

L'ordonnancement dans l'atelier à cheminement unique hybride et production par lots ("batch")

Nous étudions dans ce chapitre le problème d'ordonnancement dans l'atelier à cheminement unique de composition hybride, avec traitement par lots, aussi connu comme production par "batch". La composition hybride de l'atelier fait référence au nombre de machines disponibles pour effectuer chacune des opérations. Le traitement par lots permet la réduction des temps de traitement grâce à une structure en multibroches, qui permet de traiter plusieurs produits au même temps.

Ce problème d'ordonnancement est composé de plusieurs niveaux de décision : d'abord, la composition des lots détermine les dates de disponibilité du lot à traiter. Ensuite, ces lots sont ordonnancés pour effectuer les opérations définies dans l'atelier. Pour les opérations disposant de plusieurs machines (ou équipements) nous devons également considérer les décisions d'affectation pour définir quel lot est traité sur quel machine.

Ces niveaux de décision et leur interaction rendent ce problème particulièrement difficile, principalement car la composition des lots a un impact sur les caractéristiques des lots à ordonnancer, qui est déjà un problème NP-difficile [109]. Par conséquent, le critère d'optimisation (la minimisation du retard total) est impacté par les différents niveaux de décision.

Ainsi, ces niveaux de décision peuvent être résumés sur deux problématiques à considérer : la composition de lots et leur ordonnancement dans l'atelier hybride à cheminement unique.

Au regard de la complexité du problème, l'essentiel des méthodes de résolution dévelop-

pées dans ce chapitre sont approchées. Nous proposons également un modèle mathématique pour compléter la définition du problème.

Pour les méthodes approchées, en raison de la composition hybride de l'atelier décrit, la résolution de ce problème utilise certaines des conclusions issues du chapitre 3, notamment dans l'ordonnancement des opérations disposant de plusieurs équipements identiques. Ainsi, pour les opérations avec machines en parallèle, les méthodes permettant d'identifier la meilleure heuristique à appliquer par rapport aux données d'entrée est utilisée.

Les méthodes développées ont été testées sur des instances théoriques inspirées du cas réel décrit dans le contexte de recherche (chapitre 1). Les résultats obtenus sont utilisés pour évaluer et comparer les différentes méthodes proposées. Les performances relatives aux solutions et aux caractéristiques des méthodes de résolution sont évaluées grâce aux critères définis dans le chapitre 1.

4.1 Introduction

La gestion des ateliers de production où plusieurs opérations sont exécutées sur des produits afin d'obtenir un produit final, est un problème combinatoire dont le caractère NP-difficile a été démontré pour le cas de minimisation du retard total [43]. Si ces opérations peuvent être exécutées par plusieurs équipements, le problème a été prouvé NP-difficile même à partir du cas avec deux opérations [55].

La version étudié dans ce chapitre inclût un niveau de décision supplémentaire pour construire les lots de traitement (définir les données d'entrée du problème d'ordonnancement), et déplace également l'évaluation du retard total jusqu'à une décomposition des lots afin de livrer les commandes initiales. En effet, nous considérons dans cette version du problème, que les produits arrivent à l'atelier dans des commandes, et par conséquent, le retard est calculé sur la date de livraison de ces commandes (qui peut nécessiter le traitement de plusieurs lots).

Le traitement par lots est autorisé afin de permettre l'utilisation des équipements du type multibroches : dans un seul passage, un équipement peut traiter plusieurs produits, dans un délai identique à celui nécessaire pour traiter un seul produit à la fois. La composition de lots est néanmoins contrainte de respecter un seul type de produit par lot, car plusieurs types de produits sont considérés.

La structure générale de ce problème, correspond à des cas réels dans l'industrie dans des différents domaines et il a attiré l'attention des chercheurs par sa complexité et l'adaptabilité

à des cas industriels [109].

Tel que nous l'avons décrit dans le chapitre 1, l'objectif est la minimisation du retard total, mesuré sur la date de livraison de la commande (chaque commande contient des produits de différent type).

Ce problème peut être considéré comme nouveau, car aucune référence ne traite, à notre connaissance, un atelier avec toutes les contraintes décrites. Dans la littérature, les problèmes similaires ordonnent des ateliers du type hybride avec la composition des lots de traitement, mais aucune ne considère pas le critère de minimisation de retard sur la livraison de commandes. Par rapport à l'objectif fixé, seulement quelques références minimisent le retard total [109]. Le problème d'ordonnement à cheminement unique et minimisation du retard total a été traité par [79]. Depuis, aucune autre référence traite ce problème.

Dans la composition des lots, nous considérons des incompatibilités entre les produits : différentes natures de produits sont définies et leur traitement diffère d'un type de produit à l'autre, en suivant une unique séquence. La nature des produits, détermine les temps de traitement par opération.

Dans le contexte de recherche introduit dans le chapitre 1, nous avons décrit le fonctionnement du laboratoire d'analyses physico-chimiques, comme étant un système ressemblant l'atelier décrit précédemment. Les développements de ce chapitre sont appliqués à ce cas d'application dans le chapitre 5.

Ce chapitre propose des méthodes de résolution pour le problème d'ordonnement décrit, avec des algorithmes qui incluent les décisions liées à la composition des lots et à leur ordonnancement. La section suivante présente le problème traité avec la nomenclature à utiliser. Un modèle mathématique et des méthodes de résolution heuristiques sont présentés dans des sections ultérieures.

4.2 Présentation du problème

Dans la définition classique du problème d'ordonnement à cheminement unique, un ensemble de produits N , doivent suivre un ensemble d'opérations S à effectuer sur des machines qui sont exclusivement dédiées à chacune des opérations. Ce problème est considéré NP-difficile à l'exception de certains cas avec deux opérations [40]. Chacune des machines ne peut traiter qu'un seul produit à la fois, ou bien chaque produit ne peut être traité que par une seule machine. Nous considérons ici la version standard de ce problème d'ordonnement, les stocks entre les opérations sont de capacité infinie, et les produits et machines sont

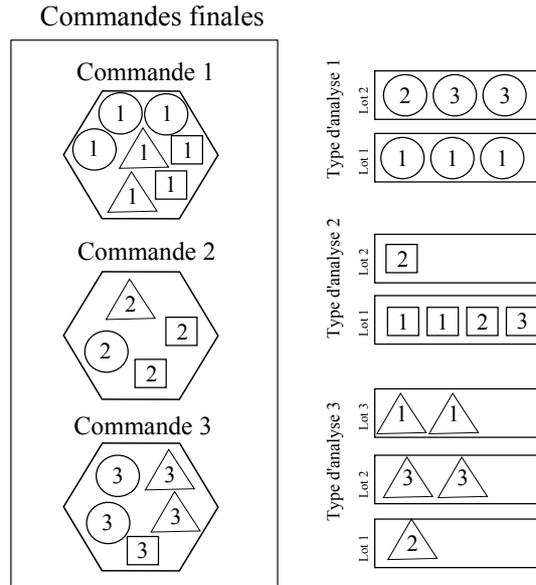


FIGURE 4.1 – Exemple de composition des lots

disponibles pendant toute la période considérée.

Pour le cas étudié dans ce chapitre, une seule séquence des opérations est utilisée pour tous les produits traités, où chaque opération est exécutée une seule fois sur chaque produit. Les opérations sont exécutées par une ou plusieurs machines (composition hybride de l'atelier). Chaque machine ne peut effectuer qu'un seul type d'opération.

Nous considérons que les produits arrivent à l'atelier dans des commandes. Une commande est composée de plusieurs produits. Différents types de produits sont considérés. Chaque commande peut contenir différents types de produits. Toute commande o a une date d'arrivée (R_o) et une date de fin souhaitées (D_o). Chaque commande est considérée finie quand tous ses produits ont été traités.

De la même manière, nous pouvons définir des dates d'arrivée (r_i) et de fin souhaitée (d_i) pour chacun des produits.

Le traitement des produits peut être effectué individuellement : un seul produit est traité à la fois par chaque équipement disponible. Afin de réduire les délais de livraison des commandes, le traitement par lot est autorisé. Ainsi, des produits de différentes commandes peuvent être traités ensemble, pendant toutes les opérations de l'atelier. Ces lots peuvent être formés à condition qu'ils soient rassemblés dès la première opération, et qu'ils soient relatifs au même type de produit. Pour être traité, des compatibilités entre les produits d'un seul lot doivent être respectés : si I_u est l'ensemble de produits du lot u , et g_i le type de produit du produit i , donc $g_i = g^* \forall i \in I_u$, où g^* est le type de produit traité dans le lot u .

La figure 4.1 inclut un exemple de la procédure pour la construction des lots de traitement. Pour cet exemple, un total de 16 produits arrivent à l'atelier, dans trois commandes (Commandes 1, 2 et 3). Un total de 3 différents types produits sont à considérer, représentés dans la figure avec des cercles, triangles et carrés. Les lots sont composés, en respectant la contrainte d'unicité du type produit par lot.

Le principal avantage du traitement par lots est l'utilisation complète de la capacité des équipements, car nous utilisons des équipements multibroches : plusieurs produits peuvent être traités au même temps. Ainsi, sur une seule machine, traiter μ produits (du même type) demande la même quantité du temps que traiter un seul produit du même type. Les temps de traitement sont définis par type de produit, et par opération : pour une opération s et un type de produit g^* , le temps de traitement d'un lot de ce type est donné par p_{s,g^*} .

Nous considérons un total de G types de produit, et une taille limite des lots égale au nombre de broches dans les équipements μ . La taille de lots peut ainsi varier dans un intervalle défini dans l'équation 4.1 pour un lot u composé de produits du type g^* .

$$b_u \in [1, \min(N_{g^*}, \mu)] \quad (4.1)$$

Où N_{g^*} est le nombre total de produits du type g^* .

