre problème, il prend donc en compte les machines du système repartis sur les différents étages :

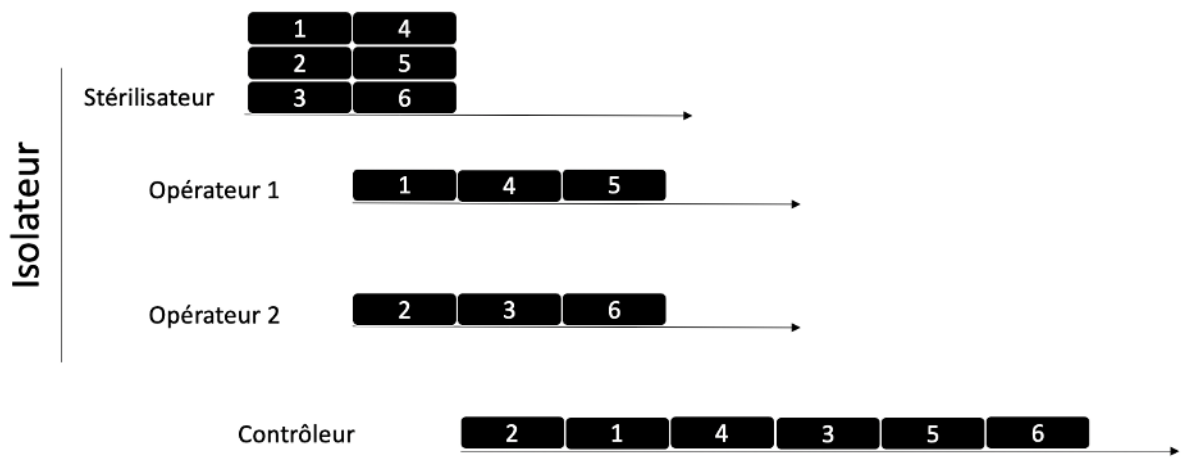


Figure 3.1 – Schéma du problème du Flow Shop hybride

2.2 Méthodes de résolution dans la littérature

2.2.1 Méthodes exactes

Les méthodes exactes sont des méthodes qui permettent de trouver la solution optimale à des problèmes d'optimisation combinatoires en un temps polynomial. On s'assure donc en utilisant une méthode que la solution trouvée est bien la solution optimale. Néanmoins, en raison de la complexité de certains problèmes, il est parfois impossible de réussir à trouver une solution en temps polynomial en raison du grand nombre de solutions possibles. On s'expose alors à des temps de calcul trop importants ce qui pose un problème de performance. Nous allons voir quelles sont les principales méthodes exactes utilisées dans le cadre du problème du flow shop hybride :

L'optimisation linéaire

L'optimisation linéaire consiste à optimiser une fonction objectif sous forme linéaire. On décrit également les contraintes du problème sous forme de fonctions linéaires par rapport aux variables.

Il convient d'évoquer le travail de Thibault Devron en ce qui concerne son approche des méthodes exactes pour résoudre le problème d'ordonnancement proposé. Il a conçu un modèle linéaire en tentant de formaliser les contraintes qui se posaient à lui. Dans son PFE, Thibault Devron a pu montrer les limites d'une telle approche en soulevant notamment la trop grande complexité du modèle. Du fait de la multiplicité des contraintes et de leurs complexités, le solveur n'a pu proposer de solution en un temps raisonnable. En effet, plusieurs contraintes s'écrivaient comme l'agrégation de plusieurs paramètres et le nombre de variables était lui aussi également très important. Grâce à ses travaux, il a pu mettre en évidence la difficulté à linéariser les différentes contraintes et par la même occasion, il a pu montrer que le modèle produit était finalement une version dégradée du problème original. En effet, le fait de vouloir traduire les contraintes en inégalités linéaires ne pouvaient se faire sans compromis.

La programmation linéaire reposant sur la méthode du Simplexe peut cependant être utilisée pour résoudre un problème d'ordonnancement tel que le flow shop hybride si on ne tient pas compte des contraintes propres à la production de chimiothérapies comme la taille des batchs de stérilisation, la date due, ou encore la durée de vie des matières premières. Il est donc possible d'utiliser un modèle linéaire pour représenter un problème d'ordonnancement dès lors que celui-ci est facilement représentable par des contraintes linéaires et dispose d'un nombre limité de variables. Cela devient plus compliqué avec un problème comme celui que nous devons traiter.

Programmation par contraintes

La programmation par contraintes peut être une alternative à la programmation linéaire dans le cadre d'un problème de grande taille. L'objectif est de décrire le problème sous forme de contraintes, linéaires ou non, permettant de définir un ensemble pour chacune des variables du système. On limite ainsi le nombre de valeurs que peuvent prendre les variables liées par les contraintes. Le principe de la programmation par contraintes est d'explorer les solutions réalisables en réduisant petit à petit l'ensemble des valeurs que peuvent prendre les variables au cours de l'exécution. On s'assure alors, à la fin de l'exploration que la solution proposée est optimale. Une telle méthode permet également de montrer qu'un problème n'a pas de solutions si aucune n'est proposée par le modèle à la fin de l'exploration.

Thibault Drevon a également pu expérimenter ce type de programmation pour résoudre le problème posé suite aux essais infructueux avec la programmation linéaire. Ce type de programmation lui a permis de formaliser plus simplement les contraintes puisque l'utilisation de fonction non linéaire n'était plus proscrite. Cependant, son expérimentation de la programmation par contraintes n'a pas non plus porté ses fruits puisque le modèle proposé était trop complexe et impossible à lancer. On note une nouvelle fois le problème de la complexité globale trop importante empêchant de réaliser un modèle viable. L'attribution des flacons de matières premières s'avère être l'une des contraintes les plus difficile à résoudre et le choix d'une méthode adéquate pour traiter ce problème s'avère trop compliqué.

Programmation dynamique

Le principe de la programmation dynamique est de découper le problème posé en plusieurs sous-problèmes qui seront alors résolus uns à uns pour combiner leur résultat et obtenir la solution optimale.

Une méthode de programmation dynamique a été proposée par [7, Salvador, 1973] pour résoudre le problème du flow shop hybride à k étages. Sa modélisation du problème a permis d'obtenir une solution optimale à partir de deux sous-solutions.

Néanmoins, il semble à nouveau compliqué de proposer ce type de modélisation au vu de la complexité du problème. En effet, le flow shop hybride résolu par Salvador ne correspondait pas vraiment au problème posé ici puisque les contraintes sur les affectations de flacons n'existaient pas. On se retrouve alors à nouveau avec un problème plus simple à résoudre et abordable avec une méthode exacte.

2.2.2 Méthodes approchées

Les méthodes approchées, à l'inverse des méthodes exactes, ne permettent pas de trouver assurément une solution optimale à un problème combinatoire posé. Cependant, il existe un avantage majeur à utiliser ce type d'approche pour des problèmes de grande taille, comme c'est le cas ici. Là où les méthodes exactes peuvent prendre énormément de temps à être résolus, notamment lorsque l'on parle d'un problème d'ordonnancement avec 150 jobs par jours sur un horizon de 30 jours, les méthodes approchées peuvent permettre d'obtenir une bonne solution en une temps limitée. Tout l'intérêt de ces méthodes qu'on appelle « métaheuristique », repose donc sur les temps de traitement réduit, bien qu'il soit impossible de savoir si la solution fournie est la solution optimale. Il est parfois nécessaire de faire des compromis quand la complexité du problème l'impose et les méthodes approchées sont grandement utilisées dans la littérature pour résoudre un grand nombre de problèmes combinatoires et notamment les problèmes d'ordonnancement.

Par ailleurs, une métaheuristique nécessite généralement une ou plusieurs solutions initiales. Ces dernières sont générées grâce à des heuristiques qui permettent généralement d'obtenir une bonne

solution sans être exceptionnelle, mais qui présente l'avantage de se trouver en un temps très réduit.

Enfin, le codage de la solution est très important puisqu'il conditionne les traitements de nos métaheuristiques. Plus le codage est simple, plus les opérateurs de nos métaheuristiques le seront, l'intérêt est toujours d'obtenir des temps de traitements intéressants. Un mauvais codage de la solution pourrait donc conduire à de mauvais résultats en termes de performance ce qui remettrait en cause l'un des avantages majeurs des méthodes approchées, les temps de traitements.

Faisons une rapide revue des principales méthodes approchées :

Recherche Tabou

La recherche tabou est l'une des méthodes les plus connues et les plus utilisées car elle est facile à mettre en œuvre et elle permet d'obtenir des résultats de bonne qualité. Le principe est le suivant : On dispose d'une solution initiale générée par une heuristique. Bien sûr, plus la solution initiale est bonne, plus on a de chances de trouver la solution optimale et surtout de minimiser les temps de traitements. A partir de cette solution, on va explorer le voisinage, c'est-à-dire qu'on va explorer les solutions qui ressemblent à notre solution initiale. Pour cela, on définit des opérateurs de voisinage comme un swap entre deux éléments de notre solution par exemple. Il existe un grand nombre d'opérateurs et certains sont spécifiques aux problèmes qu'ils traitent. Ce mécanisme consiste donc en une modification locale de la solution que l'on appelle mouvement. Contrairement à la recherche locale classique qui interdit un mouvement qui dégrade la meilleure solution connue, la recherche taboue l'autorise en mémorisant les mouvements précédents pour éviter de boucler à l'infini sur les mêmes mouvements. L'intérêt d'autoriser des mouvements dégradant la solution actuelle est de sortir d'un optimum local. En effet, le problème de la recherche locale classique est de s'arrêter dès lors qu'une solution n'améliore plus la meilleure solution connue. Cela conduit généralement à un optimum local, ce qui signifie qu'on passe à côté de l'optimum global. Pour sortir de ces optima locaux, on doit autoriser l'algorithme à chercher des solutions moins bonnes mais se rapprochant de la solution optimale. Le fait de mémoriser les derniers mouvements est censé éviter à l'algorithme de retomber dans un optimum local déjà exploré, cependant il est nécessaire de bien paramétrer son algorithme afin de ne pas chercher indéfiniment, tout autorisant une marge de solutions moins bonne assez importante pour sortir de l'optimum local.

