



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

École doctorale IAEM

THÈSE

PRÉSENTÉE POUR L'OBTENTION DU GRADE DE
DOCTEUR DE L'UNIVERSITÉ DE LORRAINE
(MENTION AUTOMATIQUE)

PAR
EMMANUEL ZIMMERMANN

MODÈLES D'OPTIMISATION ET D'ÉVALUATION DE
SYSTÈME DE PILOTAGE INTELLIGENT EN CONTEXTE
DE FLUX FORTEMENT PERTURBÉS PAR LES RE-
PRISES

APPLICATION AU CAS DE LA SOCIÉTÉ ACTA-MOBILIER

Soutenue le 12 septembre 2019

Président :

- Professeur Yves Sallez (Université Polytechnique Hauts de France)

Rapporteurs :

- Professeur Samir LAMOURI (Arts et métiers ParisTech)
- Maître de Conférences HdR Marie-Ange MANIER (UTBM)

Examineurs :

- Professeur Patrick Charpentier (Université de Lorraine)
- Maître de conférences HdR Hind Bril El Haouzi (Université de Lorraine)
- Maître de conférences HdR Philippe Thomas (Université de Lorraine)

Invités :

- Docteure Mélanie Noyel D.S.I. Acta-Mobilier
- Professeur André Thomas
- Maître de conférences Christelle Bloch (IUT Belfort-Montbéliard)
- Monsieur Alain Genet D.G. Acta-mobilier

TABLE DES MATIÈRES

I.	Contexte.....	8
A.	Observation du contexte industriel.....	9
1.	Description générale de l'entreprise.....	9
2.	Organisation des activités de production.....	11
3.	Description du système d'information.....	14
B.	Problématique Acta-mobilier.....	16
II.	Analogie (état de l'art)	22
A.	Problèmes d'ordonnancement d'atelier	22
B.	Architectures de pilotage d'atelier.....	29
1.	Les systèmes centralisés	29
2.	Les systèmes décentralisés	30
3.	Les systèmes hybrides.....	32
C.	Les mécanismes de coordination	39
III.	Proposition (Modélisation)	43
A.	Métamodèle et architecture du système.....	43
B.	Ordonnancement global	48
C.	Optimiseurs locaux.....	49
1.	Optimisateur réactif.....	49
2.	Optimisateur proactif.....	67
D.	Communication et synchronisation.....	73
IV.	Application industrielle	77
A.	L'ordonnancement global.....	77
B.	Choix de pilotes	78
C.	Ordonnanceur pour la partie découpe.....	79

D.	Ordonnanceur dynamique pour le robot de laquage	100
1.	Description du cas d'étude	100
2.	Modèle de simulation du robot de laquage.....	102
V.	Conclusions et perspectives.....	115
I.	ANNEXE 1 : Les développement informatique dans l'entreprise.....	131
A.	Les ordres de fabrication (OF)	131
B.	LE M.E.S. et l'affichage dynamique des bonnes pratiques de travail en temps réel	133
C.	Le logiciel d'intégration automatique des commandes (I.A.C.).....	136
II.	ANNEXE 2 : Méthodes d'optimisation d'ordonnancement.....	137
A.	Branch and Bound (BnB)	138
B.	Tabu search (TS).....	139
C.	Algorithme génétique (GA)	139
D.	Colonies de fourmis (ACO).....	140
E.	Particle swarm optimization (PSO)	141
III.	ANNEXE 3 : Méthodes d'évaluation de fonction multi-objectifs.....	145
A.	Somme pondérée des critères d'optimisation	145
B.	« Vector Evaluated Genetic Algorithm » (VEGA).....	145
C.	Approches basées Pareto	146
D.	Procédure d'analyse hiérarchique (AHP)	146
IV.	ANNEXE 4 : résultats comparaison MA et GA.....	149

Remerciements

Je souhaite remercier toutes les personnes que j'ai rencontré durant ces trois années tant au sein de l'entreprise Acta-mobilier que du laboratoire de recherche en automatique de Nancy et qui m'ont aidé à traverser cette étape de ma vie, qui fut riche en émotion, en rencontre et en découverte.

Je tiens à remercier tout particulièrement :

- Monsieur Philippe Thomas, maître de conférences HdR, Université de Lorraine qui a dirigé cette thèse et qui a su me redonner confiance dans des moments où le moral était au plus bas.
- Madame Hind Bril El Haouzi, maître de conférences HdR, Université de Lorraine qui a co-dirigé cette thèse et qui par son dynamisme et sa positivité contagieuse m'a aidé à prendre confiance en moi et à être fier de la valeur de mes travaux.
- Le professeur André Thomas, qui m'a ouvert les portes du monde de la recherche et m'a permis de prendre contact avec l'entreprise Acta-Mobilier.
- Monsieur Alain Genet, président directeur général de l'entreprise Acta-Mobilier qui m'a confié cet ambitieux projet.
- Madame Mélanie Noyel, qui a encadré mes travaux du côté entreprise. Ses conseils, son soutien et son management m'ont permis de progresser énormément tant d'un point de vue humain que technique. Sa vision à long termes et sa ténacité à ne jamais abandonner ses objectifs ont été de grandes sources d'inspiration pour moi.

Je remercie également :

- Le professeur Samir Lamouri, ENSAM Arts et Métiers ParisTech
- Madame Marie-Ange Manier, maître de conférences HdR, UTBM,

pour l'intérêt qu'ils ont manifesté pour mon travail et pour avoir accepté d'être les rapporteurs.

- Le professeur Patrick Charpentier, Université de Lorraine
- Le professeur Yves Sallez, Université Polytechnique Hauts de France
- Madame Christelle Bloch, maître de conférences, IUT Belfort-Montbéliard,

pour avoir accepté d'être membres de mon jury.

Pour finir, je remercie ma famille, mes amis et mes collègues pour leur soutien et leur présence à mes côtés dans les bons comme les mauvais moments.

Introduction

Cette thèse, effectuée dans le cadre d'une convention CIFRE (Conventions Industrielles de Formation par la Recherche), est le fruit de la collaboration entre l'entreprise Acta-Mobilier, basée à Monéteau dans la périphérie d'Auxerre, d'une part, et le CRAN (Centre de Recherche en Automatique de Nancy) et plus particulièrement le département ISET (Ingénierie des Systèmes Eco-Techniques) d'autre part. Acta-Mobilier est un fabricant de façades laquées et texturées haut de gamme pour les cuisines, les salles de bains, l'agencement de stands et de magasins.

Ces travaux s'inscrivent directement dans la continuité de ceux menés par Mélanie Noyel en 2015 (Noyel, 2015) et tentent de répondre à une problématique d'ordonnement de la production soumise à de fortes perturbations des flux. L'hypothèse défendue par le CRAN et en particulier le département ISET est qu'une architecture de pilotage hybride, combinant des systèmes de pilotage centralisés et distribués, constitue une réponse pragmatique et intéressante aux problèmes d'ordonnement sous perturbations.

Le document est construit de la manière suivante :

La partie I décrit le contexte d'étude de la thèse en explicitant, tout d'abord le cadre général dans lequel l'entreprise évolue, puis en mettant le focus sur son activité de production et en décrivant le système d'information de l'entreprise. Ensuite, nous définissons la problématique industrielle et rappellerons les résultats et les conclusions tirés suite à la thèse de Mélanie Noyel pour finir par présenter la problématique scientifique de nos travaux, à savoir l'implémentation d'un système de pilotage hybride basé sur le contrôle par le produit.

La partie II présente un état de l'art sur les problèmes d'ordonnement d'atelier d'une part, en prenant en compte, tout particulièrement, le cas des flux fortement perturbés par des boucles de réparations, et d'autre part, sur les architectures de pilotage de production, en se focalisant sur les architectures dites hybrides et tout particulièrement ceux qui s'appuient sur le contrôle par le produit. Une section est également destinée aux techniques de coordination aussi bien verticales qu'horizontales entre les différentes entités décisionnelles.

La partie III présente la solution que nous nous proposons de mettre en place, à savoir la réalisation d'un système de pilotage hybride basé sur le contrôle par le produit. Ce système sera constitué de plusieurs niveaux de décisions :

- Le niveau le plus haut, l'entité décisionnelle correspond au plan de production.
- Le plus bas, les entités décisionnelles seront les produits eux-mêmes.
- Les niveaux intermédiaires des ordonnanceurs ou optimisateurs que nous nommons locaux, à comprendre dans le sens où ils chercheront à réaliser l'ordonnement soit de postes spécifiques, soit de groupement de postes.

D'abord la méthodologie scientifique appliquée est décrite, puis nous continuerons par la description du méta-modèle et de l'architecture choisies. Ensuite l'étude se concentrera sur la réalisation de l'ordonnement, dans un premier temps, d'un point de vue global, avec l'élaboration du plan de production. Ensuite, nous décrirons les ordonnanceurs locaux (niveaux intermédiaires de prise de décision) que nous mettrons en place. Après, il conviendra d'explorer la manière de faire communiquer les différents acteurs du système de pilotage.

Dans la partie IV, la validation de notre proposition réalisée avec des modèles de simulation, une plateforme de test puis en situation réelle dans l'entreprise sera décrite. Nous commencerons par l'ordonnement global, puis nous décrirons les pilotes sélectionnés pour la réalisation des ordonnanceurs locaux. Ensuite, les mises en application des deux ordonnanceurs locaux réalisés seront étudiées. Pour finir, une méthode de synchronisation inter-ordonnanceurs locaux est présentée

La partie V fait le point sur l'état des travaux, tire les conclusions vis-à-vis du projet et ouvre sur les perspectives d'amélioration ainsi que sur les pistes de recherche qui n'ont pu être étudiées durant la thèse.

I. Contexte

Ce chapitre a pour but de présenter le cadre industriel, le contexte scientifique et les travaux antérieurs qui ont conduit à la réalisation de cette thèse. Durant cette thèse, nous avons choisi de suivre une approche inductive. Comme décrit par Jason Luckerhoff et François Guillemette dans « La place de la mobilisation des résultats de la recherche dans les approches inductives Volume 5, numéro 1, hiver 2018 : « Une approche inductive, autant en recherche qualitative qu'en recherche quantitative, consiste à donner priorité aux données, à l'expérience vécue, au terrain, pour ensuite avoir recours aux savoirs constitués dans un processus de construction de connaissance ». Le principe de fonctionnement de cette approche est le suivant :

Dans un premier temps, il convient d'observer la situation actuelle afin de recueillir des données et des informations.

Ensuite, en analysant et compilant les données nous avons pu constater l'impact des perturbations sur le flux de production et la complexité relative à la traçabilité des produits (**Figure 1** cadre 1 : cas réel).

Après, nous avons cherché à extraire les problèmes scientifiques sous-jacent. Afin de pouvoir les positionner par rapport à la littérature (**Figure 1** cadre 2 : analogie).

Nous avons déterminé un cadre de référence dans lequel inscrire notre proposition qui est celui des systèmes de pilotage hybride que nous expliciterons par la suite.

Nous passons ensuite, à l'étape de modélisation (**Figure 1** cadre 3 : modélisation) où nous cherchons à adapter les méthodes de résolution des problèmes déterminés à la seconde étape à notre cas d'étude.

Puis, nous finissons par la partie application (**Figure 1** cadre 4 : application), où nous décrivons les résultats obtenus, dans un premier temps sur un modèle de simulation et/ou un environnement de test. L'objectif est de ne pas nuire au bon fonctionnement de l'entreprise et d'obtenir des premiers résultats sur lesquels s'appuyer, pour convaincre l'entreprise et obtenir le feu vert pour passer à l'implémentation dans l'entreprise. Ensuite, lorsque cela a été possible, nous présenterons les résultats obtenus en conditions réelles.

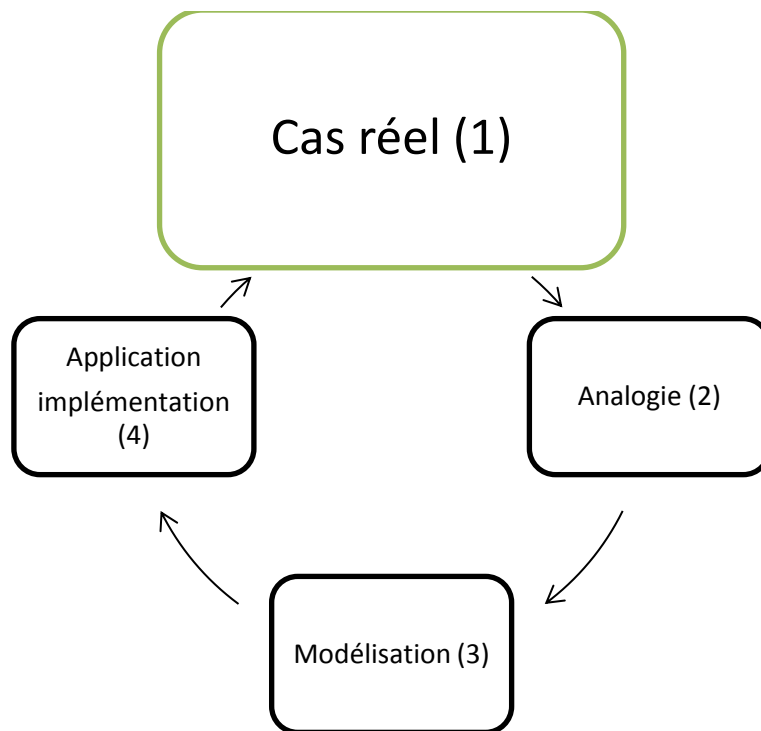


Figure 1 Démarche de la thèse

A. Observation du contexte industriel

1. Description générale de l'entreprise

Acta-mobilier est une société possédant deux activités distinctes mais utilisant les mêmes ressources humaines et structurelles :

- Elle est agenceur de stands, vitrines et magasins pour de grandes marques telles que Chanel, Guerlain, THG ou encore DS automobile.
- Elle est sous-traitante pour les fabricants de cuisines et salles de bain.

Pour les deux activités le taux de reprise est très élevé (supérieur à 30%) à cause du niveau de qualité exigé par les clients et des limitations technologiques de l'outillage de l'entreprise. En plus des délais de livraison très courts que promet l'entreprise, elle a choisi également de ne jamais décliner aucune commande et ce, quelle que soit la charge de production actuelle.

La qualité de service et le savoir-faire de l'entreprise ont pu faire d'elle le leader français sur le marché de la façade laquée haute qualité. Sur le marché nationale les entreprises comme Schmidt, Mobalpa et Pyram lui font confiance.

L'entreprise a pris le pari, il y a quelques années, de développer son activité à l'export en particulier en Allemagne, Autriche, Suisse et Royaume-Uni. Le pari était risqué à cause de l'exigence qualité attendue par ses clients mais payant car aujourd'hui l'export représente 60% du chiffre d'affaires de l'activité. Sa production hebdomadaire est d'environ 4000 pièces produites en MDF (fibres de moyenne densité). Afin de conserver son leadership sur le marché, elle propose très régulièrement de nouvelles finitions telles que les aspects béton, pierre, métal et textile. Une autre décision stratégique qui a permis à l'entreprise de survivre dans ce marché difficile a été de suivre une politique de personnalisation de masse offrant la possibilité aux clients de passer des commandes de pièces dans les dimensions et le coloris de leur choix. Cette politique oblige l'entreprise à produire en juste à temps sans aucun stock de produit fini ou semi-fini, en y incluant les laques. Pour répondre à ce problème, l'entreprise a, par exemple, décidé d'internaliser cette opération en créant son propre laboratoire de fabrication de laque de manière à être moins dépendant des temps de livraison. Le choix de travailler en **Juste à Temps**, sans stock, peut conduire à une augmentation des délais de fabrication. Or ces délais se doivent de rester les plus courts possible. En effet, pour se démarquer de ses concurrents, l'entreprise garanti un délai de livraison de deux semaines après validation de la commande. D'autres contraintes sont liées à cette politique de Juste à Temps menée par Acta-Mobilier comme la prise en compte de la forte variabilité hebdomadaire des commandes clients et l'obligation de gérer de multiple flux simultanément. Seules une très forte **réactivité de l'équipe de production et une polyvalence technique** accrue ont pu lui permettre de mettre en place une telle politique.

Par ailleurs, l'obtention et le renouvellement de la certification allemande « TÜV » permet à la société d'assurer un gage de qualité de production à ses clients. Elle possède également les certifications ISO9001, ISO14001, OHSAS 18001 et s'inscrit dans une démarche de progrès continu KAIZEN. Depuis peu, elle est également suivie par un organisme de conseil afin de maîtriser et d'appliquer les concepts du LEAN. Acta-Mobilier fonde sa stratégie interne sur quatre valeurs qui sont la **responsabilisation de chaque collaborateur, la satisfaction client, l'amélioration continue et le fonctionnement par processus.**

2. Organisation des activités de production

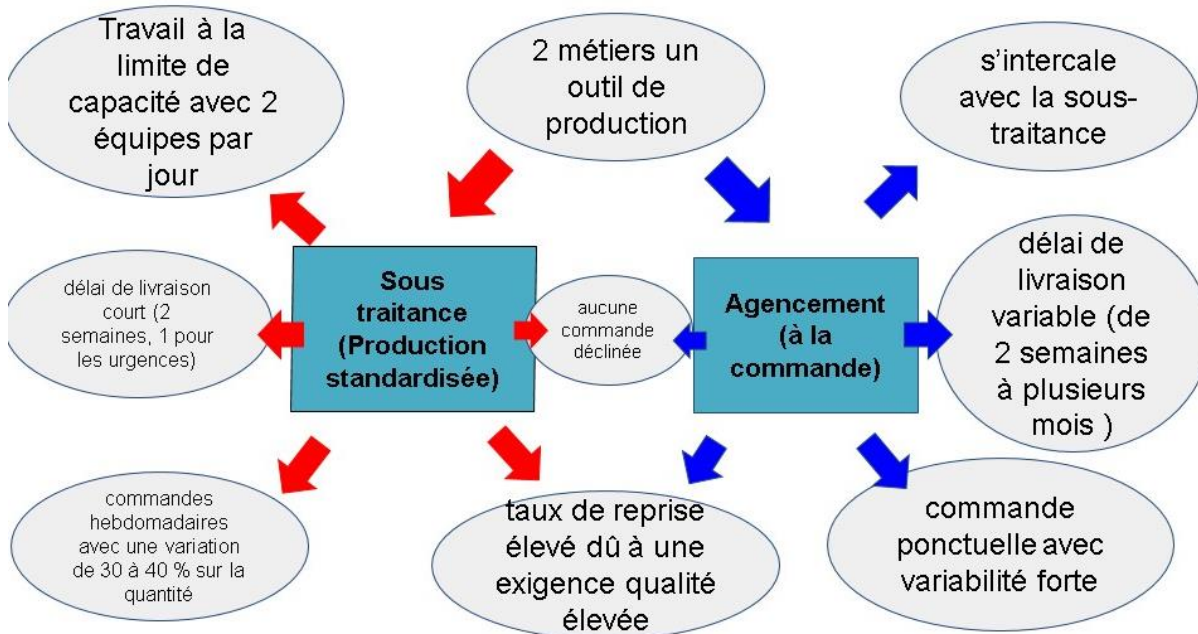


Figure 2 Une entreprise, deux métiers

La **Figure 2** présente les caractéristiques du fonctionnement de l'entreprise. Les deux activités de la société partagent les mêmes ressources. Ces deux activités ont chacune leurs particularités :

- L'activité de sous-traitance peut être considérée comme **standardisée** car malgré le large choix de couleurs et de dimensions, les processus de fabrication sont relativement standards. De plus les commandes clients sont qualifiables de périodiques. Cette production est constituée de commandes hebdomadaires ou bimensuelles dont les quantités peuvent varier de 30 à 40% d'une semaine sur l'autre et ce pour chaque client. La politique suivie est de livrer en deux semaines pour les commandes normales et une semaine pour les commandes urgentes. Cette activité correspond actuellement à un travail à la limite de la capacité en deux équipes de 8 heures. En d'autres termes, cela signifie que pour prendre en charge des projets venant de l'autre activité ou une demande plus élevée que la moyenne pour cette activité, le temps de production et/ou les ressources doivent être augmentés (intérimaires, équipe de nuit ou travail le week-end).

- L'activité d'agencement est beaucoup plus ponctuelle avec des délais de livraison très variables et des volumes de commande totalement aléatoires. Cette activité s'intercale avec la sous-traitance selon son degré d'urgence.

La chaîne de production d'Acta-Mobilier se décompose en six ateliers (**Figure 3**) :

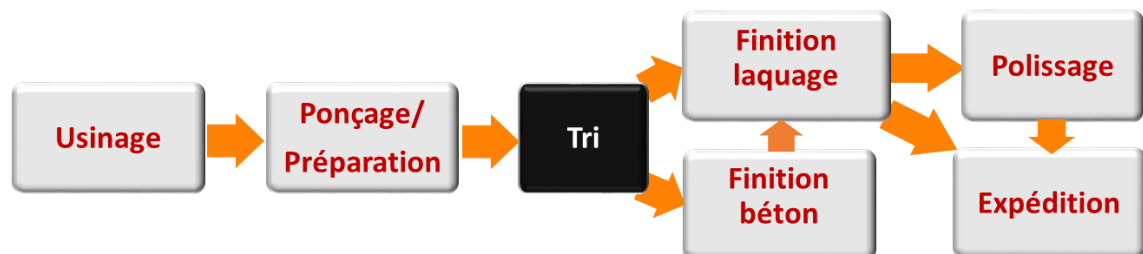


Figure 3 Division en secteurs de l'entreprise

- **L'atelier usinage** dont l'activité principale consiste en la découpe de panneaux de plusieurs matières (Mélaminé, brut, hydro, ...), sous différentes épaisseurs (3mm, 8mm, 12mm, 15mm, 19mm, 22mm, 25mm, 50mm ou 100mm) sur deux centres d'usinage, l'un utilisé principalement pour l'activité d'agencement et le second pour la sous-traitance. Par l'intermédiaire d'un logiciel, des schémas d'imbrication sont calculés afin de maximiser le nombre de pièces réalisables par panneau, et ainsi minimiser le taux de chute. Les pièces trop longues pour les tables des centres d'usinages sont débitées sur une scie (environ 20% de la production). Par la suite, les pièces nécessitant des perçages ou des découpes supplémentaires sont dirigées vers des machines ou outillages spécifiques puis rejoignent leurs lots avant de se diriger vers le secteur suivant, la préparation.
- **L'atelier préparation** regroupe le ponçage et l'apprêtage des pièces. Le ponçage peut être soit manuel, soit réalisé par une machine selon la complexité de la forme des pièces. L'apprêtage peut se faire suivant la volonté du client sur une seule ou sur les deux faces des pièces. Il est possible d'avoir dans une même commande ces deux types de produits.

- **L'étape dite de « tri »** fait partie de la préparation mais a été mise à part, car elle joue un rôle bien spécifique : son objectif est d'effectuer une vérification avant envoi en phase de laquage ou bétonnage de la bonne correspondance des produits entre la demande client et la production tant au niveau quantités que dimensions. En outre, comme sous-entendu par la phrase précédente, elle sert aussi d'aiguilleur entre les différents flux de finition (laquage, béton ou apprêté poncé). À ce poste, l'opérateur sépare les lots, contrôle les pièces puis les regroupent en nouveaux lots.
- **L'atelier laquage** regroupe deux types d'application de laque : manuel et robotisé. Pour le laquage manuel, lorsque les deux faces sont à laquer, un ponçage est effectué avant le laquage de chaque face. Pour le laquage robotisé, lorsque les deux faces sont à laquer, les deux faces sont poncées l'une après l'autre avant laquage. Les types de finitions réalisées sont assez diversifiées (brillante, satinée, microbille, ultramate, perlée, nacrée, ...). Mais pour simplifier, nous pouvons nous contenter de regrouper toutes ces catégories sous les deux familles principales brillante et satinée.
- **L'atelier polissage** ne travaille que sur les produits laqués brillants. Sur ce secteur les produits passent d'abord sur une machine de ponçage, puis une machine de polissage et enfin une machine de lustrage. Après le passage en lustrage, les éventuels défauts ressortent de manière apparente. Si tel est le cas les pièces sont reprises sur un poste de ponçage manuel pour supprimer les défauts, puis la laque est réappliquée et la pièce subit à nouveau l'ensemble du processus poly-lustrage.
- **L'atelier béton** comprend toutes les opérations spécifiques à l'application de la finition béton : l'application du « grip », le pendant de l'apprêt pour la laque, puis le bétonnage des pièces.
- **L'atelier expédition** gère le dernier contrôle qualité sur les produits, puis leur filmage et leur conditionnement. Il permet le regroupement de toutes les pièces d'une commande en provenance des différents ateliers : laquage, polissage, béton ou directement après tri.

Chaque atelier ou unité autonome de production est géré par un responsable de production. Des outils informatiques de suivi de production et de traçabilité ont été développés, par Mme Noyel dans le cadre de sa thèse puis de son poste actuel de DSI, ainsi que par nos soins pendant ma première mission dans l'entreprise, pour les aider dans leurs missions (Annexe1). À l'heure actuelle, et compte tenu de la complexité des flux, les responsables de production n'ont comme outil de pilotage de production, qu'un simple PDP (plan directeur de production) ou planning de charge à capacité infinie. L'application de cet ordonnancement revient simplement à ranger les commandes clients par délai de livraison.

3. Description du système d'information

Habituellement, les entreprises qui en possèdent, fondent leur système d'information sur leur E.R.P. et Acta-Mobilier ne déroge pas à cette règle. L'entreprise utilise la version « ManufacturingPMI » de Cegid. La plupart des E.R.P. contiennent divers modules, dans l'entreprise, celui-ci est utilisé pour gérer les commandes fournisseurs et les commandes clients, la création des articles et la gestion des gammes de fabrication fait également partie des modules utilisés. Les articles sont élaborés sous cette codification : MMTT-FCCCHHHLLLLLOO/CCCCCC. MM définit le modèle de la pièce, à savoir sa forme globale. TT correspond aux différents types possibles (emplacement des perçages, des chanfreins, ...). F établit la finition à appliquer sur le produit (laque brillante, satinée, structure béton, structure pierre, ...). CCC donne la couleur du produit. HHHH exprime la hauteur du produit et LLLL sa largeur. OO est le code option du produit, il peut représenter plusieurs choses mais la plupart du temps il correspond à l'épaisseur du produit. Enfin, le code complémentaire CCCCCC est utilisé pour certains produits particuliers pour fournir les informations utiles supplémentaires définissant celui-ci, tel qu'une troisième dimension pour les pièces à retour.

Les travaux de la thèse de Noyel sur la partie traçabilité ont démontré, l'impossibilité de faire du suivi produit par produit. En effet, les technologies à notre disposition actuellement ne nous permettent pas d'apposer sur chaque produit quelques moyens d'identification que ce soit. C'est la raison pour laquelle Noyel s'est dirigée vers l'utilisation des lots élémentaires (groupement de produits ayant la même gamme de fabrication) pour la traçabilité. Certains E.R.P. fournissent des modules d'ordonnancement et de mises en lot des produits, cependant il s'est avéré que la codification des articles choisie empêchait l'utilisation de celui-ci. Les deux seules solutions permettant de palier à ce problème sont :

- Réaliser ou faire réaliser un développement spécifique.
- Revoir entièrement le système de codification (refonte complète de l'E.R.P., impliquant un temps d'inactivité non-négligeable de l'entreprise).

Pour cette raison, une application annexe a été développée, celle-ci appelée « lotisseur » instanciée dans une base de données les lots élémentaires à partir des données des commandes client issues de l'E.R.P. Ensuite le M.E.S. gère la réalisation du suivi de production et après nos travaux, l'aide à la décision (se référer aux annexes pour plus de précision). Un logiciel s'occupe de la gestion de la production des laques fabriquées en interne par l'entreprise et un autre valide la dernière étape avant l'envoi des commandes aux clients en étiquetant les produits avant palettisation et en gérant l'expédition. La **Figure 4** résume les interactions entre les différents éléments du système d'informations.

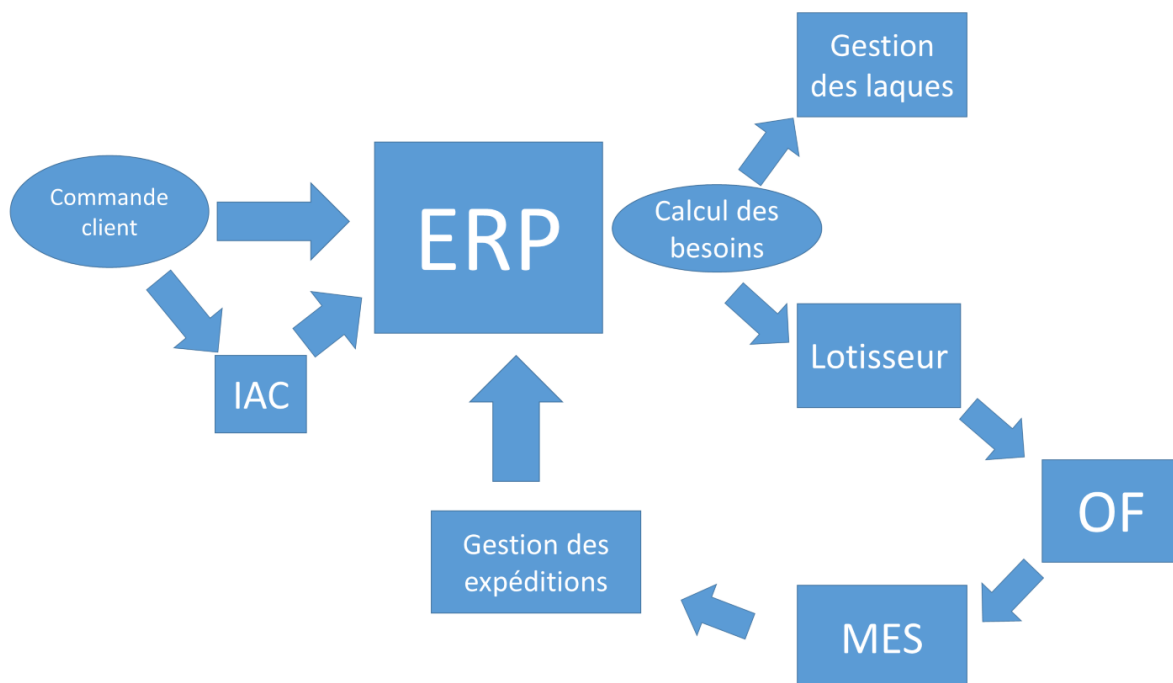


Figure 4 Interactions des éléments du système d'informations

B. Problématique Acta-mobilier

De prime abord, Il parait illusoire d'établir un planning prévisionnel robuste précis sur un horizon supérieur à trois semaines. Même pour réaliser un ordonnancement sur une seule semaine de production, en se concentrant uniquement sur l'activité de sous-traitance et en définissant des plages de production interdites à la sous-traitance (afin de prendre en considération l'agencement), on se heurte à d'autres difficultés qui rendent toute planification prédictive rapidement obsolète. En effet, les boucles de retour causées par les réparations sont tellement fréquentes qu'elles sont en mesure de dérégler l'ensemble du flux en créant des saturations de file d'attente et retarder l'ensemble des commandes. De plus, comme le montre la **Figure 5**, chaque atelier a ses propres spécificités et ne prends pas nécessairement sur les mêmes paramètres pour optimiser le flux. Par exemple, dans le secteur usinage, les critères à optimiser sont les taux de chute à la découpe et les temps de setup ce qui conduit naturellement à regrouper les lots par épaisseur et matière alors qu'au secteur laquage, les critères à optimiser sont la consommation de laque et les changements de série ce qui conduit à regrouper les lots par couleur et nombre de faces à peindre (une ou deux).

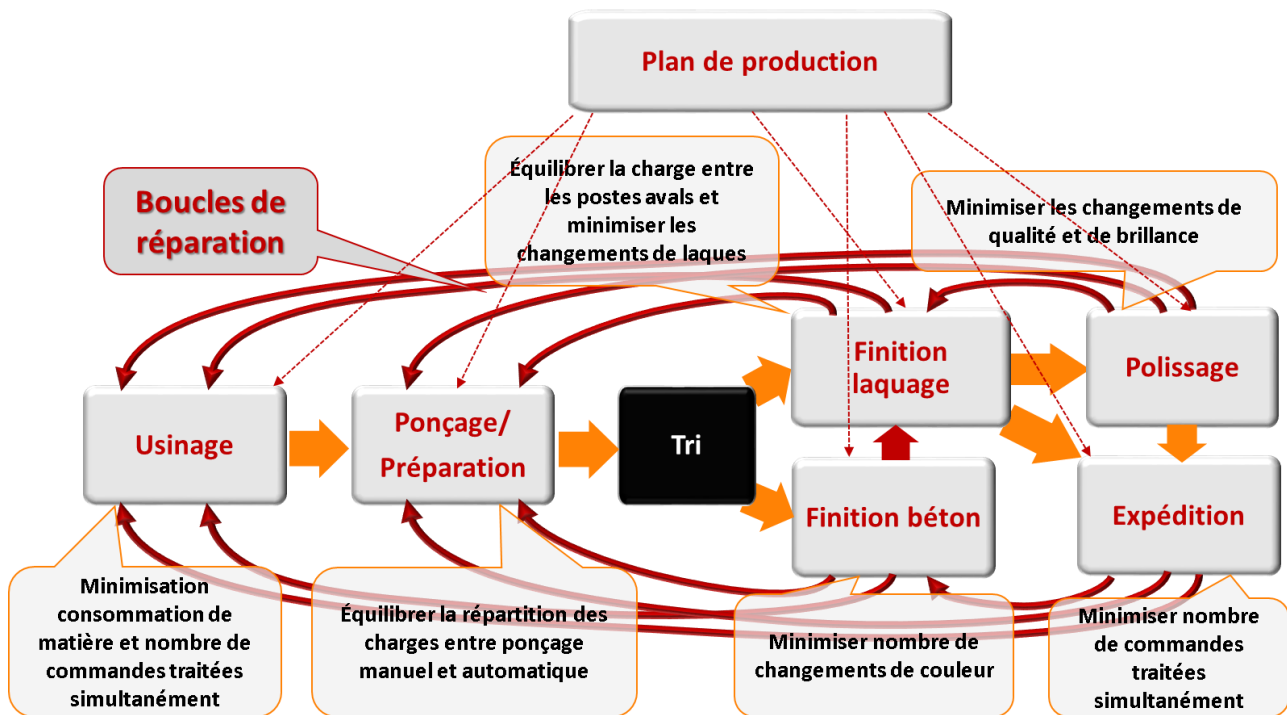


Figure 5 Objectifs des différents secteurs

L'objectif global reste toutefois de livrer en temps et en heure le maximum de clients avec la meilleure rentabilité possible. Or, en plus du phénomène de perturbation du flux induit par les boucles de réparations, la sortie de ces pièces à réparer de leur lot d'origine entraîne, à cause d'une traçabilité incomplète et de l'intervention de certains facteurs humains des pertes de pièces dans l'atelier.

Une étude réalisée dans les travaux de Noyel (Noyel, 2015) a mis en lumière le fait qu'au mieux, seulement 50% du temps de traversée des produits dans l'atelier était à valeur ajoutée. Ce qui signifie qu'il est potentiellement imaginable de réduire par deux le temps global de production.

Fort de cette remarque, il paraît alors évident que des outils et modèles d'ordonnancement sont nécessaires pour améliorer la productivité globale. En outre, il avait été envisagé par Mélanie Noyel au cours de sa thèse de réaliser une planification globale prédictive la plus juste possible (compte tenu de la variabilité du flux de produits dans l'atelier, la probabilité qu'une planification établie reste valable est très faible, ainsi « juste » signifie nécessitant le moins de remaniement possible) puis de donner les moyens nécessaires au système pour qu'il puisse soit s'auto-réguler dynamiquement, soit être réglé manuellement par des opérateurs via des outils d'alerte et d'aide à la décision.

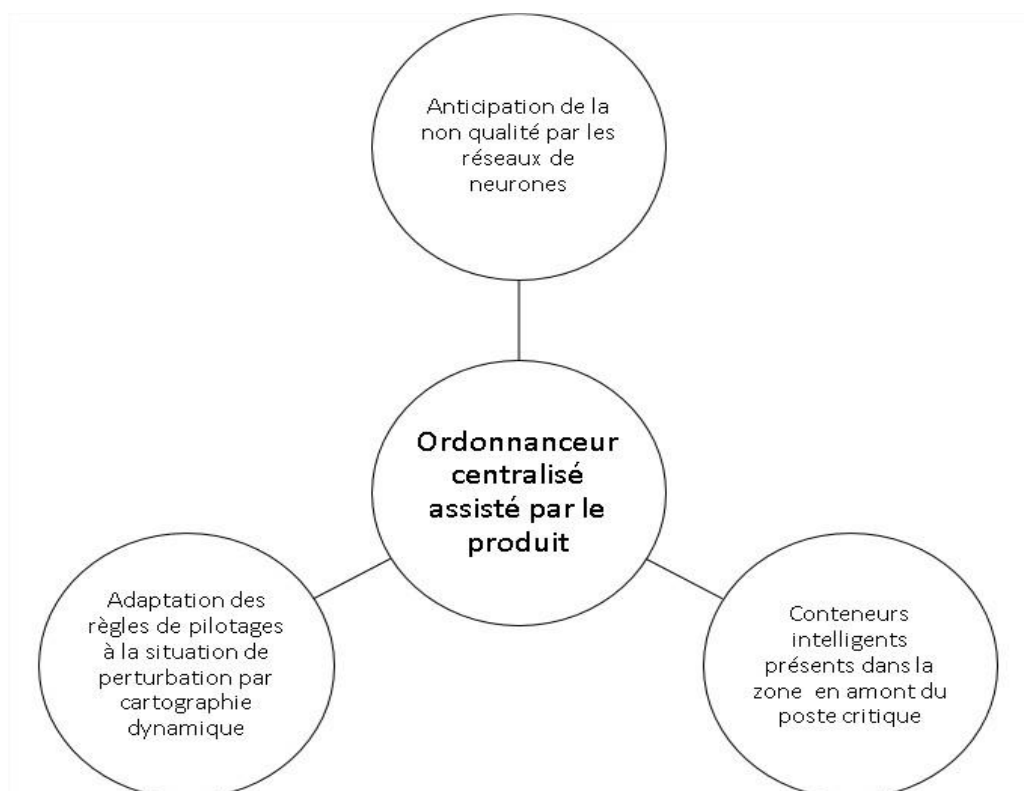


Figure 6 Axes de recherche thèse Mélanie Noyel

Dans sa thèse, Noyel a identifié trois axes principaux sur lesquels travailler afin de proposer une solution globale capable d'apporter une réponse intéressante à la problématique de l'entreprise, qui est de réaliser un ordonnancement à plusieurs niveaux :

- Global, afin de donner une vision de la charge hebdomadaire et de son avancement ainsi que de fournir des jalons pour évaluer rapidement une déviation.
- Intermédiaire pour optimiser au mieux la productivité utilisant les regroupements cités précédemment.
- Local, qui soit dynamique et ainsi capable de réagir aux différents aléas voir même de les anticiper dans certains cas.

Le schéma **Figure 6** présente ces trois axes, à savoir :

- L'anticipation de la non qualité en utilisant des réseaux neuronaux.
- L'adaptation dynamique des règles de pilotages en fonction de l'état du flux via une carte de contrôle.
- L'implantation de conteneurs intelligents dans la chaîne de production.

L'objectif à atteindre en combinant ces axes de recherche était la création d'un ordonnanceur centralisé assisté par le produit. Les deux premiers points ont été traités lors de sa thèse. Le troisième a été initié mais n'a pas pu être finalisé. Le projet actuel cherche à mettre en œuvre les axes déjà étudiés et à terminer la mise en place du troisième, pour atteindre un objectif que nous avons fait évoluer. En effet, nous en expliquerons les raisons plus tard dans le document, mais nous avons choisi d'aller plus loin que le développement d'un ordonnanceur spécifique. Notre objectif est d'élaborer un système de pilotage basé sur le contrôle par le produit pour toute l'entreprise en mettant en œuvre, si nécessaire, plusieurs ordonnancements, relatifs à des postes ou des zones de l'entreprise, pour atteindre les objectifs globaux.

De manière à simplifier la réalisation d'un tel ordonnancement, sans pour autant altérer le fonctionnement de l'entreprise, l'utilisation des lots élémentaires était nécessaire. Tous les travaux internes d'identification et de traçabilité menés jusqu'à aujourd'hui s'appuient sur leurs existences. L'une des missions pour lesquelles j'ai d'ailleurs dû œuvrer a été la fiabilisation de l'identification, l'amélioration et le déploiement de l'outil de traçabilité de ces lots élémentaires. Notre objectif est de réaliser un ordonnancement « suffisamment souple » pour prendre en compte les spécificités relatives à chaque atelier tout en assurant qu'au poste d'expédition tous les lots d'une même commande arrive quasiment en même temps. Une des contraintes à laquelle devra se soumettre l'ordonnancement est de s'assurer que la charge de tous les ateliers soit équilibrée. En outre, celui-ci doit être réactif face aux différents aléas et en particulier à l'ensemble des problèmes liés à la qualité.

Par conséquent, un des principaux axes sur lesquels agir est la réduction du taux de non-qualité, équivalent au taux de reprises effectuées sur les produits. À savoir qu'un certain nombre de ces défauts peuvent être corrigés ou évités. Comme expliqué en début de chapitre, le taux de reprise est très élevé, à cause des fortes exigences qualité attendues par les clients, mais cela est également dû au fait que lorsqu'un défaut est généré sur une pièce, celui-ci n'est pas forcément repérable immédiatement. Il est possible qu'un défaut créé lors de l'usinage ne devienne visible qu'après laquage ou lustrage, soit après plusieurs jours de traitement. Lors des précédents travaux, il a été montré qu'il était possible de prédire l'apparition de certains de ces défauts, cette prédiction reposant sur la collecte et l'exploitation en temps réel de données de production. Nous avons donc pour objectif de tirer parti de cette capacité de prédiction afin de la prendre en compte lors de l'élaboration de réordonnements dynamiques.

Après avoir fait état de la situation initiale, en décrivant les enjeux et les spécificités de l'entreprise, nous passons, à présent, à la seconde partie où nous cherchons les problématiques traitées dans la littérature, se rapprochant au plus près de notre cas **Figure 7**.

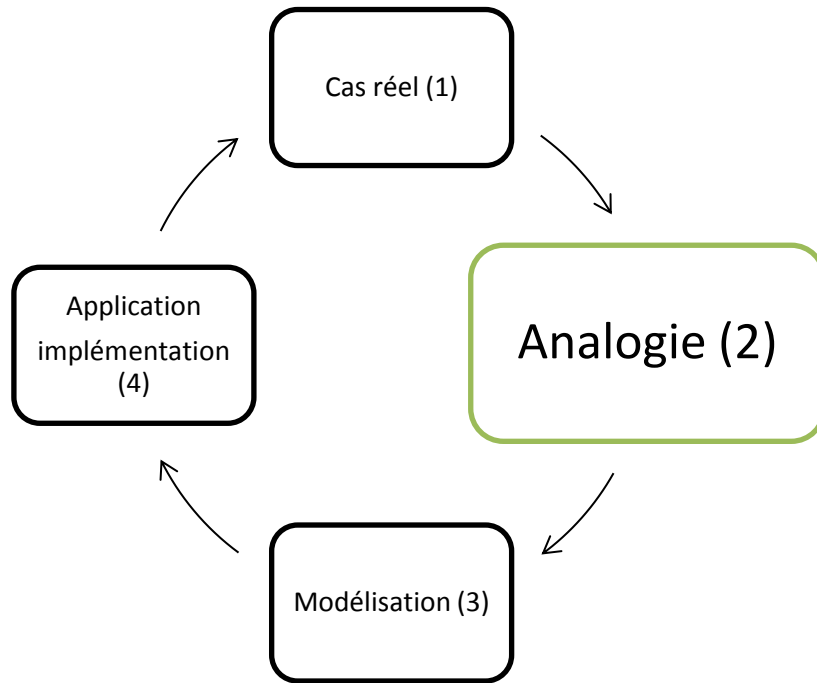


Figure 7 Analogie

II. Analogie (état de l'art)

A. Problèmes d'ordonnancement d'atelier

Comme dit à la fin du chapitre précédent, les problèmes scientifiques soulevés par la problématique industrielle appartiennent à la catégorie d'ordonnancement de chaîne de production, thématique vaste ayant fait l'objet de nombreux travaux. Avant de définir plus spécifiquement le type d'atelier qui correspond à notre problématique, il convient de faire un rappel générique des problèmes d'ordonnancement d'atelier. La définition d'un problème d'ordonnancement est : « programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution » (Carlier and Chrétienne, 1982).