Les temps de préparation des lots est compris dans le temps de traitement, et indépendante des séquences définis : un lot de type g^* peut être traité sur la même machine après d'un lot de type g^* sans temps de supplémentaire.

L'objectif étant la minimisation du retard, deux définitions sont envisageables : La première définition prend en compte la totalité des produits N . Si c'_i est la date de fin réelle du traitement du produit i , le retard de ce produit peut être calculée d'après l'équation 4.2 :

$$t_i = \max(0, c'_i - d_i) \quad (4.2)$$

Où t_i correspond aux retard du produit i . Cette définition ne prend pas en compte les commandes. Étant donné que les produits font partie d'une commande, une deuxième définition du retard peut être établie :

$$T_o = \max(0; C_o - D_o) \quad (4.3)$$

Où T_o est le retard de la commande o , et C_o est la date de fin réelle maximale des produits de la commande o .

Par la suite de ce chapitre, le retard est calculé d'après la deuxième définition (équation 4.3).

Une solution complète pour le problème décrit est composée de la description des lots de traitement, et les plannings pour chaque machine pour toutes les S opérations. Les décisions d'ordonnancement peuvent être de deux natures différentes : un planning général ou un planning indépendant par opération. La première option utilise un seul planning pour l'ensemble d'opérations, où les produits sont traités dans un ordre unique. Ce type de planning correspond à l'ordonnancement de permutation. La deuxième option permet des changements entre les phases, générant ainsi, autant des plannings que des opérations dans l'atelier. Les méthodes proposées dans ce chapitre considèrent ces deux types d'ordonnancement. Ainsi, les méthodes proposées peuvent être implémentées sur ces deux types d'ateliers.

Avec la nomenclature à trois champs présente en [75] et expliquée dans le chapitre 1, le problème décrit est connu comme $FHm, ((PM^{(k)})_{k=1}^m) / r_i, batch / \sum T_i$. Désormais nous référons ce problème comme $HFSP$ («Hybrid Flow Shop Problem»).

Étant donné que le problème $HFSP$ a été prouvé NP-difficile pour certains problèmes avec deux opérations [55], pour la version décrite dans ce chapitre, nous proposons des méthodes de résolution approchées car la taille des problèmes est trop importante pour une résolution exacte.

Afin de simplifier la présentation de ce problème, la nomenclature suivante est utilisée par la suite :

- $S \leftarrow$ Nombre total d'opérations
- $O \leftarrow$ Nombre de commandes reçues
- $G \leftarrow$ Nombre de types de produit considérés
- $m_s \leftarrow$ Machines disponibles pour l'opération s
- $\mu \leftarrow$ Nombre limite de broches dans les équipements
- $N \leftarrow$ Nombre total de produits (toute commande et type confondus)
- $N_g \leftarrow$ Nombre total de produits de type g
- $R_o \leftarrow$ Date d'arrivée de la commande o
- $D_o \leftarrow$ Date de fin souhaitée de la commande o
- $r_i \leftarrow$ Date d'arrivée du produit i (correspond à la date d'arrivée de la commande associée)
- $d_i \leftarrow$ Date de fin souhaitée du produit i , (correspond à la date de fin souhaitée de la commande associée)
- $g_i \leftarrow$ Type de produit du produit i
- $\beta_{i,g} \leftarrow$ Donnée binaire pour déterminer si le produit i est de type g
- $\gamma_{i,o} \leftarrow$ Donnée binaire pour déterminer si le produit i appartient à la commande o
- $\pi_{s,g} \leftarrow$ Temps de traitement d'un lot de type g , dans l'opération s

- $\mathcal{N} = [1, \dots, N] \leftarrow$ Ensemble des produits à traiter
- $\mathcal{O} = [1, \dots, O] \leftarrow$ Ensemble de commandes à traiter
- $\mathcal{S} = [1, \dots, S] \leftarrow$ Ensemble d'opérations dans l'atelier
- $\mathcal{G} = [1, \dots, G] \leftarrow$ Ensemble de types de produit

Par rapport à la composition des lots de traitement, les équations 4.4 et 4.5 sont définies pour déterminer le temps de traitement et la date d'arrivée d'un lot u , par rapport aux produits affectés : pour un lot u , composé d'un ensemble I_u de produits, le temps de traitement du lot u dans l'opération o ($p_{u,s}$), et la date d'arrivée (L_u) sont calculés avec les équations 4.4 et 4.5 respectivement.

$$p_{u,s} \leftarrow \pi_{s,g} \forall s \in \mathcal{S} \quad (4.4)$$

$$L_u \leftarrow \max(r_i \forall i \in I_u) \quad (4.5)$$

Sous certaines conditions, le problème décrit et étudié dans ce chapitre peut correspondre à d'autres problèmes combinatoires :

Condition	Problème
$O = 1$	
$G = 1,$ $m_s = 1 \ s = 1, \dots, S$	$F//C_{max}$ Maximal polynomiale solvable [60]
$O = 1,$ $G = N$	$FHm, ((PM^{(k)})_{k=1}^m) // \sum T_i$ Pas de traitement par lots ni des dates d'arrivée différentes
$O = N,$ $G = N$	$FHm, ((PM^{(k)})_{k=1}^m) / r_i / \sum T_i$ Pas de traitement par lots
$1 < O < N,$ $G = 1$	Problème d'assignation
$1 < O < N,$ $0 < G < N,$ $\mu > \max_{1 < g < G} (N_g),$ $\max_{1 < o < O} (R_o) = 0$	$HFSP$ Pas de traitement par lots ni dates d'arrivée des différentes
$S = 2,$ $G = N,$ $m_s = 1 \ s = 1, \dots, S,$ $D_o = D^*, \ o = 1, \dots, O$	$F/r_i/C_{max}$ Minimal NP-Hard [81]

La plupart de ces cas ne seront pas traités dans ce mémoire, à l'exception du problème

$FHm, ((PM^{(k)})_{k=1}^m) / r_i / \sum T_i$ qui sera utilisé pour les développements heuristiques présentés dans une section ultérieure.

La suite du chapitre est déclinée sur plusieurs sections. La section 4.3 présente un modèle mathématique qui décrit le fonctionnement de l'atelier hybride à cheminement unique. Les méthodes de résolution proposées sont présentées dans les sections 4.4 et 4.5.2. Afin d'identifier les méthodes les plus efficaces et performantes pour ce problème, des tests sur des instances théoriques ont été réalisés. Ces instances sont inspirées du cas d'application réel.

4.3 Modélisation mathématique : Modèle avec variables mixtes

Afin de compléter la définition donnée dans la section précédente, nous proposons un modèle mathématique pour résoudre le problème $HFSP$. Ce modèle est composé de variables binaires pour les décisions de constitution de lots et leur ordonnancement, et de variables entières positives pour les paramètres relatifs aux temps. Sans perte de généralité, nous supposons que les données de chaque instance sont des entiers positifs. Dans un souci de généralisation du problème traité (plannings de permutation et non permutation), les variables de décision permettent l'association produit/lot et l'ordonnancement des lots de manière indifférente pour chaque opération $s \in S$.

L'affectation des produits aux lots de traitement est représentée par la variable $x_{i,u}$:

$$x_{i,u} = \begin{cases} 1 & \text{Si le produit } i \text{ fait partie du lot } u \\ 0 & \text{Sinon} \end{cases} \quad (4.6)$$

Les décisions de l'ordonnancement des lots de traitement par opération sont représentées par la variable $y_{u,v,s}$:

$$y_{u,v,s} = \begin{cases} 1 & \text{Si le lot } u \text{ précède le lot } v \text{ pendant l'opération } s \\ 0 & \text{Sinon} \end{cases} \quad (4.7)$$

Les autres variables utilisées dans le modèle sont :

- $T_o \leftarrow$ Retard de la commande o
- $\alpha_{u,g} \leftarrow$ Variable binaire pour déterminer si le lot u contient des produits de type g
- $c'_{u,s} \leftarrow$ Date de sortie du lot u de l'opération s
- $M \leftarrow$ Entier suffisamment grand
- $N + 1 \leftarrow$ Lot supplémentaire pour signaler la fin du planning sur une phase
- $0 \leftarrow$ Lot supplémentaire pour signaler le début du planning sur une phase

Pour compléter les plannings dans les phases avec plusieurs équipements, les lots de traitement sont affectés à la première machine disponible. Le modèle suivant décrit le problème traité dans ce chapitre.