C'est la méthode qu'on choisit Thibault Drevon et Alexis Robbes [6] dans le cadre de leurs travaux avec l'UBCO. Dans le cas de Thibault Drevon, il s'agit de la méthode qui lui a permis d'obtenir les meilleurs résultats et les meilleures performances. Cela s'explique par la puissance d'un tel algorithme et la possibilité d'affiner les paramètres et les différents critères de sélection de l'algorithme pour parvenir à une solution toujours meilleure. Son expérimentation de la recherche taboue montre de bons résultats en ce qui concerne l'ordonnancement, donc la capacité à respecter les dates dues sans trop de retard, et également en ce qui concerne la perte de reliquats sur un horizon de deux jours. Sur un horizon de 14 et 30 jours, les résultats sont plus mitigés, cependant ils pourraient être améliorés en modifiant l'opérateur de voisinage et en interdisant notamment le swap entre deux jobs ordonnancés sur deux jours différents ce qui dégrade forcément le critère d'ordonnancement de notre solution. Cependant, les résultats globaux sont bons et prometteurs, il y a ici une piste à poursuivre pour tenter d'améliorer les solutions obtenues.

Alexis Robbes a également expérimenté la méthode la recherche taboue dans le cadre de ses travaux avec l'UBCO. Il expose ses résultats dans son article « Minimizing the costs induced by perishable resource waste in a chemotherapy production unit » qui est également conjointement réalisé avec l'aide Jean-Charles Billaut et Yannick Kergosien, les encadrants de ce projet, et également Virginie André. Le principe de son traitement est le suivant : on résout séparément le problème d'ordonnancement et d'affectation des jobs aux molécules. Le second problème est résolu via l'utilisation d'un solveur en utilisant une modélisation linéaire très efficace pour ce type de

problème. On obtient ainsi plusieurs solutions réalisables pour l'affectation des jobs au flacons de molécules. Ensuite sa méthode d'ordonnancement permet de réaliser des permutations non destructives entre les jobs. C'est-à-dire que l'on ne peut intervertir deux jobs que si la permutation ne modifie pas l'assignement de flacon du deuxième job. En outre, à chaque permutation, on vérifie que la solution obtenue respecte les solutions du problème d'affectation des flacons de molécule aux jobs pour ne pas avoir à résoudre à nouveau ce problème. Tout comme pour le cas de Thibault Drevon, les résultats sont bons sur un horizon de temps limité, il devient cependant plus compliqué d'obtenir des résultats aussi concluants sur un horizon de temps élargi. Cela est dû à la complexité plus importante ainsi qu'au nombre de jobs à traiter qui augmente de 150 par jour de production.

La recherche taboue constitue donc une piste intéressante dans les problèmes d'ordonnancement en général mais également pour notre problème. Plusieurs approches peuvent être testées et chacune dispose de leurs avantages et de leurs inconvénients, cependant il est important de noter que dans les deux cas déjà expérimentés dans le cadre la production de chimiothérapies de l'UBCO, les résultats sont encourageant et montre la possibilité de satisfaire les deux contraintes de notre problème : la satisfaction des patients (minimisation de la somme des retards des jobs) et la minimisation des pertes pour éviter de gâcher des produits onéreux.

Algorithme Génétique

Une deuxième méthode bien connue dans l'ensemble des méthodes approchées est l'algorithme génétique, également appelé algorithme évolutionnaire. Cet algorithme s'inspire, comme son nom l'indique, de la théorie de l'évolution de Darwin. On applique le principe de sélection naturelle afin de conserver, à chaque itération, les individus répondant le plus aux critères imposés, ici l'ordonnancement et l'affectation des molécules. Ces individus seront les plus forts et les plus adaptés à leur environnement. Le processus d'un tel algorithme est le suivant :

- On initialise tout d'abord une population, d'une taille fixée en paramètre.
- On sélectionne ensuite deux parents, selon une politique définie en amont. Cela peut être une sélection par tournoi qui choisit les meilleurs parents dans des paires choisies au préalable, ou bien une sélection grâce une probabilité pour chaque parent, proportionnelle à l'adaptation, c'est-à-dire que la probabilité d'être choisi augmente proportionnellement à la valeur de la fonction objectif. Il existe plusieurs autres critères de sélection qu'il peut être intéressant d'étudier pour choisir le plus adapté à la situation.
- A partir des parents sélectionnés on va générer des enfants par croisement. Comme pour le principe de la reproduction, on va croiser les chromosomes de chaque parent afin d'obtenir un enfant combinant les caractéristiques de chacun. Ce croisement se fait en échangeant des portions de chromosomes de chaque parent, via un ou plusieurs points de croisement. Pour donner un exemple concret, imaginons que nos individus soient représentés par des séquences de valeurs d'une taille fixe. Alors on découpe les séquences de chaque parent en deux parties ou plus et on échange les portions ainsi découpées entre les deux parents, en vérifiant que les croisements effectués n'entraînent pas de doublons de valeurs dans les solutions obtenues. A la suite du croisement, on obtient donc deux enfants qui peuvent être potentiellement meilleurs que leurs parents.
- Enfin, la dernière étape, toujours relativement à la théorie de l'évolution, consiste à faire muter aléatoirement certains individus, en échangeant aléatoirement des portions de chromosomes par exemple. Le pourcentage de mutation au sein de la population doit tout de même rester faible, sans quoi on tombe dans l'aléatoire complet et le principe d'évolution de la population n'est plus respectée. Il est tout de même important d'effectuer des mutations pour ne pas tomber dans des optima locaux.
- Enfin, après avoir obtenu autant d'enfants que l'on a de parents, on choisit les meilleurs parmi les parents et les enfants afin de construire la nouvelle population sur laquelle on va répéter le processus.

Au cours des itérations on sauvegarde la meilleure solution connue de sorte à pouvoir effectuer nos comparaisons au fur et à mesure.

Au terme, en fonction des solutions obtenues, et notamment dans notre cas on pourra être amené à choisir une solution proposant le meilleur compromis parmi les meilleures solutions.

Pour ce projet, nous avons choisi conjointement avec l'encadrant de travailler avec un algorithme génétique puisque cela constitue une approche encore inexplorée pour notre problème. Néanmoins, dans la littérature, il existe plusieurs algorithmes génétiques utilisées pour des problèmes de type flow shop hybride et ces approches pourront nous être utile lors de l'implémentation de notre algorithme.

3 Le problème d'affectation des molécules (Bin Packing avec conflits)

3.1 Présentation du problème

Le problème original du bin packing est le suivant : on cherche à ranger des objets de tailles variés dans des paniers ayant une certaine contenance. Le but est d'utiliser le moins de paniers possibles pour ranger tous les objets. Dans notre cas, si on considère que les paniers sont nos flacons de molécules et que les objets à ranger sont les jobs à affecter aux flacons, alors on se ramène au problème du bin packing. Il existe cependant une subtilité dans la présentation du problème, en effet dans notre cas, on ne peut pas affecter les jobs aux molécules comme on le souhaite. Par exemple, on ne peut pas affecter deux jobs réalisés simultanément dans deux isolateurs différents à un même flacon de molécule, c'est physiquement impossible. On parle de donc de bin packing avec conflits. On définit alors un graphe de conflits qui représente chaque jobs avec des arcs liant les jobs qui entrent en conflits les uns avec les autres.

Le problème du bin packing est également un problème NP-difficile ce qui complexifie encore plus le problème général. Il existe plusieurs méthodes pour résoudre le problème du bin packing, notamment des méthodes exactes comme pour le problème du flow shop hybride mais également des heuristiques célèbres.

On peut notamment citer la méthode exacte qui utilise la modélisation linéaire pour résoudre le problème via l'utilisation d'un solveur. C'est le choix qu'a fait Alexis Robbes dans son travail pour l'UBCO qu'on a expliqué plus en détails dans la partie sur l'ordonnancement. Cette méthode est fiable et très efficace pour résoudre ce problème, ce qui s'explique par le fait que les contraintes du problème se formalisent très bien en fonctions linéaires. On parle de relaxation linéaire ou relaxation continue des contraintes du problème.

L'inconvénient de cette méthode se situe dans la réalisation de la résolution : l'utilisation d'un solveur. Si on souhaite utiliser les résultats fournis par le solveur, on doit trouver un moyen de faire communiquer notre algorithme génétique de résolution du problème d'ordonnancement avec le solveur. Cela peut s'avérer compliqué à mettre en œuvre et entraîner des temps de traitement long en plus d'imposer l'utilisation de deux outils différents simultanément.

En revanche, il existe également des heuristiques de très bonne qualité, que l'on peut facilement intégrer à notre algorithme génétique puisqu'il s'agit d'algorithmes simples que l'on peut inclure à notre traitement au moment de l'évaluation de notre solution notamment.

3.2 Les heuristiques de résolution du Bin Packing avec conflits

Dans cette section nous allons évoquer rapidement deux heuristiques intéressantes pour résoudre le problème du bin packing avec conflits :

3.2.1 First-Fit Decreasing

Cette méthode fournit de bons résultats et est simple à mettre en œuvre. Il est nécessaire de disposer de tous les jobs et de tous les flacons de molécules disponible avant de lancer l'exécution de l'algorithme. On devra donc avoir préalablement construit un ordonnancement avant de pouvoir attribuer les jobs aux flacons.

On commence par trier les jobs par volume requis décroissant, ainsi, on dispose des jobs demandant le plus grand volume de matières en premier. Une fois le tri effectué, on parcourt nos flacons disponibles et on affecte le job en cours au premier flacon contenant le volume nécessaire, tout en vérifiant que le flacon en question ne dispose pas déjà de jobs affectés avec lesquels le job courant entrerait en conflit.