Les problèmes d'ordonnancement d'atelier se définissent suivant trois caractéristiques, notées $\alpha|\beta|\gamma$ qui sont respectivement :

- Le type d'atelier.
- Les informations relatives aux réglages et autres conditions de fonctionnement de l'atelier.
- Le(s) critère(s) à optimiser, notation définie par (Lawler et al., 1989).

Ainsi nous pouvons retrouver les types suivants d'ateliers (α) :

- 1 : une seule machine est considérée et tous les travaux (jobs) se résument à une seule tâche (opération) (Abdul-Razaq et al., 1990).
- Pm : m machines en parallèle, où les jobs sont aussi composés d'une seule opération mais avec la possibilité de la réaliser sur différentes machines.
- Qm : m machines en parallèle, ayant éventuellement des temps de transformation différents (SHMOYS et al., 1995).
- Fm : lorsque la gamme de fabrication est identique et linéaire pour tous les jobs, le problème est appelé « flowshop » (Nawaz et al., 1983)
- Jm : lorsque la gamme est linéaire mais variable suivant les jobs, il s'agit à présent d'un problème de « jobshop » (Blazewicz et al., 1993)

- *Om* : lorsque la gamme n'a pas de contrainte par rapport au nombre d'utilisation de chaque machine, le problème étudié est alors un « openshop » (Gonzalez, 1976).

Les différentes conditions d'ateliers généralement considérées sont les suivantes (β)

(Allahverdi et al., 2008) :

- *Rj* : pas de job prêt à l'instant 0 ou date de mise à disposition dynamique
- *Nwt* : pas d'attente
- *zbfrr* : pas de zone tampon
- *block* : avec blocage
- *prmp* : avec préemption
- *prec* : avec contraintes de précédence
- *brkdown* : avec pannes machine

Conditions spécifiques aux réglages :

- *STsi* : temps de réglage indépendant de la séquence
- *SCsi* : coût de réglage indépendant de la séquence
- *STsd* : temps de réglage dépendant de la séquence
- *SCsd* : coût de réglage dépendant de la séquence
- *STsi, b* : temps de réglage indépendant de la séquence des lots
- *SCsi, b* : coût de réglage indépendant de la séquence des lots
- *STsd, b* : temps de réglage dépendant de la séquence des lots
- *SCsd, b* : coût de réglage dépendant de la séquence des lots
- *Rsi* : temps de retrait indépendant de la séquence
- *Rsd* : temps de retrait dépendant de la séquence

Pour les critères de performance (γ) communément mesurés, nous avons :

- *Cmax* : makespan
- *Lmax* : retard maximale
- *TSC* : coût total de réglage
- *TST* : temps total de réglage
- *TFT* : temps d'écoulement total
- *TCT* : temps total de complétion
- *TE* : avance totale
- *TL* : retard total

- *MFT* : temps d'écoulement moyen
- *MCT* : temps de complétion moyen
- *MT* : retard moyen
- *NLJ* : nombre de jobs en retard
- *WTE* : avance totale pondérée
- *WTT* : retard total pondéré
- *WMFT* : temps d'écoulement moyen pondéré
- *WTFT* : temps d'écoulement total pondéré
- *WMCT* : temps de complétion moyen pondéré
- *WTCT* : temps de complétion total pondéré

L'une des spécificités de l'entreprise est d'être composée de six ateliers clients et fournisseurs les uns des autres mais possédant tous leurs propres critères d'optimisation et leviers d'action possibles. Chacun d'eux mériterait une définition de son problème d'ordonnancement pourtant s'il fallait synthétiser cet ensemble en un seul problème, on le classerait dans la catégorie suivante :

La chaîne de fabrication de l'entreprise correspond à l'étude d'un problème de type « openshop », à cause des éventuels bouclages internes prévus dans la gamme de fabrication de certaines familles de produits. Il faut considérer la présence de pannes machines et l'arrivée en dynamique de nouveaux jobs à ordonnancer, de plus il faut inclure les temps de réglages qui sont dépendants de la séquence de lots. Les critères généraux de performance à mesurer seraient le temps total de réglage, le retard total et le makespan, ainsi le problème s'écrirait : $O_m | \{r_j, brkdown, STsd, b\} | \{C_{max}, TL, TST\}$.

Cependant la complexité du problème global n'est pas encore complètement décrite par cette modélisation, en effet, ce qui amplifie davantage la difficulté du problème réside dans les fortes perturbations auxquelles est soumise la chaîne de production. Il faut donc resserrer le zoom sur les problèmes d'ordonnancement dynamique. Nous nous retrouvons donc face à un problème d'ordonnancement dynamique soumis à des incertitudes difficilement mesurables (ou tout du moins modélisables par des lois probabilistes), difficulté généralement contournée par la gestion de stocks de sécurité. Cela est impossible dans le cas de l'entreprise car elle produit en juste à temps en proposant une diversité de dimensions et de couleurs, impliquant l'incapacité de créer du stock.

Dans la catégorie des problèmes d'ordonnancement dynamique, deux objectifs sont à considérer : la flexibilité et la robustesse (Billaut, 2004). La réalisation des algorithmes cherche à établir le meilleur compromis entre ces deux objectifs.

La flexibilité correspond à l'insertion de degrés de liberté dans l'ordonnancement (flexibilité temporelle, séquentielle, d'affectation ou de mode d'exécution) afin d'autoriser certaines permutations sans perturber l'ordonnancement. La robustesse, quant à elle, de manière, peut être définie, de manière simplifiée, comme la mesure de la sensibilité d'un ordonnancement face à l'incertitude des données et des aléas (Billaut, 2004).

Il y a, selon la littérature, deux types d'ordonnancement dynamique qui tiennent compte d'éléments perturbateurs tels que l'apparition de nouveaux jobs après que le planning ait été défini ou les multiples flux de réparations pouvant survenir à tout instant (Tanimizu et al., 2006) : le séquençement dit « temps-réel » et le séquençement dit « réactif ».

Le séquençement nommé « temps réel » (Hatono et al., 1998; Shin and Kuroda, 1998) : sa particularité est de ne pas avoir de planning préétabli. Son avantage premier est d'être efficace (à comprendre au sens de rapidité de réponse) et d'évoluer rapidement face à des événements imprévus, cependant il souffre de problème d'optimisation vu que son objectif est de fournir une réponse acceptable aussi rapidement que possible.

Le séquençement nommé « réactif » (Du et Chiou, 2000; Jain et Elmaraghy, 1997) : utilise la plupart du temps des algorithmes génétiques ou des heuristiques d'optimisation basées sur les colonies de fourmis pour la ré-optimisation du planning initial dès que cela est nécessaire. Les performances du séquençement réactif dépendent pourtant du modèle choisi pour réaliser le planning initial ainsi que des moyens de le développer en utilisant un algorithme d'ordonnancement prédictif (Zhao et al., 2013), qui propose un ordonnancement, évalue cette prédiction et selon son niveau de fiabilité, l'applique ou la ré-évalue.

Les méthodes de résolution des problèmes d'ordonnancement sont diverses et variées, dans cette étude nous nous focaliserons quasiment exclusivement sur les techniques d'ordonnancement avec groupement de produits, appelées technologie de groupe (GT), définies comme « une philosophie industrielle qui identifie et exploite les similitudes inhérentes des pièces et des processus de fabrication » (Ham et al., 2012), compte tenu du fait que nous travaillons avec des lots élémentaires. Les problèmes traitant de l'ordonnancement de lots (aussi désignés par l'appellation « Scheduling with batching ») ont été principalement définis par (Potts and Kovalyov, 2000). Ils indiquent que les travaux (jobs) peuvent être rassemblés lors de leur passage sur une machine s'ils possèdent les mêmes configurations ou que la machine est en mesure d'effectuer plusieurs travaux en simultané. Ainsi un lot est défini « comme un ensemble de travaux planifiés de façon contiguë sur une machine, partageant une même configuration ». **Ce choix résulte des travaux précédents menés pendant la thèse de Noyel avec l'instanciation des lots élémentaires sur lesquels nous voulions construire notre proposition et qui seront la clé de voûte de l'architecture de pilotage que nous souhaitons réaliser et dont nous parlerons plus loin.**

La GT lorsqu'elle est appliquée dans des problèmes d'ordonnancement soulève des problèmes dits d'ordonnancement de groupe (GS). Dans le cadre spécifique des problèmes flowshop, on parlera alors de Flow Shop Group Scheduling (FSGP). (Neufeld et al., 2016), en accord avec la littérature explique que : « la technologie de groupe trouve sa principale implémentation en milieu industriel via l'industrie cellulaire (CM) »... « Résoudre ce type de problématique revient à traiter trois problèmes de planification, traités la plupart du temps indépendamment ». Ces trois étapes sont :

- La formation des cellules, en regroupant les postes de travail ainsi que la génération des familles de produits.
- L'agencement de l'atelier pour positionner les cellules de production.
- L'ordonnancement des tâches dans et entre les cellules, cette sous-partie du problème correspond à l'étude de la classe de problème des ordonnancements de groupe.

Dans notre étude, nous ne traiterons que du groupement des produits en famille et de l'ordonnancement de ces groupes, car aucun réagencement de l'atelier n'est souhaité dans l'entreprise, la partie formation de cellules de production, quant à elle, serait très limitée vis-à-vis du parc machine de l'entreprise.

Notre but principal est de fournir un ordonnancement dynamique capable d'adapter la production aux perturbations qu'elle peut subir. Une possibilité envisagée pour atteindre cet objectif est de modifier, à la volée, la taille des lots de fabrication ou encore de les réordonner, chaque fois que les objectifs ou les données d'entrée changent. Cette catégorie de problème consiste à élaborer des lots de fabrication afin de minimiser les gaspillages de temps et de production. Il y a, au moins, deux méthodes d'interprétation de ce problème :

- Le lot-sizing : Prendre l'ensemble des produits à réaliser et les subdiviser en lots de taille optimale (Fernandes et al., 2016), (Ramya et al., 2016). Nous ne nous intéresserons pas particulièrement à cette classe de problème car nous considérons, d'après nos hypothèses, que les lots élémentaires qui sont nos données d'entrées sont des regroupements minimaux déjà optimisés vis-à-vis des postes aux contraintes les plus importantes. Par conséquent, il serait incohérent de chercher à éclater des regroupements déjà établis (lots élémentaires, créés pour être des groupes optimaux de produits vis-à-vis de la gamme de production) pour les remplacer par de nouveaux, valables uniquement à ce poste de travail.
- Part-family : Rassembler les lots élémentaires en des groupements plus importants avec des caractéristiques communes (Taboada and Coit, 2007), (Co, 1990; Han and Ham, 1986). Ceux-ci cherchent à grouper les produits suivant les critères de production (tel que l'épaisseur ou la gamme de fabrication).

Un des problèmes qui doit être traité pour l'entreprise est la composition des lots de fabrication, à savoir : comment regrouper au mieux les lots élémentaires plutôt que le nombre de produits à mettre dans des lots de fabrication. La classe de problème « part-family » est la plus adaptée à notre cadre d'étude. Il a été prouvé dans (Logendran and Sriskandarajah, 1993) que, déjà dans le cas d'un problème à 2 machines à séquences indépendantes, la catégorie des FSGP est NP-difficile. C'est pourquoi, les méthodes heuristiques sont privilégiées pour les résoudre. De nombreux travaux traitent de cette classe de problème et de ses différentes méthodes de résolution possibles, une grande partie d'entre elles utilisent des techniques de clustering :

- (Bowers et al., 1995) les utilisent afin de minimiser les temps de changement de série dans des problèmes, indépendants de la séquence. Dans le cas de l'entreprise, une contrainte forte par rapport au nombre de commandes à intégrer dans un lot de fabrication existe. Par contre, les temps de changement de série sont, selon les postes de charge, soit :
 - Indépendants de la séquence (pour l'atelier usinage, passer d'une série à une autre consiste à prendre les outils de coupe adaptés à l'épaisseur de panneau à travailler et à changer de type de panneau à débiter).
 - Fortement dépendant (au laquage, le changement de coloris influe sur le temps de nettoyage des buses et des circuits d'amorçage).
- (Taboada and Coit, 2007), (Co, 1990; Han and Ham, 1986) ont travaillé sur du clustering multicritères. Ces travaux montrent que les problèmes de groupement peuvent complètement différer d'une production à une autre. Dans notre cas, bien que le nombre de lots élémentaires dans chaque lot de fabrication ne soit pas une contrainte forte, l'écart-type doit rester faible afin de ne pas créer des lots de fabrication disproportionnés et donc permettant de lisser la charge de production. Des méthodes comme les algorithmes génétiques pour résoudre ce problème ont été étudiées par (Lee-Post, 2000), par exemple.

Pour synthétiser, ce type de problème d'ordonnancement peut s'écrire de la manière suivante : L'ensemble des jobs à réaliser J est divisible en F familles de produits, en notant n_f le nombre de job pour la famille f . Le temps de setup séparant deux jobs d'une même famille est considéré comme nul. En revanche, lorsqu'un job d'une famille f_j est traité après un job d'une famille f_F le temps de réglage sur la machine m (Setup time) est non nul et noté S_{mkj} ou S_{m0j} si k est le premier travail. Ainsi les problèmes indépendants de la séquence possèdent la propriété suivante : $\forall (k, j) \in F \rightarrow S_{mkj} = S_{m0j} = S_{mj}$. De même, les problèmes indépendants de la machine $\forall (m, n') \in F \rightarrow S_{mkj} = S_{nkj} = S_{kj}$.

Les contraintes et propriétés des sous-ensembles de la chaîne de production sont :

- L'usinage, où il serait logique de rassembler les lots aux propriétés géométriques similaires.
- Le laquage, où un regroupement par couleur et rapprochement des teintes similaires serait un facteur d'amélioration.

- Les expéditions où il faudrait avoir à disposition la totalité des produits d'une même commande dans un intervalle de temps réduit, autant que faire se peut.

Leur simple évocation suffit à nous convaincre que traiter cette problématique comme un seul et unique problème d'ordonnement aux objectifs différents voir même contradictoires est d'une bien trop grande complexité, en particulier, si nous cherchons à obtenir des performances satisfaisantes tant d'un point de vue global que localisé à chaque atelier voir même à chaque poste de charge.

Nous pensons donc qu'il vaut mieux subdiviser le problème d'ordonnement global en plusieurs sous-problèmes que nous qualifierons de locaux. Il faudra par la suite, mettre en place un système capable de mettre en interaction et faire coordonner ces ordonnanceurs locaux pour garantir un optimum global, tout comme cela a déjà été traité dans plusieurs travaux, comme ceux de Dumetz (Dumetz et al., 2017) par exemple. C'est pourquoi nous allons étudier les architectures de pilotage d'atelier.

B. Architectures de pilotage d'atelier

Au début des années 2000, l'équipe Systèmes Contrôlés par le Produit (SCP) du laboratoire CRAN a orienté ses travaux sur les systèmes intelligents de production, sous la direction du Pr Gérard Morel, convaincu de l'apport que peut avoir le produit dans la prise de décision. Les travaux de (Valckenaers, 2001) et de (McFarlane et al., 2003) vont également dans ce sens. D'un point de vue structurel, les systèmes de pilotage de production peuvent être répartis en trois catégories.

1. Les systèmes centralisés

Ils ont vu le jour dans les années soixante avec l'apparition, tout d'abord, des systèmes implémentant la logique MRP (Material Requirement Planning) puis MRP² (Manufacturing Resource Planning) ainsi que par la théorie des contraintes (ToC) (Vollmann et al., 1997), dont les résultats démontrent que la bonne gestion des postes goulots d'étranglement permet d'obtenir des résultats indéniables.

La notion de CIM (Computer integrated Manufacturing) commence à se développer au sein des communautés scientifiques. La finalité de ce système est de rassembler et faire collaborer l'ensemble des acteurs du système de production en supervisant l'ensemble des opérations de l'entreprise. Les CIMs se voulaient être en mesure de réagir avec flexibilité aux changements et de proposer de solutions de meilleure qualité. Tel ne fut pas le cas, en effet, nombre de systèmes se révélèrent trop rigides, inaptes à proposer des solutions dynamiques face aux changements.

2. Les systèmes décentralisés

Ils ont émergé au début des années quatre-vingts en s'inspirant de la philosophie JàT (Juste à temps) et la mise en service des approches telles que le Six Sigma ou le Lean Manufacturing dont la plus célèbre représentante est la méthode Kanban pour les méthodes de management visuels. Dans les années quatre-vingt-dix, le paradigme des Intelligent Manufacturing System (IMS) (Yoshikawa, 1995) remet en cause celui des CIM, conduisant vers des architectures plus décentralisées de manière à obtenir plus de réactivité que leurs prédécesseurs. Nous pouvons par exemple citer :

- Les architectures holoniques (HMS), groupement d'entités récursives nommées « holons », ces entités sont autonomes, coopératives et définies comme une décomposition fonctionnelle via des concepts orientés objets. Les holons se créent de manière dynamique, mais prédéfinie durant la phase de conception, ils ont chacun des rôles spécifiques mais peuvent réagir aux changements de statut des autres entités grâce à leurs propriétés de coopération et de négociation. Les holarchies possèdent de fortes propriétés d'auto-organisation et des facilités pour communiquer (Valckenaers et al., 1994).

- Les architectures biologiques (BMS) s'inspirant des comportements de la biologie (animalier, végétal, corporel, ...) bien connus pour leurs capacités de réactivité contre les modifications environnementales (Ueda, 1992). La bionique correspond à imiter le vivant, comme par exemple dupliquer l'organisation cellulaire, naturellement fondée sur un modèle hiérarchique ; reproduire les colonies ou les essaims d'animaux faisant parti des systèmes à population sociales construits comme un modèle d'hétéarchie où tous les habitants peuvent participer de manière égale aux décisions. Les entités de base sont les cellules ou « modelon » basés sur des concepts comme les enzymes ou l'ADN. Chaque nouvelle cellule se crée par reproduction et réorganisation. Elles possèdent toutes des fonctions prédéfinies mais pouvant être changées pendant la phase d'exploitation. Grande capacité de réaction grâce aux approches biologiques. Les cellules se regroupent en organes pour réaliser des fonctions spécifiques.
- Les architectures fractales (FMS) inspirées par la théorie des fractales (Mandelbrot, 1975), ces architectures ont la particularité de ne pas posséder de structure formelle tout en étant capable de s'adapter aux aléas de leur environnement (Tharumarajah, 1996). La Fractal est une unité autonome et autosimilaire ayant pleine connaissance de son environnement. Les nouvelles fractales se créent par reproduction en utilisant le phénomène d'auto-organisation. Les objectifs de chaque fractal sont définis à l'avance mais peuvent être réassignés dynamiquement. La flexibilité est assurée par la restructuration du système en employant des processus d'auto-organisation et d'auto optimisation. Comme toutes les entités sont récursives, la capacité de restructuration est applicable à tous les niveaux.

3. Les systèmes hybrides

Les systèmes dits hybrides, cherchant à tirer parti des avantages des deux approches précédentes, à comprendre comme un mélange de choix tantôt centralisés tantôt distribués, ne cessent de se développer depuis lors. Ceux-ci sont au cœur des problématiques actuelles, car les architectures de pilotages d'atelier constituent l'un des grands enjeux de la quatrième révolution industrielle. (Cardin et al., 2018) dit que : « il devient alors nécessaire de définir des systèmes informatiques d'entreprises dotés d'architectures de contrôle connectées, interopérables, flexibles et réactives. L'une des approches possibles est alors d'utiliser et d'intégrer verticalement les architectures de contrôle holoniques ». Quelques-uns des derniers représentants de ces systèmes que l'on peut citer sont : ADACOR²(Barbosa et al., 2015), POLLUX(Jimenez et al., 2017) ou encore ORCA (Pach, 2014).

Dans le cadre d'applications industrielles, les thèses dirigées par le Pr André Thomas ont permis de démontrer que les architectures hybrides de pilotage sont les seules pouvant amener une réelle avancée (Haouzi, 2017). L'étude de ce type d'architectures est également un sujet d'actualité aux niveaux nationaux et internationaux. Nous pouvons citer, par exemple, les travaux de (Cardin et al., 2017) sur la coopération et les interactions entre les décisions centralisée et distribuées, ceux de (Valckenaers, 2019) ou de (Barbosa et al., 2015).

L'implémentation de tels systèmes complexes adaptables est, dans la majorité des cas, basée sur les paradigmes de l'intelligence artificiel distribuée (AID) et la programmation orientée objet (POO).

Dans le cadre du pilotage d'ateliers aux flux perturbés, on peut distinguer deux, voire trois types de processus relativement au pouvoir d'anticipation face aux événements (Berchet et al., 1999) :

- La prédiction mettant en œuvre des algorithmes pouvant faire appel à des modèles de simulation pour déterminer a priori l'état du système dans un futur plus ou moins proche à partir de son état actuel et de sa base de connaissance acquise au cours du temps.
- La réactivité, agissant suite à un écart à la consigne, cherche à corriger les effets de tout événement perturbant (connu ou non) le bon déroulement des activités.
- La proactivité, mixant des techniques réactives et prédictives afin d'agir par anticipation.

- Nous pouvons également citer l'interactivité, généralement centré sur l'interaction homme-machine.

Des hybridations comportementales entre réactif et prédictif ont commencé à être étudiés afin de tirer parti des avantages des deux types (Haouzi, 2008; Klein, 2008; Pannequin, 2007). L'évolution de la technologie a également permis l'essor de solutions dites proactives cherchant à anticiper les perturbations afin de les prévenir (Talibi et al., 2013) par l'intermédiaire de simulation on-line. Des techniques issues du domaine de l'intelligence artificielle, basées sur des modèles d'apprentissage, tels que les réseaux de neurones, sont typiquement utilisées à de telles finalités. Notamment, elles peuvent être mises à contribution pour prédire, si dans les conditions actuelles de fonctionnement, la production risque d'être non-qualitative et le cas échéant, proposer une alternative viable. Le réseau de neurones développé dans (Noyel et al., 2016) peut être cité en exemple pour ce type d'application. Cette manière de distinguer les types de processus décrit la vision fonctionnelle des systèmes. Il existe également des aspects dits : organique et évolutionniste des systèmes de contrôle.

Quel que soit l'architecture choisie, il faut que le système de pilotage soit défini de manière suffisamment générique pour intégrer tout nouvel élément et assez modulable pour accepter des modifications d'objectifs divers et variées. Aujourd'hui les architectures hybrides commencent à être nombreuses et il serait fastidieux et peut-être hors propos d'en faire un état des lieux exhaustif. Nous allons donc procéder de la manière suivante, à partir de la problématique que nous avons explicitée dans le précédent chapitre et les considérations que nous avons mis en évidence vis-à-vis de l'ordonnancement, nous allons définir une liste de critères avec lesquels nous allons comparer un certain nombre de ces architectures.

Un ordonnancement dynamique hybride implique que certaines décisions seront prises de manière centralisée et d'autres de manière distribuée, les questions qui en découlent sont : quel mode de contrôle choisir pour quelle décision ? Comment donner l'intelligence aux produits pour qu'ils prennent les décisions en local ? Pour que l'ordonnancement multi niveaux soit pertinent et cohérent, il faudra également assurer une transmission d'information multidirectionnelle entre ces niveaux mais aussi entre les entités d'un même niveau.

Ci-dessous, se trouve la liste des caractéristiques, découlant de la problématique et du mode de fonctionnement de l'entreprise, que doit posséder, selon nous, le système de pilotage que nous utiliserons.

Liste des critères :

- **Intégration dans le système de l'entreprise** : gestion des informations envoyées par les ressources, les produits et l'ERP en temps réel.
- **Modularité** : mise en place de l'architecture dans l'entreprise, pouvoir implémenter le système petit à petit sans arrêter la production.
- **Évolution temporelle** : nécessité qu'à tout moment un nouvel équipement installé soit compatible avec l'ensemble des composants du système déjà présents.
- **Réactivité (rapidité d'exécution et prise en charge des aléas)** : l'ensemble des composants du système et de leurs interactions se doivent d'être rapide afin de pouvoir faire face au mieux aux perturbations dans les plus courts délais possibles.
- **Répétabilité** : garantie du bon fonctionnement des algorithmes, d'une transmission correcte et juste des informations.
- **Interaction homme-machine** : les éléments du système doivent être en mesure de prendre en compte les décisions humaines, de les accepter et de les faciliter (voir *Annexe I.B*).
- **Respect des performances et indicateurs globaux** : dans un système composé de plusieurs optimisateurs, il est essentiel qu'en plus de chercher à atteindre leurs propres objectifs, ceux-ci essayent également de réaliser les objectifs globaux chers à l'entreprise (satisfaction client, qualité élevée et respect des délais).

Vis-à-vis de cette liste de caractéristiques que nous avons établie, nous avons tenté de comparer un certain nombre de systèmes de pilotage afin d'orienter notre décision dans la sélection d'une architecture à utiliser, le résumé de cette étude est présenté dans le tableau ci-dessous.

Architectures Critères	ORCA-FMS (PACH, et al., 2014)	SoHMS (QUINTANILLA, et al., 2014)	PROSIS (PUJO, et al., 2009)	ADACOR² (DEBNATH, et al., 2015)	Herrera (Herrera et al., 2011)	Pollux(Jimenez et al., 2017)
Intégration	++	?	+	+/-	+	++
Modularité	?	?	+	+	++	+
Évolution temporelle	+	?	+	+/-	++	+
Réactivité	+	+	+	++	+	++
Répétabilité	?	?	?	++	?	?
Interaction H-M	+/-	+	-	?	+/-	?
Respect des performances	+	+	+	?	+	+

Tableau 1 Comparaison des architectures hybrides

Résumé :

- ORCA-FMS : Le système est capable de s'interfacer avec E.R.P. et M.E.S. La généralité des entités du système permet que tout nouvel équipement puisse facilement s'intégrer à l'existant. Ce système est pourvu d'un changement de mode qui peut s'activer dès que nécessaire afin de réagir en cas de perturbation. La couche de contrôle global permet d'assurer le suivi des objectifs globaux.
- SoHMS : Son processus a été conçu avec une phase réactivité. L'architecture inclut des holons ressources opérateurs, afin d'assurer une interaction entre humain et machine. Pour atteindre les objectifs globaux, chaque holon ressource a la capacité de refuser certaines disponibilités mais doit en contrepartie proposer des solutions alternatives.
- PROSIS : Le système fait la passerelle avec l'E.R.P. via les I_Products et les I_Resources. Sa capacité d'auto-organisation lui offre la possibilité de mettre en place le système petit à petit et d'incorporer de nouveaux éléments sans nuire au bon fonctionnement de l'ensemble. La réactivité du système est garantie par sa définition même. Son développement a été prévu pour des systèmes entièrement automatisés dans lesquels l'implication de l'homme cherche à être minimisé. Les entités nommés « ACE » assurent l'interaction entre toutes les parties prenantes pour assurer la réalisation des objectifs communs.
- ADACOR² : A priori, le système est capable de s'intégrer avec un E.R.P. et/ou un M.E.S. Ses capacités d'auto-organisation lui permettent une intégration au fur et à mesure et l'ajout de nouveaux éléments, cependant comme le système fonctionne avec une liste de comportement pour son pilotage, un nouvel élément peut nécessiter de revoir cette liste. En plus de la liste de comportement, un moteur de sélection permet de s'adapter aux perturbations. Un stabilisateur de nervosité permet au système de ne pas sortir de sa zone d'action habituelle.

- HERRERA : La partie « données » qui est construite en miroir de la partie physique intègre les données issues de l'E.R.P. Sa mise en place peut se faire étape par étape sans gêner le fonctionnement de l'entreprise. La généricité des entités doit permettre de prendre en compte de nouvelles ressources ou de nouveaux produits au fil de l'eau. Les décisions prises localement en temps réel doivent assurer la réactivité. De base, aucun agent « humain » n'a été introduit dans l'architecture mais leur prise en compte ne semble pas impossible. Les décisions globales quant à elle assure la bonne réalisation des objectifs globaux.
- POLLUX : Le système a été conçu de manière à s'interfacer avec les outils du système d'information. La définition des entités et de leurs interactions semble permettre une mise en place progressive et l'ajout simple de nouveaux composants. Comme pour ORCA-FMS un mode de basculement existe, il s'active soit périodiquement soit suite à un événement perturbateur. La couche supérieure (coordination) de l'architecture a été prévue pour assurer l'atteinte des objectifs globaux.

Conclusion :

Les outils, critères et modèles de décisions sont extrêmement contraints par les choix/préférences du concepteur, ainsi que leurs capacités et les spécifications des problèmes étudiés. Le CRAN a exploré, avec les travaux d'Herrera (Herrera et al., 2011), un cadre générique basé sur le paradigme VSM (Viable System Model) introduit par(Beer, 1984), celui-ci expliquait qu'une organisation ne doit pas être vu comme une entité pyramidale. Au contraire, elle doit être considérée comme collaborative divisée en trois parties : l'unité de pilotage, le système opérant et l'environnement. Dans l'unité opérante, le même schéma est appliqué et prendra à son propre niveau une partie des décisions. Cela donnant au système une propriété de récursivité dans sa structuration.

Bien que conscient des aspects sur lesquels d'autres architectures semblent peut-être plus prometteuses que celle-ci, nous avons souhaité partir du cadre, défini par le modèle d'Herrera, sur lequel nous avons bâti notre proposition car ce modèle bien que servant de référence pour l'équipe ISET de notre laboratoire n'a jamais vraiment eu d'implémentation réelle en milieu industriel(Haouzi, 2017; Thomas, 2014). Nous voulions donc tenter une concrétisation de cette architecture et selon les résultats et les difficultés rencontrées pouvoir valider ou invalider celui-ci.

Dans le métamodèle d'Herrera, l'intelligence distribuée aux entités, est celle des systèmes MRP2 où les décisions centralisées ont été distribués entre elles. La **Figure 8** est une représentation de son modèle. Cette représentation définit dans les grandes lignes la structure que doit avoir le système de pilotage, mais tout l'enjeu repose sur la manière dont les niveaux de hiérarchie vont être définis. Les spécificités qui les différencieront seront notamment les données qu'ils seront à même de posséder et de traiter, le tout, contraint par les décisions auxquelles chacun des niveaux d'agrégation aura le droit de participer.

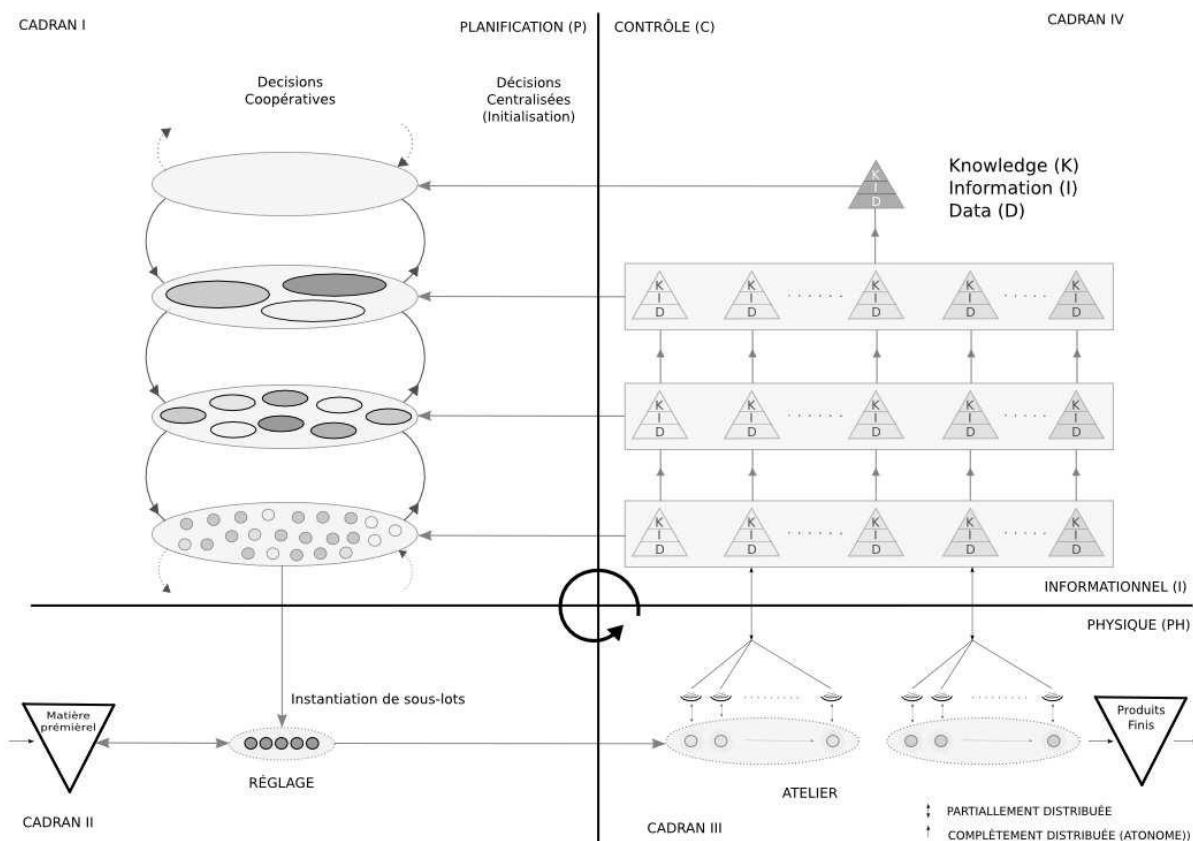


Figure 8 Méta modèle VSM de Carlos Herrera

Après avoir établi dans quelle classe de problème nous nous trouvons et l'architecture qui nous permettrait de mettre en œuvre les méthodes de résolution à ces problèmes, un bref état de l'art sur les mécanismes de coordination et les techniques synchronisation est présenté. Les méthodes et outils qui nous permettront de répondre aux problèmes d'optimisation des ordonnanceurs locaux sont présentés en **Annexe II** et **Annexe III**.

C. Les mécanismes de coordination

Avant de parler de synchronisation, il faut nécessairement aborder le thème de la communication et de la prise de décision partagée. Nous pouvons, par exemple, citer les travaux de (Smith, 1994) sur les Contract Nets Protocol (CNP), méthode typique de communication des systèmes multi-agents, ou ceux de (Jennings, 1995) sur la négociation ou encore les travaux réalisés sur les techniques de consensus (Olfati-Saber et al., 2007). Une première approche permettant de faire converger plusieurs entités autonomes pourrait être de leur faire respecter un ensemble de règles communes, que l'on nomme comportements sociaux.

Les systèmes multi-agents, faisant partie du domaine de l'intelligence artificielle distribuée, sont des groupes « d'agents » capables d'analyser et de se comporter en réponse à leur environnement, ses systèmes sont élaborés pour réaliser les objectifs définis par leurs concepteurs (Ferber, 1997). Ils sont considérés comme représentants emblématique des méthodes d'implémentation des architectures de pilotage intelligent. Des plateformes logicielles spécifiques au développement des systèmes multi-agents ont été développées, en accord avec les normes FIPA (Foundation for Intelligent Physical Agents) (O'Brien and Nicol, 1998). Au CRAN, nous avons choisi d'utiliser la plateforme JADE (Java Agent DEvelopment framework) (Bellifemine, 1999). Cette plateforme est constituée de trois composantes basiques : un environnement d'exécution, une bibliothèque de classes et des outils graphiques utiles à l'administration des agents.

Le choix d'un modèle de développement guide grandement les mécanismes d'interaction et surtout de communication entre les éléments du système. Il reste tout de même à déterminer les mécanismes de coordination. Dans sa thèse, Pannequin (Pannequin, 2007) décrit certains des mécanismes existants. Pour une architecture hybride, il évoque deux types de coordination :

- La coordination verticale : un mécanisme hiérarchique, principalement présent dans les systèmes où les centres décisionnels sont structurés de manière pyramidale avec des flux d'informations, majoritairement verticaux. Les consignes se transmettent du sommet de la pyramide vers les centres de décisions inférieurs et la remontée des résultats s'effectuent en miroir de la descente. Ce mécanisme est donc définissable par deux processus l'agrégation et la désagrégation des objectifs. Il est à noter que dans le processus d'agrégation des données vont nécessairement se perdre. La coordination hiérarchique est caractéristique d'un système centralisé (Genin et al., 2008).

- La coordination hétérarchique : ce type de coordination s'inspire majoritairement des phénomènes sociaux humains ou animaliers et caractérise plutôt des systèmes décentralisés. Le CNP, exemple typique de coordination de ce type, s'inspire du fonctionnement d'un marché : un émetteur envoie n appels à proposition avec un délai maximum de réponse admissible. Parmi les récepteurs, m répondent soit par une proposition soit par un refus. Lorsque le délai est dépassé, l'émetteur réalise une évaluation des propositions obtenues et sélectionne celle(s) qui lui convient puis notifie tous ses interlocuteurs de sa décision. Les émetteurs des propositions retenues renvoient une information de réussite ou d'échec de la demande à l'émetteur d'origine. Nous pouvons aussi citer des mécanismes type **enchère** pour les inspirations humaines. La stigmergie (avec des modèles tels que les colonies de fourmis) ou le fonctionnement par essaim (tel que le modèle Particle Swarm Optimisation) pour les modèles animaliers.

Lorsque l'on se place dans le cadre d'une entreprise dont la chaîne de production est divisible en plusieurs ateliers dont les objectifs peuvent être distincts voir même opposés, chercher à les synchroniser pour les faire converger vers un objectif commun tel que la minimisation des retards, des coûts ou la maximisation de la qualité nécessite de mettre en place des mécanismes de coordination. Selon (Bareigts, 2000), la coordination se définit comme « un ensemble de règles de fonctionnement établies par un ou plusieurs acteurs en vue de réaliser une tâche en commun ». Les problèmes de coordination en industrie ont fait l'objet de multiples études (Bhatnagar et al., 1993). Par exemple, l'importance de la coordination de plusieurs optimisateurs basés sur une optimisation globale pour une ligne de production automobile est décrite par (Muhl, et al., 2001). Parmi les différents mécanismes de coordination de chaîne de production, nous pouvons notamment citer (Bhatnagar et al., 1993):

- Flux poussé : la planification s'effectue en amont, par flux poussé (les activités sont toutes planifiées les unes après les autres, en commençant par les plus urgentes). Il s'agit là d'une technique simple ne nécessitant pas de gestion particulière, le suivi d'une règle de pilotage FIFO est suffisant pour l'appliquer, il faut veiller à ce que les données d'entrées des étapes en aval correspondent aux données produites par les étapes en amont.

- La planification en deux phases : une phase de propagation de la demande où la planification est faite avec un approvisionnement infini puis suivi par une planification à capacité réelle. Elle a, par exemple, été utilisée par (Gaudreault et al., 2010) dans ses travaux sur la planification de l'ensemble de la chaîne de production dans l'industrie du bois. Cette technique de planification a l'avantage d'être plus proche de la réalité.
- La planification goulot en premier : cette technique est une amélioration de la méthode précédente (planification en deux phases démarrant par le poste goulot) également utilisée dans (Gaudreault et al., 2010). Son avantage, par rapport à l'autre méthode, est qu'elle permet de maximiser l'utilisation de la ressource limitante et d'éviter les ruptures de chaîne.
- Rassemblement et vérification : méthode proposée par (Madureira et al., 2003) dans son étude des EJSSP (Extended Job Shop Schedule Problems) traitant des problèmes d'ordonnancement dynamique de type Job Shop. Elle suit une procédure de résolution qui commence par résoudre l'ordonnancement de chaque machine du problème séparément afin d'avoir des cas plus simples à traiter, puis à les rassembler et à valider qu'une solution les rassemblant est réalisable. Dans le cas où la solution n'est pas réalisable un mécanisme de réparation nommé « IMACM (Inter-Machine Activity Coordination Mechanism) » est employé en coordonnant les activités des machines. Cette coordination s'effectue en prenant en considération les précédences dans les opérations ainsi que les autres contraintes du système tout en conservant l'ordre d'attribution des tâches pour chaque machine. Il s'agit là d'une solution vers laquelle nous pourrions tendre une fois l'architecture mise en place.

Conclusion :

À l'issue de ce chapitre, nous avons déterminé les outils que nous souhaitons utiliser afin de résoudre les problèmes liés à notre cas d'étude. Nous mettrons donc en œuvre de l'algorithme génétique, des méthodes de sélection distribuées par AHP et commencerons la synchronisation de nos ordonnanceurs par un fonctionnement FIFO puis nous essayerons de tendre vers une technique plus aboutie. Maintenant que les choix sont faits, nous pouvons passer à l'étape suivante de notre démarche, qu'est la modélisation (**Figure 9**). Le chapitre suivant présente donc la manière dont nous avons adapté ces cadres de résolution générique à nos problématiques.

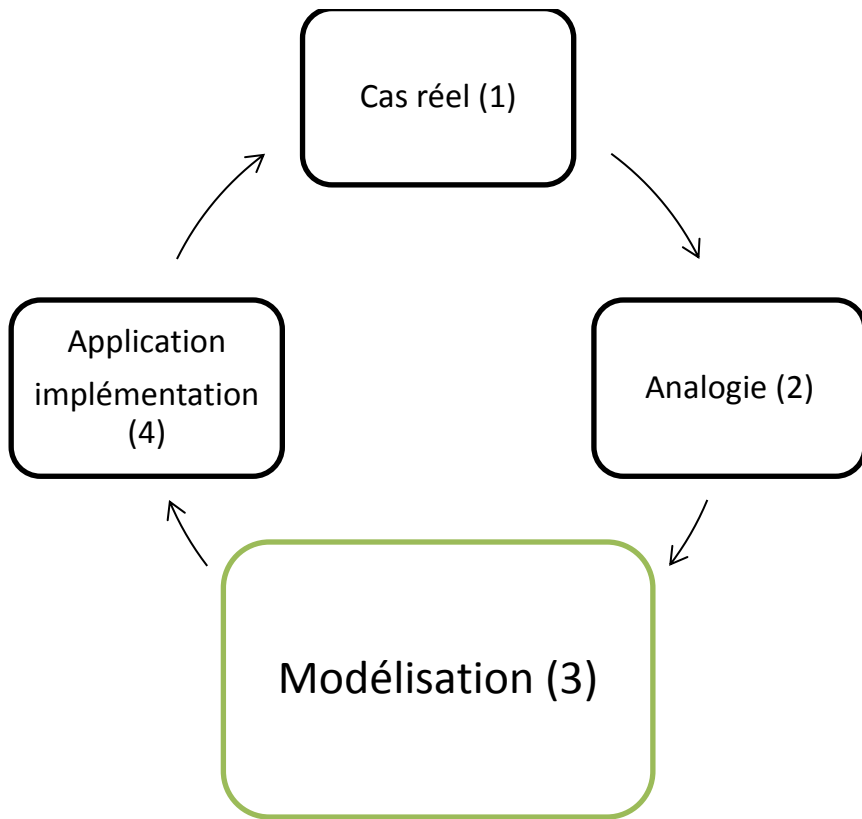


Figure 9 Modélisation

III. Proposition (Modélisation)

A. Métamodèle et architecture du système

De manière à améliorer la précision et la réactivité d'un ordonnancement, notamment en cas de fortes perturbations, le passage par un échange d'informations fréquent entre le système d'information et les produits est nécessaire afin d'avoir des données justes à chaque fois qu'une prise de décision est nécessaire. Pour ce faire, les produits doivent non seulement communiquer avec le système d'information mais aussi avec l'environnement extérieur. Ils doivent communiquer avec les différentes ressources du système (les postes de charges, les secteurs, la chaîne de production) pour avoir connaissance des niveaux d'occupations des postes suivants ou des machines en panne, ce qui permet de ré-évaluer la planification judicieusement. Afin de maximiser la réactivité, il faut permettre aux produits de prendre certaines décisions par eux-mêmes, en consensus, par exemple : choisir lequel des produits de la file d'attente d'une machine doit être priorisée, à l'instant T, connaissant son état, celui de chacun des autres candidats, l'état de la machine sur laquelle il doit être travaillé ainsi que de n'importe quel élément quantifiable, inhérent au système d'information, pouvant jouer un rôle dans la prise de décision. On pense ici, par exemple, à prendre en considération la capacité d'anticiper la production de non-qualité (THOMAS et al., 2018). Cette faculté s'avère particulièrement utile dans le cas d'une chaîne de production soumise à de fortes perturbations causées par la non-qualité. Ce point sera détaillé dans le paragraphe III.C.2.