Fonction objectif :

$$\min Z = \sum_{o=1}^O T_o \quad (4.8)$$

Sous contrainte :

$$\sum_{u=1}^N x_{i,u} = 1 \quad \forall i \in \mathcal{N} \quad (4.9)$$

$$x_{i,u} + \alpha_{u,g} (1 - 2\beta_{i,g}) \leq 1 - \beta_{i,g} \quad \forall i \in \mathcal{N}, \forall g \in \mathcal{G} \quad (4.10)$$

$$\sum_{i=1}^N x_{i,u} \leq \mu \quad \forall u \quad (4.11)$$

$$r_i * x_{i,u} - L_u \leq 0 \quad \forall i, \forall u \in \mathcal{N} \quad (4.12)$$

$$(\alpha_{u,g} * \pi_{s,g}) - p_{u,s} \leq 0 \quad \forall g \in \mathcal{G}, \forall s \in \mathcal{S}, \forall u \in \mathcal{N} \quad (4.13)$$

$$L_u + p_{u,1} - c'_{u,1} \leq 0 \quad \forall u \in \mathcal{N} \quad (4.14)$$

$$c'_{u,s} + p_{u,s} + M * y_{u,v,s} - c'_{v,s} \leq M \quad \forall s \in \mathcal{S}, \forall u, v \in \mathcal{N} \text{ où } u, \neq v \quad (4.15)$$

$$c'_{u,s-1} + p_{u,s} - c'_{u,s} \leq 0 \quad \forall s \in \mathcal{S} \text{ avec } s > 1, \forall u \in \mathcal{N} \quad (4.16)$$

$$\sum_{u=1, u \neq v}^N y_{u,v,s} = 1 \quad \forall s \in \mathcal{S}, \forall u, v \in \mathcal{N} \quad (4.17)$$

$$\sum_{v=1, v \neq u}^N y_{u,v,s} = 1 \quad \forall s \in \mathcal{S}, \forall u, v \in \mathcal{N} \quad (4.18)$$

$$\sum_{v=1}^N y_{0,v,s} = m_s \quad \forall s \in \mathcal{S} \quad (4.19)$$

$$\sum_{u=1}^N y_{u,N+1,s} = m_s \quad \forall s \in \mathcal{S} \quad (4.20)$$

$$\gamma_{i,o} (c'_{u,S} - D_o) - M (1 - x_{i,u}) - T_o \leq 0 \quad \forall i, \forall u \in \mathcal{N}, \forall o \in \mathcal{O} \quad (4.21)$$

$$T_o \geq 0, c'_{u,s} \geq 0, \alpha_{u,g} \geq 0, p_{u,s} \geq 0, L_u \geq 0 \quad (4.22)$$

$$x_{i,u} \in \{0, 1\} \quad \forall i, \forall u \in \mathcal{N} \quad (4.23)$$

$$y_{u,v,s} \in \{0, 1\} \quad \forall s \in \mathcal{S}, \forall u, \forall v \in \mathcal{N} \quad (4.24)$$

Dans ce modèle, l'équation (4.8) décrit la fonction objective : minimiser la somme des retards des commandes O . Pour garantir l'affectation de tous les produits à un lot, nous utilisons l'équation (4.9). La contrainte (4.10) est relative à la relation lot/type de produit : tout lot doit être composé de produits d'un seul type de produit. La contrainte (4.11) limite la taille des lots au nombre de broches dans les machines. Nous utilisons la restriction (4.12) pour calculer les dates de disponibilité L_u du lot u . L'équation (4.13) établit les temps de traitement pour l'opération s , d'après le type de produit du lot u . Les contraintes (4.14) et (4.15) calculent le temps de sortie pour la première opération et le reste respectivement. Ces deux contraintes utilisent des valeurs modifiées des limites données aux dates de lancement de chaque opération. L'équation (4.16) limite les dates des fin pour les cas où les lots passent directement d'une opération $s - 1$ à l'opération suivante s . Le nombre de successeurs et de prédécesseurs d'un lot est contrôlé avec les contraintes (4.17) et (4.18), et limité à 1 pour chaque opération s . Le nombre de machines par opération est limité par l'équation (4.19) à M_s . Cette contrainte calcule le nombre de lots dans la première position de toutes les machines pour l'ensemble de opérations S . D'une manière similaire, l'équation (4.20) limite le nombre de lots en dernière position de toutes les machines de chaque opération. Le retard par commande est calculé avec l'équation (4.21), sur la base du temps de sortie de la dernière opération S . La restriction (4.22) vérifie que les variables auxiliaires (temps de traitement par lot, dates de disponibilité et sortie, entre autres) ont des valeurs supérieures à 0. Pour finir, les variables binaires sont définies dans les équations (4.23) et (4.24).

Au total, ce modèle est composé de $N^2 + (G + 3S + 3) + N(GS + 3S + G + 4) + 2S$ contraintes, où N est le total de produits, G les différents types de produits, et S le nombre d'opérations. Au maximum, $N^2(1 + S) + NG$ variables binaires sont nécessaires avec $3N + 2NS$ variables standard.

Nous avons testé ce modèle sur des instances issues des données réelles du cas d'application décrit (le laboratoire d'analyses). Dans une limite de temps fixée à 1800 secondes, nous avons résolu des instances de petite taille (8 produits et 4 types de produit) de ce problème avec le logiciel CPLEX d'ILOG-IBM. Comme nous l'avons détaillé dans un chapitre précédent (chapitre 3), ce logiciel a des bonnes performances dans la résolution de modèles linéaires combinant des variables entières et réelles. Le résultat a été validé avec une procédure d'énu-

mération complète. Pour les instances de plus grandes tailles, le temps de résolution dépasse les limites fixées, et devient donc inutilisable dans un contexte d'application industriel.

Afin de trouver des solutions dans un intervalle de temps acceptable, nous proposons, dans les sections suivantes, des méthodes de résolution approchées diverses, lesquelles considèrent les deux types d'ordonnancement mentionnés auparavant (permutation et non permutation). Parmi les méthodes exposées, nous incluons la méthode utilisée actuellement dans le laboratoire d'analyses physico-chimiques pour la gestion des échantillons. Cette méthode est formalisée pour la première fois.

Afin d'inclure un point de comparaison, la méthode utilisée actuellement pour la gestion d'échantillons dans le laboratoire d'analyses LPC est donnée dans la sous-section 4.4. Après, les méthodes adaptées et testées sont décrites.

4.4 Méthode de résolution actuelle

Le problème complet, décrit dans ce chapitre, n'a pas été traité, à notre connaissance, dans aucune référence scientifique. Une étude bibliographique sur les problèmes voisins au *HFS*P avec traitement par lots, est présenté dans le chapitre 2. Afin de comparer les méthodes présentées dans ce chapitre, avec algorithme complet incluant toutes les contraintes décrites, nous présentons dans cette section la méthode de résolution actuellement utilisée dans le cas industriel du contexte de recherche : le laboratoire physico-chimique (*LPC*) du centre de recherche du groupe Soufflet.

Avec la nomenclature du problème d'ordonnancement traité dans ce chapitre, les échantillons arrivant au laboratoire correspondent aux produits de l'atelier. Ces échantillons arrivent dans des commandes liées au développement d'un nouveau biocatalyseur. Les lots de traitement sont composés d'échantillons suivant une même analyse dans le laboratoire. Les machines multibroches sont des équipements d'analyses avec plusieurs capteurs pour effectuer un total de μ analyses simultanément. Une description plus détaillée du fonctionnement de ce laboratoire est donnée dans le chapitre 5.

Le fonctionnement du laboratoire LPC dépend des analyses effectuées sur les différents échantillons envoyés. Lors de sa création, les équipements et l'infrastructure de ce laboratoire ont été conçus selon la conception classique d'un laboratoire de recherche. Les critères utilisés dans la gestion de qualité des temps de mouvements n'ont pas fait partie des critères utilisés dans la prise de décision du positionnement des différents composants du laboratoire. De la même manière, les procédures ont été construites sur des protocoles physico-chimiques pour

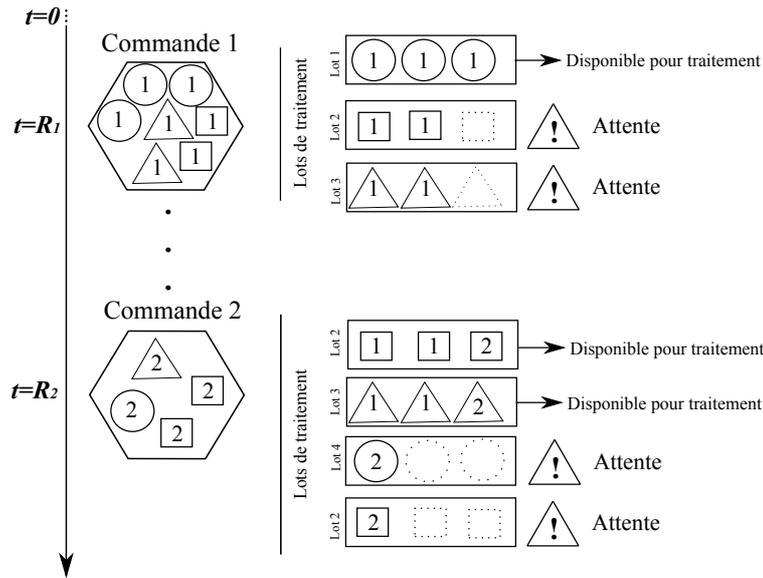


FIGURE 4.2 – Exemple de fonctionnement de la solution actuelle

la réalisation des mesures analytiques. Ainsi, le fonctionnement du laboratoire était garanti, sans pour autant s'intéresser à sa productivité.

C'est ainsi que la méthode actuelle pour la gestion des échantillons a été fixée. Pour cette méthode, le responsable du laboratoire a défini un paramètre B qui détermine le nombre d'échantillons (produits) faisant partie d'un même lot. Cette valeur est fixée. Dès qu'un lot est complet (un total B produits concernant le même type de produit sont présents), il est autorisé à commencer l'analyse. L'ordonnancement de ces lots de traitement est défini par rapport à la position dans le stock entre opérations : le lot attend dans une file d'attente jusqu'à ce que l'équipement soit libéré. L'ordre suit une loi du type premier arrivé, premier servi pour la première opération («First-In-First-Out» (*FIFO*)). Pour les opérations restantes, cette loi se conserve, mais nous l'appellerons *FCFS* («First-Come-First-Served»). Tel qu'il a été décrit en [103], si la méthode *FCFS* est utilisée pour finir l'ordonnancement dans les opérations, le planning résultat est un ordonnancement à permutation.

Un exemple du fonctionnement de cette solution est donné dans la figure 4.2. Dans cette figure, deux commandes arrivent à l'atelier à deux dates différentes R_1 et R_2 . Les Lots sont composés respectant la contrainte d'unicité du type de produit traité. Etant donné que cette méthode fixe une valeur B pour la taille du lot (exemple $B = 3$), même si des produits sont affectés, le lot ne peut pas être disponible pour son traitement avant de compléter la quantité définie par B . Cela génère des attentes.