3.2.2 Best-Fit Decreasing

Cette méthode fournit également de bons résultats et est tout aussi simple à mettre en œuvre que la première. On doit là aussi disposer de tous les jobs affectés aux machines, puis les trier en ordre décroissant par volume requis.

Le principe est alors d'affecter le job courant au flacon dont le volume restant est le minimum par rapport aux autres flacons tout en vérifiant également qu'il n'y a pas de conflits avec d'autres jobs du flacon.

Ces deux heuristiques sont les plus connues et les plus efficaces. Nous essaierons donc l'une et l'autre dans notre problème afin de vérifier laquelle des deux nous permet d'obtenir les meilleurs résultats.

4 Optimisation multiobjectif

4.1 Présentation du problème multiobjectif

Le problème que l'on traite dispose de plusieurs objectifs qui sont en contradiction :

- D'un côté, on doit minimiser la somme des retards de chaque job pour satisfaire au mieux les patients.
- De l'autre côté, on doit minimiser les pertes, ce qui revient à minimiser le nombre de flacons ouverts en cherchant à affecter le plus de jobs possibles à chaque flacon.

Ces deux objectifs sont contradictoires puisque d'un côté, on va avoir tendance à rapprocher les jobs entre eux dans l'ordonnancement pour terminer la production le plus rapidement possible. On ne laissera pas de temps de latence entre deux jobs s'il est possible de les réaliser directement l'un après l'autre par exemple. De l'autre côté, pour l'affectation des jobs aux flacons, on va justement avoir tendance à mettre des temps d'attente et à faire commencer les jobs plus tard. Si on reprend l'exemple des deux jobs qui se suivent, on peut par exemple imaginer un troisième job sur un autre isolateur que les deux premiers. Ce job commencerait sa production peu avant la fin de la complétion du premier job sur l'autre isolateur. Imaginons maintenant que le deuxième et le troisième job aient besoin de la même molécule, si on s'arrange pour mettre assez de temps d'attente entre le premier et le deuxième job dans le premier isolateur, on pourra potentiellement réutiliser le flacon ouvert pour le troisième job pour produire le deuxième. Cela permet d'éviter d'ouvrir deux flacons pour réaliser deux jobs qui pourraient ne nécessiter l'ouverture que d'un flacon. On comprend donc que les deux objectifs de ce problème sont contradictoires, on va donc devoir trouver au final une solution qui est un bon compromis entre ces deux objectifs.

C'est ce qu'on appelle l'optimisation multi-objectif. On va chercher à optimiser chacune des fonctions objectif de notre problème pour trouver un ensemble de solutions optimales. Cet ensemble de solutions optimales est appelé front de Pareto. On cherchera donc à trouver des solutions appartenant à cet ensemble, ce sont des solutions qu'on peut qualifier de non dominées. Cela signifie que pour chaque solution du front de Pareto, aucune d'entre elle ne peut être considérée comme moins bien qu'une solution n'appartenant pas à cet ensemble. Voici un schéma illustrant un front de Pareto pour un problème où les deux fonctions objectif sont notées $f1$ et $f2$:

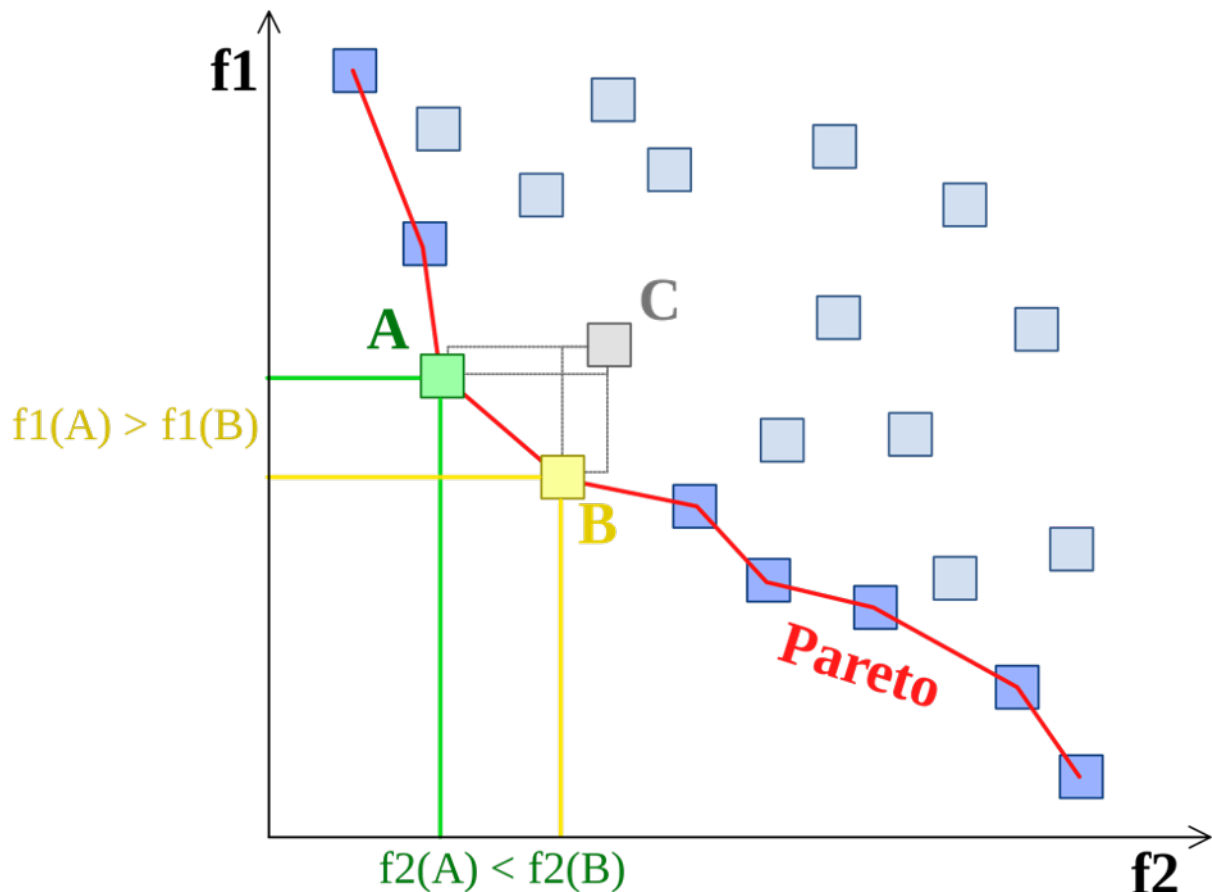


Figure 3.2 – Exemple de front de Pareto pour une fonction à double objectifs

On constate deux choses :

- Aucune des solutions du front de Pareto n'est meilleure que l'autre. En effet, si on prend deux solutions de cet ensemble, peu importe lesquelles, alors l'une des deux sera meilleure pour l'une des fonctions objectif et inversement.
- Aucune des solutions du front de Pareto n'est moins bien que les solutions qui ne s'y trouvent pas. Les solutions ne faisant pas partie du front de Pareto sont dites « dominées ». C'est le cas de la solution C sur l'illustration, elle est battue par A et B puisque ces deux solutions minimisent plus les deux fonctions objectif.

Il existe plusieurs approches pour résoudre ce type de problème.

4.2 Les différentes approches

- L'approche choisie par Thibault Drevon dans son PFE, était d'établir une combinaison linéaire des deux critères à l'aide de coefficients en donnant un poids plus important à la

satisfaction du critère d'ordonnancement. C'est une méthode simple et qui peut se montrer efficace. On peut toujours expérimenter différents poids sur les deux critères et choisir celui qui apporte les résultats les plus convaincants. L'intérêt de cette méthode est de se ramener à un problème mono-objectif et donc limiter la complexité du modèle. Cependant, on ne pourra pas accéder à l'ensemble des solutions du front de Pareto à moins de tester l'ensemble des poids possibles ce qui serait une perte de temps. On obtient tout de même des solutions du front de Pareto en réalisant cette méthode.

- Une autre façon de se ramener à un problème mono-objectif est de définir un ordre lexicographique. On priorise alors certaines fonctions objectif par rapport à d'autres. Dans notre cas, on pourrait vouloir prioriser l'objectif de satisfaction du critère d'ordonnancement qui sera toujours prioritaire à l'autre. On cherchera donc d'abord à minimiser la fonction objectif du critère d'ordonnancement, puis on cherchera ensuite à minimiser le critère d'affectation des jobs aux flacons. Le problème est le même que pour la combinaison linéaire. On ne peut pas obtenir toutes les solutions du front de Pareto.
- Il est également possible de conserver une approche multiobjectif, nous permettant ainsi d'obtenir l'ensemble des solutions du front de Pareto. C'est notamment le rôle des algorithmes Greedy et NSGA2. Ces algorithmes sont des approches plus complexes, qui nécessitent plusieurs évaluations et de nombreux traitements supplémentaires. Cependant, dans l'optique de l'utilisation d'un algorithme génétique, l'utilisation de l'algorithme NSGA2 pourrait être judicieuse puisqu'il s'agit également d'un algorithme génétique qui ajuste les populations générées au front de Pareto. On aurait alors besoin d'adapter notre algorithme pour le faire correspondre au NSGA2.

5 Rapprochements avec le problème original

Nous avons passé en revue les différents problèmes qui se posent à nous ainsi que les principales méthodes permettant de les résoudre dans la littérature.

Pour ce qui est du problème de l'ordonnancement, nous avons vu qu'il pourrait être judicieux d'utiliser un algorithme génétique qui offre de bonnes performances et qui n'a encore jamais été utilisé dans le cadre précis de notre problème.

Si on choisit un algorithme génétique pour tenter de trouver la solution optimale à notre problème, ou du moins s'en rapprocher, on devra également résoudre le problème d'affectation des jobs aux flacons de molécules, le Bin Packing. On pourrait intégrer une heuristique de résolution du Bin Packing adaptée à notre problème dans la génération des solutions initiales avant de laisser l'algorithme génétique explorer plusieurs populations par croisement puis mutation. On pourrait aussi essayer de mettre en œuvre un opérateur au sein de l'algorithme génétique permettrait de trouver la meilleure solution du Bin Packing une fois l'ordonnancement réalisé suite au croisement et aux éventuelles mutations.