Un tel comportement de la part des produits relève du thème des produits intelligents et soulève l'interrogation suivante : quelles sortes de données doivent être embarquées par ceux-ci, comment les conjuguer à celles détenues par le système d'information lui-même ? Afin d'améliorer les performances, il faut assurer une synchronisation parfaite entre les différents éléments d'ordonnancement du système. Les problèmes de synchronisation font l'objet du paragraphe III.D. Comme indiqué dans le chapitre précédent, pour répondre à cette problématique, nous proposons de mettre en place un système de pilotage contrôlé par le produit sur la base de l'architecture proposée par Herrera (**Figure 8**) que nous avons fait évoluer pour correspondre à notre cas d'étude (**Figure 10**). Cependant la mise en place d'une architecture telle que celle-ci devrait faciliter cette étape de synchronisation, compte tenu du fait que toutes les entités décisionnaires du système doivent être construites sur le même schéma.

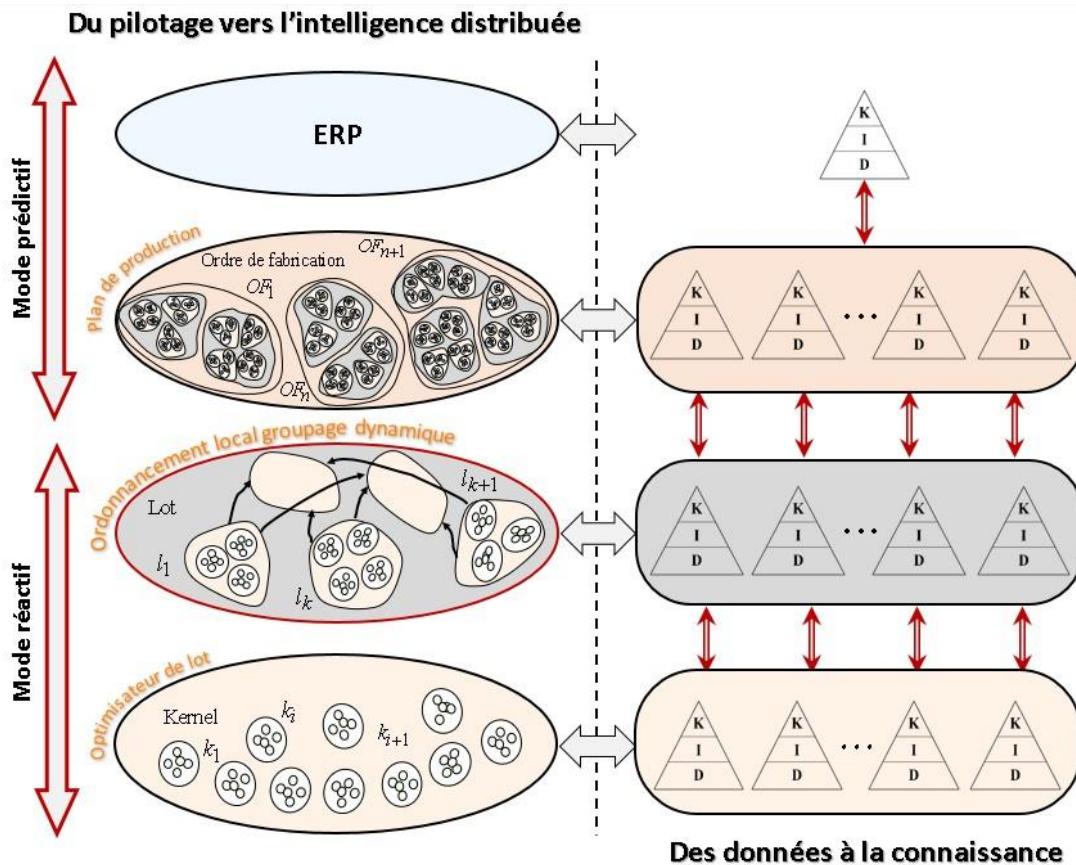


Figure 10 Méta-modèle VSM

La partie de gauche présente les différentes activités de planification avec les différents niveaux de récursivité du système. Dans les plus hauts niveaux, nous avons le plan de production et les optimisateurs centralisés qui fonctionnent de manière prédictive. Ensuite, dans les niveaux inférieurs, nous avons les optimisateurs réactifs, voir proactifs en descendant jusqu'à la prise de décision entre les produits.

La partie de droite, correspond au contrôle informatique : les informations remontant du terrain telles que les différentes décisions prises par les produits ainsi que les perturbations, ces informations sont notées suivant le modèle « Data, Information and Knowledge » introduit par (Zins 2007). Il est à noter que chaque entité de chaque niveau d'hybridation du côté planification possède son jumeau du côté informatisé.

Plus que de proposer une nouvelle architecture théorique, nous cherchons à mettre en place un système viable, répondant à une problématique réelle, en s'inspirant des systèmes contrôlés par le produit (SCP), s'appuyant sur le concept des produits intelligents capable si possible de s'adapter aux quatre phases décrites dans le chapitre précédent, à savoir : la régulation, l'adaptation fonctionnelle, l'adaptation structurelle et l'évolution structurelle.

Dans ces travaux, un méta-modèle récursif a été proposé afin d'améliorer l'efficacité de la ligne de production, réduire le délai et fournir un planning adaptatif capable de prendre en compte les réparations et les pannes machine. Les éléments du système s'inspirent du concept de cellule cybernétique défini par Beer (Stafford, 1984).

Les « cellules » (terme utilisé par Beer pour définir l'entité décisionnaire de plus bas niveau, les plus petites unités de l'architecture), doivent représenter suivant les possibilités envisageables pour la chaîne de production à piloter, les regroupements minimaux de produits, pouvant aller jusqu'au, dans le cas le plus extrême, lot unitaire. Dans l'application industrielle présentée au chapitre IV, il n'était technologiquement pas faisable de descendre à ce niveau de granularité. Le modèle proposé est prévu pour fonctionner de telle manière que le fonctionnement unitaire ne soit, finalement, qu'un cas particulier de groupement minimal. L'idée est que chaque élément du système soit pourvu « d'accréditation » spécifiques à son niveau de hiérarchie pour participer à la prise de décision, communiquer et qu'il possède une connaissance de lui-même et des autres en adéquation avec ses activités. En plus de communiquer entre eux, les produits sont en interaction avec d'une part les ressources et de l'autre les opérateurs. Pour créer une concordance entre les éléments, il faut que les autres acteurs communicants suivent la même logique que les produits. La **Figure 11** décrit cette évolution simultanée des trois entités en interaction suivant le changement d'horizon dans le niveau de décision. L'axe d'horizon désigne le niveau décisionnel sur lequel s'applique le niveau d'agrégation. Nous avons trois niveaux : opérationnel, tactique et stratégique. Le trinôme en interaction (ressources, produits et humain) évolue suivant cet horizon, en représentant l'évolution du niveau d'agrégation.

Nous proposons dans le modèle, de combiner :

- Des techniques de groupement (clustering) ou de détermination de quantité économique (lot-sizing) servant à agréger des cellules (afin de faire naître des entités plus cohérentes vis-à-vis de certains postes de charges ou groupe de postes pour faciliter les calculs et adapter la prise de décision).
- De l'ordonnancement dynamique permettant de remplir les objectifs des différents postes de charges et optimiser le flux de production.

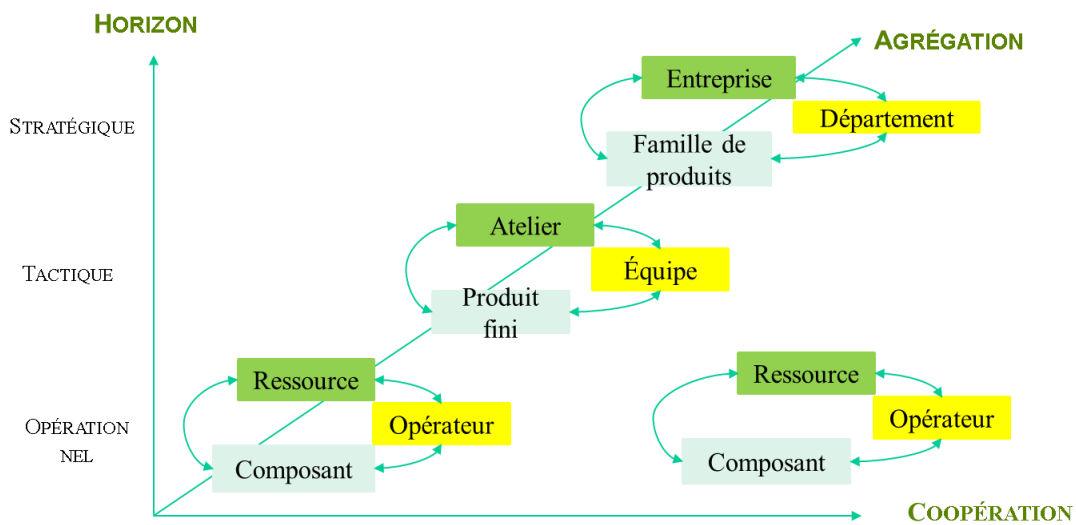


Figure 11 Évolution des entités collaborantes

Sachant que lorsque la chaîne de production est perturbée, elle est en constante évolution, il est nécessaire que les ordonnancements soient réactifs et se recalculent aussi souvent que nécessaire afin de rester pertinent et utilisable en permanence.

Notre objectif est d'une part de réaliser un ordonnancement global simple (très peu de contraintes avec une périodicité supérieure à la journée) qui servira de cadre et de base aux optimisateurs locaux. Puis d'autre part, de construire les optimisateurs locaux pour les deux postes goulot cités en début de chapitre (usinage et laquage) et de les relier par un simple FIFO. Les paragraphes suivants, en décrivent la mise en œuvre, en commençant par l'ordonnanceur global.

Pour mettre en œuvre notre proposition nous avons choisi des pilotes sur lesquels démarrer notre implémentation et chacun sera décrit par la suite. Une analyse du flux « Acta » faite sur le logiciel Arena, a mis en évidence que les postes les plus critiques sont l'usinage et le laquage. Suivant les circonstances, l'un ou l'autre de ces postes est susceptible de devenir goulot d'étranglement. De plus, nous avons constaté que les objectifs des autres secteurs étaient atteignables en ne réalisant que des ordonnanceurs pour les deux postes cités précédemment (**Figure 12**). Ainsi, nous appliquerons un premier ordonnanceur, à base d'algorithme génétique, sur le secteur usinage et le séquençement généré sera considéré valable jusqu'à l'étape de tri ou les lots de fabrication seront alors divisés. Ensuite, nous mettrons en place un ordonnanceur proactif, sur le laquage, qui modifiera le flux initial, poussé, si un risque de produire de la non-qualité apparaît. Cette décision se prendra, en mettant en œuvre un processus AHP afin de déterminer suivant une liste de critères quel est le lot le plus opportun à produire. Nous avons choisi de ne pas travailler sur le flux béton pour le moment, c'est pourquoi, il a été retiré du schéma.

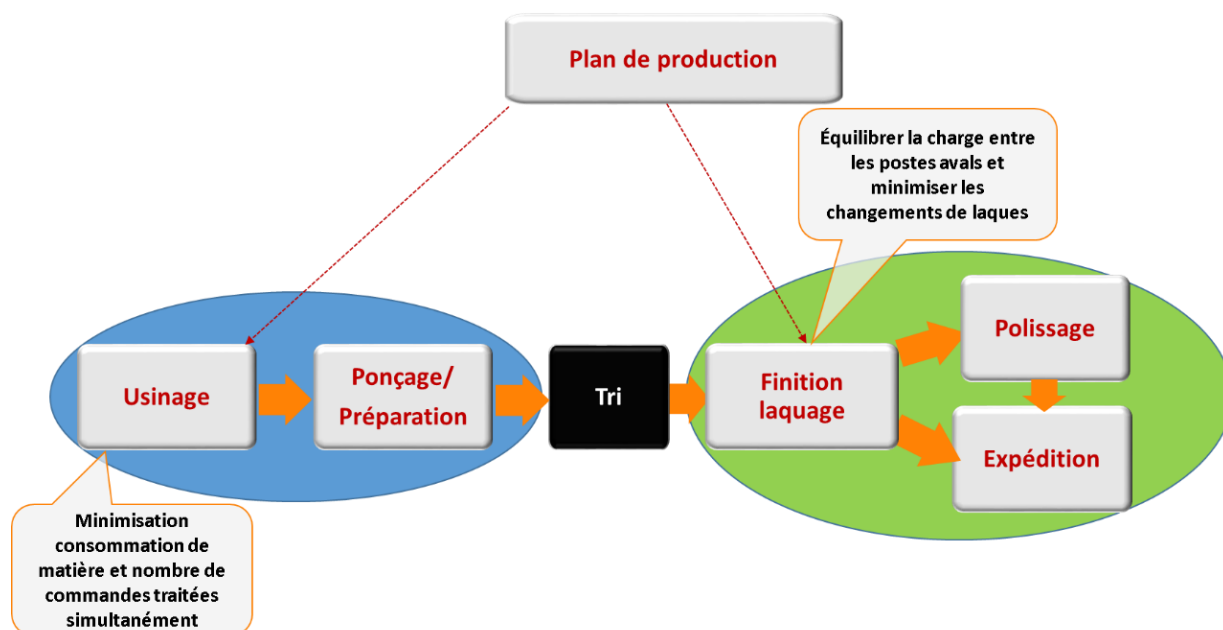


Figure 12 Postes étudiés

L'implémentation s'est faite en plusieurs temps. En effet, il était inenvisageable de tester directement des prototypes dans l'entreprise pour différentes raisons :

- Parce qu'elle ne possède pas encore les ressources technologiques pour mettre en œuvre certaines sous-parties de la proposition.

- Pour ne pas interférer avec la production et le taux de service que l'entreprise garanti à ses clients.

Ainsi, nous avons commencé par une phase de simulation et/ou transposition sur plateforme de test puis nous avons ensuite chercher à les implémenter in vivo.

B. Ordonnancement global

Lorsque l'on étudie le cas d'une chaîne de production ayant des objectifs différents et même parfois contradictoires entre les différents postes de travail, la réalisation d'un ordonnancement global est très complexe à mettre en place à cause de la diversité des objectifs d'optimisation. Cela est encore plus vrai lorsque celle-ci est soumise à d'importantes perturbations comme celles conduisant certains produits à être complètement refabriqués, faisant qu'un planning prédictif n'est juste qu'à l'instant T où il est généré. Nous nous proposons donc de réaliser un ordonnancement très simplifié à considérer comme un guide fournissant des repères :

- Jalons de l'avancée de la production.
- Les données d'entrées pour des ordonnanceurs locaux.
- Les dates limites d'arrivée des produits en fin de chaîne.

Sa réalisation se limite à rechercher les postes les plus représentatifs et produire un ordonnancement au plus tard n'utilisant que ces postes. Une étude des flux et de la répartition de charge peut permettre d'identifier les postes critiques (goulot, poste fournisseur à clients multiples, ...). En se focalisant sur ceux-ci, un planning macroscopique simple peut être réalisé en faisant un ordonnancement au plus tard qui ne considère que ces postes clés et affecte des temps de production moyennés allant du poste concerné jusqu'au suivant auquel il est préférable d'ajouter une marge de sécurité afin de déterminer une plage acceptable d'arrivée pour chaque produit. Le plus grand enjeu est d'établir une plage correcte afin de contraindre les optimisateurs spécifiques (locaux) et de créer une synchronisation entre eux et cette esquisse. Nous allons donc, à présent, passer à la réalisation des optimisateurs locaux.

C. Optimisateurs locaux

Chaque optimisateur local doit être considéré comme un problème entier d'ordonnancement à résoudre. Une partie de la réflexion porte sur la méthode de résolution à employer pour trouver une solution avantageuse dans un délai acceptable ainsi qu'à déterminer quel acteur du système est le plus apte à être le décisionnaire (en d'autres termes à déterminer à quel niveau de la hiérarchie systémique la prise de décision doit être effectuée). Il convient d'établir également quel type d'ordonnancement il est envisageable de réaliser (réactif, prédictif ou proactif). Cependant, il aurait également pu être envisageable de proposer une solution générique prenant en entrée les variables et les contraintes mais il ne s'agit pas là d'une des finalités de cette thèse. Suivant les contraintes et les fonctions objectifs à réaliser, certains optimisateurs sont plus aptes à être implémentés que d'autres, en effet si l'on cherche, par exemple, à faire intervenir les caractéristiques de chaque produit dans la prise de décision, il est plus intéressant d'élaborer une décision distribuée pro-active. Dans cette étude, nous nous sommes focalisés sur l'étude d'un optimisateur réactif et d'un optimisateur proactif qui répondent le mieux à nos besoins.

1. Optimisateur réactif

Cet optimisateur a été développé dans l'objectif de minimiser la consommation matière et les temps de réglages d'un poste d'usinage devant travailler avec plusieurs spécificités de matières premières (dimensionnel : épaisseur d'un panneau de MDF, d'une tôle d'acier, la matière : acier inoxydable, panneau de particule, ...) et dont chaque changement nécessite un réglage prenant un temps non négligeable. L'objectif est de maximiser le regroupement de produits semblables tout en conservant la proximité des produits appartenant à une même commande client. Cet optimisateur a aussi été élaboré dans l'idée de pouvoir facilement re-diviser les groupements effectués au poste en aval pour les réassembler différemment.

a) *Modélisation mathématique*

De la description de ce problème, nous pouvons considérer le problème d'optimisation de l'atelier usinage comme la combinaison de deux classes de problème connus et longuement étudiés séparément (**Figure 13**) :

- Le « lot-sizing » : réaliser des lots de fabrication (que nous noterons MB pour Manufacturing Batches) pour réduire les coûts de production tout en évitant de produire des lots trop denses qui augmenteraient le nombre de commandes client traités simultanément et qui pourraient ajouter également des tâches à non-valeur ajoutée.

- Le « scheduling » : proposer un planning des MBs afin d'optimiser l'étape de dégroupage/regroupage. Nous imaginons une étape de découplage après laquelle, les lots de fabrication n'auront plus d'utilité. À ce moment, ceux-ci seront décomposés pour être à nouveau regroupés suivant de nouvelles caractéristiques optimisant les postes aval.

Cependant l'intersection de ces deux catégories a fait l'objet de beaucoup moins de travaux.

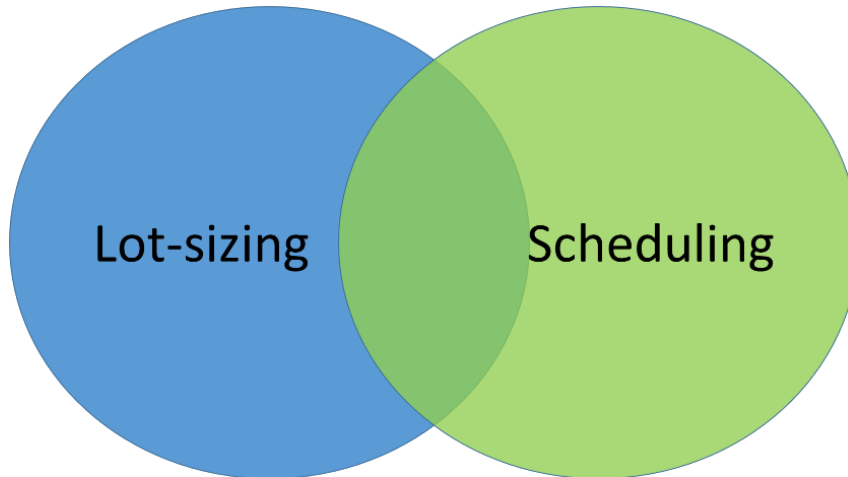


Figure 13 Intersection de deux classes de problème

Ce problème appartient à la catégorie de l'ordonnancement de groupe de produits sur un atelier type Flow shop avec dépendance de la séquence. Nous allons commencer par présenter un modèle mathématique représentant le cas général de cette catégorie de problème d'ordonnancement.

Nous avons un atelier flow shop de M machines ($m=1, 2, \dots, M$) doit produire G groupes de tâches ($g= 1, 2, \dots, G$). Chaque groupe g contient N_g tâches et P_{jgm} est le temps de process de la tâche j du groupe g sur la machine m . Par conséquent, le nombre total de tâches à réaliser est égal à $N = \sum_{g=1}^G n_g$.

Les temps de réglages sont dépendants de la séquence, c'est-à-dire qu'un temps de réglage important est nécessaire sur chaque machine dès lors qu'un changement de groupe a lieu. S_{mfg} est le temps de réglage requis sur la machine m lorsque le groupe g passe après le groupe f . Les temps de réglage entre les tâches d'un même groupe sont négligeables.

Dès que la première tâche du groupe g est entamée sur la machine m , l'ensemble du groupe est produit sans interruption puis la production passe au groupe suivant. Les réglages peuvent être fait par anticipation. Il n'y a pas de relation de précédence entre les groupes ou entre les tâches d'un groupe. La préemption n'est pas autorisée : lorsqu'une tâche débute sur une machine m , celle-ci ne peut passer à la machine $m+1$ que lorsqu'elle a fini d'être réalisée sur m . Toutes les tâches sont prêtes à être réalisées au début de la planification. Toutes les machines sont disponibles 100% du temps.

Cependant, certaines simplifications peuvent être prises par rapport au modèle précédent afin d'établir le modèle mathématique correspondant à notre problème d'ordonnancement. Le fait que le séquençement fourni à la première machine contraint complètement ceux des postes suivants (fonctionnant en FIFO), nous permet de simplifier le problème à l'étude d'une unique machine. Or (Giglio, 2015) propose un modèle pour ce type de problème, sur lequel nous allons poser notre cas d'étude.

Considérons les commandes C_i avec i le numéro de la commande. Celles-ci sont divisées en lots élémentaires de telle sorte que ' K_{ij} ' soit le lot élémentaire j de la commande i . Nous chercherons à regrouper les lots élémentaires possédant les mêmes configurations (ici épaisseur pour cause de simplification) dans des lots de fabrication (batches notés l). Schématiquement représenté sur la **Figure 14**.

Commandes (C_i) \longrightarrow Lots ($K_{i,j}$) \longrightarrow Batches (l)

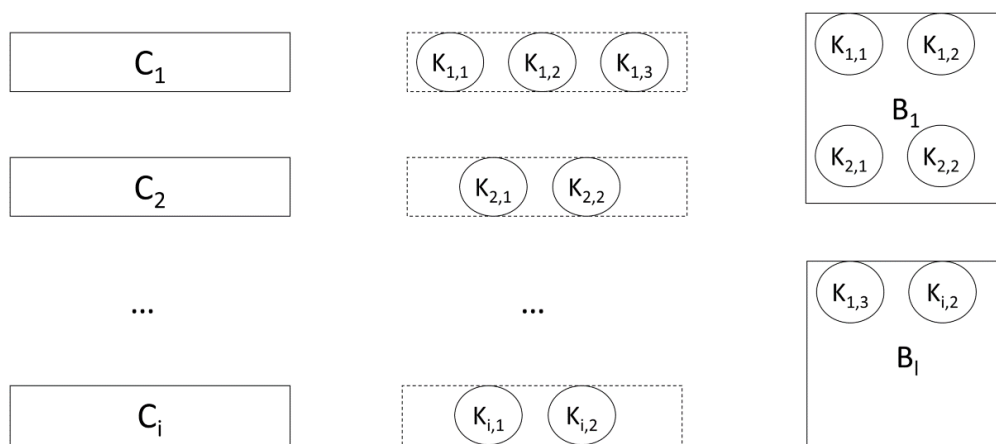


Figure 14 Répartition commandes-lots-batches

Le système est cadré par les contraintes suivantes : nous limiterons la surface totale d'un batch (égale à la somme des surfaces de l'ensemble des lots que contient le batch) à une surface maximale notée S_{max} . De plus, nous limiterons le nombre de commandes desquelles proviennent les lots d'un batch à un maximum noté N . Chaque commande est affectée d'un délai de livraison qu'elle ne doit pas dépasser. Nous rappelons également qu'un batch ne peut être constitué que de lots de même configuration. L'optimisation que nous cherchons à atteindre est la suivante :

- Maximiser le nombre de lots au sein d'un batch.
- Minimiser le nombre de setup.
- Minimiser l'écart de position maximal entre les batches issus des mêmes commandes (distance maximale séparant deux lots $K_{i,j}$ et $K_{i,j'}$ d'une commande i dans le séquençement).

Nous supposerons pour cette modélisation que les temps de setup sont indépendants de la séquence. Comme dit précédemment, notre modèle est donc construit comme le mélange d'un modèle de lot-sizing et d'un modèle de scheduling et nous allons répartir les variables de décision suivant le modèle duquel elles sont extraites.

Les paramètres du modèle :

- C : nombre de commandes
- nk_i : nombre de lots dans la commande i
- d_i : due date de la commande i
- e_{ij} : épaisseur du lot K_{ij}
- $S_{i,j}$: surface en m^2 du lot K_{ij}
- tt_{ij} : durée de traitement du lot K_{ij}
- $St_{uv} = St$: durée de setup
- S_{max} : surface maximal d'un batch
- M : nombre total de pièces
- N : nombre de commandes dans un batch

Les variables de décision de notre modèle :

Lot-sizing	
b_l	1 si le batch l est actif 0 sinon
X_{ijl}	1 si le lot $K_{ij} \in$ au batch l 0 sinon
Z_{il}	1 si le batch l contient des lots de la commande i 0 sinon
n_l	nombre de lots dans le batch l
L	nombre de batches actifs
e_l	épaisseur du batch l

Tableau 2 Variables de décision lot-sizing

Scheduling	
$\delta_{ll'}$	1 si le batch l' est le suivant direct du batch l 0 sinon
$Y_{ll'}$	1 si l' se trouve après l dans le séquençement 0 sinon
$Sup_{ll'}$	1 si la séquence $l-l'$ induit un setup 0 sinon
$tdpb_i$	date de début du premier batch de la commande C_i
$tddb_i$	date de début du dernier batch de la commande C_i
$tfpb_i$	date de fin du premier batch de la commande C_i
$tfdb_i$	date de fin du dernier batch de la commande C_i
$W_i = tfdb_i - tdpb_i$	durée de traitement de la commande i
t_l	date de début de traitement du batch l
tb_l	durée de traitement du batch l

Tableau 3 Variables de décision scheduling

Nous chercherons à minimiser trois fonctions :

- Le nombre de batches instanciés : $\sum_{l=1}^L b_l$
- Le nombre de setup nécessaires pour le séquençement de la production :

$$\sum_{l=1}^L \sum_{l'=1}^L Sup_{ll'}$$
- La distance (ici, il s'agit d'un temps) entre les lots originaires d'une même commande :

$$\sum_{i=1}^C W_i$$

Ainsi, la fonction « objectif » globale, normée afin d'avoir toutes les sous-fonctions comprises dans le même intervalle, que nous cherchons à minimiser s'écrit :

$$\min z = \frac{\sum_{l=1}^L b_l}{\sum_{i=1}^C n_{k_i}} + \frac{\sum_{l=1}^L \sum_{l'=1}^L Sup_{ll'}}{L} + \frac{\sum_{i=1}^C W_i}{(L-1) \sum_{i=1}^C n_{k_i}} \quad (3)$$

Le calcul de l'optimisation doit s'effectuer sous les contraintes suivantes :

Lot-sizing	
$\forall l, \quad b_l \leq b_{l-1} \quad (4)$	Suppression des solutions symétriques
$\forall l, \quad \sum_{i=1}^c \sum_{j=1}^{nk_i} X_{ijl} \leq Mb_l \quad (5)$	Identification des batches instanciés
$\forall (i, j), \quad \sum_{l=1}^M X_{ijl} = 1 \quad (6)$	Assurer que chaque lot est affecté une fois et une seule
$\forall (i, j, i', j', l) \text{ avec } (i, j) \neq (i', j'),$ $X_{ijl}e_{ij} \leq M(1 - X_{i'j'l}) + X_{i'j'l}e_{i'j'} \quad (7)$	Tous les lots au sein d'un même batch ont la même épaisseur
$\forall (i, j, l),$ $e_l \geq X_{ijl}e_{ij} \quad (8)$	
$\forall l, \quad \sum_{i=1}^c \sum_{j=1}^{nk_i} X_{ijl}S_{ij} \leq Smax \quad (9)$	Limiter la capacité maximale (en termes de surface, pour le cas présent) d'un batch
$\forall (i, l), \quad \sum_{j=1}^{nk_i} X_{ijl} \leq MZ_{il} \quad (10)$	L'affectation de la variable Z_{il} prenant la valeur 1 si le batch « l » contient un lot de la commande C_i
$\forall (i, l), \quad \sum_{j=1}^{nk_i} X_{ijl} \geq Z_{il} \quad (11)$	
$\forall l, \quad \sum_{i=1}^c Z_{il} \leq Nb_l \quad (12)$	Respecter la contrainte limitant le nombre maximal de commandes dans le batch l à la valeur N

Tableau 4 Contraintes du lot-sizing

Il est notable que les contraintes (4) à (8) sont représentatives d'un problème de type Sac à dos.

Scheduling	
$\forall l', \sum_{l=1}^M \delta_{ll'} \leq b_{l'} \quad (13)$	Déterminent le séquençement en suivant une logique type voyageur de commerces
$\forall l, \sum_{l'=1}^{\infty} \delta_{ll'} \leq b_l \quad (14)$	
$\forall l, \delta_{ll} \leq 0 \quad (15)$	
$\forall (l, l') \text{ avec } l \neq l',$ $Y_{ll'} \geq 1 - M(1 - \delta_{ll'}) \quad (16)$	Assurent la correspondance entre les $Y_{ll'}$ et les $\delta_{ll'}$ (17) et (18) assurent la transitivité de la relation de précédence
$\forall (l, l', \beta) \text{ avec } (l \neq l' \neq \beta),$ $(Y_{ll'} + Y_{l'\beta}) \geq Y_{l\beta} \quad (17)$	
$\forall (l, l', \beta) \text{ avec } (l \neq l' \neq \beta),$ $(Y_{ll'} + Y_{l'\beta}) \leq Y_{l\beta} + 1 \quad (18)$	
$\forall (l, l') \text{ avec } (l \neq l'),$ $Y_{ll'} + Y_{l'l} = 1 \quad (19)$	
$\forall (l, l', \beta) \text{ avec } (l \neq l' \neq \beta),$ $Y_{ll'} + Y_{l'\beta} + Y_{\beta l} \leq 2 \quad (20)$	Suppriment la possibilité de créer des boucles dans le séquençement
$\forall (l, l', \beta) \text{ avec } (l \neq l' \neq \beta),$ $\delta_{ll'} + \delta_{l'\beta} + \delta_{\beta l} \leq 2 \quad (21)$	
$\sum_{l=1}^M b_l = L \quad (22)$	Évalue le nombre de batches actifs

Tableau 5 Contraintes liées au scheduling

Union lot-sizing et scheduling	
$\forall l, \quad tb_l = \sum_{i=1}^c \sum_{j=1}^{nk_i} X_{ijl} tt_{ij} \quad (23)$	Effectue le calcul de la durée de traitement d'un batch
$\forall (l, l') \text{ avec } (l \neq l'),$ $t_{l'} \geq t_l + Sup_{ll'} St + tb_l - M(1 - Y_{ll'}) \quad (24)$	Garantissent que deux batches ne soient pas produits simultanément
$\forall (l, l') \text{ avec } (l \neq l'),$ $t_l \geq t_{l'} + Sup_{ll'} St - MY_{ll'} \quad (25)$	

Tableau 6 Contraintes relatives à l'union des deux problèmes

Minimiser l'écartement entre lots d'une même commande	
$\forall (i, l), \quad tddb_i \geq t_l - M(1 - Z_{il}) \quad (26)$	Définir les dates de début et de fin de traitement de la commande Ci
$\forall (i, l), \quad tdpb_i \leq t_l + M(1 - Z_{il}) \quad (27)$	
$\forall (i, l),$ $tfdb_i \geq tddb_i + tb_l - M(1 - Z_{il}) \quad (28)$	
$\forall (i, l),$ $tfpb_i \leq tdpb_i + tb_l + M(1 - Z_{il}) \quad (29)$	
$\forall i, \quad tfdb_i \leq d_i \quad (30)$	Empêche la date de fin de traitement de dépasser la date de livraison
$\forall i, \quad W_i = tfdb_i - tdpb_i \quad (31)$	Calcule la durée effective de traitement d'une commande Ci

Tableau 7 Contraintes pour minimiser l'écartement

Setup	
$\forall (l, l') \text{ avec } l \neq l',$ $MSup_{ll'} \geq -M(1 - \delta_{ll'}) + e_l - e_{l'} \quad (32)$	Contraintes à associer conjointement à la fonction « objectif » visant à minimiser $Sup_{ll'}$. Elles permettent de fixer la valeur de $Sup_{ll'}$ à 0 ou 1 suivant le fait que e_l et $e_{l'}$ soient égaux ou non.
$\forall (l, l') \text{ avec } l \neq l',$ $MSup_{ll'} \geq -M(1 - \delta_{ll'}) - e_l + e_{l'} \quad (33)$	

Tableau 8 Contraintes de setup

Ensemble de définition	
$\forall(i, j, l, l'),$ $(b_l, X_{ijl}, \delta_{ll'}, Y_{ll'}, Z_{il}, Sup_{ll'}) \in \{0,1\}$ (34)	Booléens
$\forall(l, i),$ $(n_l, L, dd_i, N, C) \in \mathbb{Z}^+$ (35)	Entiers naturels positifs
$\forall(l, i, j),$ $(e_l, t_l, tb_l, tddb_i, tdpb_i, tfdb_i, tfpb_i, e_{ij}) \in \mathbb{R}^+$ (36)	Réels positifs

Tableau 9 Contraintes ensemblistes

a) *Analyse de la complexité*

Une fois le modèle formel de ce problème posé, il convient d'en analyser la complexité. Pour cela, nous avons décomposé le problème en cinq segments :

- Le problème de batching : Les équations (4) à (12) sont celles relatives à la distribution des lots au sein des batches en assurant les contraintes de configuration identique pour les lots dans un même batch et les limitations sur la surface maximale de batch ainsi que sur la diversité maximale autorisée des commandes au sein d'un batch. Comme dit précédemment, ce problème est identifiable à un problème de type « Sac à dos ». En s'appuyant sur les travaux de (Lagoudakis, 1996), nous savons ce problème comme NP-complet et de complexité $O(2^{poly(n)})$ auquel nous ajoutons une complexité de $O(n^5)$ pour la contrainte (7).
- Pour le problème d'ordonnancement, étant donné que les contraintes (13) à (22) dérivent de la modélisation de (Giglio, 2015) où l'on se rapproche d'un problème de voyageur de commerces sans boucles fermées et où tous les points ont une entrée et une sortie. En effet, à part les premiers et derniers batches, tous ont un prédécesseur et un successeur. Or (Papadimidriou, 1977) a démontré que même lorsque les distances sont euclidiennes, le problème reste NP-complet. Ainsi la complexité de ce problème est en $O(n!)$.
- Pour que les problèmes de groupement et de séquençement soient traités de manière égale, il fallait qu'ils soient rassemblés sous une même logique, c'est ainsi fait via les équations (23) à (25) qui lient les variables X_{ijl} et $Y_{ll'}$. Cette combinaison est de complexité $O(n^3)$.

- L'objectif de minimisation de l'espacement entre les lots d'une même commande, représenté par la variable W_i , était nécessaire à considérer pour le problème car pour les postes qui seront en aval de celui étudié, la consigne est d'attendre que l'ensemble des lots formant une commande soient présents pour être traités. De ce fait, lorsqu'un lot de la commande C_i contenu dans le batch « I » est traité, il lui faut patienter dans la file d'attente du poste aval jusqu'à ce que les autres lots de C_i soient traités. Les équations (26) à (31) assurent la prise en compte de cette caractéristique et la complexité correspondante est, elle aussi, de l'ordre de $O(n^3)$.
- La prise en compte du setup, modélisée par (32) et (33), ne doit se produire que quand deux batches consécutifs sont de configurations différentes. Complexité d'ordre $O(n^3)$.

Ces segments, mis bout à bout, nous donnent ainsi un problème de complexité $O(n!)$

$$O(n!) + O(2^{poly(n)}) + O(n^5) + 3 * O(n^3) \approx O(n!) \quad (37)$$

La complexité algorithmique du problème a pu être identifiée. Cette considération seule, ne suffit pas lorsque l'on cherche à résoudre un problème à l'aide d'un solveur. En effet, il faut s'attarder sur le nombre de variables et de contraintes introduites par le système lors de l'utilisation d'un solveur dont l'un des meilleurs représentants est GLPK (GNU Linear Programming Kit) dont les capacités ne sont pas infinies. À cela, il faut bien évidemment ajouter la complexité du système à résoudre et la capacité mémoire des calculateurs. Nous avons établi deux fonctions afin d'approximer le nombre de variables et de contraintes intervenant dans notre cas :

$$Nx(i, j, l) = 3l^2 + ij l + il + 5i + 4l \quad (38)$$

$$Nc(i, j, k, l) = k^2 l + 3l^3 + 6kl + i^2 j + 13l^2 + il + k + 3l \quad (39)$$

Avec Nx le nombre de variables, Nc le nombre de contraintes, i le nombre de commandes, j le nombre maximal de lot dans une commande, k le nombre de lots total du système et l le nombre de batches maximal souhaité.

Ainsi, si on limite l'étude à l'optimisation pour la production des commandes d'une seule semaine pour un seul client, nous obtenons 45 commandes, contenant au plus 5 lots par commande donnant un total de 80 lots, et souhaitant limiter le nombre de batches à 15, nous arrivons à $Nx = 5010$ variables et $Nc = 127175$ contraintes.

Fort de cette analyse, il est évident que sans relaxation d'un certain nombre de contraintes, il est impensable d'obtenir des solutions intéressantes en des temps de calcul ne devant pas excéder plus d'une dizaine de minutes. Et même, en relaxant les contraintes, les chances d'atteindre nos objectifs semblent minces. Cela nous a conforté dans l'idée de passer par des techniques évolutionnistes pour résoudre ce type de problème.

b) Simplification du modèle

Ainsi, nous proposons ci-dessous une modélisation de notre problème sous une forme plus semblable à des problèmes d'ordonnancement multicritères de groupes de produits que nous résoudrons avec un algorithme génétique.

Soient les données suivantes :

- N : Nombre de contremarques (commandes client)
- M : Nombre de lots de commandes client (FPRB)
- F : Nombre de lots de fabrication (batches) (MB)
- P_m : Nombre de MB dans le FPRB m

La première fonction à optimiser, F1, correspond à la minimisation de la distance séparant les MB appartenant au même FPRB nommé $l_{e,m}$ où e est une caractéristique matière, par exemple l'épaisseur du lot, et m le numéro de la commande client :

$$F1 = \min_i \left(\sum_{j=1}^F a_{i,j} \max(x_j - x_i - 1, 0) \right) \quad (40)$$

Tel que :

$a_{i,j}$ exprime le fait que les deux MB $i = l_{e,m}$ et $j = l'_{e',m'}$ appartiennent tous deux au même FPRB avec :

- $a_{i,j} = 1$ si $m = m'$; 0 sinon (41)
- x_i est la position du MB i dans la séquence.

La seconde fonction F2 représente la minimisation du nombre de setup pour le séquençement considéré :

$$F2 = \min_i \sum_{i=1}^F \sum_{j=1}^F y_{i,j} \quad (42)$$

Où :

- $y_{i,j} = 1$ si $e \neq e'$ et j est placé juste après i , 0 sinon (43)

En conséquence, la fonction globale à minimiser s'exprime de la manière suivante :

$$F = \alpha F1 + \beta F2 \quad (44)$$

Avec α et β , des ratios combinatoires, déterminés par réglages suite à plusieurs expérimentations jusqu'à obtenir un résultat satisfaisant, leurs valeurs respectives étant de 2/3, 1/3.

L'optimisation se fait suivant les contraintes suivantes :

- $F1 \geq 0$ (45)
- $F2 \geq M$ (46)
- $\forall i, x_i \in (0, F)$ (47)
- $\forall (i, j), y_{i,j} \in \llbracket 0,1 \rrbracket$ (48)
- $\forall i, \sum a_{i,j} = P_k$ où k est le numéro de FPRB du MB i (49)
- $\forall j, \sum a_{i,j} = P_k$ où k est le numéro de FPRB du MB j (50)

Comme précisé dans le paragraphe **III.C.1.a)**, le modèle est à l'intersection de deux classes de problèmes. Ainsi donc, dans un premier temps, nous allons nous concentrer sur la partie clustering. Et après, nous traiterons de la partie ordonnancement de notre problème.

(1) Proposition d'une heuristique pour le Clustering

L'un des objectifs étant de s'assurer que tous les produits d'une même commande client soient présents simultanément au poste aval, le groupement s'est fait en deux phases :

- Dans la première phase, les commandes client possédant le plus de caractéristiques matières communes sont groupés en respectant la contrainte consistant à limiter le nombre de commandes client différentes dans un même groupe à cinq. Ces groupements sont nommés lots de commandes client, notés FPRBs (pour Finished Product Reference Batches)

- Dans la seconde phase, les lots élémentaires formant les FPRBs, ayant la même caractéristique matière sont regroupés dans des lots de fabrication, notés MBs, et représenté par la lettre du lot de commande client FRPB concaténé à un chiffre désignant sa position sur l'ensemble des lots de fabrication contenus dans le lot de commande client (par exemple : si on considère le lot de commande client A constitué de 4 lots de fabrication, le premier lot de fabrication à être généré sera le lot A1/4). La **Figure 15** ci-dessous illustre ce processus de création.