La taille du lot (B) a été fixée de manière empirique, en utilisant la valeur μ relative aux nombre de broches dans les équipements.

Dans l'objectif de minimiser le retard total, cette politique pénalise des lots car leur date de lancement est attardé artificiellement en attendant à compléter les B produits du même type. Sachant que les commandes et leur composition suivent des lois aléatoires, certains lots peuvent attendre pendant des périodes de temps qui ont des forts impacts dans le calcul du retard total sur les commandes.

Étant donné que sous-utiliser les équipements d'un centre de recherche implique un surcoût négligeable par rapport aux coûts d'une analyse complète, construire des lots de traitement de taille inférieure à μ peut accélérer et dynamiser ces sites particulières.

La suite de ce chapitre est déclinée de façon à inclure les différents composants des méthodes heuristiques adaptées à la résolution du problème *HFS*P avec traitement par lots.

4.5 Heuristiques à deux niveaux

Afin de résoudre le problème d'ordonnancement décrit dans les sections précédentes, nous avons décomposé le problème en deux niveaux de décision. Dans un premier temps, les décisions relatives à la construction de lots de traitement sont prises afin de réduire les temps de traitement, grâce au traitement simultané des produits contenus dans un même lot. Chaque équipement étant composé de plusieurs broches, les temps de traitement sont identiques pour exécuter l'opération sur un ou plusieurs produits à la fois.

Une fois les lots de traitement construits, les dates de disponibilité du lot, et les temps de traitement (liés au type de produit) sont définies. Les équations (4.5) et (4.4) sont utilisées à ces fins. Ces informations conforment les données d'entrée d'un problème d'ordonnancement d'un atelier à cheminement unique de composition hybride (*HFS*P). Cela correspond au deuxième niveau de décision. Ces deux niveaux de décisions sont représentés dans la figure 4.5.

En raison de la complexité du problème et des performances limitées de la méthode de résolution du modèle mathématique linéaire, les méthodes présentées ici sont des métaheuristiques adaptées.

L'intégration de ces deux niveaux décisionnels a été effectuée sur une structure de recherche métaheuristique de sélection naturelle. Pour cela, nous avons adapté un algorithme génétique, dont les solutions représentent les décisions relatives à la construction des lots de traitement. L'évaluation de ces solutions est directement liée au deuxième niveau de décision : l'ordonnancement. Nous nous sommes concentrés principalement sur la résolution de

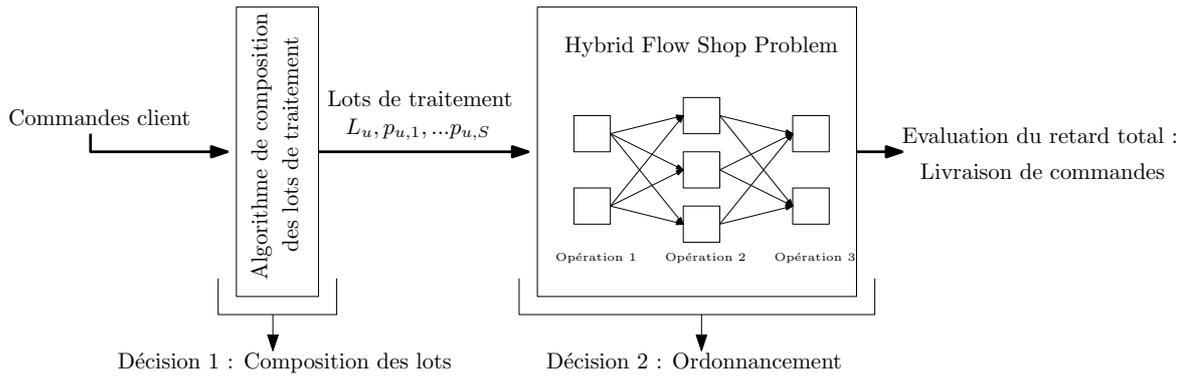


FIGURE 4.3 – Niveaux de décision

ce dernier, car il serait à l'origine de la plus part des retards. L'algorithme 12 montre la structure suivie par les méthodes à deux niveaux.

Algorithme 5 Algorithme à deux niveaux

- 1: $Population \leftarrow$ Population initiale : solutions du problème composition des lots (Premier niveau)
 - 2: **Pour** chaque $Solution$ en $Population$ **Faire**
 - 3: $T_{solution} \leftarrow$ Fonction objectif : Minimisation du retard total (Deuxième niveau)
 - 4: **Fin Pour**
 - 5: **Pour** chaque $Generation = 1$ à $MaxGeneration$ **Faire**
 - 6: $Population' \leftarrow$ Solutions issues des croisements de $Population$ (Premier niveau)
 - 7: $Population'' \leftarrow$ Solutions issues des mutations de $Population$ (Premier niveau)
 - 8: **Pour** chaque $Solution$ en $P' \cup P''$ **Faire**
 - 9: $T_{Solution'} \leftarrow$ Fonction objectif : Minimisation du retard total (Deuxième niveau)
 - 10: **Fin Pour**
 - 11: $Population \leftarrow$ Sélection des solutions parents pour la génération suivante (Premier niveau)
 - 12: **Fin Pour**
-

Cet algorithme reprend les étapes classiques d'un algorithme génétique, où les lignes 3 et 9 font appel à la fonction objectif de la solution. Cette fonction est une évaluation complète de l'objectif du problème d'ordonnancement des lots de traitement (le retard total), lequel est un problème NP-difficile [109].

Le problème d'ordonnancement est traité avec la meilleure méthode issue de la littérature pour le problème *HFSP* avec minimisation du retard total [79], et avec un ensemble de méthodes qui reprennent les conclusions du chapitre 3, où nous cherchons, pour les opérations concernées, la méthode de minimisation du retard dans l'ordonnancement de machines en

parallèle qui s'adapte le mieux en vue des données d'entrée (dates d'arrivée et temps de traitement des lots).

Le reste de cette section est divisé en trois parties. La première partie explique en détail l'algorithme utilisé pour la constitution des lots de traitement. L'évaluation des solutions de cet algorithme impliquent une résolution d'une instance du problème *HFSP*. La résolution de ce problème a été abordée à partir de différentes méthodes. Les différentes méthodes de résolution sont expliquées dans deux sous sections. La première considère des solutions où les ordonnancements obtenus sont à permutation. La deuxième partie, propose des méthodes pour des ordonnancements sans permutation. Les performances de ces méthodes sont testés sur des instances issues du cas réel expliqué précédemment, avec la structure présente au laboratoire LPC, qui correspond à un atelier avec 3 opérations, avec 2 équipements pour la première opération, 3 pour la deuxième, et 2 pour la dernière.

4.5.1 Composition de lots de traitement par sélection naturelle

Cette sous-section est dédiée à la méthode développée pour le premier niveau de décision.

Très utilisés dans la résolution de problèmes combinatoires difficiles, les algorithmes génétiques font partie des métaheuristiques à population, qui appliquent le concept de sélection naturelle, afin de garantir une performance minimale dans la solution obtenue à la fin du processus générationnel. Nous avons adapté un algorithme génétique pour les décisions relatives à la composition des lots de traitement.

Le fonctionnement d'un algorithme génétique repose sur une structure générationnelle, qui commence l'exploration de l'espace de solutions à partir d'un ensemble de solutions dites solutions initiales. Des modifications sont ensuite effectuées sur ces solutions afin de chercher de nouvelles solutions pouvant obtenir de meilleures performances. Pour cela, il est nécessaire d'établir une structure homogène qui rassemble l'information nécessaire pour construire une solution. Cette structure correspond à la représentation de la solution.

Représentation de la solution

Avec l'objectif d'inclure toutes les combinaisons possibles, nous avons choisi une représentation qui permet d'affecter chaque produit i à tout lot compatible (contrainte d'unicité du type du produit du lot). Ainsi, toutes les possibilités sont considérées pour tout type de produit. Pour un type de produit g^* , la représentation de la solution permet d'avoir des lots composés d'un seul produit, mais également, un lot avec tous les produits de type g^* (N_g)

<i>Produit</i>	1	2	3	4	5
<i>Type de produit</i>	<i>g</i>	<i>g</i>	<i>h</i>	<i>g</i>	<i>h</i>
<i>X</i> :	1	2	1	1	1

FIGURE 4.4 – Exemple de représentation de solution

dans la limite donné par μ (le nombre de broches par équipement) tel qu'il a été décrit dans l'équation (4.1). De cette manière, pour toute instance, si le nombre d'analyses différentes est G , au maximum, un total de N lots peuvent être définis, et dans le cas où $N_g \leq \mu \forall g \in \}$ le nombre minimale de lots est de G .

La représentation de la solution est un vecteur avec N positions, où chaque position est relative à un produit. Ce vecteur est rempli avec l'identifiant du lot auquel le produit est affecté. Pour un type de produit g , nous noterons η_g le nombre de lots composés pour ce type de produit. Si $\eta_g = 1$, un seul lot contient tous les produits du type g . Si le nombre total de produits du type g est N_g , et si $\eta_g = N_g$, chaque lot est composé d'un seul produit de type g . Un lot composé contient au minimum un produit. Les identifiants utilisés sont de entiers positifs, limités, pour un type de produit g à l'intervalle $[1, \dots, \max(\eta_g, \mu)]$.

La figure 4.4 donne un exemple de la représentation de la solution pour les décisions de constitution des lots de traitement. Dans cet exemple, 5 produits de deux types de produit différent g et h , sont affectés à 3 lots de traitement. Deux des lots correspondent au type de produit g , et un lot au type de produit h . Le premier lot du type g est composé des produits 1 ($x_{\{1\}} = 1$) et 4 ($x_{\{4\}} = 1$). Le deuxième lot contient le produit 2 ($x_{\{2\}} = 2$). Pour le type de produit h , le lot construit inclut les deux produits $x_{\{3\}}$ et $x_{\{5\}}$.