Quoiqu'il en soit, il sera sûrement nécessaire d'expérimenter plusieurs approches si cela est possible afin de déterminer laquelle est la plus adaptée à notre problème. Si on choisit de résoudre le Bin Packing au moment de la génération de la population initiale, on devra trouver la solution du Bin Packing pour chacune des solutions d'ordonnancement initiales. Dans ce cas, on pourrait rester sur une approche multiobjectif, ou bien monoobjectif en choisissant d'utiliser une combinaison linéaire des fonctions objectif pour la fonction objectif globale. Si on choisit de résoudre le problème de Bin Packing après avoir réalisé les croisements et les mutations, on se penche alors sur une approche mono objectif qui consistera à prioriser le critère d'ordonnancement plutôt que le critère d'affectation des jobs aux flacons.

4

Analyse et conception

Dans cette section, on s'attardera à étudier le problème plus en détails dans le but de fournir une analyse complète. On proposera notamment une modélisation mathématique du problème.

1 Identification du problème

Le problème que l'on cherche à résoudre est un double problème :

- D'une part, on souhaite ordonnancer des tâches, en l'occurrence les préparations de chimiothérapies. Ce problème s'apparente au problème d'ordonnancement flow-shop hybride. Cela signifie que l'on souhaite ordonnancer les tâches en différentes étapes. Les différentes étapes sont, la stérilisation, la production puis le contrôle. Nous sommes donc ici sur un atelier à 3 niveaux possédant chacun des machines parallèles c'est-à-dire que chaque niveau traite simultanément plusieurs tâches. Dans notre cas par exemple, la phase de stérilisation va traiter plusieurs paniers sur chaque isolateur, on peut donc lancer plusieurs batch simultanément sur les stérilisateur de chaque isolateur. Cela constitue le premier niveau de l'atelier, le deuxième étant les opérateurs chargés de préparer les solutions dans les isolateurs et de la même manière plusieurs opérateurs peuvent fabriquer des produits simultanément. C'est le deuxième niveau de notre atelier. Enfin, le troisième et dernier niveau de l'atelier est la phase de contrôle, ici réalisé par une seule machine ce qui pourra amener à un goulot d'étranglement dans le cas où plus de produits à contrôler parviennent des isolateurs que la capacité de contrôle du contrôleur.
- D'autre part, on cherche à minimiser les pertes, et donc à choisir un ordonnancement qui permettrait de réutiliser le plus possible les flacons de matières entamés. Cela s'apparente à un problème de consommation des ressources également connu sous le nom de Bin Packing. On va chercher à assigner des flacons de matières premières à des préparations.

Ces deux problèmes sont NP-difficile ce qui signifie qu'il n'est pas possible de trouver de solution optimale en un temps polynomial. On va donc chercher à trouver une modélisation du problème dans un premier temps, afin de définir les paramètres, les variables, les contraintes ainsi que les fonctions objectif correspondant aux retards et aux pertes à minimiser.

2 Analyse du problème

Tout d'abord, commençons par détailler les données du problème et les différents éléments qui

rentrent en jeu :

2.1 Les ressources

Pour réaliser une production de chimiothérapie, plusieurs ressources entre en jeu :

- Les molécules : ce sont les matières premières, elles se présentent sous forme de flacons de différents volumes ayant une contenance en mg de molécules. Chaque molécule dispose d'une date de péremption après ouverture et d'un prix unitaire qui, comme évoqué précédemment peut être très élevé, d'où l'intérêt d'une bonne gestion pour limiter au maximum les dépenses. Les flacons de molécules, qu'ils soient ouverts ou non sont stockés dans un réfrigérateur dont la capacité peut être considéré comme infinie. Il est également possible de conserver les flacons ouverts ou non dans les isolateurs mais la durée de conservation est moindre et doit être en adéquation avec les productions en cours pour éviter de devoir ranger les flacons dans le réfrigérateur et donc de perdre du temps sur l'ordonnancement.
- Les isolateurs : ce sont des grandes machines qui permettent de fabriquer les produits dans un environnement stérile évitant ainsi toute contamination extérieure. Ils ont une contenance de 16 paniers ce qui permet de traiter plusieurs préparations simultanément. Dans les isolateurs du CHU de Bretonneau, il y a deux lignes de production ce qui signifie que deux opérateurs peuvent travailler dans un isolateur et interagir avec tous les paniers stérilisés pour faire leurs préparations. Chaque isolateur dispose de deux stérilisateur sauf un qui sert à faire des tests cliniques comme expliqué précédemment. Se référer aux spécifications pour avoir une photo des isolateurs de l'UBCO.
- Les stérilisateur : Ils sont intégrés aux isolateurs et permettent comme leur nom l'indique de stériliser les paniers de préparations qui contiennent les flacons de molécules ainsi que des outils et d'autres solutions nécessaires pour diluer les produits et ainsi obtenir le bon dosage dans la préparation. La nouveauté par rapport aux précédents projets sur ce sujet est la présence de deux chambres de stérilisation par isolateurs ce qui permet en pratique de pouvoir lancer deux stérilisations différées comme nous avons pu le voir dans les spécifications. Conjointement avec monsieur Billaut, nous avons décidé de considérer le système comme il était précédemment, c'est-à-dire avec un stérilisateur par isolateur ce qui facilitera la modélisation et l'implémentation algorithmique par la suite. Cela est possible puisque chaque opérateur peut accéder aux paniers de chacune des chambres de stérilisation. La durée d'une phase de stérilisation varie en fonction du nombre de panier à stériliser, en moyenne elle est de 15 minutes mais celle-ci peut être diminuée si, dans l'urgence d'une préparation on remplit moins le stérilisateur mais cela reste exceptionnel.
- Les opérateurs : ce sont les personnes qui effectuent les préparations dans les isolateurs. Ils ont accès aux produits et peuvent les manipuler à l'aide de gants intégrés aux stérilisateur. Une photo des postes d'opérateur sur un isolateur de l'UBCO est disponible sur la Figure 4.1. On y voit notamment la possibilité d'accéder aux paniers provenant des deux chambres de stérilisation comme souligner dans le précédent point. Les opérateurs sont des employés et ils ont donc des durées d'indisponibilités qui rentrent en compte dans la planification des préparations. C'est notamment le cas pour les soucis liés à la durée de conservation d'un flacon ouvert dans l'isolateur qui est inférieure à sa durée de conservation dans le « vrai » réfrigérateur ou sont stockés les produits. Par exemple, on ne lancera pas de production si les opérateurs ne peuvent pas finir leur tâche avant une durée d'indisponibilité (pause ou fin de journée) ce qui aurait pour conséquence de devoir ranger les flacons ouverts dans les réfrigérateurs et donc une mauvaise optimisation des temps de productions.

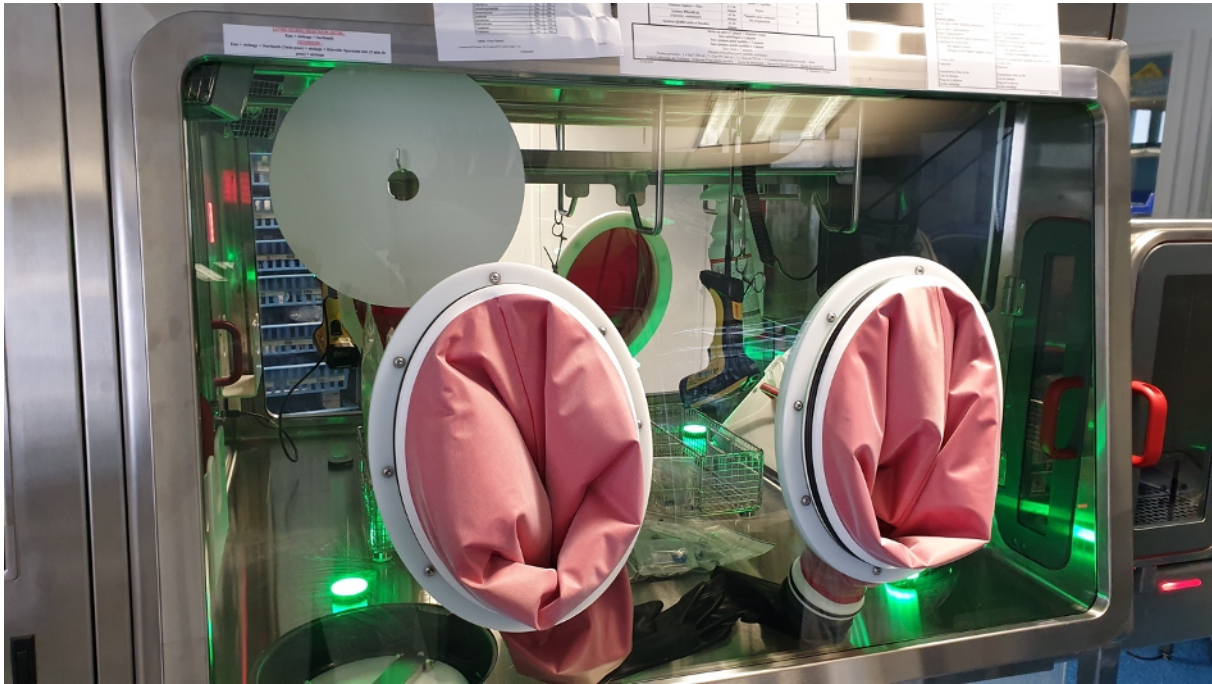


Figure 4.1 – Poste de travail d'un opérateur à l'UBCO de Tours

2.2 Les jobs

Nous venons de voir en détails les ressources dont nous disposons pour réaliser les produits de chimiothérapies, intéressons-nous maintenant aux tâches en elles-mêmes, c'est-à-dire les différentes préparations à réaliser et quels en sont les paramètres, variables ainsi que les contraintes.