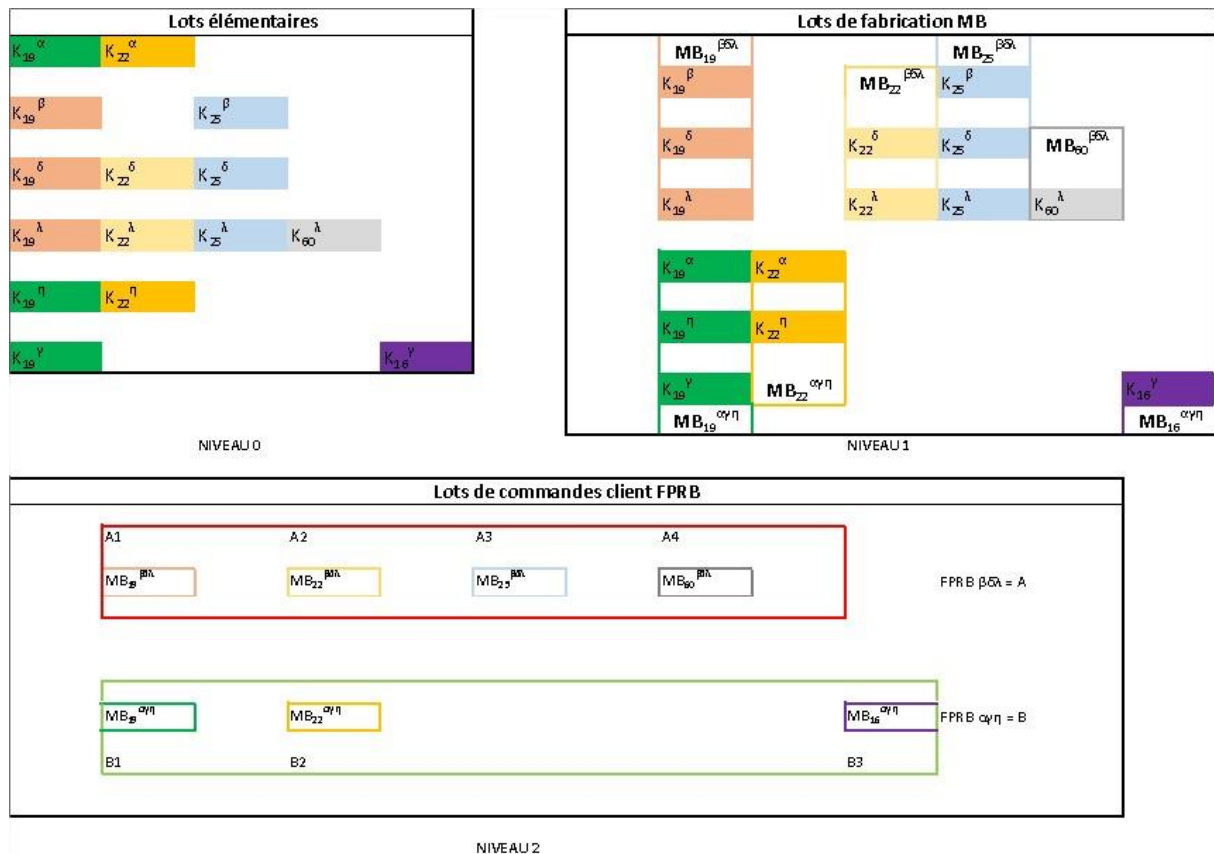


Figure 15 Le concept du clustering sur deux niveaux

Ce groupement à 2 niveaux limite le nombre de commandes client traitées simultanément ce qui en simplifie le traitement.

- (2) Utilisation de la méta-heuristique « algorithme génétique » pour l'ordonnancement

Une fois la phase de clustering achevée, il reste un nombre de F lots de fabrication à ordonner. Ceci divise, a minima, par deux le nombre d'entrées pour l'ordonnement. La technique de résolution du problème d'ordonnement sélectionnée est l'utilisation d'algorithme génétique à cause de leur capacité à fournir des résultats satisfaisant en des temps de calcul limités (Allahverdi et al., 2008). Ce choix a été fait car dans le cas d'un flux fortement perturbé par les boucles de réparations susceptible de survenir à n'importe quel moment, il est bien plus utile d'avoir une solution réactive avec un faible temps de calcul plutôt qu'optimale. En outre la probabilité de pouvoir appliquer cette planification sans aucune dérive est faible et une étape suivante de l'étude serait de faire recalculer l'ordonneur à chaque fois qu'une perturbation survient. L'éventualité de pouvoir se référer à un résultat intermédiaire non sélectionné de l'algorithme génétique lorsque des réparations s'ajoutent a été proposée. Cela semble une piste prometteuse à développer pour de futurs travaux. Cela permettrait d'évaluer la robustesse de la solution dans une situation dynamique perturbée.

L'algorithme génétique appliqué est schématisé sur la **Figure 16**. Les étapes de l'algorithme sont les suivantes :

- Initialisation de la population.
- Sélection des meilleurs chromosomes par évaluation vis-à-vis de la fonction objectif (44).
- Conservation des 25% de la population les plus favorables.
- Création de la nouvelle génération par processus de croisement et mutation à partir des chromosomes conservés.
- Création de nouvelles générations jusqu'à atteinte du critère d'arrêt (10 générations successives sans amélioration de la fonction d'optimisation).
- Renvoi du meilleur chromosome de la dernière génération produite.

Les paramètres de l'algorithme sont les suivants :

- La population initiale est constituée de 1000 individus de lots ordonnés aléatoirement. Un individu(chromosome) correspond à un séquençement dans lequel chaque lot de fabrication (gène) apparaît une fois et une seule, voir **Figure 17**.

- Le mode de sélection choisi est une sélection naturelle avec loi élitiste (choix fait pour la raison évoquée précédemment : dans notre contexte rapidité prévaut sur optimalité et l'élitisme est connue pour être la plus rapide à converger vers une solution, même si, la solution obtenue n'est pas optimale (Thierens and Goldberg, 1994)).
- Le taux de mutation est de 5%.
- Le taux de croisement est de 70%.

Le processus de croisement, semblable à celui présenté dans (Armentano and Mazzini, 2000), fonctionne par itération de la manière suivante :

En suivant la règle définie sur la **Figure 18** une nouvelle population est générée en croisant les gènes des chromosomes conservés de la génération actuelle :

À la première itération, les deux meilleurs chromosomes vis à vis de la fonction d'optimisation sont choisis pour le croisement. Ensuite, à la seconde itération, on compare deux à deux les trois meilleurs chromosomes. Et les itérations se poursuivent jusqu'à avoir atteint le taux de croisement choisi initialement puis on procède à l'étape de mutation jusqu'à atteindre à nouveau une population de 1000 individus.

La mutation est définie ici comme un échange de position réalisé entre deux gènes d'un individu. Deux mutations sont appliquées :

- Une proche où deux voisins choisis aléatoirement sont échangés.
- Une éloignée où les positions sont choisies aléatoirement.

La répartition entre ces deux mutations est de 60-40.

Les taux de mutation et de croisement ont été fixés à ces valeurs suite à des campagnes de tests jusqu'à atteindre un compromis acceptable entre efficacité et rapidité.

Le processus est illustré par la **Figure 19**.

Dans une planification où on ne considère pas plusieurs passages sur la même machine, il est incohérent de retrouver le même produit à plusieurs emplacements. C'est pourquoi aussi bien dans le processus de croisement que de mutation, une sécurité a été ajoutée, cela correspond au phénomène d'avortement assurant qu'un lot ne se trouve pas à plusieurs positions dans le séquençement. Lorsqu'un avortement se produit, le chromosome concerné est alors détruit et un nouveau est proposé à sa place.

L'algorithme stoppe dès le moment où le résultat de la fonction d'optimisation ne progresse plus après 10 générations successives.

Les ratios combinatoires de la fonction d'optimisation (44) ont été déterminés grâce à de multiples analyses et expérimentations. Les séquençements testés ont montré que pour assurer une progression fluide durant toute la phase précédant le tri, la fonction F1 doit être celle à privilégier. Les valeurs attribuées à α et β correspondent à un ratio proche des 2/3, 1/3.

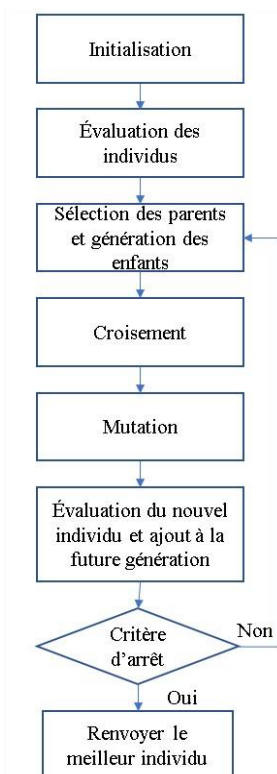


Figure 16 Diagramme d'algorithme génétique

Population initiale : 1000 individus construits aléatoirement
 1 individu = 1 chromosome = une séquence de gènes
 1 gène = 1 lot se trouvant à une position précise dans la séquence

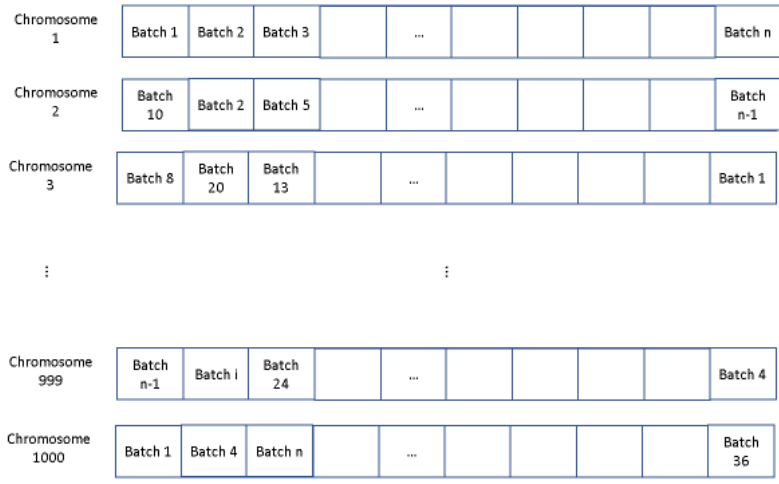


Figure 17 Population

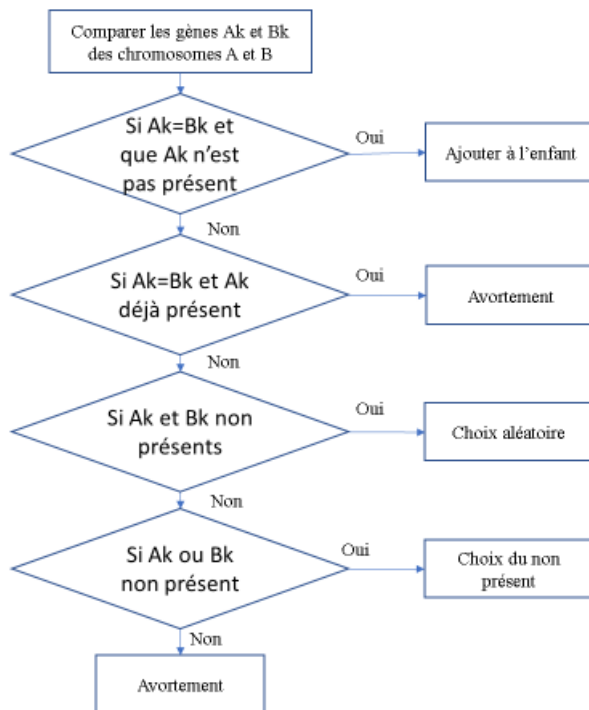


Figure 18 Choix des chromosomes dans le processus de croisement

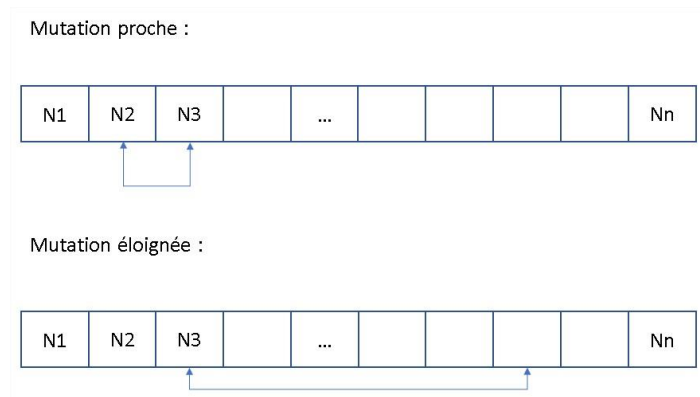


Figure 19 Principe de mutation

Nous avons donc choisi de mettre en place un algorithme génétique, géré de manière centralisée, pour traiter ce problème de type « batching and sequencing » multicritère avec une gestion en somme pondérée des objectifs. Nous présenterons les résultats que nous avons obtenus dans le chapitre suivant. Maintenant, nous allons poursuivre avec l'étude de notre second optimisateur, l'optimisateur dédié au robot de laquage.

2. Optimisateur proactif

Dans ce paragraphe, nous proposons un autre type d'optimisateur qui a la caractéristique d'être proactif : il réévalue l'ordonnancement proposé en temps réel en se basant sur des modèles de prédiction afin d'anticiper la production de non-qualité et l'éviter. Nous nous plaçons, ici, dans le cas de l'ordonnancement d'un poste de travail automatisé pour lequel un modèle de prédiction du risque de production de non-qualité a été élaboré via l'utilisation d'un réseau de neurones. Cet optimisateur base son fonctionnement sur l'interrogation des produits présents dans la file d'attente du poste, sur la récupération des données de la machine et de l'état actuel de la production en aval du poste. Pour être rentable tant en consommation matière qu'en temps d'utilisation, ce poste automatisé doit travailler sur des lots de quantité importante. Cependant si une non-qualité apparaît sur un produit, celle-ci n'est détectable qu'en sortie de machine, sept heures après réalisation de la tâche et celui-ci devra être retravaillé et donc voir son temps de process rallongé de la même durée. En outre, si ce problème survient sur un produit d'un lot, il est fort probable que le lot entier doit être retravaillé.

L'optimisateur a donc été pensé dans l'optique de limiter la production de non-qualité, tout en réduisant la consommation de matière première inhérente à un changement de production. Nous ajoutons aussi à cela, le maintien de l'équilibre d'alimentation entre les différents postes en aval que doit fournir la machine.

En partant du réseau de neurones élaboré par (Noyel et al., 2016) puis complété par la suite dans (THOMAS et al., 2018), nous proposons de l'utiliser comme moyen de déclenchement de réordonnancement, ainsi que comme critère de sélection de celui-ci. Cela constitue le socle d'une technique proactive d'ordonnancement pour l'un des postes de travail de l'entreprise.

Pour qu'il en soit ainsi, nous faisons fonctionner ce réseau de neurones avec un nombre prédéterminé de lots élémentaires présents dans la file d'attente du poste de travail en question. Ce réseau de neurones nous offre deux possibilités majeures :

- Proposer un réglage alternatif de la machine afin de réduire le risque de non-qualité. Cependant ce changement peut, en contrepartie, induire un temps de production plus important pouvant mettre les autres lots en retard.
- Effectuer un échange entre le premier lot de la file d'attente et l'un des autres présents.

La première option a été étudiée dans la thèse de Mélanie, un pilote avait même été mis en place mais malgré son intérêt, son utilisation n'est pas devenue un automatisme et est malheureusement tombé en désuétude. C'est pour cette raison que nous avons choisi d'explorer la seconde possibilité. Le calcul du réseau neuronal est initié juste avant le passage du lot sur le poste de travail. Si le risque de non-qualité est jugé comme acceptable **Figure 20** (inférieur à un seuil à établir) alors les lots sont réalisés dans l'ordre prévu initialement.

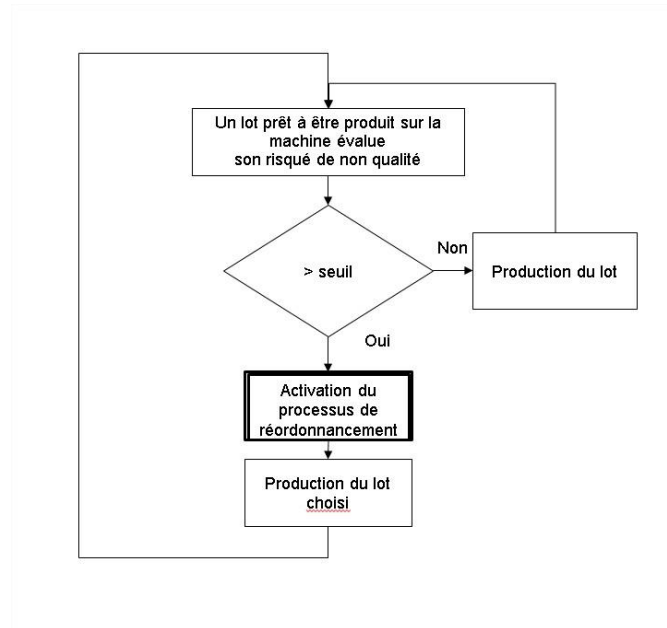


Figure 20 Risque de non-qualité détecté

Sinon, si le réseau de neurones établit que le risque dépasse le seuil pour le prochain lot qui est sensé passer, alors le système entre dans une prise de décision collaborative avec les autres lots en file d'attente du poste de travail **Figure 21**. Le but de cette négociation est de définir quel est le lot de la file d'attente le plus apte à être produit relativement aux différents critères du poste de travail. Pour notre cas d'étude, nous avons sélectionné cinq critères : le **risque de non-qualité**, le **ratio critique**, un **équilibre de la répartition des charges** entre les différents postes clients, **les temps de réglages** induits par le changement de lots et l'impact sur **la nervosité du système** (ce qui se représente par le nombre de modifications apportées à la planification originale), ce dernier critère est présent pour faire en sorte de ne pas perturber toute la logistique déjà mise en place au poste de travail (approvisionnement en laque , en panneaux ou autres composants nécessaires aux opérations de la machine (quincaillerie par exemple)).

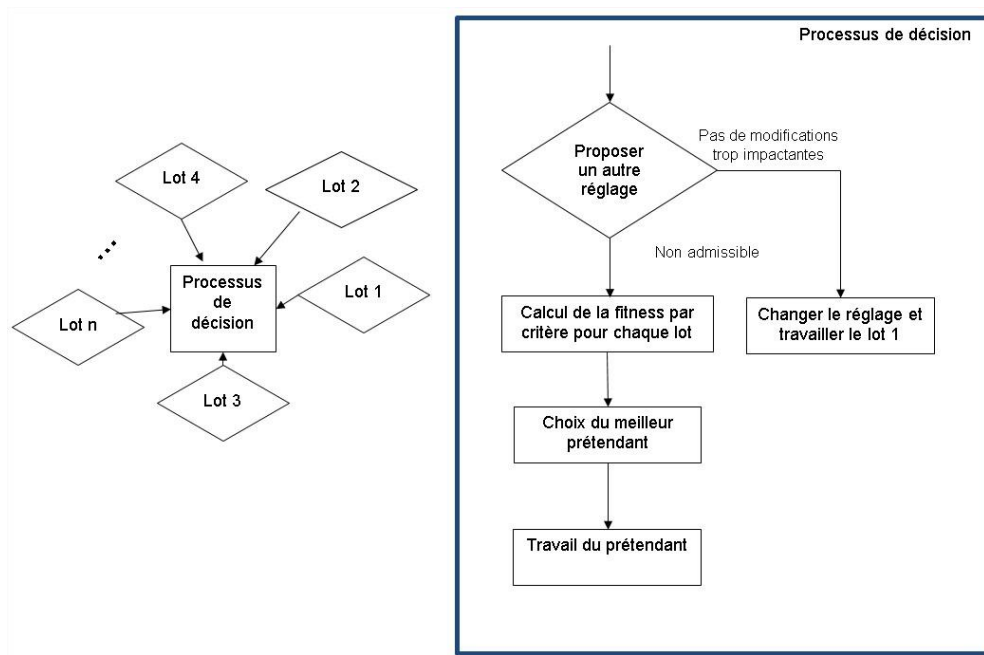


Figure 21 Processus de réordonnement

En suivant ces critères, les lots négocieront entre eux qui sera le prochain à être travaillé. Les techniques de décision collective implémentées dans des systèmes multi-agents sont nombreuses. Certaines s'inspirent d'attitudes sociales telles que le consensus (Olfati-Saber et al., 2007) ou le vote par majorité (Pitt et al., 2006), d'autres cherchent à imiter les comportements biologiques comme la robotique en essaim (Schmickl et al., 2009) avec les essaims d'abeilles (Teodorovic, 2003) ou les colonies de fourmis (Xiang and Lee, 2008). D'autres encore sont tirées de la théorie des jeux (Parsons and Wooldridge, 2000).

Pour notre étude nous nous sommes dirigés vers l'utilisation du processus de hiérarchie analytique (AHP), introduit par (Saaty and Vargas, 1980). Cette technique est particulièrement bien appropriée pour les prises de décisions complexes et offre l'avantage aux personnes décisionnaires d'établir des priorisations.

Le principe consiste à transformer des décisions complexes, basées parfois sur des critères abstraits, en des comparaisons deux à deux pour établir par la suite une synthèse. L'avantage premier est la capacité à prendre en compte aussi bien les aspects subjectifs que objectifs d'une décision. De plus, l'algorithme AHP est construit de manière à s'assurer de la consistance de l'évaluation réalisée. Pour résumer, AHP est une technique standard pour la prise de décision et plus particulièrement lorsqu'il y a des variables de décision indénumérables.

Notre implémentation de l'AHP est inspirée de (Teknomo, 2017). Chaque lot élémentaire est comparé aux autres présents dans la file d'attente, critère par critère, de manière à construire des matrices de comparaison. Ces matrices sont à diagonales unitaires et les autres valeurs sont comprises dans l'ensemble suivant :

$E = \{$

- $1/3 \Rightarrow$ le lot i est beaucoup moins intéressant que le lot j
- $1/2 \Rightarrow$ le lot i est moins intéressant que le lot j
- $1 \Rightarrow$ les deux lots ont un impact identique
- $2 \Rightarrow$ le lot i est plus intéressant que le lot j
- $3 \Rightarrow$ le lot i est beaucoup plus intéressant que le lot j

$\}$

Une matrice de prédominance doit être réalisée pour chaque critère à optimiser : Soit « A » la matrice représentant les valeurs de prédominance d'un lot par rapport à un autre vis-à-vis du critère « a » où $A_{i,j}$ exprime la valorisation du lot i par rapport au lot j . Chaque matrice de comparaison doit satisfaire aux propriétés suivantes :

- $A_{j,i} = 1 / A_{i,j}$
- $A_{i,j} > 0$

Les vecteurs colonnes de chaque matrice sont normalisés puis sont tous sommés de manière pondérée afin de représenter l'importance de chaque critère dans la décision globale. Le lot dont la somme des vecteurs est la plus élevée est désigné comme celui à être travaillé en premier. Les étapes détaillées de la méthode sont représentées sur la **Figure 22**. « c » est la variable d'itération permettant d'explorer chacun des m critères,

« K » est une matrice $n \times n$, ou n désigne le nombre de lots, qui enregistre les $A_{i,j}$ décrit précédemment.

La fonction « Comparer » donne le résultat de la comparaison entre le lot i et le lot j pour le critère c .

La fonction « Normaliser(K) » sert à normaliser les colonnes de K .

« E » est une matrice $n \times m$ représentant la somme des $K_{i,j}$ pour le lot i par rapport au critère c .

L'algorithme renvoie le lot avec la plus haute valeur.

```

n : nombre de lots ; m : nombre de critères ; K : matrice n*n ; E : matrice n*m ; c=0 ;
W : vecteur(m) des poids attribués à chaque critère

Tant que c<m :
  i=j=0
  Tant que i<n-1 :
    K(i,i) = 0
    j=i+1
    Tant que j<n :
      K(i,j)=Comparer(i,i,c)
      K(j,i) = 1/K(i,j)
    Fin tant que
  Fin tant que
  Normaliser(K)
  l = 0
  Tant que l < n :
    E(l,c) =  $\sum_j K(l,j)$ 
  Fin tant que
Fin tant que
a=0
Choisi = 0
Tant que a<n :
  Suivant =  $\sum_k E(k,a) * w_k$ 
  Si Suivant > Choisi alors
    Choisi = Suivant
  Fin si
Fin tant que
Retourner Choisi

```

Figure 22 Diagramme de fonctionnement du protocole AHP

Pour cet optimisateur, nous avons choisi d'opter pour une technique proactive, afin de bénéficier des travaux qui avaient été menés sur les réseaux de neurones. En outre, comme la première optimisation se concentrait essentiellement sur du prévisionnel avec prise de décision centralisée faisant intervenir des groupes de lots (niveau juste en dessous de l'ERP sur la **Figure 10**), nous voulions nous tourner vers une solution plus distribuée où chaque lot interviendrait localement dans la prise de décision pour faire face à un risque de perturbation potentiel (niveau inférieur de la **Figure 10**). Cette solution fait intervenir une prise de décision par une méthode type AHP. Maintenant que nous avons décrit les deux catégories d'optimisateurs que nous avons réalisés, pour compléter l'architecture il reste à discuter de la manière dont ces éléments vont coexister dans le système, c'est donc l'objet du paragraphe suivant.

D. Communication et synchronisation

Pour qu'une architecture soit viable, outre les algorithmes de décisions embarqués dans les optimisateurs et une structure à donner aux entités la composant, il faut gérer toute la partie communication entre les éléments de l'architecture. Cela englobe bien évidemment l'échange d'information entre les entités de même niveaux (les lots, les postes de travail et les hommes) mais aussi la communication avec les entités d'autres niveaux (lots élémentaires avec lot de fabrication, poste de travail avec atelier). En particulier, il faut s'attarder sur la manière dont les changements locaux vont se répercuter sur la planification globale et vice versa.

À l'heure actuelle, notre mise en place de l'architecture s'arrête à la plateforme Tracilogis dont les moyens de communication inter-entités sont cadrés par le langage de développement du système multi-agent choisi, à savoir JADE, et surtout par son implémentation. Toutes les communications passent par un système d'abonnement géré par un (Dialog Facilitator) DF, comme cela est le cas pour la plupart des architectures que nous avons évoqué dans le chapitre précédent.

Sans rentrer spécifiquement dans les détails, car à l'instant présent il ne s'agit là que d'hypothèses que nous n'avons pas testées, notre idée serait :

- Remplacer les ordres de fabrication (voir *Erreur ! Source du renvoi introuvable.*) papier par des cartes RFID passives qui contiendrait l'identifiant de lot élémentaire des produits qu'elles accompagnent et éventuellement l'ensemble des informations les concernant (leurs gammes de fabrication et leurs caractéristiques).
- Équiper les conteneurs (chariots ou palettes) de micro-ordinateurs embarqués (type Raspberry), de lecteurs RFID actifs (afin d'interroger les cartes RFID des lots qu'ils contiennent) et d'écrans (pour retransmettre aux collaborateurs des informations telles que la destination des lots ou leur degré d'urgence).
- Intégrer à chaque poste de travail, un lecteur pluggé sur le M.E.S. (voir *Erreur ! Source du renvoi introuvable.*). Celui-ci serait alors la parfaite interface entre le trio produits / ressource / homme gérant ainsi la prise de décision locale.

La remontée et la descente des données vers le système d'information pourrait ainsi se faire en temps réel étant donné que chaque poste de travail est connecté au réseau de l'entreprise.

Comme dit juste avant, Il s'agit là d'une vision purement hypothétique ne tenant compte en aucun cas de toutes les difficultés voire impossibilités techniques que pourraient induire une telle réalisation. Notre seule certitude à ce sujet est que celle-ci nécessiterait un investissement en temps et en argent non négligeable pour l'entreprise tout en créant un gap considérable dans sa manière de travailler. C'est à cause de telles considérations et de notre volonté de ne pas perturber le bon fonctionnement de l'entreprise pendant ces périodes économiques difficiles que nous avons fait le choix de valider la crédibilité de nos propositions par simulation et ou transposition sur plateforme de test.

Nous allons, à présent, nous focaliser sur la partie synchronisation entre optimisateurs. Nous avons émis l'hypothèse que, dans un premier temps, un simple écoulement naturel, tout du moins entre les deux optimisateurs que nous avons réalisés, serait suffisant. En effet, étant donné qu'ils ne fonctionnent pas par rapport aux mêmes horizons ni aux mêmes jeux de données et qu'un certain nombre de postes de travail les séparent, il ne nous paraissait pas nécessaire d'établir un mécanisme spécifique de synchronisation. Pour être plus précis, re-voyons l'ensemble des éléments à notre disposition et la manière dont ils se positionnent les uns par rapport aux autres. La **Figure 23** schématise ce processus :

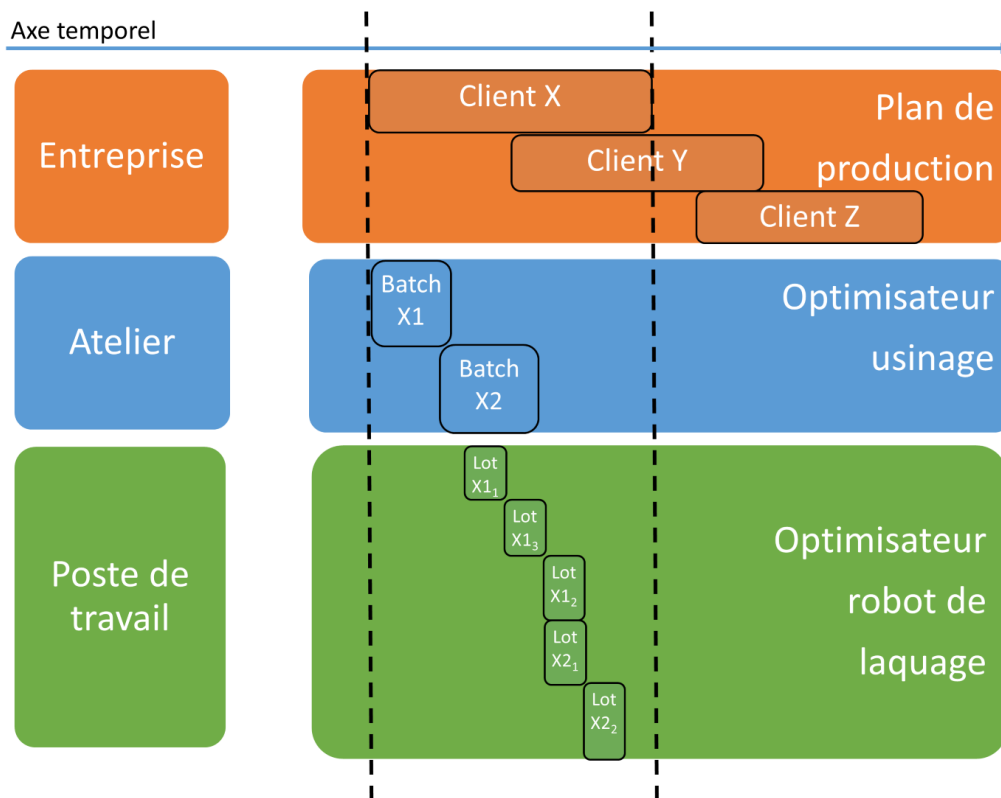


Figure 23 Enchaînement des opérations sur les optimiseurs

- Au niveau global de l'entreprise, le plan de production établie approximativement les instants de début de production de chaque client à partir de leur date de livraison au plus tard.
- Avant le commencement de l'usinage d'un client, l'optimiseur de l'atelier est lancé, afin de déterminer les regroupements en batches à réaliser et leur séquençement. Ainsi les batches seront produit un à un dans l'atelier usinage et le flux restera inchangé jusqu'à l'étape de tri.
- Les batches alors redivisés en lots élémentaires sont emmenés dans la file d'attente du robot de laquage où les opérateurs vont, a priori, les ordonner suivant deux critères, la couleur de laque qui était utilisée lors de l'opération précédente et le temps de process des lots qu'ils ont à leur disposition (il est à noter qu'un ordonnancement informatisé serait à intégrer à cette phase, mais son étude ne fait pas partie de ces travaux). C'est là, que l'optimiseur proactif du robot entre en jeu. Il agira en temps réel pour réordonnancer dès que le risque de non-qualité sera trop élevé pour le lot sur le point d'être travaillé.

Forts de ces considérations, nous avons conclu que dans les premiers temps, suivre le flux poussé généré spontanément par le fonctionnement standard de la chaîne de production était une méthode de synchronisation viable. Une méthode plus élaborée pourrait et devra être mise en place, mais fera l'objet de travaux ultérieurs. En effet, nous n'avons pas eu le temps d'aborder la partie applicative relative à cet aspect durant la thèse. Nous évoquerons dans le dernier chapitre la manière dont nous souhaiterions procéder pour la suite des travaux.

Dans ce chapitre, nous avons étudié la partie modélisation de notre problème et avons passé en revue l'adaptation des techniques que nous avons élaborées spécifiquement pour notre problématique, nous allons à présent décrire la manière dont nous les avons mises en œuvre, ainsi que les différents résultats que nous avons obtenus **Figure 24**.

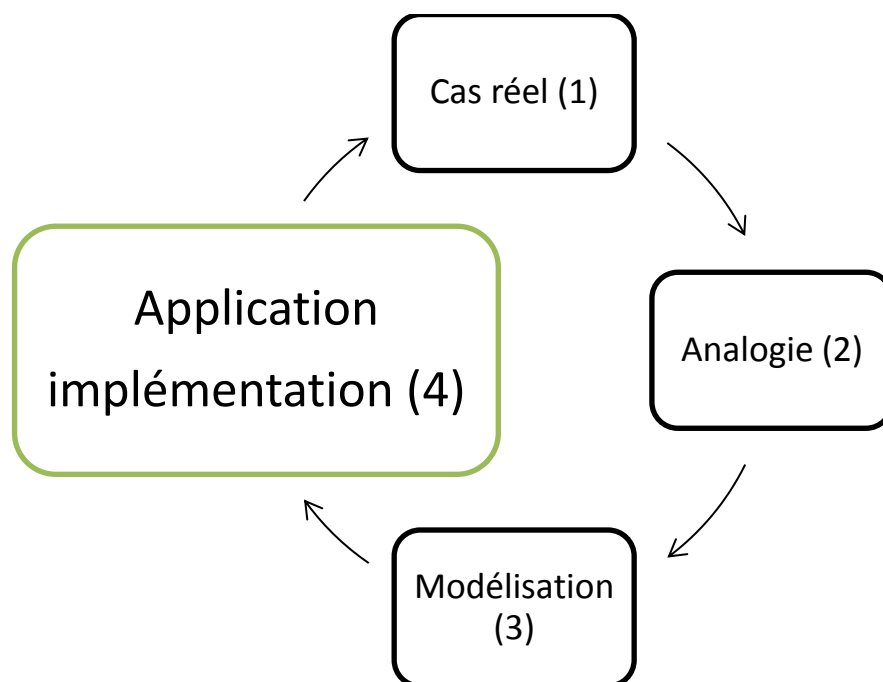


Figure 24 Application

IV. Application industrielle

A. L'ordonnancement global

Comme dit dans le chapitre précédent, la production d'un planning global sert majoritairement à titre indicatif et fournit plus particulièrement des plages acceptables pour les ordonnanceurs spécifiques locaux. De ce fait, il n'a pas été jugé nécessaire de réaliser un développement personnalisé afin de le produire et l'afficher.

Les temps de production moyens par pièce, calculés à partir des données relevées depuis le système d'information sur une période de 20 semaines, sont appliqués sur les quantités hebdomadaires à produire.

Les postes déterminés comme jalons pour cet ordonnancement sont les suivants :

- L'imbrication des pièces précédant leur découpe sur les centres à commandes numérique.
- L'application de la couche d'apprêt.
- Le passage des pièces à la phase de tri.
- Le passage des pièces en laquage (sans distinction faite sur les pièces devant passer par le robot de laquage et celles devant passer par les cabines).
- Le passage des pièces brillantes sur la lustreuse.
- Le poste d'expédition, précédant le départ des pièces de l'usine.

Une solution clé en main nommée Direct Planning produite par Volume Software a été retenue. Le travail a essentiellement été concentré sur la transmission des données au logiciel et à l'apprentissage de son fonctionnement. Cependant, il s'est avéré que le traitement direct par le logiciel ne proposait pas des résultats à la hauteur des attentes. Par conséquent, le module développé à l'origine uniquement pour la récupération et le formatage des données est devenu le module d'ordonnancement et le logiciel s'est finalement vu relayé au rang d'afficheur (**Figure 25**).

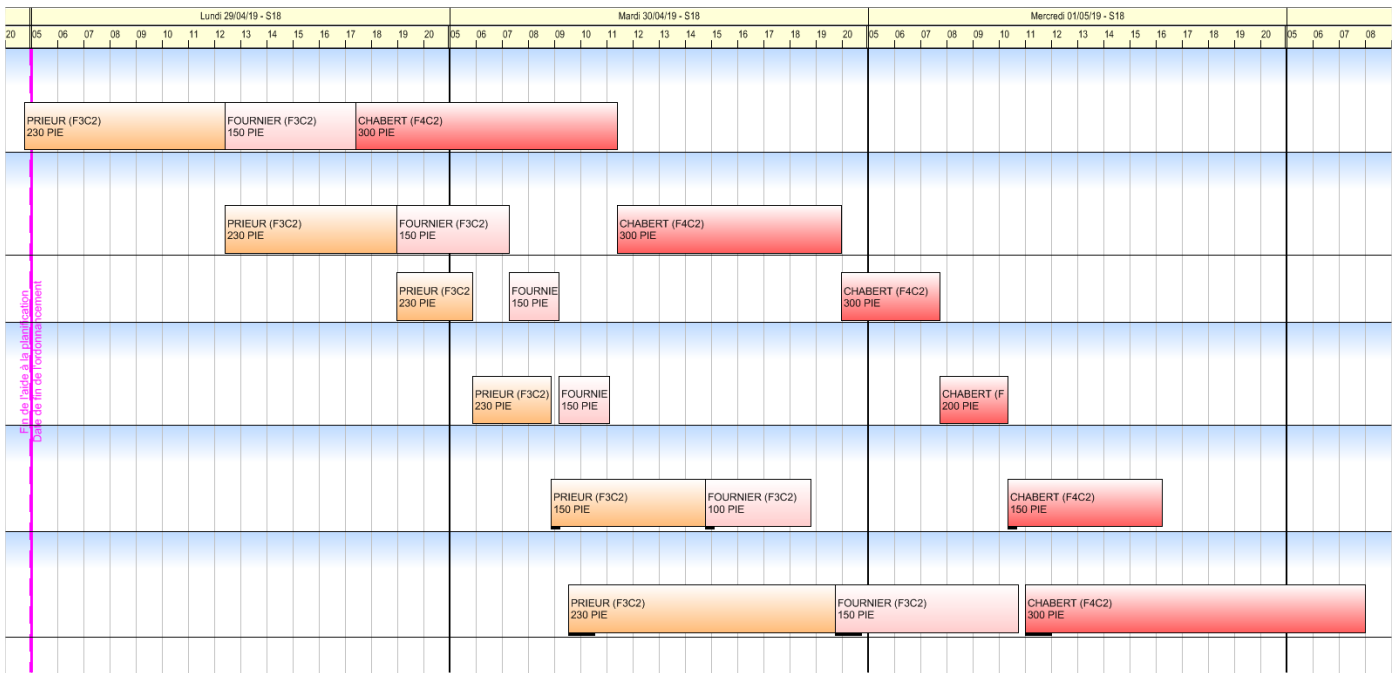


Figure 25 Rendu du logiciel Direct Planning

B. Choix de pilotes

Théoriquement, des regroupements et séquençements permettant d'atteindre les objectifs à optimiser sont envisageables pour chaque poste de travail. Il s'agirait d'un travail trop fastidieux que de chercher à réaliser un optimiseur pour chacun d'eux.

Les différentes observations sur le terrain et une simulation de la charge de travail faite sur logiciel de modélisation et de simulation ont permis de déterminer les secteurs les plus critiques à étudier : l'usinage et le laquage (voir **Figure 26**).

En effet nous avons pu constater que les lots élémentaires ayant les mêmes caractéristiques sont instinctivement regroupés par les opérateurs de l'usinage pour essayer de réduire les changements de série. Ces regroupements n'ont que peu de raison de changer jusqu'à l'étape de tri, où les dimensions et l'état de toutes les pièces vont être vérifiés. Dans le même temps, l'opérateur du poste va rediviser ces regroupements en nos lots élémentaires, pour les regrouper à nouveau selon les couleurs. Cependant, il faut tout de même veiller à ne pas faire des regroupements trop importants pour éviter que le flux soit en dents de scie. La principale difficulté consiste à déterminer jusqu'à quel point regrouper ou non les lots élémentaires afin d'obtenir le meilleur compromis entre fluidité du flux et réduction des coûts liés aux changements de série.

Dans les paragraphes suivants, l'application faite avec chacun des deux optimisateurs réalisés sera décrite. Le premier étant l'optimisateur réactif dédié à la limitation des coûts de production appliqué sur l'atelier usinage de l'entreprise.

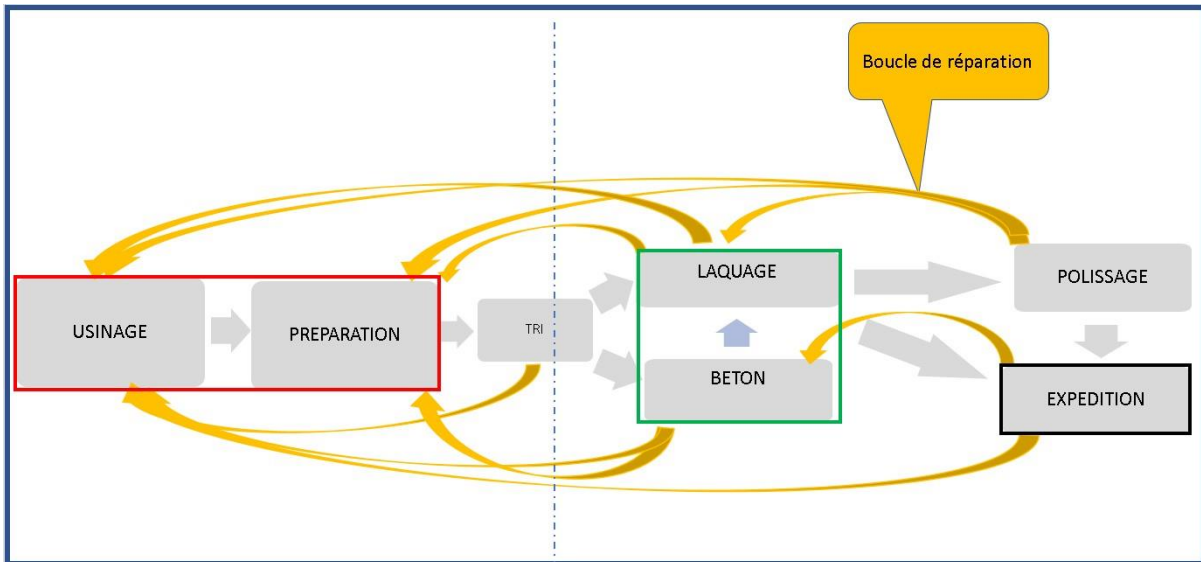


Figure 26 Boucles de réparations et regroupement

C. Ordonnanceur pour la partie découpe

Nous avons appliqué l'optimisateur local réactif visant à limiter la consommation matières et les temps de réglages à l'atelier d'usinage de l'entreprise Acta. En effet, dans le secteur usinage, des panneaux mélaminés de plusieurs épaisseurs différentes sont utilisés pour réaliser les produits demandés par les clients. Les panneaux sont découpés à dimension par un centre d'usinage. Les pertes de temps majoritaires lors de cette étape de la fabrication viennent du changement des têtes d'usinage et de la manutention nécessaire lors d'un changement d'épaisseur de panneau pour alimenter la machine. En plus de limiter le temps de réglage et de changement, rassembler les produits identiques permet de limiter les pertes matières.

a) Résultats d'implémentation

La méthode proposée a été implémentée en VB.Net et a été mise en place pour le traitement des commandes d'un unique client depuis 2016. Son implémentation a permis à l'entreprise d'améliorer sa relation avec le client et d'augmenter le volume commandé. Elle a été en mesure d'atteindre les objectifs qualité et délais d'expédition requis tout en réduisant le temps global de production en réduisant certaines étapes de non-valeur ajoutée. Le **Tableau 10** met en vis à vis les performances de l'entreprise sur le client considéré avant et après la mise en place de notre proposition sur la même période de l'année. Nous avons comparé la fin d'année 2015, où le système n'existait pas encore, à la fin de l'année 2016 où le système a été mis en place. Sur ces deux périodes, les conditions étaient similaires, avec même une augmentation de quantité commandée de 32% en 2016. Pourtant, le temps de production global a diminué d'une journée.