Population initiale

Comme toute métaheuristique inspirée de la sélection naturelle, l'algorithme génétique proposé dans cette section commence la recherche à partir d'un ensemble de solutions appelées population initiale. L'impact de cette population sur la performance globale de l'algorithme varie selon le problème combinatoire résolu. Cette population doit contenir des solutions de bonne qualité, afin d'extraire les gènes pouvant être la source de meilleures solutions. Les solutions faisant partie d'une population, sont appelées chromosomes. La population initiale doit contenir des chromosomes qui facilitent la diversité des solutions, et ainsi échapper aux optimums locaux.

La population initiale pour le problème de construction de lots est principalement com-

posée de chromosomes de solutions réalisables de bonne qualité. Étant donné que nous considérons des dates de disponibilité différentes de 0 pour les commandes arrivant au laboratoire d'analyses, des solutions aléatoires où des lots de fabrication sont composés avec des produits arrivant à des dates trop éloignées peuvent avoir un effet significativement négatif sur les performances de la solution. Pour deux produits i et j , si $r_i \ll r_j$, il est très probable que le produit i aie un retard significativement plus élevé si les deux produits sont traités ensemble, que si ils sont traités sur deux lots différents.

Ainsi, afin d'éviter ces effets, nous nous sommes inspirés de la solution actuelle (section 4.4) pour la génération de la population initiale. De cette manière, la valeur B (taille du lot) n'est plus définie par μ , mais sélectionnée aléatoirement à l'intérieur d'un intervalle : $B \in [1, \mu]$. Les produits sont donc affectés d'après leurs dates d'arrivée au laboratoire, afin de compléter chaque lot.

Cette méthode garantit des solutions où un seul lot par type de produit est proposé (dans le cas où ce nombre ne dépasse pas μ), ainsi que des cas où chaque produit est traité indépendamment.

L'évaluation de la population initiale s'effectue avec la fonction objectif.

Évaluation de la fonction objectif

L'évaluation des chromosomes établis dans les différentes populations lors de la résolution du problème de construction de lots de traitement s'effectue avec une évaluation de la fonction objectif. Cette évaluation détermine les performances de la solution, par rapport à l'objectif défini pour le problème *HFSP* (la minimisation du retard total). Cette évaluation mesure le critère à utiliser pour la sélection naturelle.

Dans notre structure à deux niveaux, l'évaluation de la fonction objectif est le deuxième niveau de décision : l'ordonnancement des lots de traitement dans le *HFSP*. Cette évaluation correspond à résoudre un problème combinatoire dont les données d'entrée dépendent du chromosome (solution) évalué.

L'évaluation de la fonction objectif pour le problème *HFSP* traité dans ce chapitre doit être directement proportionnelle au calcul du retard total. Le calcul du retard total dans l'atelier à cheminement unique (sans les décisions de la composition des lots) est un problème combinatoire NP-difficile. Dans certain cas, cette complexité implique l'utilisation de méthodes d'estimation de la fonction objectif [32], [131], à la place des méthodes d'évaluation de la fonction objectif. Dans notre cas, étant donné la structure de deux niveaux proposée, nous avons préféré l'utilisation de méthodes de résolution rapides, capables de construire des

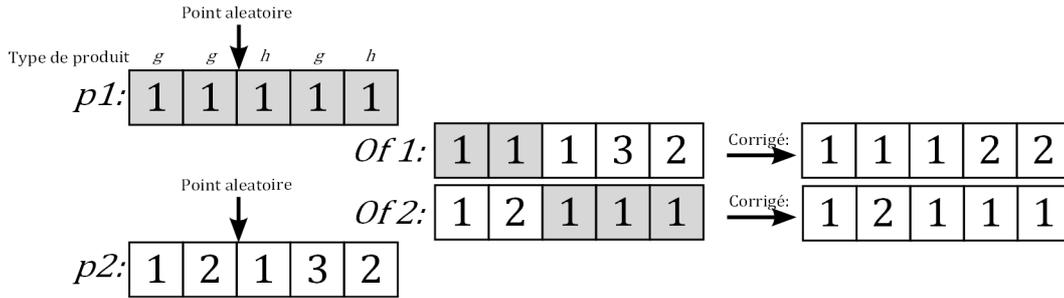


FIGURE 4.5 – Exemple de croisement pour la construction de lots

solutions de qualité acceptable avec des temps de calcul réduits. Par conséquent, le risque de sous-estimer une bonne solution est réduit, malgré une augmentation du temps de calcul.

De ce fait, nous avons hybridé des méthodes pour calculer et minimiser le retard total, avec les compositions des lots définies dans les chromosomes. Ces méthodes utilisées pour effectuer l'ordonnancement des lots sont présentées dans les sections 4.5.2 et 4.5.3.

Le résultat de l'évaluation de la fonction objectif est utilisé pour effectuer la sélection des chromosomes parents afin d'effectuer les croisements et mutations de l'algorithme génétique décrit dans cette section. La sélection de ces chromosomes est effectuée par une sélection aléatoire, où chaque chromosome a une probabilité inversement proportionnelle au retard total calculé. Cette méthode est connue comme la sélection par roulette.

Les mouvements utilisés dans le croisement et mutation sont expliqués par la suite.

Croisement

Dans la théorie des algorithmes à sélection naturelle, le croisement des deux ou plus solutions (chromosomes) parents, produit des nouvelles solutions avec des performances meilleures que celles des parents. Nous proposons une méthode de croisement, laquelle, à partir de deux parents (p_1 et p_2), génère deux nouvelles solutions (Of_1 et Of_2). La figure 4.5 montre un exemple du croisement sur un chromosome avec 5 produits de 2 types différents.

Le croisement utilisé dans cette section sélectionne aléatoirement un point d'intersection : pour le premier nouveau chromosome Of_1 , les gènes avant ce point proviennent du parent p_1 . Of_1 est complété avec les gènes du parent p_2 . Le deuxième nouveau chromosome Of_2 est construit avec le même point d'intersection, mais en inversant les rôles des parents. Une vérification est réalisée pour contrôler que les nouveaux chromosomes sont valides (affecter tous les produits à au moins un lot).

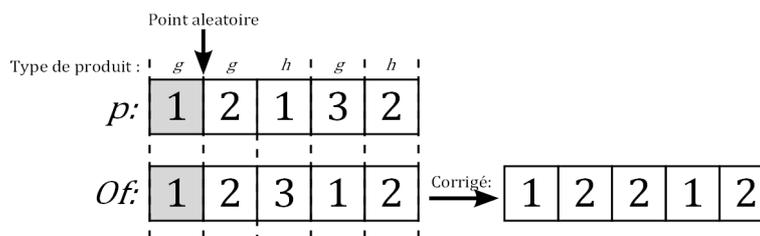


FIGURE 4.6 – Exemple de mutation pour la construction de lots

Mutation

Afin d'augmenter la diversité et sortir des optimums locaux causés pour des populations très homogènes, nous incluons dans la construction d'une nouvelle génération, des chromosomes issus d'un processus de mutation. Autre que proposer des solutions performantes, le mouvement de mutation cherche à introduire des nouveaux gènes dans la population. Ce concept appliqué au problème de construction des lots, correspond à changer pour certains produits, le lot auquel ils étaient affectés dans le chromosome parent.

L'exécution de cet algorithme de mutation commence avec la sélection d'un chromosome parent p (avec la procédure de sélection expliquée précédemment). Un point est sélectionné de manière aléatoire. Les gènes avant ce point sont copiés directement dans le nouveau chromosome Of . Le reste du chromosome est complété avec les gènes du parent sélectionné, mais copiés de manière inverse à leur position initiale. Un exemple de ce mouvement est donné dans la figure 4.6.

Les chromosomes issus de la mutation suivent une procédure de vérification et correction pour garantir le respect des contraintes du problème.

La nouvelle génération créée avec les mouvements de croisement et mutation est complétée avec les meilleurs chromosomes de la génération précédente. Nous appelons cela l'héritage.

Les sections suivantes donnent le détail des méthodes utilisées pour l'évaluation de la fonction objectif : l'ordonnement des lots de traitement construits. Ces méthodes sont divisées en deux, par rapport au type de solution obtenue. Les solutions proposées sont des ordonnancements de permutation ou de non permutation.

Les valeurs des dates de disponibilité des lots et les temps de traitement sont actualisés d'après les équations 4.4 et 4.5, selon le chromosome évalué. Ces valeurs sont les données d'entrée des méthodes de résolution des problèmes d'ordonnement *HFSP*.

Dans la figure 4.7 une représentation de la distribution des méthodes développées est incluse, avec la classification entre méthodes à permutation et les méthodes sans permutation.