Dans la suite du document on parlera de job pour décrire les préparations à effectuer. Par ailleurs, on notera « nb_jobs » le nombre de jobs à effectuer sur l'ensemble de la journée.

Les jobs seront attribués à des batchs pour passer en stérilisation dans les isolateurs, ou ils seront ensuite traités par les opérateurs.

Il est composé de plusieurs paramètres ainsi que de variables qui vont être déterminées au cours du traitement, que nous avons pu relever dans notre diagramme de classe des spécifications. Voici ces données ainsi que les variables à déterminer :

- Les paramètres :
 - Une date de validation : R_j avec $j \in [1, \text{nb_jobs}]$
 - Une date due : D_j avec $j \in [1, \text{nb_jobs}]$
 - Un volume requis : V_{rj} avec $j \in [1, \text{nb_jobs}]$
 - Un volume produit : V_{pj} avec $j \in [1, \text{nb_jobs}]$
 - Une molécule : M_j avec $j \in [1, \text{nb_jobs}]$
 - Une durée de stabilité du produit : E_j avec $j \in [1, \text{nb_jobs}]$
 - Un retard maximum : d_j avec $j \in [1, \text{nb_jobs}]$
 - Un temps de stérilisation : S_j avec $j \in [1, \text{nb_jobs}]$
 - Un temps de préparation : P_j avec $j \in [1, \text{nb_jobs}]$
 - Un temps de contrôle : C_j avec $j \in [1, \text{nb_jobs}]$
- Les variables :
 - Une date de stérilisation : D_{sj} avec $j \in [1, \text{nb_jobs}]$
 - Une date de production : D_{pj} avec $j \in [1, \text{nb_jobs}]$
 - Une date de contrôle : D_{cj} avec $j \in [1, \text{nb_jobs}]$
 - Un identifiant de batch : id_{Batch_j} avec $j \in [1, \text{nb_jobs}]$

Comme expliqué précédemment, les jobs seront répartis dans des classes de données qui seront les batchs. Ils auront chacun des variables qui permettront de les affecter à des opérateurs pour la phase de production.

2.3 Les contraintes

Lors de l'ordonnancement de la production de chimiothérapies, il y a plusieurs contraintes qui rentrent en jeu. Nous allons ici les formaliser en langage naturel, nous les modéliserons mathématiquement dans la section « Modélisation du problème ».

Tout d'abord, nous avons des contraintes propres à l'ordonnancement lui-même :

- On ne peut pas lancer un job en stérilisation avant sa date de validation (sauf cas exceptionnels) (Contrainte 1)
- On ne peut pas affecter les jobs d'un batch à d'autres opérateurs que ceux de l'isolateurs ou le batch a été stérilisé. Cela signifie qu'un job ne peut pas être commencé dans un isolateur et fini dans un autre. (Contrainte 2)
- On ne peut pas lancer la préparation d'un job durant les périodes d'indisponibilité d'un opérateur (les heures de pauses et les périodes hors travail). (Contrainte 3)
- On ne peut pas finir de préparer un job durant les périodes d'indisponibilité d'un opérateur. (Contrainte 4)
- On ne peut pas lancer le contrôle d'un job que durant les heures de travail. (Contrainte 5)
- On ne peut pas finir le contrôle d'un job que durant les heures de travail. (Contrainte 6)

Ensuite, nous avons les contraintes d'affectation des jobs aux flacons :

- On ne peut pas affecter deux jobs qui sont affectés à des isolateurs différents en même temps à un même flacon, car il n'est pas possible de faire passer le flacon d'un isolateur à l'autre sans repasser par la phase de stérilisation qui ferait perdre du temps. (Contrainte 7)
- On ne peut pas affecter deux jobs à un même flacon tel que la durée entre les deux jobs implique un dépassement de la date de péremption du flacon. (Contrainte 8)

3 Modélisation du problème

3.1 Modélisation mathématique des contraintes

On va maintenant modéliser mathématiquement le problème afin de mieux comprendre les contraintes et définir les variables, les paramètres et les contraintes plus explicitement :

On doit construire un ordonnancement sur horizon de 30 jours. Prenons comme une unité de temps la minute. Sur 30 jours on a donc $30 \times 24 \times 60$ minutes soit 43200 minutes ou 43200 unités de temps.

Conjointement avec l'encadrant, nous avons décidé d'attribuer des dates de validation R_j allant du début de la journée de travail, soit 8h le matin, ou en unités de temps à la date $480 + k \times 1440$ ou k est le jour, $k \in [0, 29]$, à 15h ou $900 + k \times 1440$, soit trois heures avant la fin de la journée, laissant ainsi le temps nécessaire pour terminer toutes les préparations de la journée. On a donc $R_j \in [480 + k \times 1440, 900 + k \times 1440]$

En ce qui concerne les dates dues D_j , on fixe celles-ci comme suit : $D_j = R_j + x$, avec $x \in [60, 90]$

Les temps de production sont eux des durées comprises dans l'ensemble 5,10,15,20.

Concernant les ressources, chaque opérateur est un employé qui a des périodes d'indisponibilités comme on a pu le voir précédemment. Les dates d'indisponibilités de chaque opérateur sont donc de 18h (fin de journée) ou $1080 + k \times 1440$ à 8h le lendemain ou $480 + (k+1) \times 1440$. En outre, nous devons aussi considérer les dates d'indisponibilités dues aux pauses des employés pour déjeuner. Si on considère que chaque employé est libre de prendre une pause d'une heure pour manger entre 11h30 et 13h00, alors on note la durée d'indisponibilité comme allant de $690 + k \times 1440$ à $780 + k \times 1440$ avec une durée de 60 minutes maximum. Il n'est donc pas possible pour un opérateur de prendre une pause après la date $720 + k \times 1440$.

Pour reprendre les contraintes qu'on a écrit en langage courant dans la section « Analyse du problème », nous devons maintenant les modéliser mathématiquement.

Commençons par modéliser les contraintes d'ordonnancement :

Contrainte 1 : $Ds_j \geq R_j$

Contrainte 2 : Cette contrainte est compliquée à modéliser mathématiquement, cependant on s'assurera dans le code qu'un job ne puisse pas être stériliser dans isolateur puis préparer dans un autre.

Comme les heures de pauses des opérateurs peuvent varier, on notera celle-ci H_o pour un opérateur o .

Contrainte 3 : $Dp_j > 480 + k \times 1440 \wedge Dp_j < H_o \wedge Dp_j > H_o + 60 \wedge Dp_j < 1080 + k \times 1440$

Contrainte 4 : $Dp_j + P_j > 480 + k \times 1440 \wedge Dp_j + P_j < H_o \wedge Dp_j + P_j > H_o + 60 \wedge Dp_j + P_j < 1080 + k \times 1440$

Contrainte 5 : $Dc_j > 480 + k \times 1440 \wedge Dc_j < 1080 + k \times 1440$

Contrainte 6 : $Dc_j + C_j > 480 + k \times 1440 \wedge Dc_j + C_j < 1080 + k \times 1440$

Passons maintenant aux contraintes concernant l'affectation des jobs aux flacons :

Pour ces contraintes, on considère à chaque fois deux jobs j_1 et j_2 qu'on devrait affecter à un même flacon.

Contrainte 7 : $\max(Ds_{j_1}, Ds_{j_2}) > \min(Dp_{j_1} + P_{j_1}, Dp_{j_2} + P_{j_2})$

Pour cette contrainte, on note St_f la durée de stabilité du flacon f en question.

Contrainte 8 : $\max(Dp_{j_1} + P_{j_1}, Dp_{j_2} + P_{j_2}) - \min(Ds_{j_1}, Ds_{j_2}) < St_f$

3.2 Fonctions objectif

Comme nous avons pu le voir précédemment, on cherche à minimiser les retards des jobs, c'est-à-dire la différence entre le temps qu'on devrait mettre pour produire une solution en sachant sa date de validation et son temps de préparation et le temps qui est réellement nécessaire à la préparation de cette solution. Cela constitue la première composante de notre fonction objectif : on note T_j le retard du job j avec $j \in [1, nb_jobs]$ et on a alors : $T_j = \max((Dc_j + C_j) - D_j, 0)$

On va donc chercher à minimiser $\sum T_j$ d'une part pour minimiser les retards de l'ensemble des jobs et ainsi satisfaire au maximum les patients en réduisant leur temps d'attente.

D'autre part, on va chercher à minimiser les pertes, c'est-à-dire les flacons ouverts, non terminés et qui ne sont pas utilisés avant leur date de péremption et qu'on doit donc jeter. On note V_r le volume restant d'un flacon ouvert, notons Vr_m le volume total restant de la molécule m parmi les flacons ouverts. On cherche alors à minimiser le rapport suivant qui correspond aux pertes d'une molécule m et qu'on note $Sp_m = \max(Vr_m/V_m, 0)$ ou V_m est le volume total disponible pour une molécule m .

On considère ce rapport puisque le volume ouvert d'une molécule correspond à un volume qui va potentiellement se perdre. On cherche donc à avoir le plus petit volume de molécule ouvert en optimisant leur utilisation. C'est là tout l'intérêt de la gestion des reliquats.

Enfin, la somme qu'on cherche à minimiser est donc $\sum Sp_m$. Il s'agit de la deuxième composante de notre fonction objectif.

Nous procéderons de la façon suivante, en suivant la procédure décrite dans la partie « état de l'art » : Tout d'abord on va construire une combinaison linéaire en attribuant des coefficients à ces deux objectifs en testant différents poids. On pourra également expérimenter l'ordre lexicographique qui consiste à résoudre successivement chaque objectif en priorisant le critère d'ordonnancement de sorte à minimiser ces deux objectifs l'un après l'autre.

Ensuite, on pourra tenter de mettre en place une optimisation multiobjectif en utilisant notamment l'algorithme NSGA2 qui permet d'obtenir l'ensemble du front de Pareto pour un problème donné.