	2015 (sans optimisation)	2016 (avec optimisation)
Nombre de pièces commandées par semaine	S49 : 456 pièces S50 : 247 pièces S51 : 596 pièces	S49 : 633 pièces S50 : 518 pièces S51 : 561 pièces
Nombre de jours séparant l'ouverture de la première palette et la fermeture de la dernière palette à l'expédition	S49 : 5 jours S50 : 5 jours S51 : 5 jours	S49 : 4 jours S50 : 4 jours S51 : 4 jours

Tableau 10 Comparaison avec et sans optimisation

Ces résultats montrent des améliorations notables pour l'entreprise mais il est difficile de quantifier l'impact direct de notre proposition sur ces performances. Pour valider l'efficacité de l'optimisateur, nous avons réalisé une simulation sur une semaine de production de l'atelier usinage avec une commande de 36 contremarques pour 4 épaisseurs de panneaux.

Le **Tableau 11** présente les résultats comparatifs entre notre méthode et deux règles d'ordonnement potentiellement utilisable par l'entreprise :

- Regrouper tous les lots d'une même contremarque dans le même lot de fabrication
- Regrouper tous les lots de même épaisseur dans le même lot de fabrication.

	Méthode proposée	Épaisseur par épaisseur	Contremarque par Contremarque
Nombre de réglages	11	4	22
W_{ip} max entre les lots d'une même contremarque	5	14	4
Nombre max de palettes ouvertes en simultané	7	16	2

Tableau 11 Comparaison entre notre approche et les anciennes techniques utilisées

Notre proposition donne un parfait compromis entre ces deux règles extrêmes de pilotage. Elle s'approche toujours de la solution la plus optimale suivant chaque caractéristique. Pour répondre à l'obligation d'avoir tous les produits d'une contremarque présents en simultané à l'étape suivante et dans l'optique d'établir un management visuel simple et standardisé pour les opérateurs humains, l'application développée conduit à un résultat aisément compréhensible par les collaborateurs grâce à sa représentation visuelle. Des documents spécifiques ont été réalisés pour la partie usinage et d'autres pour le reste du processus. Pour l'étape de tri, une application visuelle a été développée pour permettre au collaborateur de séparer les lots de fabrication et regrouper les lots élémentaires simplement dans leur contremarque en ajoutant un code couleur pour identifier les différents éléments

La **Figure 27** est une capture d'écran de cette application. Le premier niveau des compartiments présente l'ensemble des lots de fabrication formant un lot de contremarques, puis à chaque sous-niveau se présentent chacune des contremarques contenues par celui-ci. Enfin le dernier sous-niveau montre quels lots élémentaires sont à replacer dans une contremarque (par exemple : le lot de contremarque D est composé des lots de fabrication D1 D2 et D3 ; les contremarques 549483, 549484 et 594486 sont recomposables à partir des 3 lots de fabrication. La contremarque 549483 sera complète lorsque les lots élémentaires 105511 et 105503 seront rassemblés).

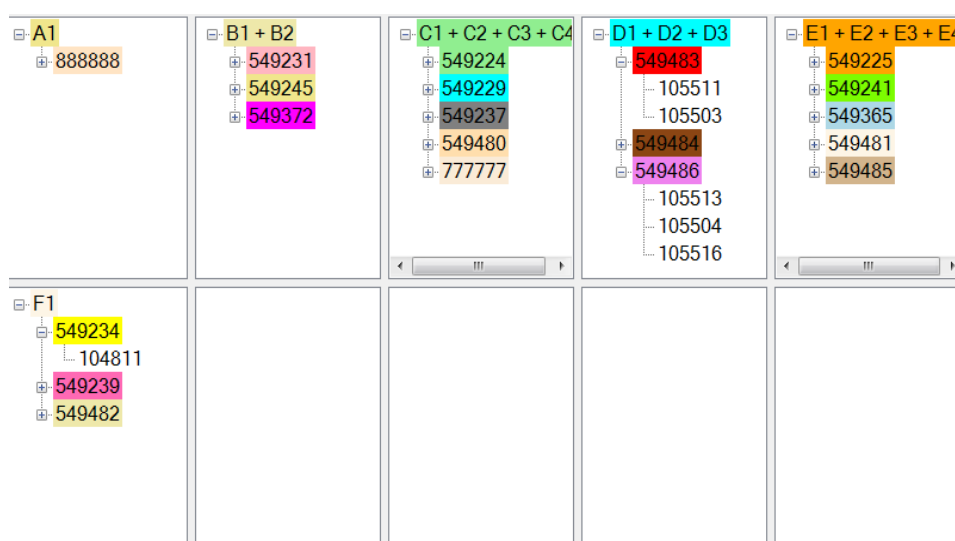


Figure 27 Une capture d'écran de l'application servant à re-trier les lots

Ce groupement à deux niveaux garantit que l'étape de tri aura suffisamment d'espace pour diviser les lots de fabrication et reconstruire les lots élémentaires des contremarques sans avoir une palette de contremarque non complète trop longtemps en file d'attente et en limitant le nombre de tâches à non-valeur ajoutée induites par cette démarche.

Il y a entre 60 et 100 lots élémentaires à grouper en lot de fabrication par semaine. Cela équivaut à un nombre de produits à réaliser compris entre 600 et 900. Cela représente un ensemble allant de 16 à 40 lots de commandes client à ordonnancer.

b) Comparaison avec la résolution par méthodes exactes

En complément de ce qui a été dit dans le chapitre précédent vis-à-vis de la résolution par méthode exacte en utilisant des solveurs comme (GLPK), nous avons tout de même souhaité effectuer une comparaison de performance.

Pour se faire, nous avons relaxé les contraintes relatives au setup et à la distance entre les lots d'une même commande : équations (26) à (33). Nous avons fait ces simplifications car relativement au temps total de traitement des batches, les temps de setup peuvent être considérés comme négligeables et surtout, le retrait de ces contraintes conduit à une réduction de 15% sur la quantité de variables du modèle.

De plus, nous avons choisi de subdiviser le cas exemple cité pour le calcul du nombre de variables (à savoir : une semaine de 45 commandes, contenant au plus 5 lots par commande donnant un total de 80 lots, et limitant le nombre de batches à 15) à plusieurs sous problèmes. Ainsi, en considérant seulement 5 commandes, toujours composé au plus de 5 lots par commande, donnant cette fois-ci un total de 12 lots et si on se limite à faire au plus 6 batches, on obtient alors $N_x = 542$ variables et $N_c = 3002$ contraintes.

Description du cas de test :

Considérons 5 commandes composées respectivement de 1,1, 2, 5 et 3 lots. La due date des commandes est fixée à $t+160h$ et le temps de traitement est fixé à $1.1 * \text{Surface (h)}$. Le **Tableau 12** décrit les caractéristiques épaisseur et surface des lots composant les commandes.

	1		2		3		4		5	
	Ép.	Surf.	Ép.	Surf.	Ép.	Surf.	Ép.	Surf.	Ép.	Surf.
C0	19	0.1								
C1	19	0.1								
C2	28	3.7	19	11.9						
C3	28	0.3	19	1.77						
C4	19	1	19	0.62	16	0.9				

Tableau 12 Épaisseur et surface du cas de test

Nous avons fait tourner le solveur sous deux configurations différentes :

- La fonction objective ne dépend que du nombre de batches à minimiser et les contraintes (26) à (33) sont supprimées.
- Seules les contraintes (32) et (33) sont supprimées et où la minimisation de W_i (espace maximal séparant les lots d'une même commande) est considérée dans la fonction « objectif ».

Résultats configuration 1 :

4 batches générés

	1	2	3	4	5
C0	B ₃				
C1	B ₂				
C2	B ₁	B ₁			
C3	B ₁	B ₃	B ₃	B ₀	B ₁
C4	B ₁	B ₂	B ₀		

Tableau 13 Batches configuration 1

Séquence : b₃b₀b₁b₂

Temps de traitement :

- b₃ de 0 à 7.909
- b₀ de 7.909 à 11.05
- b₁ de 11.05 à 16.12
- b₂ de 16.12 à 31.05

Temps séparant des lots d'une même commande :

- C₀ : 0h
- C₁ : 0h
- C₂ : 0h
- C₃ : 8.21h
- C₄ : 20 h

W_i = 28.21h

	1	2	3	4	5
C0	B ₀				
C1	B ₀				
C2	B ₃	B ₀			
C3	B ₃	B ₀	B ₁	B ₂	B ₃
C4	B ₁	B ₁	B ₂		

Tableau 14 Batches configuration 2

Résultats configuration 2 :

4 batches générés

Séquence : $b_0b_1b_2b_3$

Temps de traitement :

- b_0 de 0 à 15.36
- b_1 de 15.36 à 23.002
- b_2 de 23.002 à 26.066
- b_3 de 26.066 à 31.05

Temps séparant des lots d'une même commande :

- C_0 : 0h
- C_1 : 0h
- C_2 : 15.69 h
- C_3 : 15.69 h
- C_4 : 3.064 h

$W_i = 34.444h$

Conclusions sur les résultats obtenus avec le solveur :

La seconde configuration devant normalement minimiser W_i donne une valeur moins intéressante que celle de la première configuration. Cela s'explique par le fait que dans la seconde configuration le calcul se complexifie et les résultats obtenus deviennent moins bons. Un second cas d'étude, avec un jeu de données plus important a également été testé. Cependant celui-ci a été stoppé manuellement après des intervalles de temps X que nous avons pris croissants au fur et à mesure des itérations (d'abord 20 minutes, puis 40, puis 60 et enfin 90). Nous constatons déjà là les limites de la résolution par méthode exacte. Nous allons poursuivre par une comparaison directe entre l'algorithme génétique développé pour le problème et le modèle sous la première configuration (minimisation du nombre de batches à réaliser uniquement et non prise en compte des contraintes (26) à (33)).

Cette comparaison a été réalisée de la manière suivante :

Un jeu de données correspondant aux commandes d'un client pour une semaine standard a été choisi. Comme pour la comparaison entre les simplifications du modèle mathématiques, le jeu de données complet a été divisé en cinq sous-ensemble afin de permettre au modèle linéaire de fournir un résultat dans un temps acceptable. Chacun de ces sous-ensembles comprend entre 5 et 10 commandes.

L'ensemble des résultats des cinq groupes de données sont décrits en Annexe 4. Ci-dessous le résumé de ces cinq cas d'étude est présenté.

Critère	Modèle	Groupe 1	Groupe 2	Groupe 3	Groupe 4	Groupe 5
Nombre de batch	GA	3	9	5	9	1
	MA	3	6	4	8	1
Nombre de setup	GA	2	5	2	8	0
	MA	1	4	1	4	0
Surface moyenne par batch (m²)	GA	10.98	5.34	9.52	10.17	8.8
	MA	10.98	8.01	12.7	13.08	8.8
Temps de calcul (s)	GA	9	11	15	15	4
	MA	10	2700	2700	2700	5
Espacement max entre lots d'une même commande	GA	1	2	4	4	0
	MA	1	3	5	7	0
Non-respect des contraintes	GA	Dépassement surface max	Ok	Dépassement surface max	Ok	Ok
	MA	Dépassement du nombre de commande max par batch	Dépassement du nombre de commande max par batch	Dépassement du nombre de commande max par batch	Ok	Ok

Tableau 15 Synthèse comparaison des méthodes

Le MA donne de meilleurs résultats sur le séquençement et ainsi réduit le nombre de setup, cependant, à cause de la relaxation des contraintes prévues à l'origine, il ne cherche pas à minimiser l'espacement maximal séparant les lots d'une même commande. De ce fait, il est moins performant pour ce critère. En outre, il ne tient pas compte de la spécificité d'isoler certaines couleurs, c'est pourquoi il réduit le nombre de batches créés par rapport au GA. À cela, il faut ajouter le fait que la contrainte du nombre maximal de commandes à placer dans un batch n'a pas été respecté trois fois sur cinq par le MA tandis que le GA a dépassé par deux fois la limite de surface maximale par batch, considérée comme une contrainte plus faible. Pourtant, malgré les relaxations, 3 cas sur 5 ont été stoppés manuellement après 45 minutes de calcul pour le MA alors que le GA a toujours été capable de fournir une réponse acceptable en moins de 20 secondes. Afin de compléter notre comparaison nous avons calculé la performance relative du GA par rapport au MA :

- Perf nombre de batch = $\frac{\sum_i \frac{N_{batchGAi}}{N_{batchMAi}}}{nombre\ d'échantillon} - 1 = 17,5\%$

Cette valeur signifie que le GA fournit en moyenne un nombre de lots qui est 17,5% plus grand que celui du MA.

- Perf setup = $\frac{\sum_i \frac{N_{setupGAi}}{N_{setupMAi}}}{nombre\ d'échantillon} - 1 = 45\%$

Le GA est 45% moins performant que le MA vis-à-vis du nombre de setup généré par le séquençement.

- Perf surface = $1 - \frac{\sum_i \frac{Surface_{GAi}}{Surface_{MAi}}}{nombre\ d'échantillon} = 16,12\%$

La surface moyenne des batches produits par le GA est en moyenne 16,12% plus faible que le MA.

- Perf temps de calcul = $\frac{\sum_i \frac{T_{MAi}}{T_{GAi}}}{nombre\ d'échantillon} - 1 = 120,6$

Le GA donne des résultats 120,6 fois plus rapidement que le MA

- Perf espacement = $\frac{\sum_i \frac{E_{batchMAi}}{E_{batchGAi}}}{nombre\ d'échantillon} - 1 = 30\%$

Le GA produit des séquençements dont l'espacement maximal entre lots d'une même commande sont en moyenne 30% plus petit que ceux du MA.

Ces résultats montrent bien l'intérêt de l'utilisation d'une technique évolutionniste plutôt qu'une méthode exacte, en particulier lorsque la rapidité de réponse est un enjeu clé.

c) *Transposition sur plateforme de test*

Pour aller plus loin dans l'analyse de notre proposition et pour démontrer sa pertinence au regard d'autres solutions, nous avons cherché à implémenter notre algorithme sur une plateforme de test.

(1) La plateforme TRACILOGIS

La plateforme TRACILOGIS propose un environnement de test pour des applications de multiples scénarios industriels sur des problèmes de classification, d'identification de produits ou encore de pilotage intelligent d'atelier utilisant plusieurs modes de contrôle : centralisé, distribué ou hybride. Celle-ci permet d'implémenter, de tester et de comparer différentes techniques de traçabilité et de pilotage des chaînes logistiques en particulier dans les industries du bois. Le système physique est composé de quatre automates. Chacun d'eux gère sa propre partie de la plateforme. Les produits réalisés sont des plateaux pouvant être marqués par un point ou une ligne sur l'un des côtés. De 0 à 4 palets et de 0 à 4 pastilles peuvent être placés sur les plateaux. Le processus global de la plateforme est présenté sur la **Figure 28** et le logigramme de fonctionnement est représenté sur la **Figure 29**.

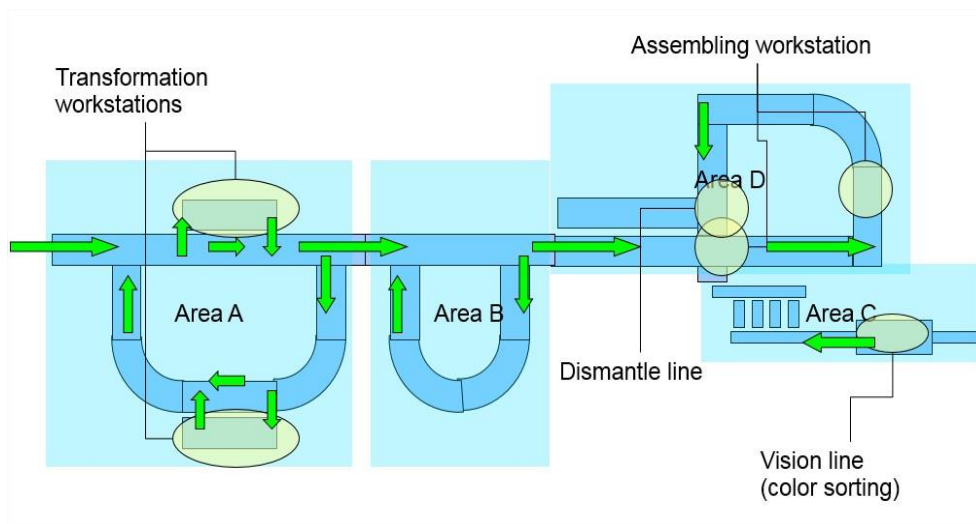


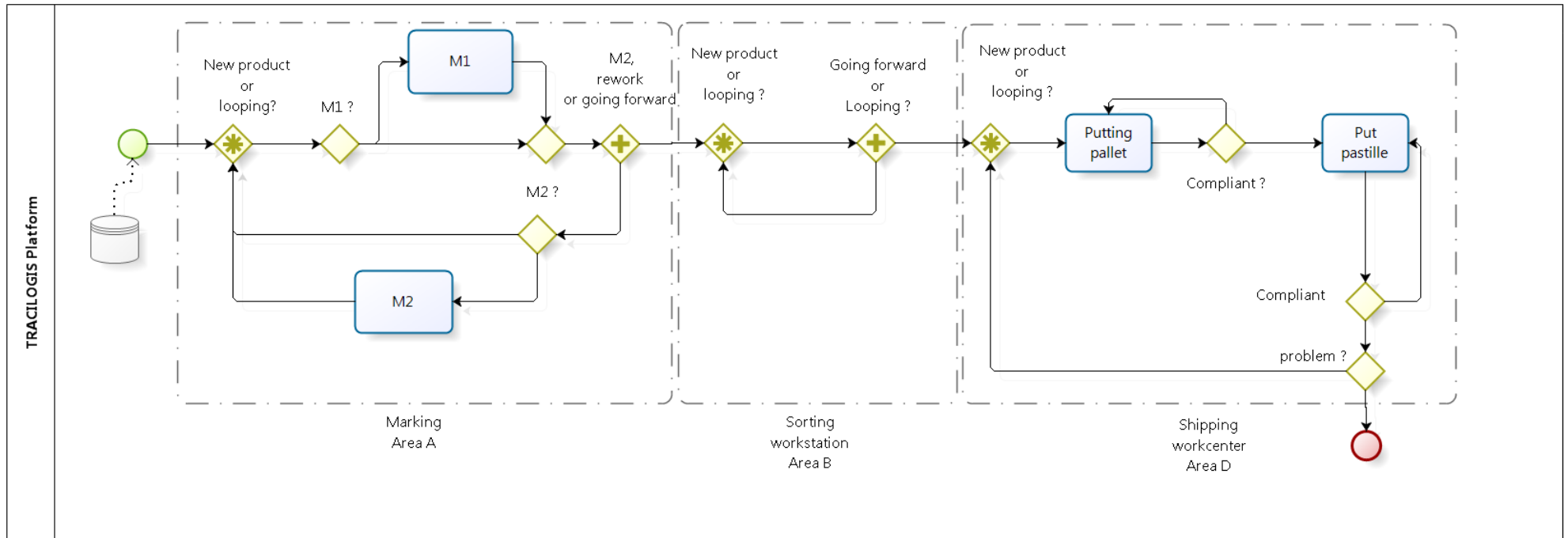
Figure 28 Diagramme de TRACILOGIS

- L'automate A gère le rail alimentant le poste de marquage. Les postes sont placés de manière séquentielle mais chacun d'eux peut être évité suivant la configuration du produit (area A)

- L'automate B contrôle une zone de bouclage permettant de réordonner les produits (area B)
- L'automate C dirige un rail de tri utilisant un système de caméra pour reconnaître la couleur des palets (area C)
- L'automate D gère une ligne d'assemblage : poser des palets et des pastilles sur les plateaux en accord avec la gamme de fabrication imposée et se dirige vers l'une des deux sorties disponibles (area D).

Les plateaux et les palets portent tous des tags RFID et des lecteurs sont placés sur chaque poste de travail de la chaîne et chaque point de découplage.

Le système multi-agents, développé sous JADE, gère l'envoi de toutes les actions à réaliser aux automates. Le transfert d'informations est déclenché par les comportements implémentés dans les agents. Les interactions se font via la transmission de messages asynchrones.



Legend:



Decoupling point
SCC change



Choice between
letting enter a new
product and looping

Figure 29 Logigramme TRACIOLOGIS

Au sein d'un système multi-agents, un nombre non nécessairement défini d'agents évoluent et chacun d'eux est en interaction avec les autres pour atteindre ses objectifs et parfois également un but commun. Leur champ d'action peut également être limité tant spatialement que temporellement ou socialement, c'est ce qui est communément nommée myopie. Tout système multi-agents cherche à maintenir une communication constante entre les agents, ne jamais détériorer le fonctionnement de l'ensemble et garantir l'organisation du savoir entre eux. À l'origine, des agents étaient instanciés pour les types d'acteur suivants :

- Les agents « ressource », passifs, attendant des demandes d'action de la part des produits.
- Les agents « produit », conscient de leur gamme de fabrication et de leur destination objective. Ils recalculent à chaque mouvement le chemin le plus court vers leur destination en suivant la méthode de Dijkstra. Ils ont connaissance de leur propre état d'avancement mais ne savent rien des autres produits. Le diagramme UML de cet agent est présenté en **Figure 30**.
- Les agents « PLC (Production Line Controller) », un par automate gère la coordination de toutes les actions réalisables dans la zone qu'il supervise.
- Les agents « RFID » sont les agents associés à tous les lecteurs RFID placés sur les points de décisions du tapis roulant (entrée et sortie de machine, point de couplage/découplage, ...).

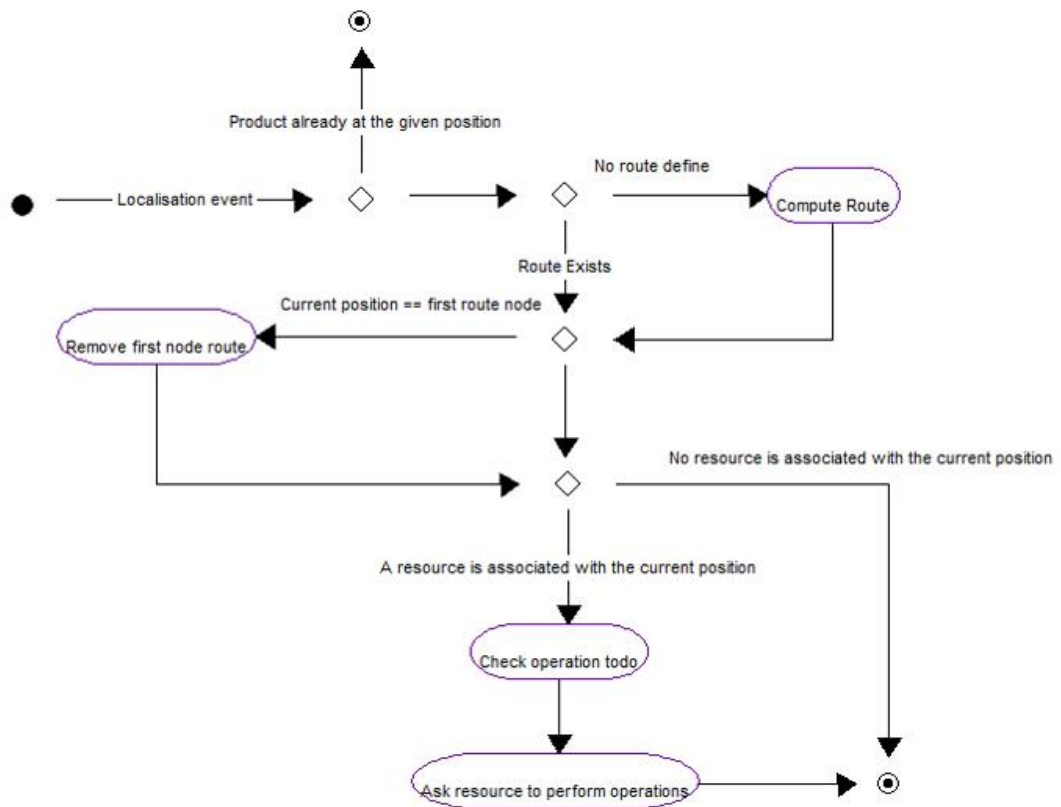


Figure 30 UML de l'agent produit

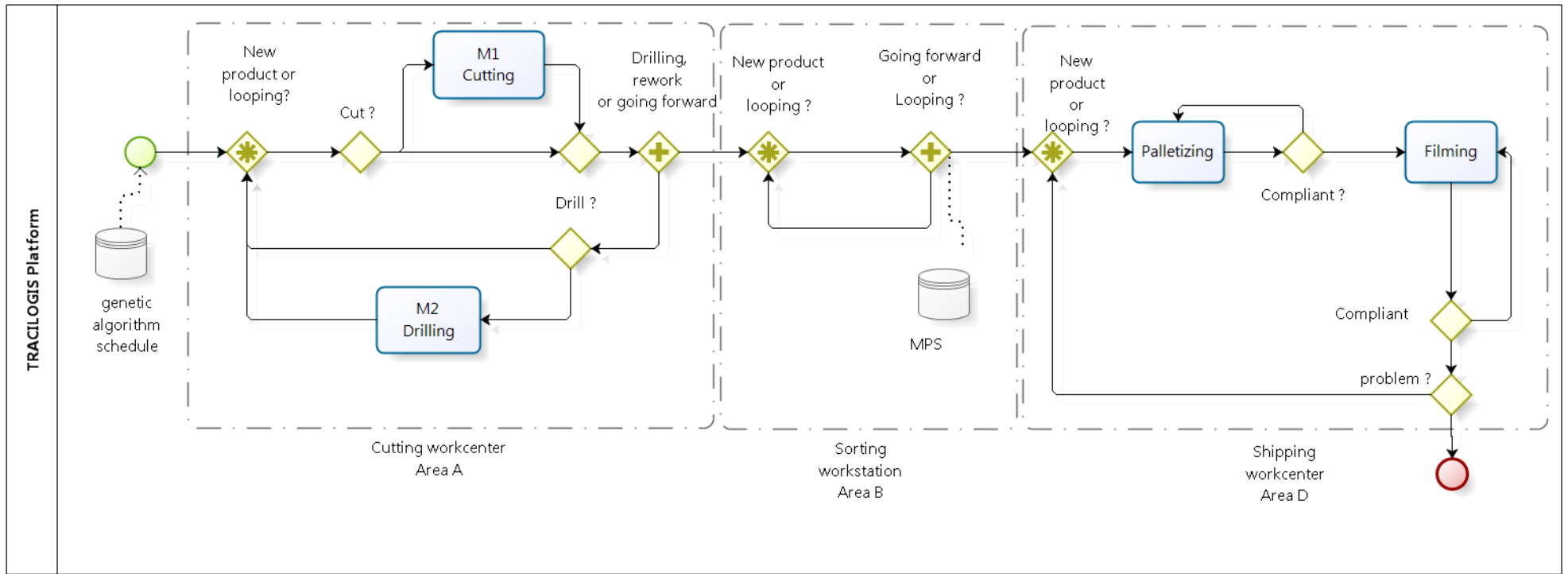
(2) La transposition de notre cas d'étude

La plateforme TRACILOGIS rend possible l'étude de nombreuses possibilités de scénarios. Mais notre objectif était, ici, de réaliser des cas de test aussi proche que possible du cas industriel de manière à démontrer l'apport et la faisabilité de notre proposition à l'entreprise sans perturber son fonctionnement. De plus, elle nous permettait de mesurer la fiabilité d'un système multi-agents en réponse à des perturbations causées par les boucles de réparation. Pour se faire, des évolutions de structure et de fonctionnement ont dû être apportées à la plateforme pour réaliser la transposition que nous décrivons ci-après :

- La zone « A » a été associée à l'atelier d'usinage. La première machine de marquage servant à simuler le flux de production standard et la seconde comme le flux des produits demandant des étapes supplémentaires d'usinage (perçage, défonçage, ...) avec un cycle plus long
- La zone « B » qui est un simple carrousel peut être identifié à l'étape de tri où les produits sont déposés, contrôlés, séparés et regroupés par couleur. L'ouvrier faisant partir les lots suivant le plan de production établi.

- La zone « D » est comparée aux étapes de finition et expédition avec la dépose de palets pour la finition (le nombre de palets à placer sur le plateau doit être mis en parallèle au temps que les produits passe au laquage chez Acta) et la dépose de pastilles (à mettre en parallèle du passage en polissage pour les brillants et du filmage pour les satinés).

La transposition est schématisée sur la **Figure 31**. La première implémentation à ajouter à la plateforme consistait à introduire la notion de réparation. Pour cela, une règle aléatoire a été insérée dans le comportement de l'agent ressource pour envoyer à l'agent produit une notification d'opération échouée. Lorsqu'un produit reçoit une telle réponse, il recalcule le plus court chemin pour refaire l'opération qui vient d'échouer. Ensuite un agent « ordonnancement » a été introduit dans le système, son objectif est de s'assurer que les produits quittent la zone B selon l'ordre du planning qui lui est, soit fourni, soit qu'il calcule, sinon il oblige les produits à tourner dans le carrousel. Pour cela, les produits communiquent avec lui et lui demandent, s'ils ont le droit de passer en zone D. Une fois ces deux améliorations rajoutées, les premiers tests ont pu être réalisés.



Legend:



Decoupling point
SCC change



Choice between
letting enter a new
product and
letting a product loop

Figure 31 Transposition d'Acta sur TRACIOLOGIS

(3) Expérimentations :

Nous avons choisi de créer une production de dix plateaux (nombre maximal de plateaux disponibles sur la plateforme). Ils ont été répartis suivant les 3 configurations suivantes : 4 produits avec une ligne faite sur la machine M1 (configuration α **Figure 32**), 3 produits avec un point fait sur la machine M1 (configuration β **Figure 32**) et 3 produits fait avec une ligne sur la machine M2 (configuration γ **Figure 32**).

La configuration α est une ligne dessinée sur le côté droit du plateau, opération faite seulement sur la machine M1.

La configuration β est un point dessiné sur le côté droit du plateau, opération seulement disponible sur la machine M1 également.

La configuration γ est une ligne tracée sur le côté gauche du plateau, cette opération est faite sur la machine M2. Pour réaliser l'opération sur M2, le produit doit d'abord passer devant M1 et emprunter le chemin de bouclage pour atteindre M2. C'est pourquoi il y a un temps de transfert égal à 10 secondes.

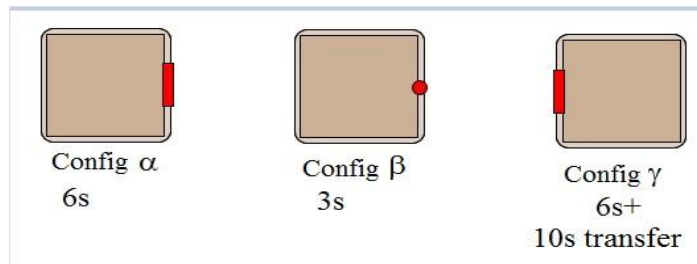


Figure 32 Liste des configurations

Les produits sont groupés en 3 commandes, ayant le même délai :

- Commande rouge : $\alpha\beta\gamma$
- Commande verte : $\alpha\beta\gamma$
- Commande bleue : $\alpha\beta\beta\gamma$

Afin de correspondre au processus d'Acta, dès qu'un produit d'une commande sort de la zone B, les produits suivants qui n'appartiennent pas à la même commande vont rester en attente jusqu'à ce que le reste de la commande en cours passe. Pour valider l'apport de la proposition, nous l'avons comparé à deux autres règles de pilotages :

- Chaque commande est lancée entière, l'une après l'autre, sans chercher à faire la moindre optimisation au niveau des opérations réalisées. Le seul facteur considéré ici est de garantir qu'une commande passe entièrement. Nous l'avons nommé « commande par commande »
- L'ordonnancement est fait en limitant le nombre de changement d'opération (temps de réglage). Ce scénario a été nommé "opération par opération"
- Notre proposition d'optimisateur utilisant l'algorithme génétique (établir un compromis entre les temps de réglages et le WIP. Dans le cas présent, le WIP est estimé par le nombre de bouclage). Celui-ci est simplement nommé « GA planning »

Un planning a été élaboré pour chaque règle de pilotage avant lancement :

- Planning 1 (commande par commande) : $\alpha\gamma\beta\beta\gamma\alpha\alpha\beta\beta\gamma$
- Planning 2 (opération par opération) : $\alpha\alpha\alpha\beta\beta\beta\beta\gamma\gamma\gamma$
- Planning 3 (GA planning) : $\beta\beta\beta\alpha\alpha\gamma\gamma\alpha\beta$

Le **Tableau 16** montre les résultats obtenus par rapport aux quatre aspects étudiés suivants :

- 1) Le nombre de boucles qui ont été nécessaires pour suivre la règle de pilotage : tant que tous les produits d'une même commande ne sont pas tous sortis de la zone B, les autres doivent boucler dans cette zone. Cela revient à mesurer ce que nous considérons comme le « work in progress »
- 2) Le makespan
- 3) Le nombre maximal de commandes en cours de réalisation simultanément. La présence des trois commandes avec chacun une couleur sert à représenter la notion de contremarque introduite dans le chapitre précédent. Ce critère est important car l'un des objectifs principaux de notre optimisateur est de minimiser le nombre de contremarque en cours de réalisation. En effet, plus il y a de contremarque en cours plus grande sera la file d'attente en zone B (équivalent au problème d'espace au sol nécessaire dans la phase de tri pour l'entreprise)
- 4) Le nombre de changement d'opérations, chaque changement d'opération est à rapprocher d'un temps de réglage.

(4) Les résultats

La première partie des tests a été réalisée sans introduire de probabilité de réparation. De par le fait, aucune source de variabilité n'a été introduite. C'est pourquoi durant cette partie des expériences nous n'avons donc réalisé que deux jeux d'expériences (le second servant de phase de contrôle).

	Commande par commande	Opération par opération	GA planning
Nombre de boucle	2	8	3
Makespan	5min 08 s	4 min 44 s	4 min 56 s
Commandes simultanées max	1	3	2
Changement d'opération	6	2	4

Tableau 16 Résultats d'expérience sans réparation

Durant cette première phase, il est évident que l'ordonnement "commande par commande" permet d'atteindre le plus faible nombre de boucle d'attente mais en contrepartie il a le nombre de changement d'opération le plus important. Il est aussi le plus long à se finir car le changement d'opération est le facteur le plus impactant sur la durée de traitement globale bien qu'il assure de ne travailler qu'une seule commande à la fois.

À l'opposé, la règle "opération par opération" minimise le nombre de changement d'opération mais fait exploser le nombre de boucle et le nombre de contremarque produite en même temps. L'algo génétique proposé a l'avantage de fournir un compromis idéal entre les deux solutions précédentes.

Dans le **Tableau 17**, nous présentons les résultats obtenus avec un taux de réparation de 20% introduit. Pour cette partie, dix jeux d'expériences ont été fait et les résultats moyens sont affichés, les valeurs qui suivent les « + » dans le tableau correspondent aux effets des boucles de réparations.

	Commande par commande	Opération par opération	GA planning
Nombre de boucle	3	12	5
Makespan	9 min 38 s	9 min 13 s	9 min 21 s
Commandes simultanées max	2	3	2
Changement d'opération	6+5	2+4	4+4

Tableau 17 Résultats d'expérience avec un taux de réparation de 20%

En comparaison avec la première partie des expériences, on peut remarquer que les résultats sont très semblables, à ceci près que les temps de production ont presque doublé. L'algorithme génétique semble robuste face aux réparations et propose toujours un bon compromis.

La mesure de la dispersion du « makespan » est montrée sur la **Figure 33**. À travers cette boîte à moustache, nous pouvons constater que le pilotage « commande par commande » est celui ayant la plus faible dispersion mais avec des résultats globalement plus mauvais que les autres. GA a la plus forte dispersion mais aussi les résultats les plus intéressants. Cela peut s'expliquer par le nombre d'itération de l'algorithme génétique, qui est limité pour maintenir le temps de calcul à une durée acceptable. Augmenter ce nombre devrait permettre de limiter le taux de dispersion et augmenter la médiane. En évaluant le temps de calcul admissible, on pourrait chercher à se rapprocher du maximum admissible pour améliorer son efficacité. De plus, en voyant que la jonction entre les médianes n'est pas linéaire, on peut conclure que le choix de la stratégie a un réel impact sur le « makespan ».

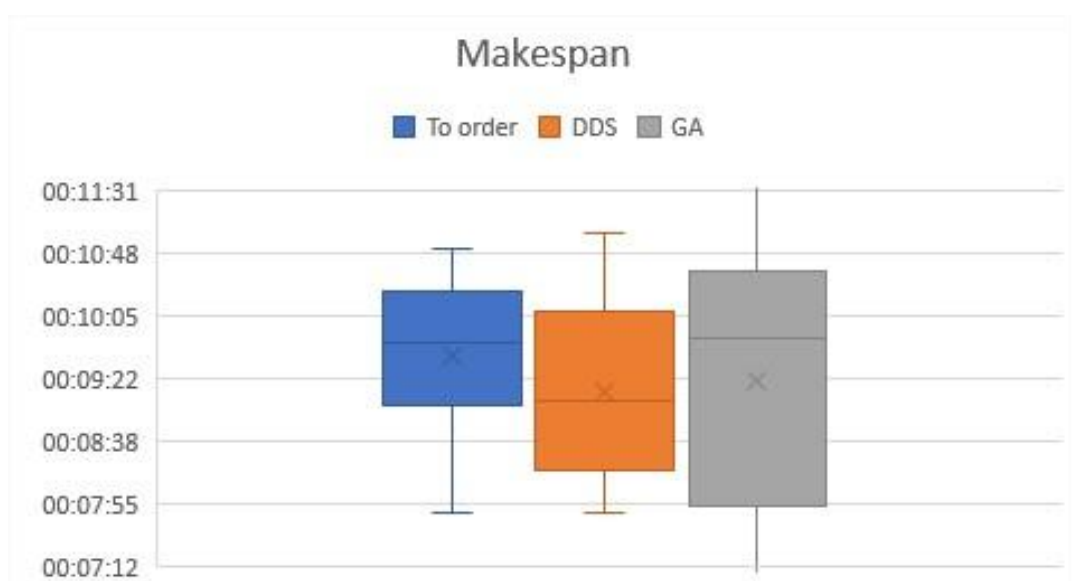


Figure 33 Mesure de la dispersion du makespan

Cette partie résume l'ensemble des travaux que nous avons mené jusqu'à présent sur cet ordonnanceur, la partie suivante est consacré aux travaux relatifs au second ordonnanceur que nous avons choisi d'étudier, celui du laquage.

D. Ordonnanceur dynamique pour le robot de laquage

1. Description du cas d'étude

Dans cette partie, nous étudions les résultats obtenus avec l'application de notre optimisateur proactif dans le cas du robot de laquage de l'entreprise Acta. Ce robot doit alimenter deux flux de production distincts : le flux des pièces brillantes, devant passer au polissage avant de rejoindre le second flux, le flux satiné au poste d'expédition. Le but est que les produits des deux flux issus d'une même commande arrivent en même temps au poste des expéditions. De plus, le robot de laquage est construit de telle manière qu'il est impossible de contrôler la qualité des produits réalisés avant que ceux-ci ne sortent du four de séchage, soit après une durée minimale de sept heures (**Figure 34**).

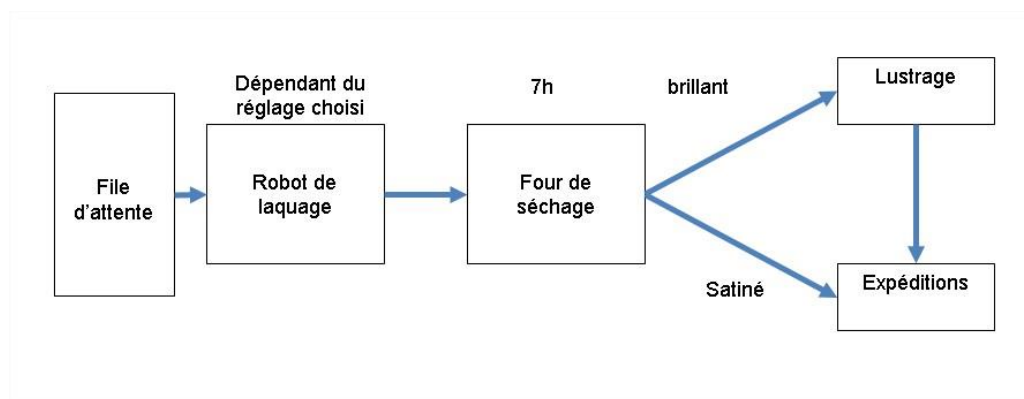


Figure 34 Workflow du robot de laquage Acta-mobilier

Comme dit précédemment Acta-mobilier souffre d'un taux de réparation très élevé, supérieur à 30 %, causé par le très haut niveau de qualité requis par ses clients. Afin de réduire ce taux, une solution envisageable est de réduire les problèmes qualité. Cependant il est très complexe d'améliorer les processus actuels déjà aux limites techniques connues et le coût qu'engendrerait l'achat d'outils à la pointe de la technologie est non-absorbable par l'entreprise. Une autre option est de faire appel à des techniques de prédiction d'apparition de non-qualité en analysant tous les facteurs pouvant influencer sur la qualité. Sur un poste de travail, tel que ce robot de laquage, où la quantité de produit à travailler à la suite doit être suffisante pour être rentable (au minimum 2.5m²), éviter la production de non-qualité est un enjeu crucial. Si un défaut qualité survient sur un produit, en général, c'est l'ensemble du lot qui est impacté par ce défaut. Pris seul, cela est déjà un problème important. Pourtant, il y a encore plus problématique : ce défaut ne peut être détecté qu'après le temps de séchage de sept heures. Certains types de défauts qualité ne demandent qu'une simple retouche faite par l'opérateur à la sortie de la machine mais d'autres, correspondant aux cas les plus fréquents, demandent un re-laquage complet des pièces (induisant un nouveau passage dans le robot et le four de séchage, soit un ajout d'au moins sept nouvelles heures au temps de process initiale). D'autant plus que la qualité du process de laquage à ce poste est contrainte par de nombreux facteurs dont certains qui sont incontrôlables tels que l'humidité ou la pression atmosphérique. Le réseau de neurones décrit dans (Noyel et al., 2016) a été élaboré afin d'analyser tous ces facteurs et de prédire la probabilité d'apparition d'un défaut qualité pour, par la suite, proposer un réglage différent du robot pouvant réduire ce risque. Notre proposition utilisera plusieurs instances de ce réseau neuronal pour réaliser une optimisation proactive. Le paragraphe suivant décrit le modèle de simulation que nous avons construit pour étudier le problème du robot de laquage et tester la proposition que nous suggérons.