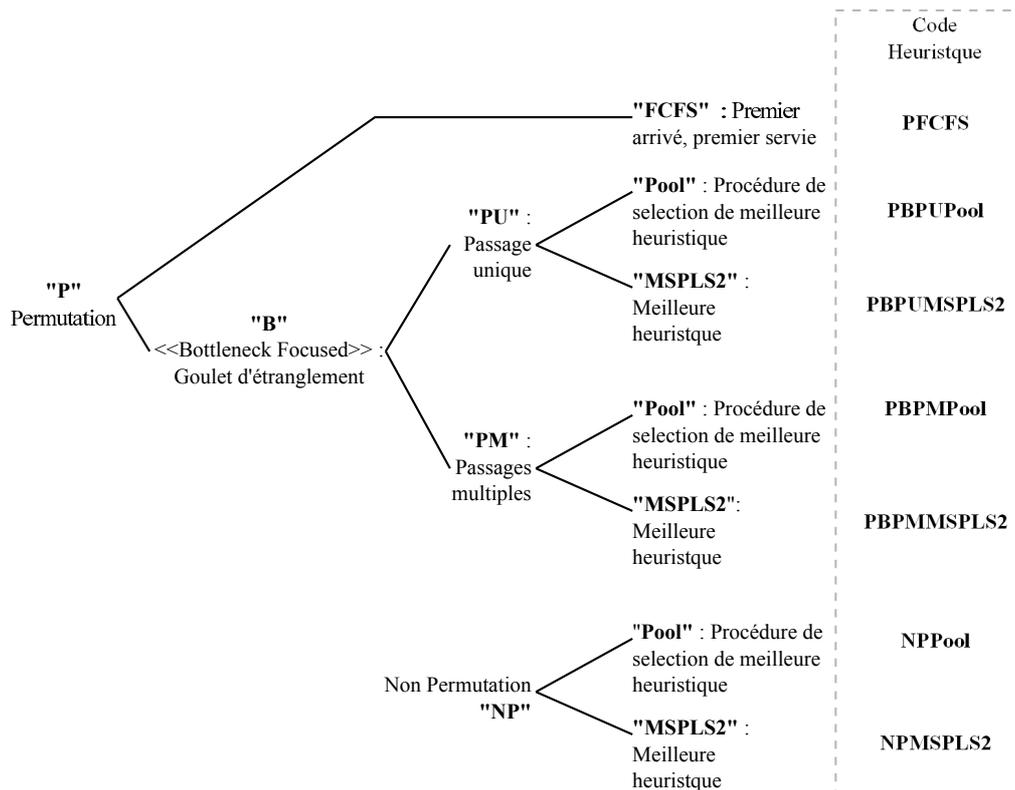


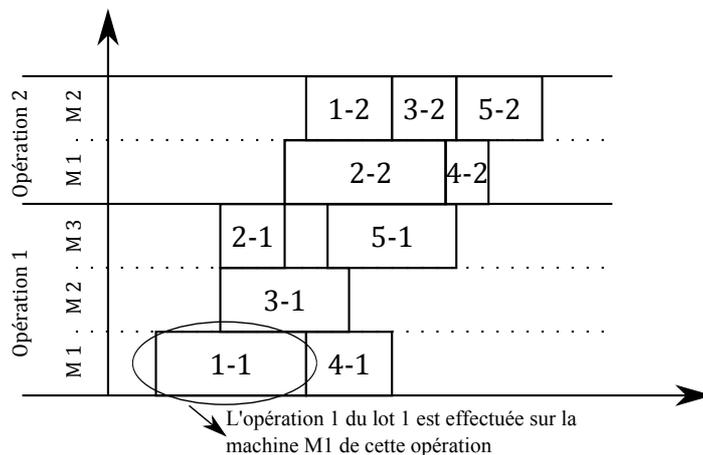
FIGURE 4.7 – Méthodes de résolution pour le problème *HFS*

Le détail des méthodes est donné dans les sections suivantes.

4.5.2 Méthodes heuristiques avec permutation (P)

Nous incluons dans les méthodes d'ordonnancement avec permutation, les méthodes qui construisent des solutions dont l'ordonnancement est détaillé pour une seule des opérations. Par rapport à la méthode et à la position l'opération pour laquelle l'ordonnancement est établi, le reste des opérations sont ordonnancées d'après un seul et unique planning, ou elles suivent la règle du *FCFS* («First-Coming-First-Served») [103]. La méthode actuellement utilisée (section 4.4) correspond à une méthode avec permutation, où un seul planning est établi, et complété avec un ordonnancement en *FCFS*.

Dans les sous-sections suivantes, nous présentons l'ensemble d'éléments des heuristiques avec permutation.

FIGURE 4.8 – Exemple d'ordonnement *FCFS*

4.5.2.1 Sélection de l'opération à prioriser

Afin d'identifier sur quelle opération parmi les S opérations de l'atelier, deux méthodes sont souvent utilisées : la solution évidente de choisir la première opération, et celle de se focaliser sur l'opération la plus chargée. La première méthode utilise les caractéristiques de la première opération de l'atelier et en déduit un ordonnancement qui est suivi pour le reste de l'atelier. Les solutions issues de cette méthode ne tiennent pas en compte l'impact de la possible différence de charge entre les opérations de l'atelier (dû principalement aux temps de traitement et au nombre de machines).

Une première méthode est celle utilisé dans le cas réel présenté, laquelle correspond à des ordonnancements construits en *FCFS* :

Premier arrivé, premier servi *FCFS* (**PFCFS**)

La méthode adaptée *FCFS* ordonnance les lots de traitement, selon leur arrivée à chaque opération. Pour la première opération de l'atelier, cette méthode est équivalente à la règle d'ordonnement du *FIFO* («First-In-First-Out»), avec les dates de disponibilité établies dans l'équation 4.5. Pour le reste des opérations, cette méthode utilise les dates de sortie des opérations précédentes pour les distribuer sur la première machine disponible (pour les opérations à équipements parallèles). La figure 4.8 donne un exemple pour 5 lots ordonnancés dans un atelier à deux opérations avec 3 et 2 machines respectivement.

La deuxième méthode pour sélectionner l'opération à prioriser dans l'ordonnement est techniquement plus intéressante :

Goulet d'étranglement - «Bottleneck focused» (B)

Cette méthode à permutation identifie l'opération qui théoriquement sera la plus chargée parmi l'ensemble d'opération de l'atelier. Cette opération représente le goulet d'étranglement de l'atelier. Il est donc logique de prioriser cette opération, afin de augmenter les performances de l'atelier, car par définition, c'est l'opération goulet qui établit la cadence de l'atelier.

Pour identifier quelle opération correspond au goulet d'étranglement, nous utilisons l'équation (4.25), qui est donnée par Lee *et al.* [79].

$$\frac{\sum_u^N p_{u,b}}{m_b} = \max_{\forall s \in \mathcal{S}} \left(\frac{\sum_u^N p_{u,s}}{m_s} \right) \quad (4.25)$$

Dans cette équation, b est l'opération goulet, si la charge moyenne des temps de traitement par machine, est la plus importante de l'atelier. En cas d'égalité, le goulet d'étranglement est la première à être exécutée.

Une fois l'opération b identifiée, il est nécessaire d'utiliser une méthode pour générer le planning pour cette opération. Étant donné cela équivaut à résoudre un problème d'ordonnancement sur des machines parallèles. nous utilisons des conclusions et des développements du chapitre 3. Dans une section ultérieure nous expliquons la mise en place de ces éléments dans les algorithmes de résolution proposés. De la même manière, il faut définir la façon dont les opérations différentes à l'opération goulet sont traitées afin d'établir le planning complet. Pour cela nous utilisons deux méthodes présentées par Lee *et al.* [79] : la méthode de liste de passage unique, et la méthode de liste de passages multiples. La sous-section suivante décrit ces deux méthodologies.

4.5.2.2 L'ordonnancement des opérations non goulet

L'ordonnancement des opérations autres que l'opération goulet est considéré comme étant une possible source de retard, malgré le fait que l'opération goulet soit optimisée. En fait, afin d'établir l'ordonnancement dans l'opération goulet, certaines définitions sont nécessaires, pour pouvoir établir les caractéristiques (données d'entrée) de ce problème d'ordonnancement. Ces caractéristiques dépendent de la position de l'opération goulet. Ainsi, trois différentes situations sont à considérer :

- $b = 1$
- $1 < b < S$
- $b = S$

Où S est le nombre total d'opérations. Pour chacune de situations possibles, la principale différence est l'origine des dates d'arrivée à utiliser. Ainsi, si $b = 1$, les dates d'arrivée à utiliser

son celles données par les dates calculées par lot avec l'équation 4.5, selon sa composition. Si l'opération goulet n'est pas la première opération, les valeurs des dates d'arrivée et les dates de fin souhaitée doivent être estimées. Pour cela nous utilisons les équations 4.26 et 4.27 respectivement.

$$Rd_u = r_u + \sum_{s=1}^b (p_{u,s}) \quad (4.26)$$

$$Dd_i = d_i - \sum_{s=b}^S p_{i,s} \quad (4.27)$$

Dans ces équations, Rd_u est une estimation de la date de sortie du lot u de l'opération $b - 1$ (celle immédiatement précédente au goulet d'étranglement) et Dd_i est une valeur approchée à utiliser pour chaque produit comme date de fin souhaitée de l'opération b . Si le goulet d'étranglement correspond à la dernière opération, les dates de fin souhaitée ne sont pas modifiées, mais les dates d'arrivée au goulet seront calculées avec 4.26.

Deux possibles types d'ordonnancement peuvent être créés avec ces définitions données : soit l'ordonnancement Ω établi pour l'opération b est réalisable, soit il ne peut pas être respecté. Cela est vérifié sur les opérations en amont (opérations 1, .. $b - 1$), en utilisant le planning Ω . Si le planning n'est pas faisable, deux méthodes de correction sont possibles : liste de passage unique ou liste de passage multiple.

Liste de passage unique (PU)

Si le planning Ω ne peut pas être suivi pour toutes les opérations en respectant les valeurs pour $Rd_u \forall u \in \Theta$ où Θ est le nombre total de lots à considérer, donc le planning est reconstruit dès l'opération 1. Ainsi, pour toutes les opérations, l'ordre est donnée par Ω avec les dates de début des opérations calculées d'après les dates de sorties de l'opération précédente.