3.3 Le problème du Flow Shop hybride

Suite à l'étude du système, il apparaît qu'il est possible de fusionner les deux premiers étages de notre atelier, on aura alors à déterminer l'affectation des jobs aux opérateurs, et on pourra reconstruire les batchs de stérilisation grâce aux dates de stérilisation. Il est plus simple de procéder ainsi puisque l'on se ramène à problème flow shop hybride à deux étages. Un étage est destiné à l'affectation des jobs aux opérateurs et aux stérilisateurs implicitement, puis le deuxième étage est celui de la phase de contrôle avec une seule machine. On a donc un étage avec plusieurs machines parallèles, et un étage avec une seule machine.

Maintenant que nous avons définis les contraintes, les paramètres et variables de jobs, ainsi que les fonctions objectif du problème, il est possible d'utiliser la notation définie par [Graham, 1979] pour caractériser notre problème du flow shop hybride :

Le problème est noté comme suit : FH2, (Pm, \emptyset) | R_j, D_j, S_j, P_j, C_j | $\alpha \sum T_j + \beta \sum Sp_m$.

3.4 Codage de la solution

Nous avons défini un flow shop hybride à deux étages, il nous faut donc trouver une façon de coder la solution pour permettre de retrouver l'ordonnancement grâce à un décodeur.

Pour le moment, nous avons décidé de procéder de la façon suivante : nous disposons d'un flow shop hybride à deux étages dont seul le premier étage contient plusieurs machines parallèles, le deuxième ne dispose que d'une machine. On va donc définir une séquence de jobs qui seront l'ensemble des jobs sur l'horizon de 30 jours. Cette séquence contiendra alors les jobs dans leur ordre de traitement et correspondra à leur date de passage sur les machines du premier étage. On ne construira pas de séquence de jobs pour la machine du deuxième étage, on se servira simplement de la première séquence pour déterminer la date de passage sur la machine du deuxième étage en utilisant une politique de premier job disponible après être passé sur le premier étage.

En ce qui concerne l'affectation des jobs aux ressources, on va construire une séquence de taille égale au nombre de jobs sur l'horizon de 30 jours, dans laquelle chaque case correspondra à un job auquel on affectera un flacon. Comme un flacon peut servir pour plusieurs préparations, il est possible que cette séquence comporte des doublons.

Avec un tel codage, on aura accès à toutes les informations nécessaires pour reconstruire l'ordonnancement au complet. Il sera cependant nécessaire de construire un décodeur pour reconstruire l'ordonnancement à partir de ces deux séquences.

Ce codage est amené à évoluer au cours des expérimentations du deuxième semestre et il est donc nécessaire d'avoir un certain recul et de ne pas s'obliger à utiliser celui-ci en particulier. Il a été défini conjointement avec l'encadrant sur la base d'une étude des données et du problème.

3.5 Algorithmes de résolution

Tout d'abord, il sera nécessaire de définir des algorithmes de génération des instances.

Ensuite, pour trouver de bonnes solutions, nous avons vu dans la partie « état de l'art » qu'il pourrait être intéressant d'utiliser un algorithme génétique.

On peut imaginer plusieurs types de fonctionnement qu'il conviendra d'étudier, voire de tester si besoin :

On pourra tout d'abord utiliser une heuristique pour générer un premier ordonnancement, sans tenir compte des affectations de jobs aux flacons. L'ordonnancement étant toujours le critère prioritaire, on pourra, après avoir trouvé un premier ordonnancement, définir une affectation des tâches aux flacons en utilisant l'une des deux heuristiques présentées dans la partie « état de l'art » pour résoudre le problème du bin packing avec conflits.

Notre algorithme génétique aura alors pour rôle de trouver d'autres solutions. On peut aussi imaginer un opérateur implicite à l'algorithme génétique pour redéfinir la meilleure affectation possible des jobs aux flacons après avoir passé les phases de croisement et de mutation, cependant, cela pourrait être coûteux en temps.

Il va être également nécessaire de mettre au point l'algorithme de traitement du décodage de la solution.

Cette section sera amenée à évoluer au fil des développements et on détaillera plus précisément le fonctionnement de chaque algorithme qu'on utilisera.

4 Génération des instances

Concernant la génération des instances, le travail réalisé par Alexis Robbes est très utile. Cependant, il est à noter que le problème qu'il a traité dans son travail avec l'UBCO est légèrement différent du nôtre.

Pour générer les instances, on va donc se servir des données récoltées par Alexis Robbes et construire les nôtres. On se servira notamment de son travail sur l'étude des stocks de molécules et sur le clustering effectué sur celle-ci afin de créer des groupes représentatifs de molécules. On reprend les 6 groupes de molécules trouvés par Alexis Robbes suite à son traitement de clustering sur l'ensemble des molécules disponibles. Ce travail de clustering est réalisé sous python et permet de générer un fichier csv contenant ces 6 groupes avec les caractéristiques pour chaque type de molécule.

On va donc se servir de ce fichier pour ajouter les molécules disponibles à notre stock initial. Ce stock initial sera contenu dans l'instance. Pour chaque molécule du stock, on dispose d'un pourcentage d'utilisation de la molécule par l'ensemble des jobs, on va donc se servir de la probabilité qu'un job utilise telle ou telle molécule pour créer 150 jobs par jours sur un horizon de 30 jours. Chaque job se voit alors attribué une molécule et un volume requis que l'on définit aléatoirement, inférieur au volume d'un flacon qu'on fixe à 30 ml.

On fixe également aléatoirement sur l'ensemble de la journée une date de validation pour chaque job.

5

Bilan et conclusion

1 Bilan du semestre 9

1.1 Faits

- Étude et prise en main problème
- Analyse du contexte et des objectifs
- Étude du fonctionnement de la production de chimiothérapie
- Étude des travaux déjà réalisés
- Études sur le problème du flow shop hybride
- Études sur le problème du bin packing
- Rédaction du cahier de spécifications
- Identification du problème
- Modélisation mathématique du problème
- Proposition d'algorithmes de résolution du problème (algorithme génétique)
- Rédaction de l'état de l'art

1.2 Reste à faire

- Reprise de l'analyse
- Définition claire de l'algorithme de résolution
- Mise en œuvre du programme :
 - Développement des différentes classes et fonctions propres à celles-ci
 - Développement des fonctions décrites dans les spécifications fonctionnelles (sauf l'algorithme génétique)
 - Développement de l'algorithme de résolution (l'algorithme génétique)
 - Mise en place des tests
 - Expérimentations (tests des développements et de performance)
- Rédaction du rapport final
- Préparation de la soutenance

1.3 Retards

Actuellement, les retards concernent surtout l'algorithme de résolution, il reste encore des points à éclaircir et cela se fera durant la phase de mise en œuvre au fil des expérimentations.

Annexes

A

Planification, gestion de projet

1 Evolution du projet

Voici la première version du diagramme Gantt pour le semestre 9 :

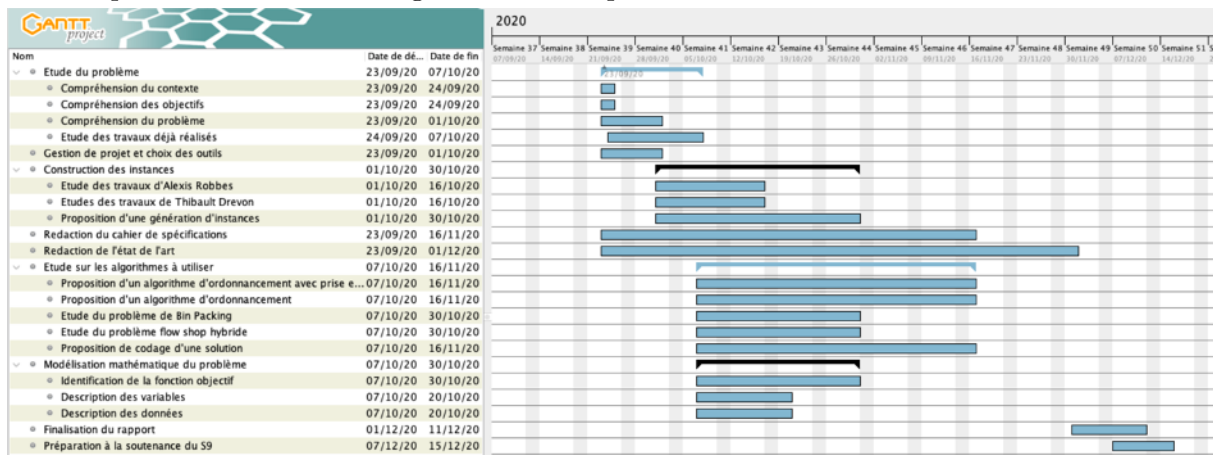


Figure A.1 – Le diagramme de Gantt du S9 initial

Suite au travail effectué et aux différentes contraintes et difficultés rencontrées, il a dû être redéfini pour correspondre aux faits réels. La rédaction du cahier de spécification et de l'état de l'art ont notamment pris plus de temps que prévus en raison de modifications de dernières minutes. La construction des instances et la proposition du codage d'une solutions ont également pris plus de temps que prévu suite à plusieurs redéfinitions en fin de semestre. Globalement, malgré les retards, le travail prévu a pu être réalisé dans son intégralité. Voici le diagramme de Gantt final du semestre 9 :