2. Modèle de simulation du robot de laquage

Ce modèle, écrit en Python, représente le robot de laquage et sa file d'attente composée de lots dont le nombre est modulable mais fixé à 5 dans un premier temps, les produits présents dans cette file d'attente sont ceux qui vont participer à la prise de décision. En effet, si le premier lot de la file d'attente présente un risque de non-qualité trop élevé (seuil fixé à 25% dans les premières expériences), les 5 lots présents dans la file vont entrer en interaction pour élire le meilleur remplaçant. À chaque fois, qu'un lot a fini d'être produit sur le poste de travail, un test de qualité est effectué, si le produit est bon, il sort du système et un nouveau produit entre dans la file d'attente, s'il est mauvais, il est remis dans la file d'attente. L'ordre dans lequel les produits entre dans la file d'attente est déterminé par un ordonnancement prédictif considéré comme donnée d'entrée du modèle. Sur la **Figure 35**, le scénario suivi est illustré : les lots de B1 à B5 sont triés suivant l'ordonnancement prédéfini de telle sorte que B1 est le lot qui doit être réalisé en premier si aucun risque de non-qualité n'est détecté. Dès qu'un lot est prêt à être produit, le robot se paramètre sur le programme qui lui correspond et réalise l'opération. Le temps opératoire est égal à la surface à réaliser multipliée par un facteur dépendant du programme choisi. Le lot quitte ensuite la machine avec un temps restant avant dépassement de la due-date réduit de sept heures (pour représenter le temps d'attente dans le four de séchage). En sortie, une fonction aléatoire est déclenchée pour déterminer si le lot est défectueux ou non. Cette fonction est pondérée par le résultat obtenu lors de l'évaluation du risque de non-qualité déterminé avant le passage du lot dans la cabine de peinture (en effet plus le risque est faible, plus faible est la probabilité que le produit sorte non conforme). S'il est déterminé non conforme, il retourne dans la file d'attente imputé d'un temps restant plus faible (le temps opératoire plus les sept heures de séchage).

Sinon, il sort du modèle en incrémentant la quantité à produire dans la file d'attente de l'un des deux postes aval (le polissage pour les lots brillant et les expéditions pour les lots satinés). Les cadences de traitement des deux postes ont été intégrées dans le modèle pour que les quantités présentes dans les files d'attente soient prises en compte en temps réel. Juste avant son passage dans le robot, le premier lot de la file déclenche son réseau de neurone afin d'évaluer son risque de non-qualité, s'il est supérieur au seuil sélectionné la procédure de sélection du meilleur remplaçant est amorcée. L'algorithme décrit dans le chapitre trois est alors effectué afin de déterminer quel est le lot qui correspond le mieux aux critères d'optimisation défini ci-dessous :

- Le risque de non-qualité.
- Le ratio critique.
- L'équilibrage de la répartition des charges entre les différents postes clients.
- Les temps de réglages induits par le changement de lots.
- L'impact sur la nervosité du système (ce qui se représente par le nombre de modifications apportées à la planification originale), ce dernier critère est présent pour faire en sorte de ne pas perturber toute la logistique déjà mise en place au poste de travail (approvisionnement en laque).

La **Figure 36** présente le scénario dans lequel le réseau neuronal de B1 a déterminé un risque trop important amenant à un réordonnancement qui a choisi B3 comme nouveau lot à passer prioritairement. Les variables du modèle sont les suivantes :

- Relatifs à la gamme de production de chaque lot :
 - La surface du lot, exprimée en m².
 - Le programme, un nombre désignant le programme sur lequel le robot doit se régler pour réaliser le lot.
 - Le temps restant avant le dépassement de la due-date, exprimé en heure.
 - Le nombre de passage, servant à comptabiliser le nombre de fois qu'un lot est réalisé par le robot.
- Relatif aux critères :
 - Chaque critère est pondéré par un facteur variant entre 0 et 1.
- Relatif au poste de travail :
 - Les quatre programmes connus par la machine ainsi que leur cadence de réalisation, exprimée en heure par m².
 - Les deux temps de réglages possibles :
 - Le premier s'applique si, au changement de lots, les finitions sont identiques.
 - Le second pour le cas contraire.
- La probabilité d'apparition est l'éventualité qu'un défaut soit constaté après réalisation.

Toutes ces variables sont redéfinies dans le **Tableau 18**.

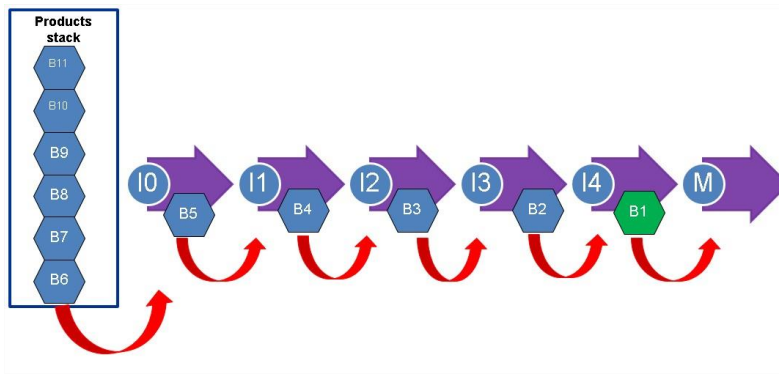


Figure 35 Fonctionnement standard

Produits		
Surface	Réel	[2,5 ;40]
Programme	Entier	1 : 1 face brillante
		2 : 2 faces brillantes
		3 : 1 face satinée
		4 : 2 faces satinées
Due date		[0 ;50]
Nombre de passage		[1 ;10]
Critères		
Poids	Réel	[0 ;1] =>choisi égal à 1 pour les expériences suivantes
Machine		
Programme 1	h/m ²	0.28
Programme 2		0.04
Programme 3		0.03
Programme 4		0.02
Réglage sans changement de finition	h	0.03
Réglage avec changement de finition	h	0.05
Modèle		
Taux d'apparition de défaut	Réel	[0;100]

Tableau 18 Liste des variables

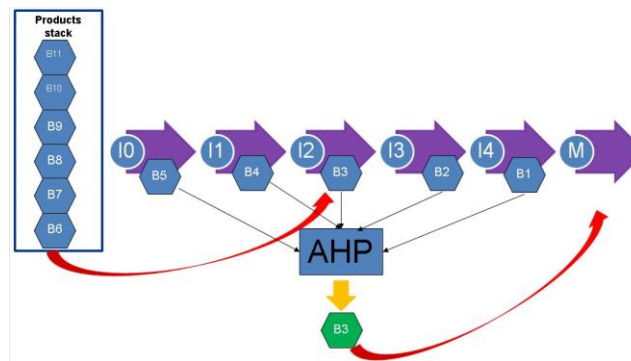


Figure 36 Si un risque de non-qualité est détecté

a) *Expérimentations par simulation*

Afin de valider l'intérêt de la solution que nous proposons, des simulations ont été réalisées sous plusieurs conditions distinctes, avec et sans utilisation de la prise de décision dynamique par AHP. Pour chacun des cas étudiés, des jeux de 10 simulations ont été faits. Les valeurs moyennes et les écarts type sont présentés dans les tableaux de résultats suivants. Afin de tirer des conclusions sur l'apport de la méthode un indicateur coût global a été mesuré. Il s'exprime en euro et est défini comme la somme des trois sous-indicateurs suivants :

- Le retard, qui représente le nombre de lots dépassant leur due-date multiplié par un coût de pénalité. Il s'agit là d'un moyen approximatif afin d'exprimer les coûts engendrés par la livraison en retard des commandes, coût complexe à représenter car un produit en retard a un impact fort sur l'image de marque de l'entreprise avec diminution de la confiance que lui porte le client et aussi parfois l'affrètement d'un camion supplémentaire.
- Le coût de réparation qui est le coût induit par le fait de retravailler un lot sur le robot (ce coût a été repris directement de celui calculé dans l'entreprise).
- Le coût en main d'œuvre qui est calculé en multipliant le makespan par le cout horaire de deux opérateurs (nombre de personne minimum nécessaire pour faire fonctionner le robot de laquage).

Dans un premier temps, notre objectif était de comparer notre proposition a une méthode classique d'ordonnancement pour justifier son intérêt. Chacune des simulations réalisées sont décrites ci-après :

- Le premier cas étudié est un jeu de données tiré de la production réelle du robot de laquage sans qu'aucune optimisation ne soit appliquée. La production d'une équipe (8 heures de travail) a été sélectionnée.

Real case	Lateness		Rework		Workforce cost		global cost	
	mean	std	mean	std	mean	std	mean	std
FIFO	1541.0	1510.2	9718.8	6563.1	107.1	23.5	11366.9	7884.5
AHP	91.4	193.1	3884.4	3245.1	80.8	9.2	4056.5	3310.6

Tableau 19 Cas réel

Ce tableau met en évidence le fait que même pris séparément, la méthode proposée offre une amélioration de chacun des coûts par rapport à un fonctionnement standard suivant une règle FIFO. Le coût de réparation est 61% plus faible qu'avec une règle FIFO pour une diminution sur le global de 64 %.

- Ensuite une simulation où la charge est équitablement répartie entre le brillant et le satiné est testée afin de déterminer l'apport lorsque l'on se place dans le cas le plus favorable vis à vis du critère d'équilibre.

Lorsque le jeu de données est déjà ordonnancé de manière à créer un équilibre parfait entre les deux flux de produits à alimenter. La méthode AHP offre une amélioration de 73 % par rapport au coût de réparation pour un taux global de 75 %.

Equal distribution	Lateness		Rework		Workforce cost		global cost	
	mean	std	mean	std	mean	std	mean	std
FIFO	2935.4	2427.5	13494.0	7713.3	118.8	25.6	16548.2	10003.4
AHP	385.7	441.5	3666.0	2238.1	81.5	8.5	4133.1	2464.3

Tableau 20 Répartition égale

- Ensuite les deux tests suivants ont été faits pour déterminer si la solution reste efficace si on met en défaut l'un des critères de décision. En ne pouvant choisir qu'entre des produits allant vers le même flux de sortie (uniquement brillant ou uniquement satiné), on met en défaut le critère d'équilibre des charges tout en optimisant le coût en réglage.

Face à cette situation, l'amélioration obtenue par l'AHP est en moyenne de 72 % par rapport au FIFO.

Only brilliant	Lateness		Rework		Workforce cost		global cost	
	mean	std	mean	std	mean	std	mean	std
FIFO	2501.3	2560.2	11544.0	6955.2	115.7	24.5	14160.9	9468.1
AHP	60.0	140.7	2964.0	1594.3	78.1	8.8	3102.1	1674.1
Only mat	Lateness		Rework		Workforce cost		global cost	
	mean	std	mean	std	mean	std	mean	std
FIFO	1677.5	2482.0	12394.0	8303.2	102.5	18.5	14173.9	10727.0
AHP	101.3	176.0	4680.0	2651.5	73.2	6.6	4854.5	2682.3

Tableau 21 Cas de non distribution (que brillant et que satiné)

- La simulation suivante teste les résultats lorsque tous les lots ont la même échéance. Lorsque tous les lots ont la même due-date, les bénéfices apportés par l'AHP se font presque exclusivement sur le coût de réparation, dans ce cas précis, l'amélioration est seulement de 18 %.

Same due date	Lateness		Rework		Workforce cost		global cost	
	mean	std	mean	std	mean	std	mean	std
FIFO	68.1	215.4	5959.2	4857.2	84.9	9.8	6112.2	4892.2
AHP	0.0	0.0	4929.6	4026.5	84.9	9.8	5014.5	4033.3

Tableau 22 Tous les produits ont la même échéance

- La non prise en compte du risque de non-qualité déterminé par le réseau de neurones a aussi été évaluée pour déterminer si même sans l'aspect prédictif l'AHP avait un impact positif sur la production.

Dans cette situation précise l'AHP fournit une solution qui est 32 % plus faible en coût de réparation mais 5 fois plus mauvaise en coût de retard.

No risk of non-quality	Lateness		Rework		Workforce cost		global cost	
	mean	std	Mean	std	mean	std	mean	std
FIFO	57.3	181.2	5545.8	5101.9	86.4	13.4	5689.5	5252.9
AHP	284.6	381.5	3775.2	1281.2	82.0	7.9	4141.7	1454.0

Tableau 23 Sans prise en compte du risque de non qualité

Pour synthétiser, nous pouvons conclure que la proposition semble apporter un réel intérêt par rapport à une règle de pilotage simple comme le FIFO (utilisée à ce jour dans l'entreprise). Nous sommes, a minima, convaincu que sa mise en application pourra réduire le taux de non-qualité produite par le robot de laquage. Il est à noter, qu'aucun réglage particulier n'ait été fait sur l'algorithme, pourtant les pistes d'amélioration sont nombreuses : en effet, nous pouvons citer le réglage des poids pour chacun des critères ou encore la variation du nombre d'entités à participer à la prise de décision comme pistes d'amélioration.

b) Améliorations

Nous sommes conscients que les performances de la méthode proposée sont étroitement liées aux paramétrages de celle-ci (ex les poids des critères et en particulier le nombre d'acteurs du processus de décision). Nous avons donc cherché à faire varier ce que nous appellerons la myopie sociale.

Pour être plus précis, nous avons fait varier le nombre de produits de la file d'attente prenant part à la prise de décision afin de comprendre de quelle manière le nombre d'entités participant à la décision pouvait influencer en bien ou en mal sur le résultat. Pour se faire, nous avons choisi de relancer les runs de simulation sur le jeu de données correspondant à une vraie semaine de production du robot de laquage, en fixant les poids de tous les critères à 1 et ne faisant varier que le nombre de produits intervenant dans la prise de décision. Nous avons réalisé les simulations pour les valeurs suivantes : 1,3,5,7,10 et 20.

Nous avons pris comme bornes inférieure et supérieure les valeurs 1 et 20 afin d'obtenir des valeurs extrêmes, même si bien évidemment celles-ci sont complètement hors propos. En effet, ne laisser qu'un seul produit communiquer revient à ne pas faire intervenir le système de prise de décision. Tant que, dans l'entreprise, le robot de laquage ne fonctionnera pas en flux tiré avec son poste en amont, cette situation se produira très rarement. De même, il est hautement improbable pour ce poste de travail d'avoir une file d'attente supérieure à 10 à moins d'un engorgement extrême causé par une panne machine.

Les **Figure 37** et **Figure 38** synthétisent les résultats obtenus pour les différents cas cités précédemment. Conformément à ce que nous espérions, lorsqu'un seul produit intervient dans la décision, les résultats sont les plus médiocres. Ensuite nous pouvons noter qu'à partir de 5 les résultats moyens stagnent alors que les temps de calcul s'agrandissent comme le montre la **Figure 39**.

Le choix de faire communiquer cinq produits pour la prise de décision en conditions standards de fonctionnement nous semble donc à la vue des résultats et ayant connaissance de l'évolution du flux habituel à l'entrée du robot de laquage être le plus pertinent. Pour mieux analyser les résultats de nos expérimentations, nous avons également réalisé une analyse de la variance à une dimension (ANOVA) pour mesurer l'effet du changement de modalité de « la myopie » sur le nombre de reworks, le retard cumulé et le temps de calcul. Les hypothèses devant être vérifiées pour réaliser ce test sont les suivantes :

- Un seul facteur de variabilité (ici la myopie) à 6 modalités (1, 3, 5, 7, 10, 20).
- Tirages aléatoires et indépendant.
- Observations supposées sans erreur.

Un test ANOVA consiste à comparer deux modalités différentes du facteur. Il génère la p-value qui doit être comparée à la valeur du risque que nous souhaitons (nous le prendrons égal à 5% ici).

Les deux hypothèses sont :

- H0 : les deux modalités considérées du facteur étudié ont le même impact sur la variable

Et son opposée :

- H1 : les deux modalités considérées du facteur étudié ont un impact différent

Le résultat du test s'obtient en comparant la p-value à la valeur du risque α choisi :

- Si p-value est supérieur au risque α ($p > 0,05$) alors H0 est acceptée avec un risque α
- Si p-value est inférieur au risque α ($p < 0,05$) alors H0 est rejetée (par extension H1 acceptée) avec un risque α .

Les résultats du test sont présentés dans les **Tableau 24** à **Tableau 26** :

Si l'hypothèse H0 est rejetée alors la case est en rouge. Par contre, si l'hypothèse H0 est acceptée, la case est en vert.

Pour le **Tableau 24**, correspondant à l'impact de la myopie sur le nombre de reworks, nous pouvons remarquer que H0 est rejeté systématiquement quand les modalités 1 et 3 sont en jeu. Cela signifie que choisir l'une de ces deux modalités, a un réel impact sur le résultat. Quand on compare les modalités 5, 7, 10 et 20 ensemble, H0 est accepté systématiquement (à l'exception du cas 5-10 qui donne tout de même une valeur proche du seuil). Par conséquent, choisir entre les modalités 5, 7, 10 et 20 n'a pas un impact différent sur le nombre de reworks. Ainsi, fixé cette modalité à 5 semble donc le meilleur compromis vis-à-vis du nombre de reworks.

Nombre de reworks	Myopie	1	3	5	7	10	20
	1	-	0	0	0	0	0
3		-	0	0	0	0	0
5			-	0.1691	0.0457	0.1755	
7				-	0.4085	0.8351	
10					-	0.6017	

Tableau 24 Résultats test ANOVA pour les reworks

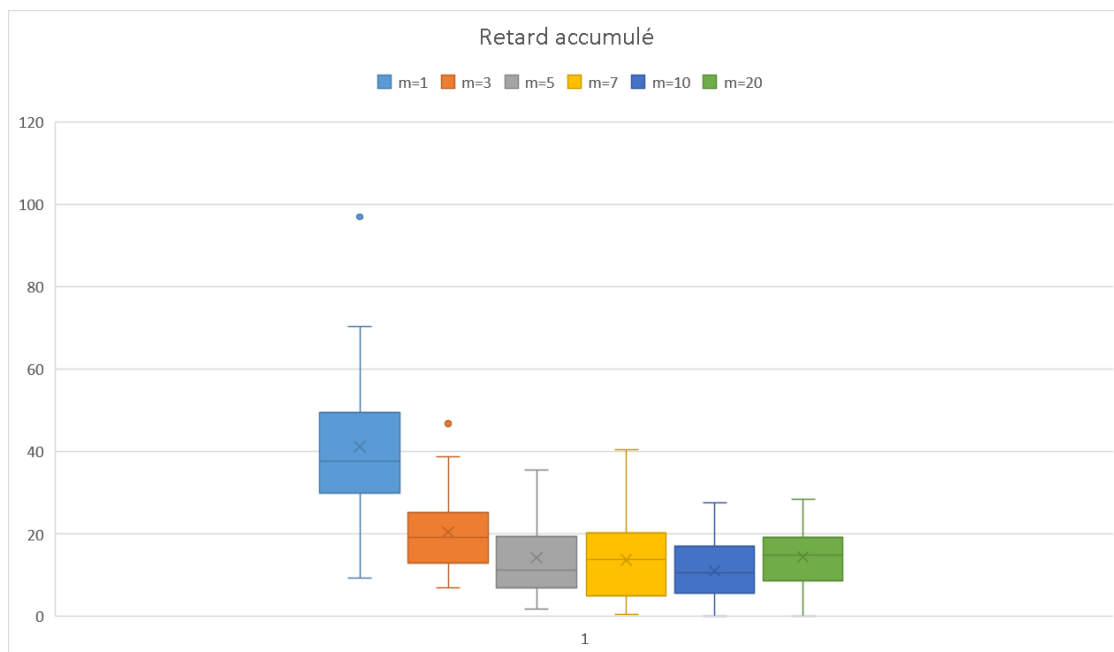


Figure 37 Impact de la myopie sur le retard

Pour le **Tableau 25**, correspondant à l'impact de la myopie sur le retard, nous pouvons remarquer que H0 est rejeté systématiquement quand les modalités 1 et 3 sont en jeu. Cela signifie que choisir l'une de ces deux modalités, a un réel impact sur le résultat. Quand on compare les modalités 5, 7, 10 et 20 ensemble, H0 est accepté systématiquement. Par conséquent, choisir entre les modalités 5, 7, 10 et 20 n'a pas un impact différent sur le retard. Ainsi, fixé cette modalité à 5 semble également être le meilleur compromis par rapport au retard.

	Myopie	1	3	5	7	10	20
Retard cumulé	1	-	0	0	0	0	0
	3		-	0.0148	0.0079	0.0001	0.0100
	5			-	0.8122	0.1562	0.9441
	7				-	0.2431	0.7420
	10					-	0.0996

Tableau 25 Résultats test ANOVA pour le retard

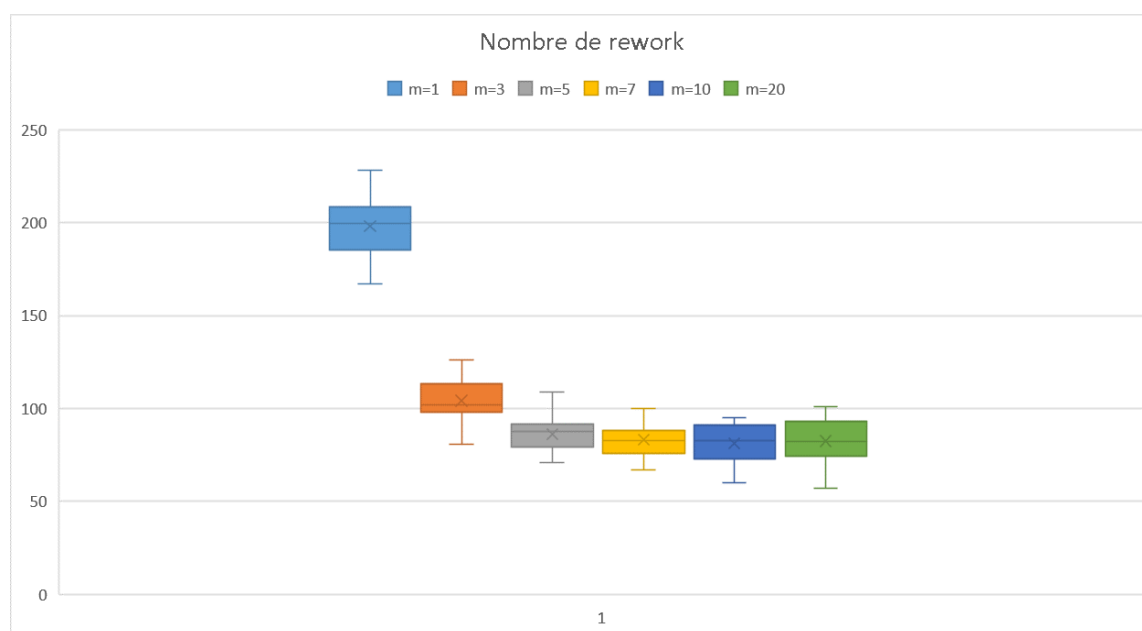


Figure 38 Impact de la myopie sur le nombre de reworks

Pour le **Tableau 26**, correspondant à l'impact de la myopie sur le temps de calcul, nous pouvons remarquer que H0 est rejeté pour toutes les combinaisons à l'exception des deux couples 1-3 et 5-7. Ce résultat était prévisible, il est normal que plus le nombre d'acteurs intervenant dans le calcul est important, plus celui-ci demande du temps.

	Myopie	1	3	5	7	10	20
Temps de calcul	1	-	0.1296	0	0	0	0
	3		-	0	0	0	0
	5			-	0.4788	0	0
	7				-	0	0
	10					-	0

Tableau 26 Résultats test ANOVA pour le temps de calcul

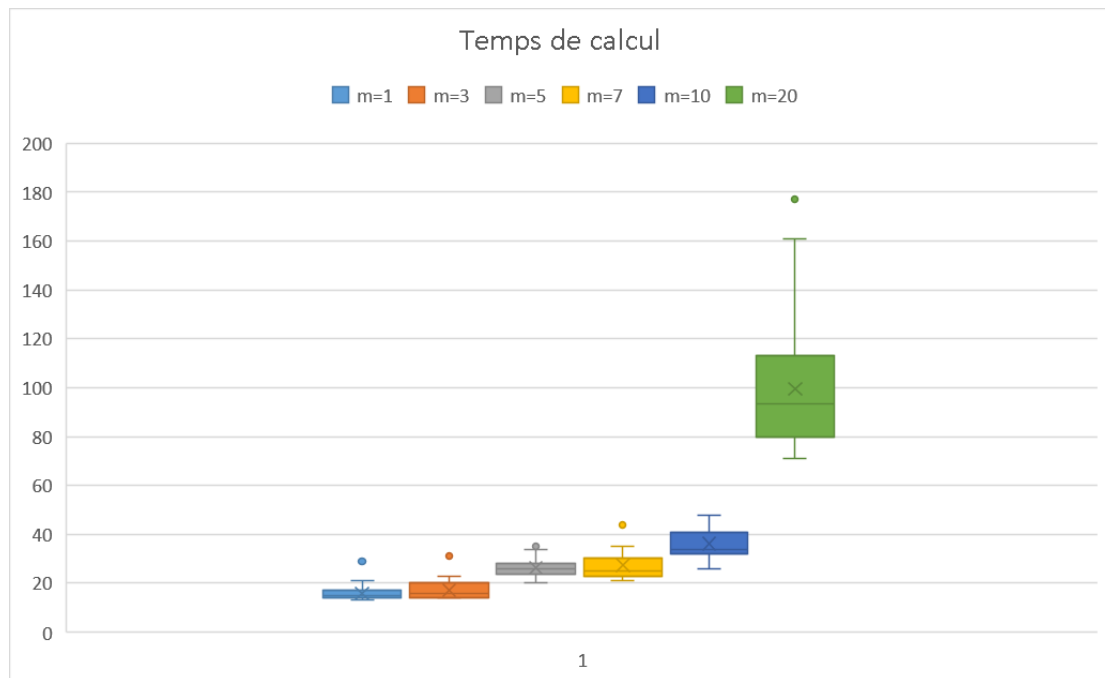


Figure 39 Impact de la myopie sur le temps de calcul

Ensuite, nous avons décidé de réaliser plusieurs tests en modifiant les poids des différents critères afin de déterminer quelle combinaison permettrait d'obtenir les meilleurs résultats. 8 cas de figures ont été comparés : le cas qui a servi à faire toutes les expériences précédentes : à savoir mise de tous les critères à 1, passage un à un de chaque critère à la valeur 10. Ensuite, nous avons essayé de placer les couples [prédiction de la qualité (NN) et due-date] à 10, [NN et équilibre des charges] à 10 et enfin le triplet [NN, due-date et équilibre des charges] à 10. Les **Figure 40** et **Figure 41** montrent les résultats obtenus pour chacun des cas de figure cités précédemment. Le cas d'étude NN = 10 donne le résultat le plus optimal tant d'un point de vue moyenne des valeurs que répartition des valeurs autour de celle-ci et cela aussi bien par rapport au temps de retard que par rapport au nombre de réparation. Nous sélectionnerons donc cette configuration pour la suite des tests.

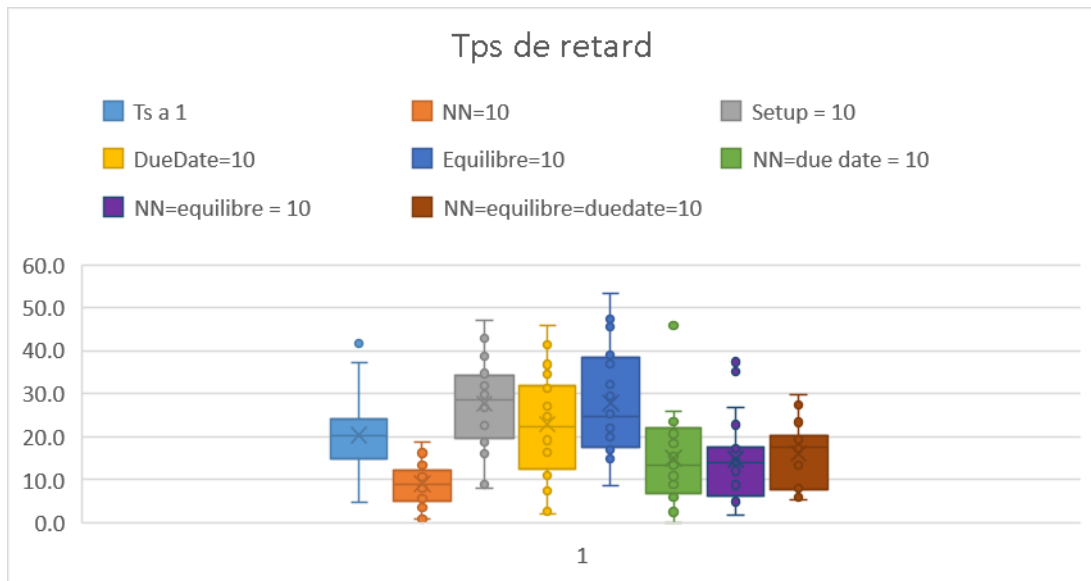


Figure 40 Comparaison des temps de retard

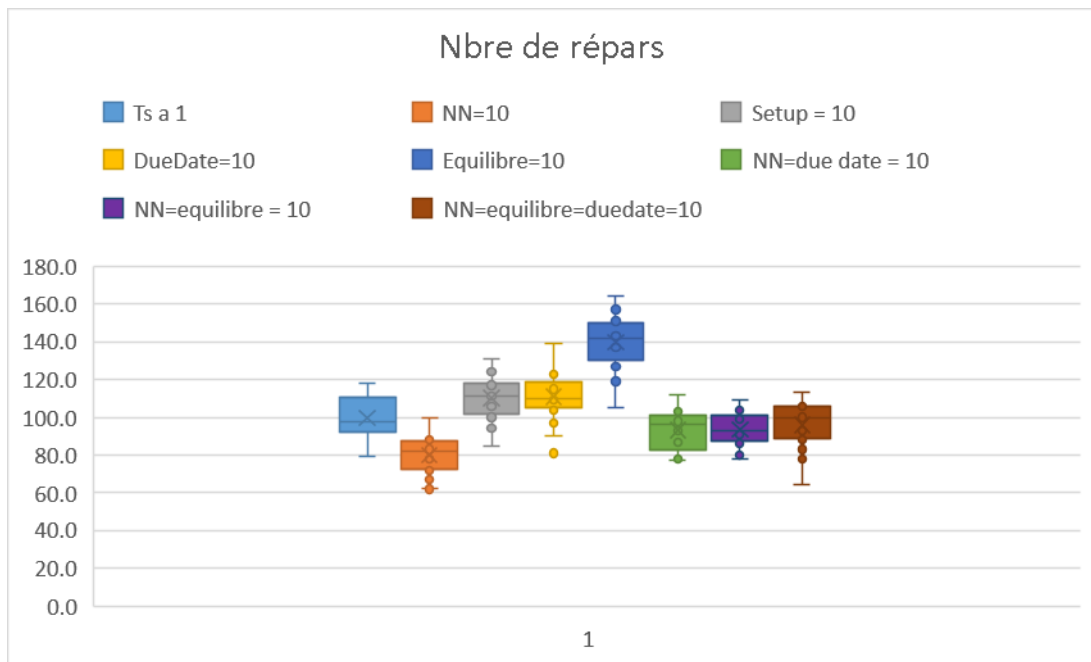


Figure 41 Comparaison du nombre de réparations

Ainsi s'achèvent les travaux que nous avons menés jusqu'alors sur cette problématique. La partie suivante récapitule l'état d'avancement des travaux à ce jour, ce qu'il nous reste encore à traiter, ainsi que les perspectives d'évolution que nous imaginons pour l'entreprise vis-à-vis de la problématique que nous avons posé.

V. Conclusions et perspectives

Cette thèse, s'inscrivant dans la continuité directe de celle de Mme Noyel, avait pour objectif la mise en application d'une architecture de pilotage hybride s'appuyant sur le contrôle par le produit. Nous avons choisi de nous inspirer d'un méta-modèle développé au sein du CRAN. Cette architecture, reposant sur un modèle bio-inspiré de type VSM, se caractérise par une structure pyramidale où chaque niveau est susceptible de prendre des décisions à son échelle. Le plus haut niveau, est le système d'information considéré dans son entièreté. À son niveau, il supervise toutes les décisions tactiques (établit, par exemple, le plan directeur de production qui servira de cadre de référence aux autres niveaux). Le plus bas niveau « d'intelligence » est distribué entre les lots élémentaires dotés de moyen de communication et d'analyse (leur intelligence est attribuée à la réactivité du système car se trouvant au plus proche du besoin). En niveaux intermédiaires, nous trouvons :

- Des optimisateurs réactifs pour superviser les sous-ateliers de la chaîne de production. Ceux-ci instancient des regroupements temporaires de lots élémentaires en lots de fabrication de manière à atteindre les objectifs de consommation matière, de temps de réglages ou de productivité du sous-atelier qu'ils supervisent.
- Des optimisateurs, plus locaux, au fonctionnement proactif (au sens où ils agissent sur l'ordonnancement d'un poste de travail et/ou sur un horizon temporel plus restreint) dont le calcul est effectué en « centralisé » mais faisant intervenir directement les lots et les informations qu'ils possèdent dans la prise de décision.

Durant nos travaux de recherche, nous avons concentré nos efforts sur un optimisateur de chacune des deux catégories citées ci-dessus, selon les besoins définis par l'étude du flux de production de l'entreprise.

Ainsi, un optimisateur réactif a été réalisé pour l'atelier nommé « usinage » dans lequel les opérations relatives au débit et au façonnage des formes des produits sont effectuées. L'atelier a été déterminé comme prioritaire car il est amené à être ponctuellement le goulot d'étranglement de la chaîne de production. Cet optimisateur est construit à l'heure actuelle de manière à réaliser un regroupement des lots élémentaires en lots plus importants et à les ordonner pour obtenir un compromis acceptable entre minimisation des consommations matières, des temps de réglage des machines de débit et la minimisation des encours. De plus, le séquençement doit garantir que les regroupements, lors de leur re-division puissent permettre de reformer rapidement les commandes clients. Cette contrainte est nécessaire, avant passage à la phase d'application de la finition. Il met en œuvre un algorithme génétique solutionnant un problème de séquençement et d'ordonnement multicritères. Pour valider notre choix d'une méta-heuristique comme méthode de résolution du problème, nous avons également tenté de le résoudre par une méthode de mathématique analytique et les résultats obtenus ont permis de confirmer que notre décision était raisonnable. Ce programme d'optimisation a été testé sur plateforme de test, dans un cas d'étude approchant le cas réel et a fourni des résultats encourageants. Une implémentation a été faite dans l'entreprise et est utilisée, chaque semaine, pour la planification d'un des clients majoritaires. Son utilisation n'a pas été généralisée car son mode de fonctionnement actuel, visant à respecter la contrainte de reconstruction des commandes avant application n'est pas un service qualité requis par l'ensemble des clients. Un switch de fonctionnement devrait être ajouté pour ne pas tenir compte de cette contrainte lorsqu'elle n'est pas requise. Cependant, un travail de fond sur le terrain (méthodes de fonctionner et de penser) est également nécessaire pour atteindre cet objectif. Ce travail a été initié durant la thèse et continue encore aujourd'hui. En outre, l'optimisateur n'a pas comme propriété, pour le moment, d'effectuer à la volée des réordonnements, mais il s'agit là d'une piste de perfectionnement qu'il nous resterait à étudier.

Un optimisateur de la seconde catégorie a été étudié pour gérer le cas du robot de laquage, celui-ci pouvant également se retrouver poste goulot dans certaines situations. Il doit fournir deux postes clients traitant chacun leurs propres familles de produits mais devant être expédiées sous les mêmes délais. Dans les critères faisant partie du réordonnancement, nous prenons également en compte certaines caractéristiques de cette machine. En effet, chaque changement de couleur implique une consommation importante de réalimentation. Il y a également la longue durée d'attente avant de pouvoir visualiser les produits après application, et ainsi, savoir si un défaut qualité est apparu. Le cas échéant, les produits sont susceptibles de devoir refaire un cycle complet de laquage. Les facteurs pouvant influencer sur l'apparition d'un défaut sont nombreux et de tous types. Ces considérations ont conduit les travaux de Noyel sur les réseaux de neurones pour élaborer un modèle d'apprentissage permettant de prédire le risque d'apparition de non-qualité. L'optimisateur utilise ce modèle afin d'évaluer les risques de non-qualité du prochain lot à produire et si celui-ci est jugé trop élevé, un processus de réordonnancement est déclenché choisissant parmi les lots présents en file d'attente, le plus adapté en considérant : le risque de non-qualité, l'impact sur l'équilibre entre les postes clients et l'impact sur la machine du changement de lot). Cet optimisateur de poste de travail recueille des informations de la machine, des produits en file d'attente et des files d'attentes des postes en aval afin d'empêcher l'apparition d'un problème. Il a été implémenté sur un modèle de simulation et a aussi été transposé sur la même plateforme de test que l'optimisateur précédent. Les résultats obtenus nous amènent à penser que son implémentation dans l'entreprise, pourra améliorer le rendement de ce poste et fluidifier le flux de production. Le temps nous a manqué pour la mise en application de cet optimisateur dans l'entreprise, il s'agit donc là, également, d'une des tâches que nous chercherons à achever par la suite.

Le manque de temps pour aller plus loin, nous a conduit à n'envisager qu'une première solution relativement sommaire pour la synchronisation des différents optimisateurs. En effet, le plan de production généré par le système d'information donne une plage de passage acceptable pour les lots dans chaque atelier et les optimisateurs se doivent de la respecter. Ensuite, nous avons considéré qu'un fonctionnement en flux poussé permettrait d'assurer une concordance entre les optimisateurs. D'autres techniques bien plus abouties, que nous n'avons pas eu le temps d'étudier, peuvent répondre à cette problématique. Cela peut même éventuellement faire le sujet d'une autre thèse.

Malgré les résultats que nous avons déjà obtenus, la route est encore longue avant de pouvoir affirmer haut et fort que l'entreprise a passé la transition qui l'amènera dans l'ère de l'industrie 4.0 et de nombreux efforts (adaptation des interfaces homme-machine et des technologies, voir Annexe I pour plus de détails) sont encore à fournir avant de pouvoir mettre en place l'architecture que nous proposons. Pourtant l'espoir perdure car même si le chemin semble long, chaque jour qui passe est un pas de plus vers la réalisation de nos objectifs.

« Quand on a choisi un chemin, aussi compliqué soit-il, on le poursuit jusqu'au bout. Sinon on ne saura jamais ce qu'il nous promet. » Yasmina Khadra, Les anges meurent de nos blessures.

BIBLIOGRAPHIE

- Abdul-Razaq, T.S., Potts, C.N., Van Wassenhove, L.N., 1990. A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Appl. Math.* 26, 235–253. [https://doi.org/10.1016/0166-218X\(90\)90103-J](https://doi.org/10.1016/0166-218X(90)90103-J)
- Allahverdi, A., Gupta, J.N., Aldowaisan, T., 1999. A review of scheduling research involving setup considerations. *Omega* 27, 219–239.
- Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y., 2008. A survey of scheduling problems with setup times or costs. *Eur. J. Oper. Res.* 187, 985–1032. <https://doi.org/10.1016/j.ejor.2006.06.060>
- Armentano, V.A., Mazzini, R., 2000. A genetic algorithm for scheduling on a single machine with setup times and due dates. *Prod. Plan. Control* 11, 713–720. <https://doi.org/10.1080/095372800432188>
- Azadeh, A., Ghaderi, S.F., Izadbakhsh, H., 2008. Integration of DEA and AHP with computer simulation for railway system improvement and optimization. *Appl. Math. Comput.* 195, 775–785. <https://doi.org/10.1016/j.amc.2007.05.023>
- Azizoglu, M., Webster, S., 2003. Scheduling parallel machines to minimize weighted flowtime with family set-up times. *Int. J. Prod. Res.* 41, 1199–1215. <https://doi.org/10.1080/0020754021000042959>
- Balasubramanian, H., Mönch, L., Fowler, J., Pfund, M., 2004. Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. *Int. J. Prod. Res.* 42, 1621–1638. <https://doi.org/10.1080/00207540310001636994>
- Barbosa, J., Leitão, P., Adam, E., Trentesaux, D., 2015. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Comput. Ind.* 66, 99–111. <https://doi.org/10.1016/j.compind.2014.10.011>
- Bareigts, C., 2000. Importance de la coordination/coopération en terme d'apprentissage organisationnel, in: *Actes Du Colloque Agent Logiciels, Coopération, Apprentissage & Activité Humaine*, ATIEF, Biarritz, France. pp. 6–7.
- Beer, S., 1984. The Viable System Model: Its Provenance, Development, Methodology and Pathology. *J. Oper. Res. Soc.* 35, 7–25.
- Bellifemine, F., 1999. JADE - A FIPA-compliant agent framework.

- Berchet, C., Habchi, G., Courtois, A., 1999. Intégration du processus de pilotage à la simulation des systèmes de production, in: Habchi, G. (Ed.), 2ème Conférence Francophone de MOdélisation et SIMulation (MOSIM'99). SCS, Annecy, France, pp. 337–343.
- Bhatnagar, R., Chandra, P., Goyal, S.K., 1993. Models for multi-plant coordination. *Eur. J. Oper. Res.* 67, 141–160. [https://doi.org/10.1016/0377-2217\(93\)90058-U](https://doi.org/10.1016/0377-2217(93)90058-U)
- Billaut, J.-C., 2004. Flexibilité et Robustesse en Ordonnancement 330.
- Birattari, M., Paquete, L., Stützle, T., Varrentrapp, K., 2001. Classification of Metaheuristics and Design of Experiments for the Analysis of Components (Technical report No. AIDA-01-05), Intellektik. Darmstadt University of Technology, Darmstadt Germany.
- Blazewicz, J., Domschke, W., Pesch, E., 1993. The job shop scheduling problem: conventional and new solution techniques. *Eur. J. Oper. Res.* 1–33.
- Blum, C., 2005. Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling. *Comput. Oper. Res.* 32, 1565–1591. <https://doi.org/10.1016/j.cor.2003.11.018>
- Blum, C., 2002. ACO Applied to Group Shop Scheduling: A Case Study on Intensification and Diversification, in: Dorigo, M., Di Caro, G., Sampels, M. (Eds.), *Ant Algorithms*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 14–27. https://doi.org/10.1007/3-540-45724-0_2
- Bowers, M.R., Groom, K., Ng, W.M., Zhang, G., 1995. Cluster analysis to minimize sequence dependent changeover times. *Math. Comput. Model.* 21, 89–95.
- Cardin, O., Derigent, W., Trentesaux, D., 2018. Contribution des architectures de contrôle holoniques à l'industrie 4.0, in: 12th International Conference on Modelling, Optimization and Simulation, MOSIM'18. Toulouse, France.
- Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T., Bril, H., 2017. Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. *J. Intell. Manuf.* 28, 1503–1517. <https://doi.org/10.1007/s10845-015-1139-0>
- Carlier, J., Chrétienne, P., 1982. Un domaine très ouvert : les problèmes d'ordonnancement. *RAIRO - Oper. Res.* 16, 175–217. <https://doi.org/10.1051/ro/1982160301751>
- Chan, F., Chung, S., Wadhwa, S., 2005. A hybrid genetic algorithm for production and distribution. *Omega* 33, 345–355. <https://doi.org/10.1016/j.omega.2004.05.004>
- Co, H.C., 1990. Cluster analysis approach to batching in Flexible Manufacturing systems. *Comput. Integr. Manuf. Syst.* 3, 79–82.
- Coello Coello, C.A., 1999. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowl. Inf. Syst.* 1, 269–308. <https://doi.org/10.1007/BF03325101>

- Colorni, A., Dorigo, M., Maniezzo, V., 1991. Distributed Optimization by Ant Colonies 6.
- Dorigo, M., Di Caro, G., 1999. The Ant Colony Optimization Meta-Heuristic. Presented at the Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, IEEE, Washington, DC.
- Du, T.C., Chiou, R.J., 2000. Applying version management of object-oriented database technology in reactive scheduling. *Int. J. Prod. Res.* 38, 1183–1200. <https://doi.org/10.1080/002075400189086>
- Dumetz, L., Gaudreault, J., Thomas, A., Bril El-Haouzi, H., Lehoux, N., Marier, P., 2017. Planning and coordination for decentralised business units in a lumber production company.
- Ferber, J., 1997. Les systèmes multi-agents : un aperçu général 31.
- Fernandes, N.O., Land, M.J., Carmo-Silva, S., 2016. Aligning workload control theory and practice: lot splitting and operation overlapping issues. *Int. J. Prod. Res.* 54, 2965–2975.
- Gaudreault, J., Forget, P., Frayret, J.-M., Rousseau, A., Lemieux, S., D'Amours, S., 2010. Distributed operations planning in the lumber supply chain: Models and coordination. *Int. J. Ind. Eng. Theory Appl. Pract.* 17, 168–189.
- Genin, P., Lamouri, S., Thomas, A., 2008. Multi-facilities tactical planning robustness with experimental design. *Prod. Plan. Control* 19, 171–182. <https://doi.org/10.1080/09537280801896250>
- Giglio, D., 2015. A MILP model for single machine family scheduling with sequence-dependent batch setup and controllable processing times. *ArXiv150107396 Cs Math*.
- Glover, F., 1989. Tabu Search—Part I. *ORSA J. Comput.* 1, 190–206. <https://doi.org/10.1287/ijoc.1.3.190>
- Gong, Y.-J., Chen, W.-N., Zhan, Z.-H., Zhang, J., Li, Y., Zhang, Q., Li, J.-J., 2015. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Appl. Soft Comput.* 34, 286–300. <https://doi.org/10.1016/j.asoc.2015.04.061>
- Gonzalez, T., 1976. Open shop scheduling to minimize finish time. *Jacm* 665–679.
- Ham, I., Hitomi, K., Yoshida, T., 2012. *Group Technology: Applications to Production Management*. Springer Science & Business Media.
- Han, C., Ham, I., 1986. Multiobjective cluster analysis for part family formations. *J. Manuf. Syst.* 5, 223–230.
- Haouzi, H.B.E., 2017. Contribution à la conception et à l'évaluation des architectures de pilotage des systèmes de production adaptables : vers une approche anthropocentrée pour la simulation et le pilotage. Université de Lorraine.
- Haouzi, H.E., 2008. Approche méthodologique pour l'intégration des systèmes contrôlés par le produit dans un environnement de juste-à-temps: Application à l'entreprise TRANE 156.