Liste de passage multiple (PM)

Afin de corriger le planning Ω , deux itérations sont définies. Pour cela, il est nécessaire d'établir un planning complet jusqu'à l'opération b , avec la méthode PU . Ces itérations ont été définies dans [79] :

1. Pour les lots avec les plus hautes Rd_u , $Rd_u = c_{u,b-1}^{\{\Omega\}}$, où $c_{u,b-1}^{\{\Omega\}}$ est la date de sortie du lot u de l'opération $b - 1$, selon le planning Ω exécuté d'après PU .
2. Pour les lots avec les plus hautes Rd_u , $Rd_u = r_u + \max \left\{ Rd_u - (c'_{u,s}); \sum_{s=1}^b (p_{u,s}) \right\}$

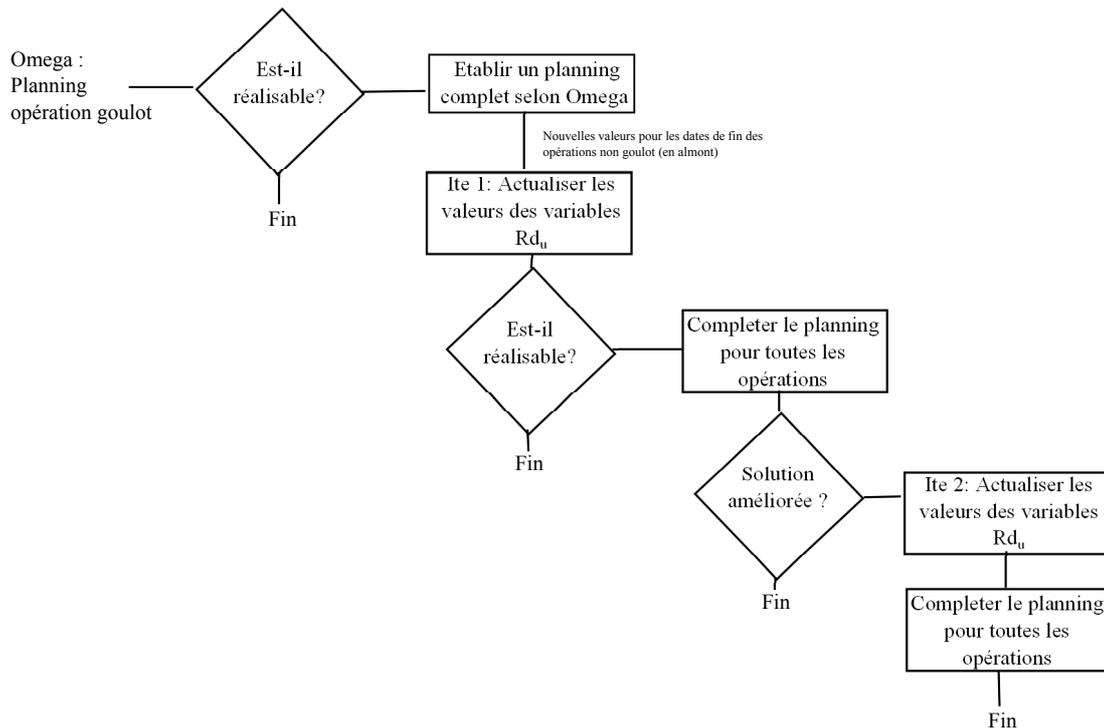


FIGURE 4.9 – Multiples passages

La première itération corrige les valeurs de Rd_u afin de trouver un planning dans l'opération goulet qui soit réalisable. La deuxième itération cherche à réduire le retard en modifiant les valeurs référence pour l'opération goulet. La figure 4.9 montre le fonctionnement de cette méthode.

4.5.3 Méthodes heuristiques sans permutation (NP)

Afin de considérer des différentes méthodes et structures de solution, et principalement à des fins de comparaison, nous avons testé des méthodes sans permutation. Il est signalé par plusieurs auteurs que la mise en place des méthodes d'ordonnancement sans permutation est souvent très compliquée, car elle implique que pour chaque opération de l'atelier, un planning est établi.

Malgré cela, nous avons testé ce type de méthode. Ainsi, pour chaque opération, un problème d'ordonnancement de machines parallèles est établi. Les méthodes de résolution de ce problème ont été étudiées dans le chapitre 3. Leur application au problème traité dans ce chapitre est donnée dans la sous-section suivante. Nous rappelons que ces méthodes sont identiques pour les ordonnancements avec et sans permutation.

4.5.4 Algorithmes d'ordonnement de machines parallèles

Afin de résoudre les problèmes d'ordonnement de machines parallèles trouvés dans la résolution du problème *HFSP*, nous avons utilisé les conclusions du chapitre 3 pour proposer deux méthodes à comparer. Dans [79], cette partie du problème *HFSP* est traitée avec une adaptation de l'algorithme *KPM* introduit par Koulamas [65]. Biskupt *et al.* [23] ont montré que leur algorithme *BHG* était dans tous les cas meilleur que la méthode *KPM*. En 2012, Yalaoui [140] a adapté cette méthode pour considérer le cas avec des dates de disponibilité différentes de 0 pour les tâches arrivant à l'atelier. Cette méthode appelée *ABHG* était signalée comme étant la plus performante pour la résolution de ce problème. Les conclusions du chapitre 3 signalent des méthodes ayant des performances supérieures que celles obtenues avec la méthode *ABHG*. C'est à partir de ces conclusions que nous proposons les méthodes présentées par la suite.

4.5.4.1 Meilleure heuristique : MSPLS2

Parmi les conclusions du chapitre précédente, nous avons établi que l'heuristique *MSPLS2* obtenait les meilleurs résultats, avec des temps d'exécution acceptables. Cette méthode est donc utilisée dans la résolution de problèmes tel qu'il a été décrit dans les sous-sections précédentes.

4.5.4.2 Procédure de sélection de la meilleure heuristique (Pool)

La deuxième contribution du chapitre 3 est le développement d'une méthode permettant de sélectionner, selon les caractéristiques des instances résolues, la méthode la plus adaptée à sa résolution. Cette méthode est donc appliquée, en cherchant à minimiser le retard total en identifiant, pour l'opération traitée, l'heuristique ayant obtenue les meilleures performances (parmi les méthodes testées dans le chapitre 3).

4.6 Résultats expérimentaux

Afin de déterminer la méthode de résolution trouvant les meilleures performances parmi les méthodes exposées dans les sections précédentes, nous avons effectué des tests comparatifs. Pour cela, des instances théoriques ont été créées sur des données issues du cas réel.

Au total, nous comparons 8 méthodes de résolution :

- Méthode actuelle

- *AG – PFCFS* : Algorithme génétique avec ordonnancement à permutation avec la règle *FCFS*
- *AG – PBPUPool* : Algorithme génétique avec ordonnancement à permutation, passage unique, et ordonnancement par procédure de sélection
- *AG – PBPUMSPLS2* : Algorithme génétique avec ordonnancement à permutation, passage unique, et ordonnancement par *MSPLS2*
- *AG – PBPMPool* : Algorithme génétique avec ordonnancement à permutation, passages multiples, et ordonnancement par procédure de sélection
- *AG – PBPMMSPLS2* : Algorithme génétique avec ordonnancement à permutation, passages multiples, et ordonnancement par *MSPLS2*
- *AG – NPPool* : Algorithme génétique avec ordonnancement sans permutation, et ordonnancement par procédure de sélection
- *AG – NPMSPS2* : Algorithme génétique avec ordonnancement sans permutation, et ordonnancement par *MSPLS2*

Les paramètres utilisés dans l'algorithme génétique ont été fixés pour l'ensemble de méthodes testées, afin de réduire le bruit des paramètres changeants. Nous avons utilisé les conclusions de [83] pour déterminer ces paramètres :

- Taille de la population : 50
- Taux de croisement : 0.8
- Taux de mutation : 0.1
- Taux d'héritage : 0.1

Pour tester toutes les méthodes sous des conditions similaires, le critère d'arrêt a été défini sur le nombre de solutions testés associées au niveau de décision produit-lot. Nous avons défini cette limite à $100 * N$, avec N le nombre produits à affecter.

Les instances générées sont conçues avec 20 types de produits différents, sur un atelier de 3 opérations, avec 2 machines dans la première opération, 3 dans la deuxième et 2 dans la dernière opération. Afin de tester les méthodes développées sur des instances proches du comportement réel du laboratoire analyse, des instances avec 8, 50, 200, 450, 800 et 1250 produits sont considérées, issus de 4, 10, 20, 30, 40 et 50 commandes. Les dates de disponibilité des produits dans la première opération sont générées selon les équations 4.28.

$$R_o = rand(0, \min(\pi_{s,g_i} \forall s \in \mathcal{S}, \text{ssi } \gamma_{i,o} = 1)) \quad (4.28)$$

Où π_{s,g_i} est le temps de traitement l'opération s , pour les produits de type g_i dans la commande o . Les dates de fin souhaitées sont définies par rapport à l'équation 4.29.

$$D_o = R_o + \lambda * (\max_{\forall s \in \mathcal{S}}(\pi_{s,g_i}, \text{ssi } \gamma_{i,o} = 1)) * (rand(0, \max_j(R_j \forall j \in N Lots))) \quad (4.29)$$

Heuristique / $O - N$	4-8	10-50	20-200	30-450	40-800	50-1250
<i>AG - PFCFS</i>	100,00%	0,00%	0,00%	0,00%	60,00%	0,00%
<i>AG - PBPUMSPLS2</i>	100,00%	0,00%	40,00%	0,00%	0,00%	0,00%
<i>AG - PBPUPool</i>	100,00%	0,00%	20,00%	20,00%	0,00%	0,00%
<i>AG - PBPMPool</i>	100,00%	20,00%	0,00%	80,00%	20,00%	40,00%
<i>AG - PBPMMSPLS2</i>	100,00%	20,00%	40,00%	0,00%	0,00%	20,00%
<i>AG - NPPool</i>	100,00%	20,00%	0,00%	0,00%	60,00%	40,00%
<i>AG - NPMSPLS2</i>	100,00%	40,00%	0,00%	0,00%	0,00%	0,00%
Actuelle	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%

TABLE 4.1 – Indicateur de performance $RBKS_q$

Où λ est un coefficient dans l'intervalle $[1, 1.9]$.