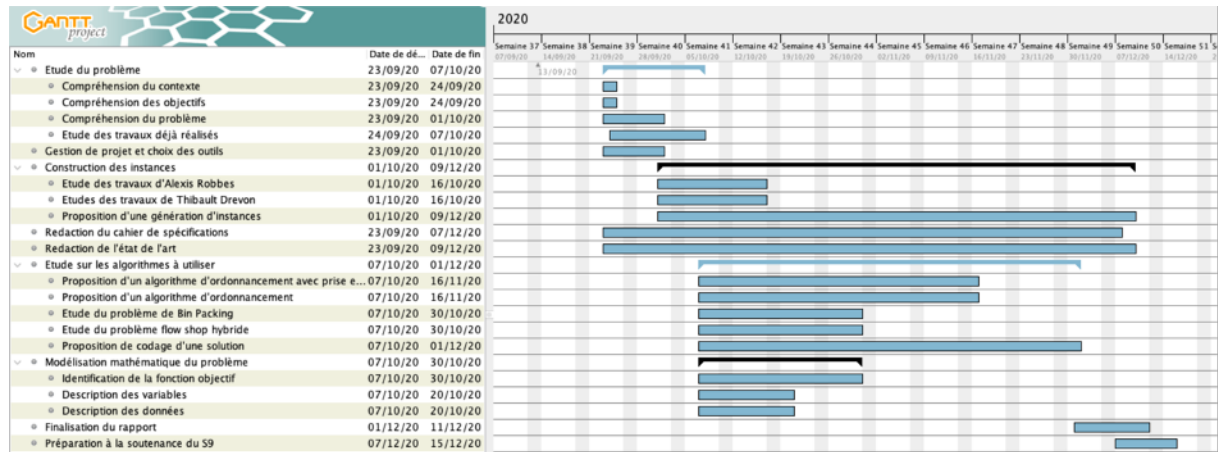


Figure A.2 – Le diagramme de Gantt du S9 final

B

Description des interfaces

1 Interfaces matérielles/logicielles

Concernant le matériel utilisé pour ce projet, l'auteur utilise son propre matériel et réalisera le projet sur un ordinateur portable tournant sous Mac OS et équipé d'un Intel Core i5 double cœur cadencé à 2 GHz et de 8 Go de RAM LPDDR3 cadencé à 1867 MHz. En outre, l'ordinateur est équipé d'un SSD de 256 Go qui peut aussi permettre d'optimiser les performances contrairement à un disque dur classique.

Ce travail est pour le moment effectué dans le cadre de la pandémie de Covid en distanciel à domicile. Il est cependant possible que d'ici la fin du projet, les expérimentations puissent être réalisées à Polytech sur des machines plus puissantes dont la configuration est inconnue à l'auteur. Si les expérimentations venaient à être effectuées sur une autre machine, cette section sera mise à jour.

2 Interfaces homme/machine

Le but de ce projet est purement algorithmique et dans une optique de recherche opérationnelle. L'intérêt n'est donc pas de concevoir une interface mais plutôt de concevoir un traitement qui optimisera au mieux les temps de traitement et qui permettra d'obtenir les meilleures solutions possibles.

On ne s'attardera donc pas trop à développer une interface poussée, mais on mettra tout de même en place une petite IHM qui permettra à l'utilisateur d'interagir avec le programme et ses différentes fonctions. On mettra par exemple un bouton permettant de générer les instances, puis un autre pour résoudre le problème. On permettra à l'utilisateur de sauvegarder les résultats dans un fichier du nom de son choix et à l'emplacement de son choix en mettant en place une fenêtre d'enregistrement de fichiers classique.

Les fonctionnalités de l'interface s'arrêteront là puisque ce n'est pas l'objet de ce projet.

3 Interfaces logiciel/logiciel

Dans le cadre de ce projet, il n'y aura pas d'utilisation de base de données, ou même de

communication avec un logiciel externe à celui développé.



Cahier de Spécification

1 spécifications Fonctionnelles

1.1 Fonctionnalités à développer

1.1.1 Fonction d'initialisation du stock (parseur)

Présentation :

- Nom de la fonction : `initialisationStock`.
- Rôle de la fonction : lit le fichier contenant les caractéristiques sur les molécules pour initialiser le stock en créant et en ajoutant les molécules.
- Priorité : primordiale.

Description :

- En entrée, on fournit le chemin du fichier csv à lire. Il n'y a pas de sortie, la fonction se chargera de d'initialiser le stock de l'instance qui est un attribut de celle-ci.
- Pour initialiser le stock, on se sert donc de l'objet stock de l'instance, dans lequel on va créer et ajouter des objets molécules qu'on va eux-mêmes initialiser à partir des données du fichier qu'on va lire.
- Le traitement de cette fonction consiste à lire chaque ligne du fichier, le format csv nous permet de découper les informations de chaque ligne facilement. Chaque ligne nous permet de récupérer toutes les informations pour créer la molécule, son coût et sa durée de stabilité.
- Concernant la gestion des erreurs pour cette fonction, on indiquera à l'utilisateur si le fichier ne correspond pas au bon format, ou bien si le contenu du fichier n'est pas bon.

1.1.2 Fonction de génération des instances

Présentation :

- Nom de la fonction : `generationInstance`.

- Rôle de la fonction : créer les jobs de chaque jour en fonction du nombre de job précisé ainsi que du nombre de jour. On initialise également le nombre d'opérateurs, les machines de notre atelier.
- Priorité : primordiale.

Description :

- Pour le moment, on génère nous même les instances, les entrées sont donc les suivantes : un nombre de jours, un nombre de job et un nombre d'opérateurs. A l'avenir, on pourra imaginer des instances stockées dans des fichiers et qu'on pourrait lire pour initialiser l'instance. Auquel cas l'entrée serait simplement le chemin du fichier en question.
- Pour initialiser l'instance, on va devoir créer des objets jobs.
- Le traitement de cette fonction consiste à créer des jobs en fonction du nombre de jours et du nombre de jobs souhaités. Pour créer chaque job, on utilise le même temps de contrôle et le même temps de stérilisation, correspondant aux données récoltées à l'UBCO, à savoir respectivement 10 et 15 minutes. Pour le temps de production, on définit une valeur qui peut prendre, avec la même probabilité, 5, 10, 15 ou 20 minutes comme on l'a indiqué dans le rapport.
- Concernant la gestion des erreurs pour cette fonction, il ne devrait pas y en avoir, à moins qu'au cours de la phase de mise en œuvre, on crée une fonction de génération des instances à partir d'un fichier source avec le même type d'exceptions que pour la fonction précédente.

1.1.3 Fonction de génération d'un job

Présentation :

- Nom de la fonction : Job.
- Rôle de la fonction : créer les jobs de chaque jour en fonction du nombre de job précisé ainsi que du nombre de jour. On initialise également le nombre d'opérateurs, les machines de notre atelier.
- Priorité : primordiale.

Description :

- Il s'agit du constructeur de la classe job, la fonction n'a donc pas non plus de sortie. Les entrées sont les suivantes : un temps de stérilisation, un temps de production (défini aléatoirement dans la génération de l'instance), un temps de contrôle, et un objet stock, celui de l'instance.
- Lors de la création d'un job, il y a une interaction avec le stock passé en entrée et avec les molécules contenues dans le stock précédemment initialiser.
- Le traitement de cette fonction consiste à créer un job en se servant des paramètres passés en entrée. Le traitement devient particulier lorsqu'on doit assigner une molécule au job. En effet, chaque molécule n'est pas autant utilisée que les autres, on a un pourcentage pour chaque molécule qui correspond à sa part d'utilisation pour l'ensemble des jobs. On va donc se servir de cette probabilité d'utilisation de chaque molécule pour en assigner une au job. On fixe également le volume requis aléatoirement, et on fixe un volume produit, lui fixe de 30 ml pour l'instant.
- Concernant la gestion des erreurs pour cette fonction, il ne devrait pas y en avoir non plus, à moins de passer des valeurs négatives en paramètres pour les temps, cependant, dans une utilisation normale, cela n'est pas censé arriver.

1.1.4 Fonction de génération d'une solution initiale

Présentation :

- Nom de la fonction : generationSolution.
- Rôle de la fonction : C'est la fonction qui va nous permettre de générer une solution initiale, ce sera notre heuristique.
- Priorité : primordiale.

Description :

- En entrée de cette fonction, on passe une instance à résoudre, en sortie on obtient une solution.
- Cette fonction interagit avec les instances et tous les objets qu'elle contient, à savoir les jobs et le stock contenant les molécules et leur quantité.
- Le traitement de cette fonction consiste à générer une première solution. Pour cela, on va se baser sur un principe de « FAM » pour First Available Machine afin de construire un ordonnancement en attribuant chaque job un par un (préalablement triés chaque jour par date de validation croissante) à la première machine disponible. A la suite de cela, on effectuera l'affectation des jobs aux molécules en se basant sur l'une des deux heuristiques de Bin Packing présentées dans la partie « état de l'art ». Enfin, on fournira la solution générée.
- Concernant la gestion des erreurs pour cette fonction, si l'instance fournie est correcte, il n'y aura pas non plus d'exception.

1.1.5 Fonction de résolution (recherche d'une solution optimale)

Présentation :

- Nom de la fonction : resolution.
- Rôle de la fonction : C'est la fonction qui va implémenter l'algorithme génétique permettant de résoudre le problème pour obtenir la meilleure solution possible (pas forcément la solution optimale). C'est la fonction la plus importante de notre programme et celle que nous allons être amené à modifier à de nombreuses reprises, aussi le processus de traitement décrit ci-dessous va surement être modifié durant la phase de mise en œuvre.
- Priorité : primordiale.

Description :

- En entrée de cette fonction, on passe la solution générée grâce à l'heuristique.
- Cette fonction interagit avec la solution, les jobs, les flacons, les opérateurs et également avec le stock.
- Le traitement de cette fonction consiste à utiliser un algorithme génétique pour résoudre le problème. On cherchera donc à générer une population initiale par rapport à la solution initiale. Le traitement détaillé de cette fonction reste à déterminer et cette section sera mise à jour durant la phase de mise en œuvre.
- Concernant la gestion des erreurs pour cette fonction, si la solution fournie est correcte, il n'y aura pas non plus d'exception.

1.1.6 Fonction de décodage d'une solution

Présentation :

- Nom de la fonction : decodageSolution.
- Rôle de la fonction : C'est la fonction qui va nous permettre de décoder la solution qu'on aura trouvé.
- Priorité : primordiale.