- Hatono, I., Nishiyama, T., Umano, M., Tamura, H., 1998. Performance evaluation of distributed real-time scheduling systems using distributed production system simulator, in: Okino, N., Tamura, H., Fujii, S. (Eds.), *Advances in Production Management Systems*. Springer US, Boston, MA, pp. 423–434. https://doi.org/10.1007/978-0-387-35304-3_37
- Herrera, C., Thomas, A., Belmokhtar, S., Pannequin, R., 2011. A viable system model for product-driven systems, in: *International Conference on Industrial Engineering and Systems Management, IESM 2011*. Metz, France, p. CDROM.
- Jain, A.K., Elmaraghy, H.A., 1997. Production scheduling/rescheduling in flexible manufacturing. *Int. J. Prod. Res.* 35, 281–309. <https://doi.org/10.1080/002075497196082>
- Jimenez, J.-F., Bekrar, A., Zambrano-Rey, G., Trentesaux, D., Leitão, P., 2017. Pollux: a dynamic hybrid control architecture for flexible job shop systems. *Int. J. Prod. Res.* 55, 4229–4247. <https://doi.org/10.1080/00207543.2016.1218087>
- Johnson, C., 1980. Constructive critique of a hierarchical prioritization scheme employing paired comparisons, in: *Proceedings IEEE Conference on Cybernetics and Society (Cambridge)*. pp. 373–378.
- Kannan, S., Slochanal, S.M.R., Padhy, N.P., 2005. Application and Comparison of Metaheuristic Techniques to Generation Expansion Planning Problem. *IEEE Trans. Power Syst.* 20, 466–475. <https://doi.org/10.1109/TPWRS.2004.840451>
- Kennedy, J., Eberhart, R., 1995. *Particle Swarm Optimization* 7.
- Klein, T., 2008. Le kanban actif pour assurer l'interopérabilité décisionnelle centralisé/distribué Application à un industriel de l'ameublement 175.
- Lagoudakis, M., 1996. *The 0-1 Knapsack Problem - An Introduction Survey*.
- Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., 1989. Sequencing and scheduling : algorithms and complexity.pdf, in: *(Designing Decision Support Systems Notes. Presented at the (Designing decision support systems notes, Eindhoven: Technische Universiteit Eindhoven*.
- Lee-Post, A., 2000. Part family identification using a simple genetic algorithm. *Int. J. Prod. Res.* 38, 793–810.
- Li, X., Gao, L., 2016. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *Int. J. Prod. Econ.* 174, 93–110. <https://doi.org/10.1016/j.ijpe.2016.01.016>
- Lin, S.-W., Ying, K.-C., Wu, W.-J., Chiang, Y.-I., 2016. Multi-objective unrelated parallel machine scheduling: a Tabu-enhanced iterated Pareto greedy algorithm. *Int. J. Prod. Res.* 54, 1110–1121. <https://doi.org/10.1080/00207543.2015.1047981>

- Little, J.D.C., Murty, K.G., Sweeney, D.W., Karel, C., 1963. AN ALGORITHM FOR THE TRAVELLING SALESMAN PROBLEM.
- Logendran, R., Sriskandarajah, C., 1993. Two-machine group scheduling problem with blocking and anticipatory setups. *Eur. J. Oper. Res., Cellular Manufacturing Systems* 69, 467–481. [https://doi.org/10.1016/0377-2217\(93\)90029-M](https://doi.org/10.1016/0377-2217(93)90029-M)
- Madureira, A., Ramos, C., Silva, S. do C., 2003. Using genetic algorithms for dynamic scheduling, in: *114th Annual Production and Operations Management Society Conference (POMS 2003)*.
- Mahmoodi, F., Dooley, K.J., Starr, P.J., 1990. An investigation of dynamic group scheduling heuristics in a job shop manufacturing cell. *Int. J. Prod. Res.* 28, 1695. <https://doi.org/10.1080/00207549008942824>
- Mandelbrot, B.B., 1975. On the geometry of homogeneous turbulence, with stress on the fractal dimension of the iso-surfaces of scalars. *J. Fluid Mech.* 72, 401–416. <https://doi.org/10.1017/S0022112075003047>
- McFarlane, D., Sarma, S., Chirn, J.L., Wong, C.Y., Ashton, K., 2003. Auto ID systems and intelligent manufacturing control. *Eng. Appl. Artif. Intell.* 16, 365–376. [https://doi.org/10.1016/S0952-1976\(03\)00077-0](https://doi.org/10.1016/S0952-1976(03)00077-0)
- Momoh, J.A., Jizhong Zhu, 2003. Optimal generation scheduling based on AHP/ANP. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 33, 531–535. <https://doi.org/10.1109/TSMCB.2003.811122>
- Nawaz, M., Enscore, E.E., Ham, I., 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega* 11, 91–95. [https://doi.org/10.1016/0305-0483\(83\)90088-9](https://doi.org/10.1016/0305-0483(83)90088-9)
- Neufeld, J.S., Gupta, J.N.D., Buscher, U., 2016. A comprehensive review of flowshop group scheduling literature. *Comput. Oper. Res.* 70, 56–74. <https://doi.org/10.1016/j.cor.2015.12.006>
- Noyel, M., 2015. *Contrôle intégré du pilotage d'atelier et de la qualité des produits*. Université de Lorraine, Nancy.
- Noyel, M., Thomas, P., Thomas, A., Charpentier, P., 2016. Reconfiguration process for neuronal classification models: Application to a quality monitoring problem. *Comput. Ind.* 83, 78–91. <https://doi.org/10.1016/j.compind.2016.09.004>
- O'Brien, P., Nicol, C., 1998. FIPA — Towards a Standard for Software Agents. <https://doi.org/10.1023/A:1009621729979>
- Olfati-Saber, R., Fax, J.A., Murray, R.M., 2007. Consensus and Cooperation in Networked Multi-Agent Systems. *Proc. IEEE* 95, 215–233. <https://doi.org/10.1109/JPROC.2006.887293>
- Pach, C., n.d. ORCA: Architecture hybride pour le contrôle de la myopie dans le cadre du pilotage des Systèmes Flexibles de Production 183.

- Pannequin, R., 2007. Proposition d'un environnement de modélisation et de test d'architectures de pilotage par le produit de systèmes de production 157.
- Papadimidriou, C., 1977. The Euclidean travelling salesman problem is NP-complete. *Theor. Computer Sci.* vol 4, p237-244.
- Parsons, S., Wooldridge, M., 2000. *Game Theory and Decision Theory in Multi-Agent Systems* 14.
- Pitt, J., Kamara, L., Sergot, M., Artikis, E., 2006. Advance Access published on January 23, 2006 doi:10.1093/comjnl/bxh164 Voting in Multi-Agent Systems, *The Computer Journal*.
- Potts, C.N., Kovalyov, M.Y., 2000. Scheduling with batching: A review. *Eur. J. Oper. Res.* 120, 228–249. [https://doi.org/10.1016/S0377-2217\(99\)00153-8](https://doi.org/10.1016/S0377-2217(99)00153-8)
- Ramya, R., Rajendran, C., Ziegler, H., 2016. Capacitated lot-sizing problem with production carry-over and set-up splitting: mathematical models. *Int. J. Prod. Res.* 54, 2332–2344.
- Saaty, T.L., 1990. How to make a decision: The Analytic Hierarchy Process. *Eur. J. Oper. Res.* 48, 9–26.
- Saaty, T.L., Vargas, L.G., 1980. Hierarchical analysis of behavior in competition: Prediction in chess. *Behav. Sci.* 25, 180–191. <https://doi.org/10.1002/bs.3830250303>
- Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., Crailsheim, K., 2009. Get in touch: cooperative decision making based on robot-to-robot collisions. *Auton. Agents Multi-Agent Syst.* 18, 133–155. <https://doi.org/10.1007/s10458-008-9058-5>
- Sha, D.Y., Hsu, C.-Y., 2006. A hybrid particle swarm optimization for job shop scheduling problem. *Comput. Ind. Eng.* 51, 791–808. <https://doi.org/10.1016/j.cie.2006.09.002>
- Shin, H., Kuroda, M., 1998. An autonomous job shop scheduling system under dynamic production environment considering machine breakdowns, in: Okino, N., Tamura, H., Fujii, S. (Eds.), *Advances in Production Management Systems*. Springer US, Boston, MA, pp. 399–410. https://doi.org/10.1007/978-0-387-35304-3_35
- SHMOYS, D.B., WEIN, J., WILLIAMSON, D.P., 1995. Scheduling parallel machines on line. *SIAM J. Comput.* 1313–1331.
- Taboada, H.A., Coit, D.W., 2007. Data clustering of solutions for multiple objective system reliability optimization problems. *Qual. Technol. Quant. Manag.* 4, 191–210.
- Talibi, Z., El Haouzi, H.B., Thomas, A., 2013. The relevance study of adaptive kanban in a multicriteria constraints context using data-driven simulation method, in: *Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference On*. IEEE, pp. 1–7.
- Teknomo, K., 2017. Analytic Hierarchy Process AHP Tutorial [WWW Document]. *Anal. Hierarchy Process*. URL <http://people.revoledu.com/kardi/tutorial/AHP/AHP.htm> (accessed 4.5.18).

- Teodorovic, D., 2003. Transport Modeling by Multi-Agent Systems: A Swarm Intelligence Approach.
- Tharumarajah, A., 1996. Comparison of the bionic, fractal and holonic manufacturing system concepts. *Int. J. Comput. Integr. Manuf.* 9, 217–226. <https://doi.org/10.1080/095119296131670>
- Thierens, D., Goldberg, D., 1994. Convergence models of genetic algorithm selection schemes. *Parallel Probl. Solving Nature—PPSN III* 119–129.
- Thomas, P., 2014. Exploitation de données pour la modélisation. 177.
- THOMAS, P., HAOUZI, H.B.E., SUHNER, M.-C., 2018. Using a Classifier Ensemble for Proactive Quality Monitoring and Control: the impact of the choice of classifiers types, selection criterion, and fusion process. *Comput. Ind.* 30.
- Ueda, K., 1992. A concept for bionic manufacturing systems based on DNA-type information, in: Olling, G.J., Kimura, F. (Eds.), *Human Aspects in Computer Integrated Manufacturing*. Elsevier, Amsterdam, pp. 853–863. <https://doi.org/10.1016/B978-0-444-89465-6.50078-8>
- Valckenaers, P., 2019. ARTI Reference Architecture – PROSA Revisited, in: Borangiu, T., Trentesaux, D., Thomas, A., Cavalieri, S. (Eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer International Publishing, Cham, pp. 1–19. https://doi.org/10.1007/978-3-030-03003-2_1
- Valckenaers, P., 2001. Editorial of the special issue on holonic manufacturing systems. *Comput. Ind.* 3, 233–234.
- Valckenaers, P., Van Brussel, H., Bonneville, F., Bongaerts, L., Wyns, J., 1994. IMS Test Case 5: Holonic Manufacturing Systems. *IFAC Proc. Vol. 27*, 31–36. [https://doi.org/10.1016/S1474-6670\(17\)45996-9](https://doi.org/10.1016/S1474-6670(17)45996-9)
- Vollmann, T.E., Berry, W.L., Whybark, D.C., 1997. *Manufacturing planning and control systems, 1997*. Irwin Boston.
- Xiang, W., Lee, H.P., 2008. Ant colony intelligence in multi-agent dynamic manufacturing scheduling. *Eng. Appl. Artif. Intell.* 21, 73–85. <https://doi.org/10.1016/j.engappai.2007.03.008>
- Yi, Y., Chang, H.-Y., Wang, J., Bai, J.-C., 2004. Soft computing for parallel scheduling with setup times, in: *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference On*. IEEE, pp. 2041–2046.
- Yoshikawa, H., 1995. Manufacturing and the 21st century — Intelligent manufacturing systems and the renaissance of the manufacturing industry. *Technol. Forecast. Soc. Change* 49, 195–213. [https://doi.org/10.1016/0040-1625\(95\)00008-X](https://doi.org/10.1016/0040-1625(95)00008-X)
- Zahedi, F., 1986. The Analytic Hierarchy Process: A Survey of the Method and Its Applications. *Interfaces* 16, 96–108.

Zhang, G., Shao, X., Li, P., Gao, L., 2009. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem.

TABLE DES ILLUSTRATIONS

Figure 1 Démarche de la thèse	9
Figure 2 Une entreprise, deux métiers	11
Figure 3 Division en secteurs de l'entreprise	12
Figure 4 Interactions des éléments du système d'informations	15
Figure 5 Objectifs des différents secteurs.....	16
Figure 6 Axes de recherche thèse Mélanie Noyel	17
Figure 7 Analogie	20
Figure 8 Méta modèle VSM de Carlos Herrera.....	38
Figure 9 Modélisation	42
Figure 10 Méta-modèle VSM.....	44
Figure 11 Évolution des entités collaborantes	46
Figure 12 Postes étudiés	47
Figure 13 Intersection de deux classes de problème	50
Figure 14 Répartition commandes-lots-batches	51
Figure 15 Le concept du clustering sur deux niveaux.....	62
Figure 16 Diagramme d'algorithme génétique.....	65
Figure 17 Population	66
Figure 18 Choix des chromosomes dans le processus de croisement.....	66
Figure 19 Principe de mutation.....	67
Figure 20 Risque de non-qualité détecté.....	69
Figure 21 Processus de réordonnancement.....	70
Figure 22 Diagramme de fonctionnement du protocole AHP.....	72
Figure 23 Enchaînement des opérations sur les optimisateurs	75
Figure 24 Application.....	76
Figure 25 Rendu du logiciel Direct Planning.....	78
Figure 26 Boucles de réparations et regroupement.....	79
Figure 27 Une capture d'écran de l'application servant à re-trier les lots	82
Figure 28 Diagramme de TRACILOGIS	89
Figure 29 Logigramme TRACILOGIS.....	91
Figure 30 UML de l'agent produit	93

Figure 31 Transposition d'Acta sur TRACILOGIS	95
Figure 32 Liste des configurations	96
Figure 33 Mesure de la dispersion du makespan	99
Figure 34 Workflow du robot de laquage Acta-mobilier	100
Figure 35 Fonctionnement standard	104
Figure 36 Si un risque de non-qualité est détecté	105
Figure 37 Impact de la myopie sur le retard	110
Figure 38 Impact de la myopie sur le nombre de reworks	111
Figure 39 Impact de la myopie sur le temps de calcul	112
Figure 40 Comparaison des temps de retard	113
Figure 41 Comparaison du nombre de réparations	113
Figure 42 Exemple d'ordre de fabrication	132
Figure 43 Menu principal du M.E.S.	133
Figure 44 Fenêtre de pointage et d'affichage de standard	135
Figure 45 Classification problèmes d'ordonnancement avec temps de réglage	137
Figure 46 Principe des colonies de fourmis (Dorigo and Di Caro, 1999)	141
Figure 47 Découpage en critère pour utilisation AHP	147

TABLE DES TABLEAUX

Tableau 1 Comparaison des architectures hybrides.....	35
Tableau 2 Variables de décision lot-sizing.....	53
Tableau 3 Variables de décision scheduling.....	53
Tableau 4 Contraintes du lot-sizing	55
Tableau 5 Contraintes liées au scheduling	56
Tableau 6 Contraintes relatives à l'union des deux problèmes.....	57
Tableau 7 Contraintes pour minimiser l'écartement	57
Tableau 8 Contraintes de setup.....	57
Tableau 9 Contraintes ensemblistes	58
Tableau 10 Comparaison avec et sans optimisation	80
Tableau 11 Comparaison entre notre approche et les anciennes techniques utilisées	81
Tableau 12 Épaisseur et surface du cas de test.....	83
Tableau 13 Batches configuration 1	84
Tableau 14 Batches configuration 2	84
Tableau 15 Synthèse comparaison des méthodes.....	87
Tableau 16 Résultats d'expérience sans réparation.....	98
Tableau 17 Résultats d'expérience avec un taux de réparation de 20%	98
Tableau 18 Liste des variables	104
Tableau 19 Cas réel	106
Tableau 20 Répartition égale.....	106
Tableau 21 Cas de non distribution (que brillant et que satiné).....	107
Tableau 22 Tous les produits ont la même échéance	107
Tableau 23 Sans prise en compte du risque de non qualité	107
Tableau 24 Résultats test ANOVA pour les reworks	110
Tableau 25 Résultats test ANOVA pour le retard	111
Tableau 26 Résultats test ANOVA pour le temps de calcul.....	112
Tableau 27 Classification des métaheuristiques (Birattari et al., 2001)	143
Tableau 28 Comparatif de métaheuristiques (Kannan et al., 2005).....	144
Tableau 29 Commandes groupe 1	149
Tableau 30 Résultats groupe 1.....	150
Tableau 31 Commandes groupe 2	151

Tableau 32 Résultats groupe 2.....	153
Tableau 33 Commandes groupe 3	153
Tableau 34 Résultats groupe 3.....	155
Tableau 35 Commandes groupe 4	156
Tableau 36 Résultats groupe 4.....	158
Tableau 37 Commandes groupe 5	159

ANNEXES

I. ANNEXE 1 : Les développement informatique dans l'entreprise

Cette annexe présente les éléments du système d'information sur lesquels nous avons travaillé et avec lesquels le système de pilotage devra communiquer.

A. Les ordres de fabrication (OF)

Une des premières missions qui nous a été confiée fut la refonte des ordres de fabrication qui d'une part représentaient un pourcentage important de la consommation de papier de l'entreprise et surtout car ceux-ci n'intégraient aucun moyen d'identification relatif aux lots élémentaires mis en place par Noyel. Ils servent donc de méthode d'identification des lots et récapitulent la gamme de fabrication des produits constituant le lot en question, de la liste complète des produits faisant partie du lot, avec leurs dimensions et de la composition des laques ou divers vernis à préparer pour leur application.

2 modèles de ces nouveaux documents ont été produits :

- Un premier regroupant ensemble tous les lots du couple {client, semaine de livraison} possédant la même gamme de fabrication pour la partie usinage-préparation (partie gauche de la **Figure 12**). En effet, pour rappel, dans cette partie de la chaîne de production, les produits à réaliser peuvent être regroupés indépendamment de leur couleur.
- Un second modèle (**Figure 42**) relatif à un unique lot élémentaire, est utilisé à partir du poste de travail « Tri » en remplacement du premier modèle. Ce modèle est plus adapté à la manière de fonctionner dans la seconde moitié de l'atelier. Nous n'avons pas souhaité établir de document regroupant les lots de même couleur pour le moment, car les gammes de fabrications sont beaucoup disparates lorsque l'on entre dans la seconde partie de la chaîne de fabrication et les cas particuliers se multiplient. Nous sommes d'avis que le système de pilotage se chargera de réaliser les groupements de lots appropriés si nécessaire.

Entête

Information dos sat

GRIS BETON SUPERHELL
FACE BETON
19MM(Cegid:BE2U)

BUCHER **S28 2015** **Qte : 7**
Surface : 0,84 m²

Ce QR Code représente le numéro d'identification d'un OF - Pour les postes non munis de lecteurs code barre, ce numéro est inscrit en dessous

Code	Libellé	Qté théo	Conso/Validation
ME-EFINITIONSUPE R01	FINITION BETON FINITION SUPERHELL 100% de POUSSIERE BETON + 2% de POLYMERISANT + 2% de GRAIS + 2% de PIGMENT NOIR DE VIGNE + 2% de BLANC DE TITANE SYNTHÉ	0,633 KG	
ME-EGRIPSUPERIE LL	GRIS FINITION SUPERHELL 100% de GRIS BRUI + 2% de POLYMERISANT + 3% de GRAIS	0,674 KG	
ME-E770	GRIS CLAIR SATINE 770 RAL 7035 100% de GR 45043 7067 + 1% de GRAIS + 2% de LAC 70	0,168 LT	
ME-V036	VERNIS 05 POUR BETON 100% de CE 4290-0003 + 1% de DR 4075 + 4% de V036 + 2% de NR 1173	0,126 LT	
ME-V037	VERNIS CRE POUR BETON 100% de VERNIS FIXATEUR COLORIF RESIN	0,126 LT	

Nomenclature

Phase	Poste	Opération	Tps théo (min)	Quantité	Conso/Validation
00010	CONTROLE TRI (212063)	Regroupement des pièces pour contrôle Tri des pièces par dimensions et couleur	0,001		
00020	PONCEUSE HEISEMANN (212078)	JEXAGE FACE	0,025		
00040	CABINE GRIP (213045)	PISTOLET GRIP SUR FACE	0,051		
00060	POSTE APPLICATION BETON (213043)	APPLI FINITION BETON SUR FACE SECON TEMPLON	0,344		
00070	CABINE GRIP (213045)	APPLI FINITION SUR CHAMANTS	0,267		
00080	PONCEUSE HEISEMANN (212078)	PONCAGE FACE	0,051		

Pour le tri : Indiquez la quantité contrôlée

Gamme de fabrication

Pied de page

23/06/2015 10:9:23 Page 1 sur 3
BUCHER(Cegid:200426) S28 2015 Lundi

BLANC PUR BRILLANT 934
MONET F4C2
BRI(Cegid:MHD)

BSKUECHEN **S28 2015** **Qte : 40**
Surface : 9,28 m²

Listing des pièces

LISTING PIECES						
Désignation	Ht	Lg	Qte	Qte contrôlée	Qte réel	OV/L.n
MONET FROM H63102894	137	304	2	[102]		531395-002
MONET FROM H63102894	137	504	6	[102][48][8]		531395-006
MONET FROM H63102894	137	894	1	[1]		531395-010
MONET COALISSANT H63102894	278	504	4	[102][8]		531395-007
MONET COALISSANT H63102894	278	894	1	[1]		531395-016
MONET COALISSANT H63102894	290	594	4	[102][8]		531395-008
MONET COALISSANT H63102894	290	894	1	[1]		531395-017
MONET COALISSANT H63102894	318	594	1	[1]		531395-010
MONET FORTE H643102894	431	594	1	[1]		531395-005
MONET FORTE H657102894	571	594	2	[102]		531395-001
MONET FORTE H657102894	571	594	2	[102]		531395-014
MONET FORTE H68102894	610	444	1	[1]		531395-003
MONET FORTE H68102894	610	594	2	[102]		531395-011
MONET FORTE H689102894	891	394	2	[102]		531395-002
MONET FORTE H689102894	891	594	2	[102]		531395-013
MONET FILEUR H689102894	891	70	4	[102][8]		531395-016
MONET FORTE H72102894	1310	504	1	[1]		531395-012
MONET FILEUR H14000198	1400	190	1	[1]		531395-019
MONET FORTE H14000258	1400	250	1	[1]		531395-020
MONET H68LL H18802894 Fabr. MONET DICHESTE H18802894X18	1603	444	1	[1]		531395-004

23/06/2015 9:26:11 Page 4 sur 5
BSKUECHEN(Cegid:200408) S28 2015 Lundi

Figure 42 Exemple d'ordre de fabrication

Initialement, ces documents étaient édités par le service commercial, lorsqu'ils considéraient que les clients n'effectueraient plus de modification ou lorsque la date limite définie pour démarrer la production du client a été atteinte. Malgré ces précautions, la probabilité qu'un client demande une modification après lancement restait non négligeable. Dans un souci de satisfaction des clients, l'entreprise ne rejette pas les demandes de changement de dernière minute. Par conséquent, certains documents devaient complètement être réimprimés. Pour limiter cette éventualité, nous sommes en train de déplacer cette impression au plus proche du besoin, à savoir juste avant de réaliser l'usinage. Il en va de même pour le second modèle, nous voulons retarder son impression au moment où le collaborateur de l'étape de Tri va effectuer l'opération de dégroupement, contrôle pièce à pièce et regroupement. Cela représentait un premier pas vers une gestion plus informatisée de la production. À terme, notre

objectif serait de remplacer ces documents par des composants d'identification dynamique avec mémoire (type puce RFID), pour étendre le champ des possibilités du contrôle par le produit.

B. LE M.E.S. et l'affichage dynamique des bonnes pratiques de travail en temps réel



Figure 43 Menu principal du M.E.S.

Le M.E.S. de l'entreprise Acta-Mobilier, mis au point par Noyel durant sa thèse a fait l'objet de plusieurs ajouts auxquels nous avons participé. En effet, dans un premier temps, sa démocratisation n'a pas été simple. Afin de favoriser son acceptation, nous avons délibérément choisi de limiter les fonctions du M.E.S. En effet, pour répondre aux problématiques les plus urgentes de l'entreprise, nous avons concentré nos efforts sur la traçabilité. Pour cela, une deuxième version de l'application a été élaborée. Dans celle-ci, les questions relatives à la productivité (calcul de TRS, ...) ne sont plus abordées. Un soin tout particulier a été apporté sur la mise en place d'un échange, une contrepartie à l'obligation de réaliser l'opération de pointage par les opérateurs. En effet, nous avons créé un système de transmission d'informations pour aider à la réalisation des tâches, que nous avons nommé « standards dynamiques ». Cela pour faire référence à ce que nous appelons « standard de travail » dans l'entreprise correspond à la meilleure manière déterminée à ce jour pour réaliser une opération à un poste donné, ceux-ci pouvant être présents sous forme de messages textuels, d'images, de vidéos ou de n'importe quel autre type de documents numériques. Et « dynamique », pour souligner le fait que ces standards s'affichent uniquement lorsqu'ils sont nécessaires et qu'ils correspondent exactement à l'opération qui doit être effectuée sur le (ou les) lot(s) qui ont été pointés. Pour cela, nous sommes partis de l'ensemble des caractéristiques intrinsèques du lot en question (couleur, famille d'article, surface, nombre de pièces, délai de livraison, criticité de la commande, ...). Nous avons ensuite déterminé pour chaque standard de travail, les conditions dans lesquelles ceux-ci devaient s'appliquer et ajouter une hiérarchie qui permet à des standards plus spécifiques de prendre le pas sur des standards plus globaux. Nous avons sauvegardé ces informations dans une base de données que l'application vient consulter lors du pointage, si les caractéristiques correspondent alors le standard de plus haute position dans la hiérarchie s'affiche.

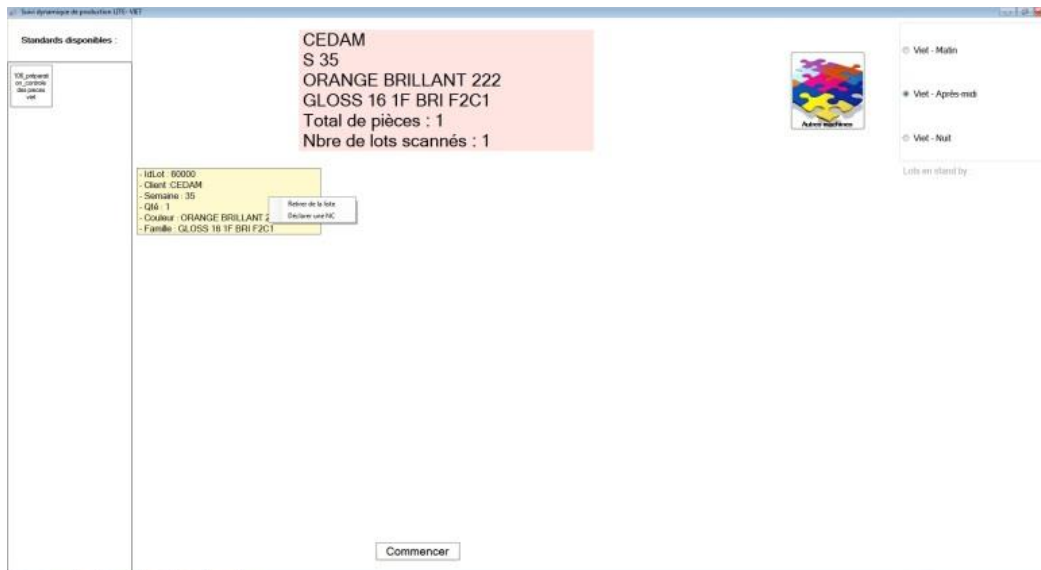


Figure 44 Fenêtre de pointage et d'affichage de standard

Ensuite nous avons concentré nos efforts sur l'identification des pièces sortant de leur lot pour cause de non-qualité. De nombreuses séances de sensibilisation à la problématique ont dû être réalisées afin de faire comprendre aux collaborateurs les enjeux sous-jacents à la déclaration des produits non-conformes.

À l'heure d'aujourd'hui, environ 80% des postes de travail de la chaîne de production sont équipés d'un ordinateur avec une instance du M.E.S. en fonctionnement. La traçabilité n'est malheureusement pas encore fiable à 100% car il subsiste des « réfractaires » que nous n'avons pas encore réussi à convaincre de l'intérêt de son utilisation ou qui considèrent la tâche trop contraignante. Nous tendons tout de même doucement vers notre objectif.

Le M.E.S. est pour nous la clé de voute, autour de laquelle, l'architecture que nous souhaitons mettre en place va se construire. En effet, en plus de porter le « jumeau numérique » de chaque poste de travail, il fera l'interface entre l'humain et le système pour l'aide à la prise de décision.

C. Le logiciel d'intégration automatique des commandes (I.A.C.)

De manière à limiter au maximum l'intervention de l'humain pour des tâches à non-valeur ajoutée et également pour réduire la probabilité d'erreur de saisie, nous avons réalisé un outil pour décrypter et traduire les fichiers de commandes envoyés par les clients en fichier intégrable directement par l'E.R.P. de l'entreprise. La spécificité de cette application est son fonctionnement par pattern à identifier dans les fichiers d'entrée pour générer les codes articles correspondants. Le logiciel demande, lorsqu'il ne possède pas de correspondance, à l'expert (le commercial en charge du client à traiter) quelle règle il doit ajouter pour traiter la ligne de commande sur laquelle il s'est arrêté. Une fois sa bibliothèque de règles créée, celle-ci représente la base de connaissance client pouvant être transmise à tout nouveau membre de l'équipe commerciale. Actuellement, il reste un certain nombre de clients qui ne nous envoient pas de fichiers de commande ou des scans de documents papiers et nous n'avons pas à notre disposition les solutions techniques pour les traiter. Lorsque ce problème sera réglé, nous serons alors en mesure d'automatiser entièrement la phase allant de la réception de la commande jusqu'au début de fabrication des produits.

II. ANNEXE 2 : Méthodes d'optimisation d'ordonnement

L'objectif, ici, n'est pas d'essayer d'établir une liste exhaustive de l'ensemble des techniques d'ordonnement existantes et de leurs avantages/inconvénients mais d'en proposer quelques-unes, de définir celles que nous avons choisi d'utiliser et les raisons qui nous ont poussés à faire ces choix. En outre, comme nous n'avons pas pour but de proposer des solutions optimales aux problèmes d'ordonnement que nous avons à traiter, mais plutôt de valider que l'architecture que nous voulions implémenter était crédible. Nous sommes conscients que d'autres méthodes pourraient mieux convenir pour solutionner nos problématiques. Le paragraphe qui suit, sert donc à proposer une taxonomie des classes de problèmes dans lesquelles ranger nos cas d'étude. Nous nous concentrerons ensuite, sur les méthodes de résolution que nous avons sélectionnées.

De nombreux travaux ont été menés sur le thème de l'ordonnement avec temps de réglage (Allahverdi et al., 2008). En accord avec la classification des problèmes d'ordonnement avec temps de réglage proposé par (Allahverdi et al., 1999) repris sur la **Figure 45**. Les problèmes que nous étudions se situent dans la catégorie des problèmes d'ordonnement de lots, indépendants de la séquence pour le premier puis dépendant de la séquence pour le second. Ces types de problème sont connus comme étant NP-difficile.

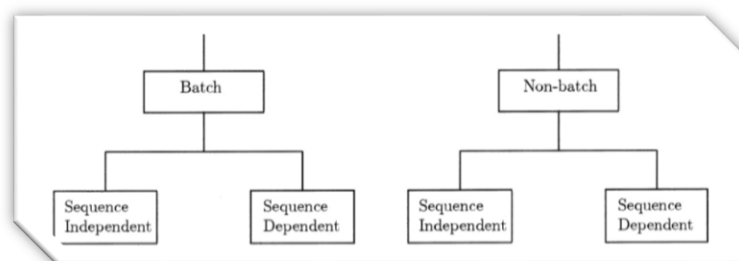


Figure 45 Classification problèmes d'ordonnement avec temps de réglage

L'ordonnement de groupe s'effectue en deux phases : séquencement des tâches au sein d'un même groupe et séquencement des groupes les uns par rapport aux autres. La première partie est réalisée généralement par de simples règles de pilotage, telle que FCFS (premier arrivé premier servi) ou SPT (temps de fabrication minimal). Pour le séquencement des groupes, on privilégie des heuristiques telles que celles présentées dans (Mahmoodi et al., 1990).

Plusieurs méthodes de résolution ont été utilisées pour répondre à ce type de problèmes telles que les algorithmes « branch and bound » (Azizoglu and Webster, 2003), « Tabu search » et aussi les algorithmes génétiques (Yi et al., 2004), (Armentano and Mazzini, 2000), (Balasubramanian et al., 2004). Les paragraphes suivants décrivent quelques-unes de ces techniques de résolution. Nous avons, délibérément, cité presque exclusivement des méta-heuristiques, car les problèmes que nous avons à traiter possèdent beaucoup trop de variables et de contraintes pour être solvables par des méthodes exactes « standards » dans des délais acceptables (nous en ferons la preuve dans la partie applicative du document). Ces méthodes bien que possédant toutes des caractéristiques qui leur sont propres, reposent sur un équilibre entre deux phases :

- L'exploration qui vise à isoler un sous-espace de solutions.
- L'exploitation qui étudie l'ensemble des éléments présents dans le sous-espace délimité par l'exploration.

Un état de l'art des différents algorithmes évolutionnaires, dont font partie ceux que nous allons évoquer, est proposé par (Gong et al., 2015). Les deux premiers présentés sont des techniques plutôt orientées centralisées alors que les suivantes peuvent tout aussi bien avoir une implémentation centralisée que distribuée.

A. Branch and Bound (BnB)

L'algorithme « Branch and Bound », ou « séparation et évaluation » en français, est un paradigme algorithmique pour les problèmes d'optimisation combinatoire ou discrète utilisé fréquemment pour résoudre les problèmes NP-difficiles. L'ensemble des solutions du problème est représenté sous la forme d'un arbre divisible en sous-ensembles de solutions similaires appelés branches. Chaque branche est évaluée, par rapport à une fonction coût et une solution temporaire, si celle-ci ne possède pas une meilleure solution, on effectue un élagage et le sous-ensemble n'est pas exploré. Sinon le sous-ensemble est redivisé en sous-ensemble et la procédure précédente est répétée.

Cet algorithme a été utilisé pour la première fois dans un problème de recherche opérationnelle pour le fameux problème du voyageur de commerce par (Little et al., 1963). Le bon fonctionnement de cette méthode repose sur l'efficacité de deux facteurs principaux : la méthode de sélection, visant à explorer les solutions en faisant intervenir des heuristiques ou des méta-heuristiques et la définition d'une « bonne » fonction d'optimisation. (Billaut, 2004) s'est notamment inspiré de la méthode du « branch and Bound » pour résoudre un problème dynamique d'ordonnement de projets sous contraintes de ressources.

B. Tabu search (TS)

La recherche Tabou est une métaheuristique, dérivant des procédés de recherche locale, conceptualisée (1986) puis formalisée (1989) par F.Glover (Glover, 1989), elle est utilisée en optimisation mathématique et donc en recherche opérationnelle. Le principe consiste à visiter les voisins proches d'une solution lorsque celle-ci a été sélectionnée, pour éviter de tomber dans un optimum local, une solution déjà explorée devient tabou (ou interdite). Les solutions précédentes sont, pour cela, conservées en mémoire. On peut citer les travaux de (Lin et al., 2016) sur les problèmes d'ordonnement multicritère de machines parallèles et en particulier la sous classe des machines parallèles non liées pour laquelle il a combiné une recherche tabou à un algorithme IPG (Iterated Pareto Greedy). Il est à noter que dans la plupart des travaux actuels concernant l'utilisation de la recherche tabou pour de l'ordonnement dynamique d'atelier, cette technique est jumelée à une autre méta-heuristique telle que les algorithmes génétiques (Li and Gao, 2016).

Avantages : capable d'apprendre du passé, évite les boucles infinies et accepte l'exploration de moins bonnes solutions pour sortir d'extrema locaux.

C. Algorithme génétique (GA)

Le concept d'algorithme génétique a été introduit en 1975 par Holland. Ces algorithmes simulent le processus d'évolution biologique, ils sont fondés sur des populations de solutions. À l'initialisation, une première population est générée par l'intermédiaire d'un mécanisme choisi par l'utilisateur (méthode de tirage aléatoire, présélection par une méthode d'optimisation, ...). Par la suite, l'algorithme crée de nouvelles populations qui ont pour objectif d'être meilleures que les précédentes et cela par l'application d'opérations génétiques nommées sélection, croisement et mutation via la combinaison des chromosomes (un chromosome étant une solution de la population en cours d'étude).

Des chromosomes tirés au sort aléatoirement par paires sont soumis au processus de croisement suivant une probabilité P_C et chaque descendant ainsi créé est lui-même soumis à une possible mutation avec une probabilité P_M généralement faible. L'adaptabilité de celui-ci à la résolution du problème est alors évaluée avec la fonction d'adaptation (fitness). Les chromosomes de la population en cours d'étude les moins adaptés à atteindre la solution sont alors remplacés par les descendants nouvellement générés. L'algorithme s'arrête lorsque la condition d'arrêt est atteinte, celle-ci pouvant être liée au nombre d'itérations, au temps de calcul ou encore à l'atteinte d'un nombre de génération successive sans amélioration. L'intérêt premier des algorithmes génétiques réside dans le fait qu'ils fournissent presque toujours de bon résultats dans les délais acceptables même s'il ne garantissent pas la récupération de la solution optimale (Madureira et al., 2003). Le premier à avoir utilisé les algorithmes génétiques pour solutionner de problèmes d'ordonnancement était (Davis, 1991), depuis la technique s'est énormément répandue pour l'ordonnancement industriel, la plupart du temps dans le cadre de problèmes statique mais est aussi efficace pour les problèmes d'ordre dynamique.

Avantages : les algorithmes génétiques convergent rapidement, la propriété de mutation permet de ne pas tomber dans des extrema locaux

Inconvénients : les réglages des probabilités de croisement et de mutation peuvent être complexes pour obtenir de bons résultats.

D. Colonies de fourmis (ACO)

Les algorithmes ACO, proposés par Marco Dorigo en 1991 dans (Coloni et al., 1991), sont comme leur nom l'indique inspirés du comportement social des fourmis, ayant la capacité de trouver le plus court chemin séparant la nourriture qu'elles veulent récupérer et leur fourmilière. Le fonctionnement, illustré **Figure 46**, est le suivant :

- 1) Une première fourmi identifie un point de ravitaillement noté (F), en empruntant un chemin quelconque noté (a) et retourne vers sa fourmilière (N) en laissant derrière elle une piste de phéromone notée (b)
- 2) Les fourmis suivantes vont emprunter de manière indifférente les quatre chemins éventuels créés par les phéromones de la première fourmi, mais naturellement un renforcement de la piste rendra le chemin le plus court plus attractif

- 3) Au fur et à mesure que le temps passe, les fourmis emprunteront le plus court chemin de plus en plus souvent jusqu'à ce que les autres chemins finissent par totalement perdre leurs pistes de phéromones.

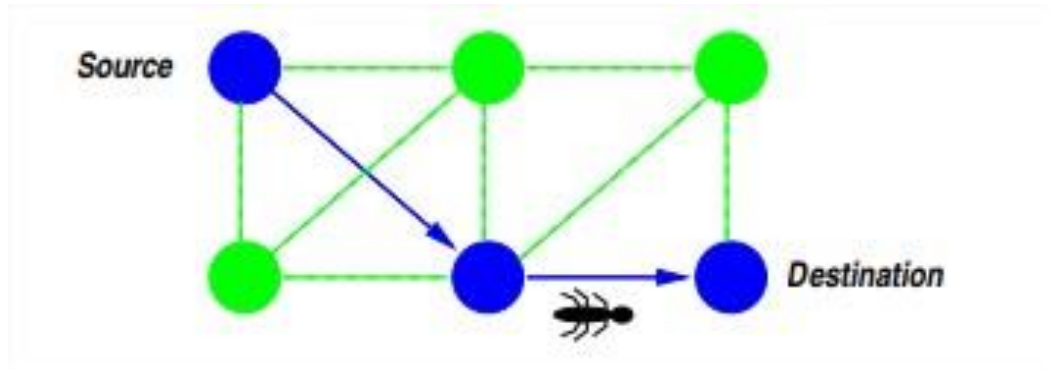


Figure 46 Principe des colonies de fourmis (Dorigo and Di Caro, 1999)

Par conséquent, les algorithmes qui en découlent suivent un principe très similaire :

Initialiser les pistes de phéromones ;

Tant que (non (critère d'arrêt)) {

 Construire solutions basées ou non sur une heuristique ;

 Mettre à jour les pistes de phéromones ;

}

Retourner meilleure piste ;

Les ACO sont très souvent utilisés dans les problèmes d'ordonnancement tels que les jobshop (Blum, 2005) ou les ordonnancement de groupes de produits (Blum, 2002). Ce type d'algorithme basé sur un comportement social vont souvent de pair avec les architectures multi-agents (Xiang and Lee, 2008).