Les temps de traitement ont été définis pour chaque opération et chaque type de produit selon les équations (4.30), (4.31), (4.32) et (4.33).

$$\chi_g = rand(1; 100) \quad (4.30)$$

$$\pi_{1,g} = rand(0.1 * \chi_g; 0.5 * \chi_g) * 100 \quad (4.31)$$

$$\pi_{2,g} = rand(1.7 * \chi_g; 2.3\chi_g) * 100 \quad (4.32)$$

$$\pi_{3,g} = rand(0.9 * \chi_g; 1.7 * \chi_g) * 100 \quad (4.33)$$

Les indicateurs de performance inclus dans le chapitre 1 sont utilisés pour comparer les performances des méthodes développées.

Pour la méthode exacte, la taille d'instance maximale pour laquelle le CPLEX a garanti des résultats optimaux dans un temps d'exécution limité à 1800 secondes est une instance de 8 produits et 4 types de produits. D'autres instances avec plus de produits non pas été résolues de manière à garantir l'optimalité de la solution.

Pour les méthodes approchées, le tableau 4.1 montre les résultats pour l'indicateur de performance $RBKS$.

Les résultats dans ce tableau sont donnés par heuristique et par nombre de produits (taille de l'instance). Ainsi, par exemple, parmi les instances avec 450 produits issus de 30 commandes ont été résolues, 40% ont été résolues avec la meilleure solution par les méthodes *AG - PBPUMSPLS2* et *AG - PBPMMSPLS2*. Dans ce tableau, plus grand est la valeur obtenue, plus efficace est l'heuristique.

Pour l'indicateur GAP , qui tient en compte de l'écart relatif à la meilleure solution connue, les résultats sont donnés dans le tableau 4.2. Pour le tableau 4.2, plus la valeur

Heuristique / $O - N$	4-8	10-50	20-200	30-450	40-800	50-1250
<i>AG - PFCFS</i>	0,00%	69,70%	24,88%	15,19%	18,03%	44,08%
<i>AG - PBPUMSPLS2</i>	0,00%	27,63%	5,40%	69,43%	70,14%	52,17%
<i>AG - PBPUPool</i>	0,00%	39,32%	13,35%	41,57%	53,95%	63,40%
<i>AG - PBPMPool</i>	0,00%	21,90%	9,62%	14,18%	55,61%	36,37%
<i>AG - PBPMMSPSL2</i>	0,00%	11,81%	10,79%	59,02%	77,54%	32,44%
<i>AG - NPPool</i>	0,00%	18,27%	18,91%	42,62%	23,39%	26,85%
<i>AG - NPMSPLS2</i>	0,00%	26,23%	22,80%	34,32%	62,28%	44,20%
Actuelle	0,00%	69,70%	24,88%	81,19%	89,67%	82,90%

TABLE 4.2 – Indicateur de performance $Gap_{q,ins}$

Heuristique / $O - N$	4-8	10-50	20-200	30-450	40-800	50-1250
<i>AG - PFCFS</i>	0,005	0,033	0,160	0,468	1,267	2,866
<i>AG - PBPUMSPLS2</i>	0,010	0,161	0,491	2,638	7,748	26,554
<i>AG - PBPUPool</i>	0,008	0,242	0,568	6,980	11,875	35,834
<i>AG - PBPMPool</i>	0,004	0,905	1,540	11,142	49,317	285,966
<i>AG - PBPMMSPSL2</i>	0,009	0,506	1,432	4,378	28,202	106,074
<i>AG - NPPool</i>	0,007	0,747	1,556	13,960	51,609	137,802
<i>AG - NPMSPLS2</i>	0,006	0,568	1,324	6,663	21,884	59,773

TABLE 4.3 – Temps de calcul moyens en secondes

de l'indicateur est proche de 0, plus performante est l'heuristique. Par exemple, pour les instances avec 1250 produits issus de 50 commandes, l'écart relatif est en moyen de 26.85% pour l'heuristique *AG - NPPool*.

Afin de comparer les méthodes proposées dans ce chapitre avec tous les critères possibles, nous incluons dans le tableau 4.3 les temps de calcul moyen nécessaires avant d'atteindre le critère d'arrêt défini précédemment. Dans le tableau 4.3, les temps de calcul sont donnés en secondes. Ainsi, par exemple pour l'heuristique *AG - PBPUMSL2*, les temps de calcul sont compris entre 0.010 et 26.55 secondes, selon la taille de l'instance résolue.

A propos de la méthode utilisée pour l'ordonnancement de machines en parallèles *Pool*, le tableau 4.4 donne la répartition moyenne des groupes. Cela correspond au taux opérations dont la procédure de sélection a trouvé chacun des groupes définis dans le chapitre 3. Nous incluons les groupes les plus représentés.

D'après l'ensemble de résultats présentés, les heuristiques à permutation obtiennent des performances plus régulières que les méthodes sans permutation. Pour les méthodes concentrées sur l'opération goulet *PB*, avec un passage unique *PU*, l'impact du type de méthode pour l'ordonnancement semble n'avoir aucune influence. En revanche, pour les méthodes

Groupe	Taux d'apparition
Groupe 10	21.12%
Groupe 4	18.12%
Groupe 14	13.41%
Groupe 3	11.76%
Groupe 15	11.48%
Groupe 5	10.18%

TABLE 4.4 – Distribution des opérations des instances résolues par rapport aux groupes (chapitre 3)

NP , ce choix et son interaction avec la taille de l'instance est à considérer.

Par rapport aux temps d'exécution, les heuristiques à passage unique PU sont les méthodes les plus rapides. L'utilisation de la procédure pour sélectionner la méthode d'ordonnement sur les opérations à équipements parallèles, a un impact sur le temps d'exécution, qui sont dans la plupart des cas, plus élevés qu'en appliquant une méthode fixée.

Les instances testées avec 50 commandes et 1250 produits, correspondent à la charge journalière du laboratoire analysé du cas industriel. Les données du temps de traitement, dates de disponibilité et types d'analyse ont été extraites de la base de données du laboratoire. Les dates de fin souhaitée ont été estimées. Ainsi, les méthodes de résolution proposées, trouvent des solutions qui réduisent considérablement le critère d'optimisation par rapport à la méthode utilisée dans le laboratoire. Ces résultats dépendent des facteurs qui ne sont pas directement liés aux décisions opérationnelles considérés ici. Ces facteurs sont étudiés dans l'annexe A.

4.7 Conclusion

Dans ce chapitre, nous avons présenté un problème de composition de lots, et leur ordonnancement dans un atelier à cheminement unique, composé de 3 opérations. Chacune de ces opérations peut être exécutée par une ou plusieurs ressources. Ce problème est composé de deux niveaux décisionnels : la composition des lots, et l'ordonnement des lots. Chacun de ces décisions correspond à un problème combinatoire. Les lots sont composés de produits arrivant à l'atelier à des dates différentes, liées à des commandes passées par les clients de l'atelier.

Nous rappelons que ce problème a été identifié dans un laboratoire d'analyses physico-

chimiques, lequel reçoit des commandes composées d'échantillons pour mesurer différents caractéristiques physico-chimiques d'un produit en développement.

Pour résoudre ce problème, avec l'objectif de la minimisation du retard total, un modèle mathématique et 7 méthodes heuristiques ont été proposés. Le retard total étant déterminé sur la date de fin réelle de la commande complète, les deux niveaux décisionnels ont un impact direct sur la performance des solutions obtenues.

La formulation mathématique établie peut être résolue pour des instances de très petite taille. Pour les instances de taille réelle, nous proposons des méthodes à deux niveaux. Dans un premier niveau, un algorithme génétique utilise une structure pour déterminer la composition des lots, sous les contraintes établies par le cas réel. L'évaluation de la fonction objectif pour cet algorithme est effectuée avec le deuxième niveau. Ce niveau résout un problème d'ordonnancement du type hybride dans un atelier à cheminement unique.

Afin d'identifier la méthode qui obtient les meilleurs résultats, un ensemble de tests est effectuée, utilisant les critères de performance définis dans le chapitre 1, et des données issues du cas réel étudié. Également, ces tests comparent les méthodes proposées contre la méthode actuellement utilisée.

Les résultats expérimentaux permettent de conclure que les méthodes proposées obtiennent, en moyenne, des solutions à 24% d'amélioration par rapport à la méthode actuellement utilisée. Concrètement, les méthodes à permutation $AG - PBPUMSPLS2$, $AG - PBPMPool$ et $AG - PBPMMSPLS2$ obtiennent les meilleures performances parmi les méthodes testées. Les méthodes sans permutation $AG - NPPool$ et $AG - NPMSPLS2$ ont eu les performances les moins intéressantes. En générale, toutes les méthodes ont obtenu des solutions qui pourraient réduire le retard total, par rapport à la situation actuelle, avec des temps de traitement acceptables, inférieures à 5 minutes pour la méthode la plus lente.

Comme il a été signalé par différents auteurs [138], [112], [62], les méthodes d'ordonnancement à permutation (P) ont des espaces de solution plus réduits par rapport aux méthodes sans permutation (NP). Cette différence des espaces de solution et la structure de la fonction objectif (qui est évaluée sur la livraison des commandes) peuvent expliquer la différence des performances entre les méthodes P et NP .

Nous proposons pour de futures études, l'utilisation de méthodes NP qui priorisent l'opération goulet, afin d'éviter de pénaliser l'ordonnancement de tâches dans cette opération avec des dates de disponibilité imposées par les opérations en amont.