Description :

- En entrée de cette fonction, on passe la solution à décoder. Il n'y aura pas de sortie mais on utilisera la fonction d'écriture dans un fichier pour enregistrer la solution. On s'arrangera pour afficher l'ordonnancement réalisé de façon à ce qu'il soit lisible facilement.
- Cette fonction interagit avec la solution fournie et tout ce qui la compose.
- Le traitement de cette fonction consiste à décoder la séquence de jobs et de flacons qui composent la solution dans le but de reconstruire l'ordonnancement en reconstituant les batchs de stérilisation, les dates de production et de contrôle, et enfin l'affectation des jobs aux flacons.
- Concernant la gestion des erreurs pour cette fonction, si la solution fournie est correcte.

1.1.7 Fonction d'écriture des résultats

Présentation :

- Nom de la fonction : `ecritureSolution`.
- Rôle de la fonction : C'est la fonction qui va nous permettre de sauvegarder une solution dans un fichier spécifié.
- Priorité : secondaire.

Description :

- En entrée de cette fonction, on passe une solution et un chemin de fichier.
- Cette fonction interagit avec les éléments d'une solution, elle sera appelée dans la fonction de décodage pour enregistrer les résultats.
- Le traitement de cette fonction consiste à sauvegarder toutes les informations dans le fichier spécifier en paramètres de manière à pouvoir les lire en dehors du programme. Par ailleurs, le fichier sera créé s'il n'existe pas, et complété s'il contient déjà une ou plusieurs solutions.
- Concernant la gestion des erreurs pour cette fonction, si le format du fichier spécifié est incorrect (le format reste à déterminer) on lèvera une exception.

2 Spécifications non fonctionnelles

2.1 Contraintes de développement et conception

Aucun type de matériel n'est imposé pour ce projet.

Le langage de programmation utilisé pour ce projet sera le C++ via l'ide CLion. Ce choix a été fait en concertation avec l'encadrant.

Aucune bibliothèque de programme particulière n'est imposée pour ce logiciel.

2.2 Contraintes de fonctionnement et d'exploitation

2.2.1 Performances

Les performances du programme fournis devront plus ou moins correspondre au programme Planif utilisé actuellement par l'UBCO pour ordonnancer sa production de chimiothérapies. Le logiciel Planif réalise un ordonnancement en cinq minutes. D'un point de vu utilisateur, on cherchera donc à conserver des temps d'exécutions similaires à ce logiciel.

2.2.2 Capacités

Concernant la capacité du logiciel, en théorie, il ne devrait pas y avoir de limite en termes de nombre de job à ordonnancer ainsi que de nombre de jours. Cependant, en pratique, on se limitera à un usage sur un horizon de 30 jours et pour un maximum de 150 jobs par jour ce qui correspond aux capacités réelles de l'hôpital.

2.3 Modes de fonctionnement

Comme on produira une petite interface permettant à l'utilisateur d'interagir avec le logiciel, on fournira un exécutable au terme du développement de sorte à ne pas avoir à passer par une ligne de commande pour utiliser ce logiciel.

Il sera également possible de passer par le terminal et d'exécuter le programme en ligne de commande si souhaité.

2.3.1 Contrôlabilité

Au niveau de l'interface, on utilisera des messages pour indiquer la fin des traitements et les temps d'exécution pour permettre à l'utilisateur d'interagir plus facilement avec le logiciel.

Si le programme est lancé en ligne de commandes, on affichera les informations du traitement au fur et à mesure de celui-ci.

Cette section pourra être complétée durant la mise en œuvre.

2.3.2 Sécurité

Concernant la sécurité du programme, il n'y aura pas contrôle spécifique. Le but est d'intégrer le traitement, s'il fournit des résultats intéressants, au logiciel Planif qui se charge de gérer les connexions et utilisations des différents utilisateurs du système.

D

Bibliographie

- [1] R.L. GRAHAM, E.L. LAWLER, J.K. LENSTRA et A.H.G. Rinnooy KAN. « Optimization and Approximation in Deterministic Sequencing and Scheduling : a Survey ». In : *Discrete Optimization II*. Sous la dir. de P.L. HAMMER, E.L. JOHNSON et B.H. KORTE. T. 5. Annals of Discrete Mathematics. Elsevier, 1979, p. 287-326. DOI : [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X). URL : <http://www.sciencedirect.com/science/article/pii/S016750600870356X>.
- [2] Houda HARBAOUI. « Ordonnancement d'un système de production industriel complexe : flow shop hybride avec des machines dédiées soumis à différentes contraintes temporelles ». Theses. Ecole nationale supérieure Mines-Télécom Atlantique ; Institut supérieur d'informatique et des techniques de communication (Hammam Sousse, Tunisie), déc. 2018. URL : <https://tel.archives-ouvertes.fr/tel-02316138>.
- [3] S. M. JOHNSON. « Optimal two- and three-stage production schedules with setup times included ». In : *Naval Research Logistics Quarterly* 1.1 (1954), p. 61-68. DOI : <https://doi.org/10.1002/nav.3800010110>. eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800010110>. URL : <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800010110>.
- [4] Asma OULED BEDHIEF et Najoua DRIDI. « FLOW SHOP HYBRIDE À DEUX ÉTAGES AVEC MACHINES DÉDIÉES ET DATES DE DISPONIBILITÉ DES JOBS ». In : *MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation*. Colloque avec actes et comité de lecture. internationale. Nancy, France, nov. 2014. URL : <https://hal.archives-ouvertes.fr/hal-01166681>.
- [5] Alexis ROBBES, Yannick KERGOSIEN et Billaut JEAN-CHARLES. « 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)*. Karlsruhe, Germany, 2019. URL : <https://hal.archives-ouvertes.fr/hal-02404168>.
- [6] Alexis ROBBES, Yannick KERGOSIEN et Billaut JEAN-CHARLES. « Minimizing the costs induced by perishable resource waste in a chemotherapy production unit ». In : *17th International Workshop on Project Management and Scheduling*. Toulouse, France, 2021. URL : <https://hal.archives-ouvertes.fr/hal-03018099>.

- [7] Michael S. SALVADOR. « A Solution to a Special Class of Flow Shop Scheduling Problems ». In : *Symposium on the Theory of Scheduling and Its Applications*. Sous la dir. de Salah E. ELMAGHRABY. Berlin, Heidelberg : Springer Berlin Heidelberg, 1973, p. 83-91. ISBN : 978-3-642-80784-8.
- [8] A. VIGNIER, J.-C. BILLAUT et C. PROUST. « Les problèmes d'ordonnancement de type « flow-shop » hybride : état de l'art ». fr. In : *RAIRO - Operations Research - Recherche Opérationnelle* 33.2 (1999), p. 117-183. URL : www.numdam.org/item/RO_1999__33_2_117_0/.

E

Glossaire

Cytotoxique substance toxique pour les cellules. [1](#)

Flow Shop problème d'ordonnancement à plusieurs étages avec machines parallèles. [4](#)

UBCO Unité de biopharmacie clinique oncologique. [1](#)

Ordonnancement de la production de chimiothérapies avec minimisation des pertes :

Jacob ADOR

Encadrement : Jean-Charles BILLAUT, Alexis ROBBES et Yannick KERGOSIEN

Objectifs

- Ordonnancement de production de chimiothérapies
- Limiter les pertes de matières premières
- Satisfaire les patients en limitant les retards



Mise en œuvre

Le but de ce projet est de mettre en place un programme permettant de gérer l'ordonnancement de la production de chimiothérapies de l'UBCO du CHU Bretonneau à Tours dans le but de limiter les pertes de matières premières.



Résultats attendus

- Programme d'ordonnancement
- Gestion des reliquats
- Sauvegarde des résultats
- Sauvegarde des instances



Ordonnancement de la production de chimiothérapies avec minimisation des pertes :

Jacob ADOR

Encadrement : Jean-Charles BILLAUT, Alexis ROBBES et Yannick KERGOSENIEN

Objectifs

- Ordonnancement de production de chimiothérapies
- Limiter les pertes de matières premières
- Satisfaire les patients en limitant les retards

Mise en œuvre

Le but de ce projet est de mettre en place un programme permettant de gérer l'ordonnancement de la production de chimiothérapies de l'UBCO du CHU Bretonneau à Tours dans le but de limiter les pertes de matières premières.

Résultats attendus

- Programme d'ordonnancement
- Gestion des reliquats
- Sauvegarde des résultats
- Sauvegarde des instances



Ordonnancement de la production de chimiothérapies avec minimisation des pertes

Résumé

Le sujet de Projet de Recherche et Développement est "Ordonnancement de la production de chimiothérapies avec minimisation des pertes". Dans ce projet nous étudions le problème d'ordonnancement de la production de chimiothérapies de l'UBCO du CHU de Tours. On s'intéresse notamment à mettre en place un ordonnancement permettant de limiter les pertes de produits cytotoxiques. Ces produits cytotoxiques sont extrêmement onéreux pour certains. L'objectif est double : limiter les pertes ainsi que les retards de production. Nous étudierons le problème en abordant l'aspect ordonnancement, mais également l'aspect affectation des ressources et nous tenterons de mettre en place un traitement efficace pour répondre aux deux objectifs de ce problème.

Mots-clés

Ordonnancement, Flow Shop, Flow Shop hybride, Chimiothérapie, CHU, Ressources, Cytotoxique, Bin Packing, Retards, Consommation

Abstract

The subject of Research and Development Project is "Scheduling of chemotherapy production with minimization of losses". In this project we study the problem of scheduling the production of chemotherapy from the UBCO of the CHU of Tours. We are particularly interested in putting in place a sequencing to limit the losses of cytotoxic products. These cytotoxic products are extremely expensive for some. The objective is twofold: to limit losses as well as production delays. We will look at the problem in terms of scheduling, but also in terms of resource allocation, and we will try to put in place effective treatment to meet both objectives of this problem.

Keywords

Scheduling, Flow Shop, Hybrid Flow Shop, chemotherapy, UHC, Resources, Cytotoxic, Bin Packing, Lateness, Consumption

Tuteurs académiques

Jean-Charles BILLAUT
Alexis ROBBES
Yannick KERGOSIEN

Étudiant

Jacob ADOR (DI5)