E. Particle swarm optimization (PSO)

L'optimisation par essaim particulaire est, également, une méta-heuristique. Élaborée par Russell Eberhart et James Kennedy (Kennedy and Eberhart, 1995), elle s'inspire de modèle du vivant et socio-psychologie. Tout comme les colonies de fourmis, cette optimisation repose sur la collaboration au sein d'une collectivité. Les particules sont placées dans l'espace de recherche et à chaque itération, elles vont se mouvoir en suivant les équations du mouvement suivantes :

- $V_{k+1} = \omega V_k + a_1(P_m - X_k) + b_2(P_v - X_k)$ (1)
- $X_{k+1} = X_k + V_{k+1}$ (2)

Où :

- V_k est la vitesse à l'instant k
- P_m est la meilleure solution trouvée jusqu'alors
- P_v est la meilleure solution du voisinage jusqu'alors
- ω est l'inertie
- a_1 et a_2 sont des paramètres choisis aléatoirement entre 0 et des limites respectives

La méthode des essaims particulaires est utilisé pour résoudre des problèmes d'ordonnement de type job shop (Sha and Hsu, 2006) et comme cité précédent elle fait aussi l'objet de combinaisons, comme dans (Zhang et al., 2009) où une hybridation entre PSO et une recherche tabou est utilisée pour résoudre des problèmes d'ordonnement multicritères de job shop flexible.

Conclusion :

Les algorithmes basés trajectoire (BnB et TS) font partis des plus efficaces sur la phase exploitation car la probabilité de passer à côté d'une solution optimale est minime par rapport à une stratégie basée population comme les autres algorithmes présentés. En contrepartie, les basés population sont plus efficaces pour la partie exploration.

Le **Tableau 27** fait une première classification d'une partie des métaheuristiques que nous avons cité précédemment. Il évalue la présence ou non d'un certain nombre de caractéristiques ayant un impact sur le choix d'une solution plutôt qu'une autre. Les caractéristiques concernées sont les suivantes :

- Suivi d'une trajectoire : les métaheuristiques ayant cette capacité permettent le retour à des solutions moins optimales mais permettant de sortir d'extrema locaux.
- L'intervention d'une population offre plus de liberté pour l'exploration de l'espace de recherche. Même si, la qualité des résultats finaux est extrêmement dépendante de la manière dont la population est manipulée.
- La conservation en mémoire des résultats antérieurs permet à certains algorithmes de diriger différemment la recherche des prochaines solutions.

- Les méthodes à visite de voisinage multiple ont l'avantage de diversifier leurs espaces de solutions.
- La possibilité que la fonction « objectif » puisse évoluer durant le processus de calcul peut rendre plus réactif l'algorithme.
- Le dernier critère de classification sert à rappeler la source d'inspiration de la métaheuristique.

Caractéristique	Recherche tabou	Algorithme génétique	Colonie de fourmis
Trajectoire	Oui	Non	Non
Population	Non	Oui	Oui
Mémoire	Oui	En partie	Oui
Voisinage multiple	Non	En partie	Non
Fonction objectif dynamique	En partie	Non	Non
Bio-inspiré	Non	Oui	Oui

Tableau 27 Classification des métaheuristiques (Birattari et al., 2001)

Nous avons pris la décision de mettre en œuvre un algorithme génétique que nous utiliserons pour le premier ordonnanceur, nous décrivons son utilisation dans le chapitre suivant. Cette décision s'est basée sur plusieurs critères :

- Notre critère prédominant est la rapidité de convergence. Les algorithmes génétiques, en particulier lorsqu'ils utilisent une loi de sélection élitiste sont connus pour converger rapidement en donnant des solutions non-optimales, mais tout de même très acceptables.
- Ensuite, cette catégorie d'algorithme est très répandue et éprouvée, de plus son implémentation est relativement aisée.
- Un bémol, toutefois, comme dit dans le paragraphe les concernant, il est vrai qu'un certain nombre d'essais a dû être réalisé afin d'ajuster les paramètres pour obtenir un résultat satisfaisant.

Dans ses travaux (Kannan et al., 2005) a réalisé une étude comparative des métaheuristiques sur le cas d'une étude de planification d'expansion de production d'énergie sur une durée de 6 ans. Les techniques sont comparées en étudiant leur taux de réussite (SR), leur nombre moyen de générations créées (ANG), le taux d'erreur, les meilleures et pires solutions générées ainsi que le temps d'exécution (ET). Le tableau suivant, est un extrait de celui réalisé par (Kannan et al., 2005).

Tech	Meilleure (10 M\$)	Pire (10 M\$)	ANG	Erreur (%)	SR (%)	ET (Min.)
GA	1,2009	1,2024	266	0-0,12	72	41,2
ACO	1,2009	1,2096	165	0-0,74	18	90
PSO	1,2009	1,2014	112	0-0,04	68	50
TS	1,2009	1,2024	421	0-0,12	40	26,7

Tableau 28 Comparatif de métaheuristiques (Kannan et al., 2005)

De ce tableau, nous pouvons conclure que, a minima, dans le cas d'étude proposé, l'algorithme génétique semble être un bon compromis entre taux de réussite, taux d'erreur et temps d'exécution. Ces résultats nous confortent également par rapport au choix d'algorithme que nous avons fait.

Lorsque l'on fait de l'ordonnancement multicritères, nous serons fatalement confrontés à un choix de méthode d'évaluation. Le paragraphe suivant est dédié à la présentation de certaines de ces techniques. Nous nous en servons dans deux situations :

- Pour établir la répartition des poids à attribuer à chaque sous fonction objectif de notre algorithme génétique.
- De manière à sélectionner une solution plutôt qu'une autre.

III. ANNEXE 3 : Méthodes d'évaluation de fonction multi-objectifs

Dans son étude des techniques évolutionnistes d'optimisation multi objectifs Coello (Coello, 1999) décrit un certain nombre de techniques pour l'évaluation des fonctions multi-objectifs résolues par algorithme génétique, nous en avons extrait quelques-unes dans ce résumé et ajouté un paragraphe sur la procédure AHP que nous utiliserons également dans nos travaux.

A. Somme pondérée des critères d'optimisation

Historiquement parlant, cette technique est la première à avoir été utilisée pour l'optimisation multicritères. Il s'agit d'une méthode d'agrégation des fonctions d'optimisation qui consiste simplement à sommer toutes les fonctions en leur attribuant des coefficients pondérateurs à chacune. Ainsi, l'optimisation multi objectif devient une simple optimisation scalaire du type : $\min \sum_{i=1}^k w_i f_i(\bar{x})$, où les $w_i \geq 0$ sont les coefficients de pondération donnant l'importance relative de chaque objectif. En général, les coefficients vérifient la propriété suivante : $\sum_{i=1}^k w_i = 1$.

Ses principaux avantages résident dans son efficacité, d'un point de vue calculatoire, et sa capacité à produire des solutions fortement non-dominées utilisables comme solutions initiales pour d'autres techniques. Elle a également l'avantage d'être simple à mettre en place et ramène le problème à une résolution scalaire.

En contrepartie, il est complexe de trouver les poids appropriés même avec une bonne connaissance du problème. Par conséquent, il faut généralement multiplier les essais et étudier l'évolution des solutions en fonction des valeurs attribuées aux coefficients.

B. « Vector Evaluated Genetic Algorithm » (VEGA)

Développé par David Schaffer comme une extension multi-objectifs du « Simple Genetic Algorithm ». À chaque nouvelle génération, un certain nombre de sous-population sont générées grâce à des sélections proportionnelles pour chaque fonction. Ainsi, pour un problème à k objectifs avec une population de N individus, k sous-populations de taille N/k seront générées. Ces sous-populations seront mélangées pour réobtenir une population de taille N sur laquelle l'algorithme génétique appliquera les processus de croisement et mutation.

Avantages : simple à implémenter

Inconvénients : La fonction objectif reste une combinaison linéaire des poids attribués à chaque objectif et de ce fait dans un espace concave certaines solutions ne seront jamais explorées. De plus, une caractéristique pouvant être désavantageuse suivant le but que l'on cherche à obtenir est le phénomène de spéciation : la technique sélectionne des individus performant par rapport à un critère particulier pour chaque sous-population, cela à pour conséquence de supprimer des solutions qui pourraient être des bons compromis entre tous les objectifs, car non-optimal vis-à-vis de chaque objectif.

C. Approches basées Pareto

L'idée est de déterminer un échantillon de la population qui est « Pareto non-dominé » par le reste de la population. Il se voit alors attribué le rang le plus élevé et est retiré de la population. Le processus est réitéré jusqu'à ce que la population soit convenablement échelonnée.

Avantages : il s'agit là, de la meilleure manière de générer un front de Pareto complet en une seule instanciation d'un algorithme génétique. De plus, elle est moins sensible à la courbure ou à la continuité du front de Pareto.

Inconvénients : le « classement Pareto » a le désavantage de ne pas avoir d'algorithme efficace pour déterminer la non-dominance d'un jeu de solution, et les performances des algorithmes se dégradent rapidement avec l'augmentation de la taille de la population ou du nombre d'objectifs.

D. Procédure d'analyse hiérarchique (AHP)

La technique nommée AHP, mise au point par Thomas L. Saaty dans les années 70, fondée sur un mélange de mathématiques et de psychologie, est le plus souvent utilisée pour de la prise de décision collective (Saaty, 1990) (Zahedi, 1986). Selon (Johnson, 1980), résoudre un problème décisionnaire avec la méthode AHP requiert quatre étapes :

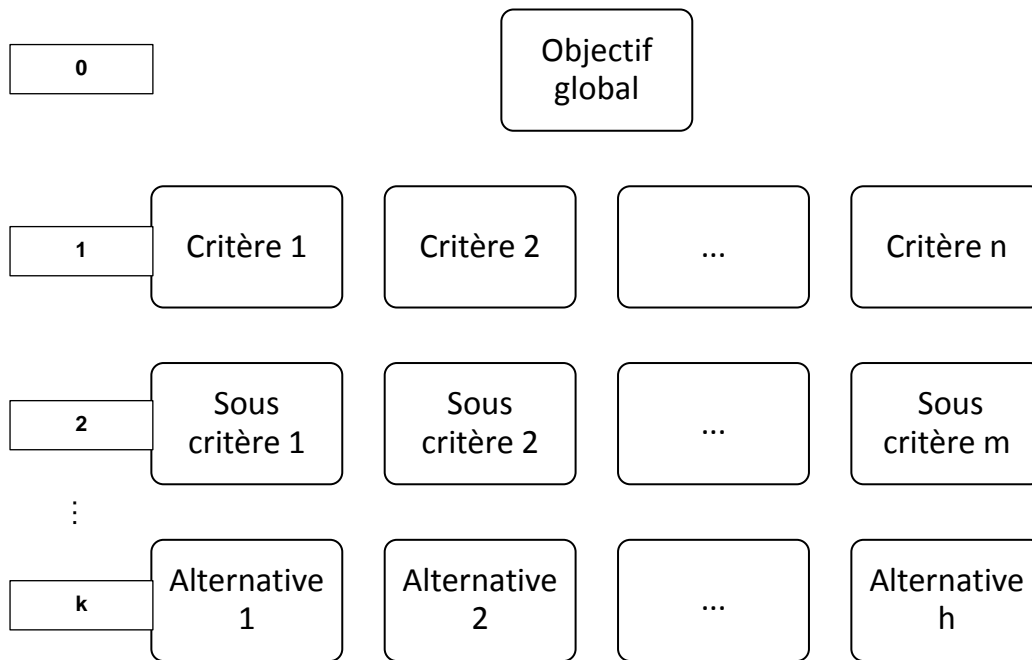


Figure 47 Découpage en critère pour utilisation AHP

- 1) « Diviser la décision en une hiérarchie de critères interconnectés » : créer une hiérarchie des critères de décision pouvant se répartir sur plusieurs niveaux.
- 2) Rassembler les données d'entrées en comparant deux à deux les critères : Mettre sous forme de matrices les comparaisons deux à deux des critères de même niveau. Pour remplir ces matrices, on affecte la diagonale à 1 et on complète la partie triangulaire supérieure avec les valeurs établies pour la comparaison, la partie triangulaire inférieure contiendra les valeurs réciproques (1 / valeur triangulaire supérieure).
- 3) Utiliser la méthode des valeurs propres pour réaliser une estimation des poids relatifs de chaque critère, cela afin de normer les matrices et de pouvoir réaliser l'agrégation de l'étape 4.
- 4) Agréger les poids relatifs afin d'obtenir une liste avec les scores de chaque alternative ».

La technique AHP est utilisée dans de nombreux domaines, dont l'ordonnancement multicritères. (Chan et al., 2005) a, par exemple, couplé l'AHP à un algorithme génétique pour résoudre un problème de production-distribution. (Azadeh et al., 2008) a quant à lui associé AHP à une analyse par enveloppement de données (DEA) dans le cadre de simulations pour déterminer des optimums alternatifs en présence de nombreuses variables quantitatives et qualitatives. (Momoh and Jizhong Zhu, 2003) propose une application d'AHP et d'Analytic Network Process (ANP) pour prioriser, ordonnancer et optimiser l'attribution des prix de puissance électrique.

Avantages :

La technique AHP permet de prendre en considération aussi bien des critères qualitatifs que quantitatifs.

Inconvénients :

Le nombre de paramètres à régler est supérieur ou égal à $(m+1) * n$, où n est le nombre de critères à considérer et m le nombre de valeurs que l'on souhaite utiliser pour la comparaison deux à deux des solutions pour chaque critère. L'implémentation peut potentiellement demander un nombre important de tests avant d'obtenir un résultat satisfaisant si l'algorithme est utilisé tel quel.

Conclusion :

Pour le premier optimisateur local, nous appliquerons une somme pondérée pour traiter le fait que le problème est multicritère.

Pour le second ordonnanceur, nous choisirons de mettre en œuvre une technique distribuée afin de valider la caractéristique d'hybridation de notre système de pilotage, pour cela, nous utiliserons la méthode AHP décrite précédemment.

Maintenant, nous allons explorer quelques-unes des méthodes permettant de synchroniser les ordonnanceurs qui nous allons réaliser.

IV. ANNEXE 4 : résultats comparaison MA et GA

Groupe 1 :

Données d'entrées :

Nombre de commandes : C=9

Commande	Nombre de lots	Épaisseur des lots	Surface des lots
C ₀	1	19	0.12
C ₁	1	19	0.12
C ₂	1	16	10.77
C ₃	1	19	0.27
C ₄	1	16	6.57
C ₅	1	19	7.9
C ₆	2	19, 16	4.19, 2.28
C ₇	1	19	0.46
C ₈	1	19	0.27

Tableau 29 Commandes groupe 1

Résultats :

Avec les 2 méthodes, on obtient trois batches.

Algorithme génétique (GA)			Mathématique analytique (MA)		
Batch 1 (19 mm)	Surface 4.3m ²		Batch 1 (16 mm)	13.05 m ²	
C ₁ lot 0			C ₂ lot 0		
C ₆ lot 0			C ₆ lot 1		
Batch 2 (16 mm)	Surface 19.62 m ²		Batch 2 (16 mm)	Surface 6.57 m ²	
C ₆ lot 1			C ₄ lot 0		
C ₄ lot 0					
C ₂ lot 0					
Batch 3 (19 mm)	Surface 9.02 m ²		Batch 3 (19 mm)	Surface 13.33 m ²	
C ₅ lot 0			C ₅ lot 0		
C ₇ lot 0			C ₇ lot 0		
C ₃ lot 0			C ₃ lot 0		
C ₀ lot 0			C ₀ lot 0		
C ₈ lot 0			C ₈ lot 0		
	C ₁ lot 0				
	C ₆ lot 0				
Séquencement :1-2-3			Séquencement :3-1-2		
W _i = 0 h			W _i =11.847 h		

Tableau 30 Résultats groupe 1

Le séquencement obtenu avec la résolution analytique est plus optimal car il réalise un setup de moins que l'algorithme génétique. En outre, le batch 2 est d'une surface totale supérieure au seuil voulu (cela est dû au fait que dans le modèle génétique, la surface totale d'un batch est une contrainte plus faible que les autres, on autorise un dépassement si la solution est de meilleure qualité). Par contre on peut constater que la contrainte qui limitait le nombre de commandes différentes à 5 dans un même batch n'est pas respectée par la méthode analytique car la contrainte a été relaxée.

Du point de vue temps de calcul, les deux méthodes ont une seconde d'écart : 9s pour le GA contre 10 pour le MA.

Groupe 2 :

Données d'entrées :

Nombre de commandes : C=9

Commande	Nombre de lots	Épaisseur des lots	Surface des lots
C ₀	2	28,19	3.67,11.88
C ₁	5	28,19,19,16,28	0.34,1.77,5.3,1.77,0.52
C ₂	3	19,19,16	0.95,0.62,0.94
C ₃	1	16	0.14
C ₄	2	16,60	1,1.65
C ₅	1	19	0.12
C ₆	2	19, 19	0.45,8.18
C ₇	2	19,19	7.03,0.17
C ₈	1	19	1.57

Tableau 31 Commandes groupe 2

Résultats :

Avec la MA, on obtient 6 batches pour 9 pour le GA. Cet écart est dû au fait que le GA tient compte d'une règle supplémentaire que nous n'avons pas incluse (certaines couleurs ont des gammes de finitions différentes des autres, pour cette raison, nous avons décidé de les isoler des autres).

Algorithme génétique (GA)			Mathématique analytique (MA)				
Batch 1(28 mm)	Surface 0.86 m ²		Batch 3(60 mm)	1.65 m ²			
C ₁ lot 0			C ₄ lot 1				
C ₁ lot 4							
Batch 2 (19 mm)	Surface 8.76 m ²		Batch 4 (19 mm)	Surface 15.76 m ²			
C ₁ lot 1			C ₀ lot 1				
C ₁ lot 2			C ₂ lot 0				
C ₅ lot 0			C ₂ lot 1				
C ₂ lot 0			C ₅ lot 0				
C ₂ lot 1			C ₆ lot 0				
			C ₇ lot 1				
			C ₈ lot 0				
Batch 3 (16 mm)	Surface 3.85 m ²		Batch 0 (16 mm)	Surface 3.85m ²			
C ₁ lot 3			C ₁ lot 3				
C ₂ lot 2			C ₂ lot 2				
C ₃ lot 0			C ₃ lot 0				
C ₄ lot 0			C ₄ lot 0				
Batch 4 (60 mm)	Surface 1.65 m ²		Batch 1 (28 mm)	Surface 4.53 m ²			
C ₄ lot 1			C ₀ lot 0				
			C ₁ lot 0				
			C ₁ lot 4				
Batch 5 (28 mm)	Surface 3.67 m ²		Batch 2 (19 mm)	Surface 16.98 m ²			
C ₀ lot 0			C ₁ lot 1				
			C ₆ lot 1				
			C ₇ lot 0				
Batch 6 (19 mm)	Surface 11.87 m ²		Batch 5 (19 mm)	Surface 5.3 m ²			
C ₀ lot 1			C ₁ lot 2				
Batch 7 (19 mm)	Surface 9.04 m ²						
C ₈ lot 0							
C ₇ lot 0							
C ₆ lot 0							

Batch 8 (19 mm)	Surface 8.17 m ²		
C ₆ lot 1			
Batch 9 (19 mm)	Surface 3.67 m ²		
C ₇ lot 1			
Séquencement :		Séquencement :	
1-2-3-4-5-6-7-8-9		3-4-0-1-2-5	
W _i = 13.893 h		W _i = 91.487 h	

Tableau 32 Résultats groupe 2

Pour le second groupe de données, les deux méthodes minimisent bien le nombre de setup, elles respectent également toutes les deux la limitation de la surface maximale d'un batch. Par contre, il y a, à nouveau, un batch du MA qui dépasse la limite des cinq commandes maximales dans le même batch. Le GA fournit sa solution en 15 secondes alors que pour ce groupe de données là le MA a été stoppé manuellement après 45 minutes.

Groupe 3 :

Données d'entrées :

Nombre de commandes : C=10

Commande	Nombre de lots	Épaisseur des lots	Surface des lots
C ₀	2	19,19	0.04,1.64
C ₁	3	19,19,19	4.74,0.05,0.79
C ₂	2	19,19	0.88,7.21
C ₃	3	19,16,19	16.37,4.36,1.67
C ₄	3	19,19,19	6.16,3.82,1.76
C ₅	2	19,19	2.19,0.11
C ₆	3	19, 19	1.88,1.75,5.22
C ₇	2	19,19,19	3.08,1.46
C ₈	2	19,19	8.39,0.81
C ₉	1	19	1.81

Tableau 33 Commandes groupe 3

Résultats :

Avec la MA, on obtient 6 batches contre 8 pour le GA.

Algorithme génétique (GA)		Mathématique analytique (MA)																		
<table border="1"> <tr> <td>Batch 1(19 mm)</td> <td rowspan="6">Surface 11.19 m²</td> </tr> <tr> <td>C₈ lot 0</td> </tr> <tr> <td>C₆ lot 0</td> </tr> <tr> <td>C₂ lot 0</td> </tr> <tr> <td>C₀ lot 0</td> </tr> </table>		Batch 1(19 mm)	Surface 11.19 m ²	C ₈ lot 0	C ₆ lot 0	C ₂ lot 0	C ₀ lot 0	<table border="1"> <tr> <td>Batch 4 (19 mm)</td> <td rowspan="10">Surface 16.98 m²</td> </tr> <tr> <td>C₀ lot 0</td> </tr> <tr> <td>C₁ lot 2</td> </tr> <tr> <td>C₃ lot 2</td> </tr> <tr> <td>C₄ lot 0</td> </tr> <tr> <td>C₄ lot 2</td> </tr> <tr> <td>C₅ lot 0</td> </tr> <tr> <td>C₆ lot 1</td> </tr> <tr> <td>C₈ lot 1</td> </tr> <tr> <td>C₉ lot 0</td> </tr> </table>		Batch 4 (19 mm)	Surface 16.98 m ²	C ₀ lot 0	C ₁ lot 2	C ₃ lot 2	C ₄ lot 0	C ₄ lot 2	C ₅ lot 0	C ₆ lot 1	C ₈ lot 1	C ₉ lot 0
Batch 1(19 mm)	Surface 11.19 m ²																			
C ₈ lot 0																				
C ₆ lot 0																				
C ₂ lot 0																				
C ₀ lot 0																				
Batch 4 (19 mm)		Surface 16.98 m ²																		
C ₀ lot 0																				
C ₁ lot 2																				
C ₃ lot 2																				
C ₄ lot 0																				
C ₄ lot 2																				
C ₅ lot 0																				
C ₆ lot 1																				
C ₈ lot 1																				
C ₉ lot 0																				
<table border="1"> <tr> <td>Batch 2 (19 mm)</td> <td rowspan="4">Surface 4.19 m²</td> </tr> <tr> <td>C₆ lot 1</td> </tr> <tr> <td>C₀ lot 1</td> </tr> <tr> <td>C₈ lot 1</td> </tr> </table>		Batch 2 (19 mm)	Surface 4.19 m ²	C ₆ lot 1	C ₀ lot 1	C ₈ lot 1	<table border="1"> <tr> <td>Batch 1(16 mm)</td> <td rowspan="2">4.36 m²</td> </tr> <tr> <td>C₃ lot 1</td> </tr> </table>		Batch 1(16 mm)	4.36 m ²	C ₃ lot 1									
Batch 2 (19 mm)	Surface 4.19 m ²																			
C ₆ lot 1																				
C ₀ lot 1																				
C ₈ lot 1																				
Batch 1(16 mm)	4.36 m ²																			
C ₃ lot 1																				
<table border="1"> <tr> <td>Batch 3 (19 mm)</td> <td rowspan="4">Surface 14.23 m²</td> </tr> <tr> <td>C₂ lot 1</td> </tr> <tr> <td>C₆ lot 2</td> </tr> <tr> <td>C₉ lot 0</td> </tr> </table>		Batch 3 (19 mm)	Surface 14.23 m ²	C ₂ lot 1	C ₆ lot 2	C ₉ lot 0	<table border="1"> <tr> <td>Batch 5 (19 mm)</td> <td rowspan="4">Surface 6.67m²</td> </tr> <tr> <td>C₁ lot 0</td> </tr> <tr> <td>C₁ lot 1</td> </tr> <tr> <td>C₆ lot 0</td> </tr> </table>		Batch 5 (19 mm)	Surface 6.67m ²	C ₁ lot 0	C ₁ lot 1	C ₆ lot 0							
Batch 3 (19 mm)	Surface 14.23 m ²																			
C ₂ lot 1																				
C ₆ lot 2																				
C ₉ lot 0																				
Batch 5 (19 mm)	Surface 6.67m ²																			
C ₁ lot 0																				
C ₁ lot 1																				
C ₆ lot 0																				
<table border="1"> <tr> <td>Batch 4 (19 mm)</td> <td rowspan="4">Surface 27.27 m²</td> </tr> <tr> <td>C₃ lot 0</td> </tr> <tr> <td>C₄ lot 0</td> </tr> <tr> <td>C₁ lot 0</td> </tr> </table>		Batch 4 (19 mm)	Surface 27.27 m ²	C ₃ lot 0	C ₄ lot 0	C ₁ lot 0	<table border="1"> <tr> <td>Batch 2 (19 mm)</td> <td rowspan="4">Surface 15.71 m²</td> </tr> <tr> <td>C₂ lot 1</td> </tr> <tr> <td>C₅ lot 1</td> </tr> <tr> <td>C₈ lot 0</td> </tr> </table>		Batch 2 (19 mm)	Surface 15.71 m ²	C ₂ lot 1	C ₅ lot 1	C ₈ lot 0							
Batch 4 (19 mm)	Surface 27.27 m ²																			
C ₃ lot 0																				
C ₄ lot 0																				
C ₁ lot 0																				
Batch 2 (19 mm)	Surface 15.71 m ²																			
C ₂ lot 1																				
C ₅ lot 1																				
C ₈ lot 0																				
<table border="1"> <tr> <td>Batch 5 (16 mm)</td> <td rowspan="2">Surface 4.36 m²</td> </tr> <tr> <td>C₃ lot 1</td> </tr> </table>		Batch 5 (16 mm)	Surface 4.36 m ²	C ₃ lot 1	<table border="1"> <tr> <td>Batch 0 (19 mm)</td> <td rowspan="7">Surface 16.1 m²</td> </tr> <tr> <td>C₀ lot 1</td> </tr> <tr> <td>C₂ lot 0</td> </tr> <tr> <td>C₄ lot 1</td> </tr> <tr> <td>C₆ lot 2</td> </tr> <tr> <td>C₇ lot 1</td> </tr> <tr> <td>C₇ lot 0</td> </tr> </table>		Batch 0 (19 mm)	Surface 16.1 m ²	C ₀ lot 1	C ₂ lot 0	C ₄ lot 1	C ₆ lot 2	C ₇ lot 1	C ₇ lot 0						
Batch 5 (16 mm)	Surface 4.36 m ²																			
C ₃ lot 1																				
Batch 0 (19 mm)	Surface 16.1 m ²																			
C ₀ lot 1																				
C ₂ lot 0																				
C ₄ lot 1																				
C ₆ lot 2																				
C ₇ lot 1																				
C ₇ lot 0																				

Batch 6 (19 mm)	Surface 9.08 m ²		<table border="1"> <tbody> <tr> <td>Batch 3 (19 mm)</td> <td rowspan="2">Surface 16.37 m²</td> </tr> <tr> <td>C₃ lot 0</td> </tr> </tbody> </table>	Batch 3 (19 mm)	Surface 16.37 m ²	C ₃ lot 0
Batch 3 (19 mm)				Surface 16.37 m ²		
C ₃ lot 0						
C ₄ lot 1						
C ₇ lot 0						
C ₅ lot 0						
Batch 7 (19 mm)	Surface 0.05 m ²					
C ₁ lot 1						
Batch 8 (19 mm)	Surface 5.78 m ²					
C ₄ lot 2						
C ₃ lot 2						
C ₇ lot 1						
C ₁ lot 2						
C ₅ lot 1						
Séquencement :		Séquencement :				
1-2-3-4-5-6-7-8		4-1-5-2-0-3				
W _i = 97.592 h		W _i = 249.181 h				

Tableau 34 Résultats groupe 3

Le séquencement proposé par le MA fait gagner un temps de setup par rapport au GA mais il y a, à nouveau, un batch (batch 4) qui ne respecte pas la limite du nombre de commandes admissibles pour le MA et un pour le GA qui ne respecte pas le seuil de surface. D'un point de vue calculatoire, le MA a été stoppé manuellement après 45 minutes tandis que le GA a retourné le résultat en 15 secondes.

Groupe 4 :

Données d'entrées :

Nombre de commandes : C=10

Commande	Nombre de lots	Épaisseur des lots	Surface des lots
C ₀	1	19	0.12
C ₁	6	28,12,19,16,19,19	2.27,0.24,6.98,4.12,0.08,0.05
C ₂	2	19,19	6.62,3.36
C ₃	2	19,19	3.09,1.29
C ₄	7	28,19,19,16,19,19,19	8.96,6.02,0.78,12.84,4.22,0.04,1.77
C ₅	4	28,12,19,16	3.51,0.36,3.68,2.95
C ₆	3	28,19,19	5.34,2.66,4.57
C ₇	1	19	0.26
C ₈	1	28	2.31
C ₉	3	28,19,16	1.34,0.98,0.85

Tableau 35 Commandes groupe 4

Résultats :

Avec la MA, on obtient 8 batches contre 9 pour le GA.

Algorithme génétique (GA)		Mathématique analytique (MA)																	
<table border="1"> <tr> <td>Batch 1(28 mm)</td> <td rowspan="3">Surface 10.29 m²</td> </tr> <tr> <td>C₉ lot 0</td> </tr> <tr> <td>C₄ lot 0</td> </tr> </table>		Batch 1(28 mm)	Surface 10.29 m ²	C ₉ lot 0	C ₄ lot 0	<table border="1"> <tr> <td>Batch 0(19 mm)</td> <td rowspan="7">14.77 m²</td> </tr> <tr> <td>C₁ lot 2</td> </tr> <tr> <td>C₁ lot 4</td> </tr> <tr> <td>C₁ lot 5</td> </tr> <tr> <td>C₃ lot 0</td> </tr> <tr> <td>C₅ lot 2</td> </tr> <tr> <td>C₉ lot 1</td> </tr> </table>		Batch 0(19 mm)	14.77 m ²	C ₁ lot 2	C ₁ lot 4	C ₁ lot 5	C ₃ lot 0	C ₅ lot 2	C ₉ lot 1				
Batch 1(28 mm)	Surface 10.29 m ²																		
C ₉ lot 0																			
C ₄ lot 0																			
Batch 0(19 mm)	14.77 m ²																		
C ₁ lot 2																			
C ₁ lot 4																			
C ₁ lot 5																			
C ₃ lot 0																			
C ₅ lot 2																			
C ₉ lot 1																			
<table border="1"> <tr> <td>Batch 2 (19 mm)</td> <td rowspan="6">Surface 14.52 m²</td> </tr> <tr> <td>C₂ lot 0</td> </tr> <tr> <td>C₄ lot 1</td> </tr> <tr> <td>C₉ lot 1</td> </tr> <tr> <td>C₀ lot 0</td> </tr> <tr> <td>C₄ lot 2</td> </tr> </table>		Batch 2 (19 mm)	Surface 14.52 m ²	C ₂ lot 0	C ₄ lot 1	C ₉ lot 1	C ₀ lot 0	C ₄ lot 2	<table border="1"> <tr> <td>Batch 1 (19 mm)</td> <td rowspan="8">Surface 16.25 m²</td> </tr> <tr> <td>C₂ lot 1</td> </tr> <tr> <td>C₃ lot 1</td> </tr> <tr> <td>C₄ lot 2</td> </tr> <tr> <td>C₄ lot 4</td> </tr> <tr> <td>C₄ lot 6</td> </tr> <tr> <td>C₆ lot 2</td> </tr> <tr> <td>C₇ lot 0</td> </tr> </table>		Batch 1 (19 mm)	Surface 16.25 m ²	C ₂ lot 1	C ₃ lot 1	C ₄ lot 2	C ₄ lot 4	C ₄ lot 6	C ₆ lot 2	C ₇ lot 0
Batch 2 (19 mm)	Surface 14.52 m ²																		
C ₂ lot 0																			
C ₄ lot 1																			
C ₉ lot 1																			
C ₀ lot 0																			
C ₄ lot 2																			
Batch 1 (19 mm)	Surface 16.25 m ²																		
C ₂ lot 1																			
C ₃ lot 1																			
C ₄ lot 2																			
C ₄ lot 4																			
C ₄ lot 6																			
C ₆ lot 2																			
C ₇ lot 0																			
<table border="1"> <tr> <td>Batch 3 (16 mm)</td> <td rowspan="3">Surface 13.69 m²</td> </tr> <tr> <td>C₄ lot 3</td> </tr> <tr> <td>C₉ lot 2</td> </tr> </table>		Batch 3 (16 mm)	Surface 13.69 m ²	C ₄ lot 3	C ₉ lot 2	<table border="1"> <tr> <td>Batch 2 (19 mm)</td> <td rowspan="6">Surface 15.46 m²</td> </tr> <tr> <td>C₀ lot 0</td> </tr> <tr> <td>C₂ lot 0</td> </tr> <tr> <td>C₄ lot 1</td> </tr> <tr> <td>C₄ lot 5</td> </tr> <tr> <td>C₆ lot 1</td> </tr> </table>		Batch 2 (19 mm)	Surface 15.46 m ²	C ₀ lot 0	C ₂ lot 0	C ₄ lot 1	C ₄ lot 5	C ₆ lot 1					
Batch 3 (16 mm)	Surface 13.69 m ²																		
C ₄ lot 3																			
C ₉ lot 2																			
Batch 2 (19 mm)	Surface 15.46 m ²																		
C ₀ lot 0																			
C ₂ lot 0																			
C ₄ lot 1																			
C ₄ lot 5																			
C ₆ lot 1																			
<table border="1"> <tr> <td>Batch 4 (19 mm)</td> <td rowspan="6">Surface 9.65 m²</td> </tr> <tr> <td>C₄ lot 4</td> </tr> <tr> <td>C₇ lot 0</td> </tr> <tr> <td>C₄ lot 5</td> </tr> <tr> <td>C₂ lot 1</td> </tr> <tr> <td>C₄ lot 6</td> </tr> </table>		Batch 4 (19 mm)	Surface 9.65 m ²	C ₄ lot 4	C ₇ lot 0	C ₄ lot 5	C ₂ lot 1	C ₄ lot 6	<table border="1"> <tr> <td>Batch 3 (28 mm)</td> <td rowspan="5">Surface 16.12 m²</td> </tr> <tr> <td>C₄ lot 0</td> </tr> <tr> <td>C₅ lot 0</td> </tr> <tr> <td>C₈ lot 0</td> </tr> <tr> <td>C₉ lot 0</td> </tr> </table>		Batch 3 (28 mm)	Surface 16.12 m ²	C ₄ lot 0	C ₅ lot 0	C ₈ lot 0	C ₉ lot 0			
Batch 4 (19 mm)	Surface 9.65 m ²																		
C ₄ lot 4																			
C ₇ lot 0																			
C ₄ lot 5																			
C ₂ lot 1																			
C ₄ lot 6																			
Batch 3 (28 mm)	Surface 16.12 m ²																		
C ₄ lot 0																			
C ₅ lot 0																			
C ₈ lot 0																			
C ₉ lot 0																			

<table border="1"> <tr><td>Batch 5 (28 mm)</td><td rowspan="5">Surface 13.43 m²</td></tr> <tr><td>C₆ lot 0</td></tr> <tr><td>C₅ lot 0</td></tr> <tr><td>C₈ lot 0</td></tr> <tr><td>C₁ lot 0</td></tr> </table>	Batch 5 (28 mm)	Surface 13.43 m ²	C ₆ lot 0	C ₅ lot 0	C ₈ lot 0	C ₁ lot 0	<table border="1"> <tr><td>Batch 4 (12 mm)</td><td rowspan="3">Surface 0.6 m²</td></tr> <tr><td>C₁ lot 1</td></tr> <tr><td>C₅ lot 1</td></tr> </table>	Batch 4 (12 mm)	Surface 0.6 m ²	C ₁ lot 1	C ₅ lot 1
Batch 5 (28 mm)	Surface 13.43 m ²										
C ₆ lot 0											
C ₅ lot 0											
C ₈ lot 0											
C ₁ lot 0											
Batch 4 (12 mm)	Surface 0.6 m ²										
C ₁ lot 1											
C ₅ lot 1											
<table border="1"> <tr><td>Batch 6 (12 mm)</td><td rowspan="3">Surface 0.6 m²</td></tr> <tr><td>C₅ lot 1</td></tr> <tr><td>C₁ lot 1</td></tr> </table>	Batch 6 (12 mm)	Surface 0.6 m ²	C ₅ lot 1	C ₁ lot 1	<table border="1"> <tr><td>Batch 6 (16 mm)</td><td rowspan="3">Surface 4.97 m²</td></tr> <tr><td>C₁ lot 3</td></tr> <tr><td>C₉ lot 2</td></tr> </table>	Batch 6 (16 mm)	Surface 4.97 m ²	C ₁ lot 3	C ₉ lot 2		
Batch 6 (12 mm)	Surface 0.6 m ²										
C ₅ lot 1											
C ₁ lot 1											
Batch 6 (16 mm)	Surface 4.97 m ²										
C ₁ lot 3											
C ₉ lot 2											
<table border="1"> <tr><td>Batch 7 (19 mm)</td><td rowspan="4">Surface 13.23 m²</td></tr> <tr><td>C₁ lot 2</td></tr> <tr><td>C₅ lot 2</td></tr> <tr><td>C₆ lot 1</td></tr> </table>	Batch 7 (19 mm)	Surface 13.23 m ²	C ₁ lot 2	C ₅ lot 2	C ₆ lot 1	<table border="1"> <tr><td>Batch 5 (16 mm)</td><td rowspan="3">Surface 15.79 m²</td></tr> <tr><td>C₄ lot 3</td></tr> <tr><td>C₅ lot 3</td></tr> </table>	Batch 5 (16 mm)	Surface 15.79 m ²	C ₄ lot 3	C ₅ lot 3	
Batch 7 (19 mm)	Surface 13.23 m ²										
C ₁ lot 2											
C ₅ lot 2											
C ₆ lot 1											
Batch 5 (16 mm)	Surface 15.79 m ²										
C ₄ lot 3											
C ₅ lot 3											
<table border="1"> <tr><td>Batch 8 (16 mm)</td><td rowspan="3">Surface 7.07 m²</td></tr> <tr><td>C₁ lot 3</td></tr> <tr><td>C₅ lot 3</td></tr> </table>	Batch 8 (16 mm)	Surface 7.07 m ²	C ₁ lot 3	C ₅ lot 3	<table border="1"> <tr><td>Batch 7 (19 mm)</td><td rowspan="3">Surface 7.61 m²</td></tr> <tr><td>C₁ lot 0</td></tr> <tr><td>C₆ lot 0</td></tr> </table>	Batch 7 (19 mm)	Surface 7.61 m ²	C ₁ lot 0	C ₆ lot 0		
Batch 8 (16 mm)	Surface 7.07 m ²										
C ₁ lot 3											
C ₅ lot 3											
Batch 7 (19 mm)	Surface 7.61 m ²										
C ₁ lot 0											
C ₆ lot 0											
<table border="1"> <tr><td>Batch 9 (19 mm)</td><td rowspan="6">Surface 9.07 m²</td></tr> <tr><td>C₆ lot 2</td></tr> <tr><td>C₃ lot 0</td></tr> <tr><td>C₁ lot 4</td></tr> <tr><td>C₁ lot 5</td></tr> <tr><td>C₃ lot 1</td></tr> </table>	Batch 9 (19 mm)	Surface 9.07 m ²	C ₆ lot 2	C ₃ lot 0	C ₁ lot 4	C ₁ lot 5	C ₃ lot 1				
Batch 9 (19 mm)	Surface 9.07 m ²										
C ₆ lot 2											
C ₃ lot 0											
C ₁ lot 4											
C ₁ lot 5											
C ₃ lot 1											
Séquencement : 1-2-3-4-5-6-7-8-9	Séquencement : 0-1-2-3-4-6-5-7										
W _i = 149.156 h	W _i = 299.728 h										

Tableau 36 Résultats groupe 4

Le séquencement proposé par le MA fait gagner quatre temps de setup par rapport au GA. La contrainte de surface est respectée dans les deux cas. D'un point de vue calculatoire, le MA a été stoppé manuellement après 45 minutes tandis que le GA a retourné le résultat en 14 secondes.

Groupe 5 :

Données d'entrées :

Nombre de commandes : C=5

Commandes	Nombre de lots	Épaisseur des lots	Surface des lots
C ₀	1	19	0.59
C ₁	1	19	4.34
C ₂	1	19	2.01
C ₃	1	19	0.45
C ₄	1	19	1.41

Tableau 37 Commandes groupe 5

Résultats :

Avec la MA et le GA on obtient un seul batch de surface 8.8m² avec un résultat obtenu en 4 secondes pour le GA contre un résultat de 5 secondes pour la MA.

RÉSUMÉ

Cette thèse CIFRE issue d'une collaboration entre Acta-Mobilier, fabricant de façades laquées haut de gamme et le CRAN. Cette thèse, s'inscrivant dans la continuité de celle de Mélanie Noyel, a pour objectif la réalisation d'une architecture de pilotage hybride s'appuyant sur le contrôle par le produit. Nous avons choisi de nous inspirer d'un méta-modèle développé au sein de l'équipe du CRAN. Cette architecture repose sur un modèle VSM, où chaque niveau est susceptible de prendre des décisions à son échelle. Le plus haut niveau supervise les décisions tactiques (par exemple, le plan directeur de production), le plus bas niveau d'intelligence est distribué entre les produits dotés de moyen de communication et d'analyse (leur intelligence est attribuée à la réactivité du système car se trouvant au plus proche du besoin).

En niveaux intermédiaires, nous trouvons d'une part des optimisateurs centralisés pour superviser les sous-ateliers de la chaîne de production, de manière à atteindre les objectifs de consommation, de temps de réglages ou de productivité des sous-ateliers qu'ils supervisent. D'une autre part, des optimisateurs pour des postes de travail spécifique faisant intervenir directement les produits et les informations qu'ils possèdent dans la prise de décision.

Un optimisateur de chacune des deux catégories en accord avec les besoins définis par l'étude du flux de production de l'entreprise ont été réalisés. Un optimisateur centralisé a été réalisé pour l'atelier usinage dans lequel les opérations relatives au débit et au façonnage des formes des produits sont effectuées. Il est construit pour réaliser des regroupements en lots de fabrication et à les ordonnancer pour obtenir un compromis entre minimisation des consommations matières, des temps de réglage des machines et la minimisation du WIP. Le séquençement doit garantir que les regroupements, lors de leur division permettent de reformer rapidement les commandes clients. Cette contrainte est nécessaire, avant passage à la phase d'application de la finition. Il met en œuvre un algorithme génétique solutionnant un problème d'ordonnement multicritères. Pour valider notre choix d'une méta-heuristique comme méthode de résolution du problème, nous avons tenté de le résoudre par une méthode de mathématique analytique et les résultats obtenus ont confirmé que notre décision était raisonnable. Cette optimisation a été testée sur plateforme de test et a fourni des résultats encourageants. Une implémentation faite dans l'entreprise, est utilisée chaque semaine pour une planification spécifique.

Un optimisateur de la seconde catégorie a été étudié pour gérer le cas du robot de laquage, celui-ci doit fournir deux postes clients ayant leurs propres familles de produits mais devant être expédiées aux mêmes dates. En outre, les points faibles de ce poste, à savoir la consommation importante de laque à chaque changement de couleur et la longue durée d'attente avant de pouvoir visualiser les produits et savoir si un défaut qualité est apparu, impliquant de devoir refaire un cycle complet de laquage. L'optimisateur utilise un modèle de prédiction de non qualité afin d'évaluer les risques relatifs au passage du prochain lot à produire et si celui-ci est jugé trop élevé, un processus est déclenché choisissant parmi les lots présents en file d'attente, le plus adapté en considérant plusieurs facteurs. Cet optimisateur de poste de travail recueille des informations de la machine, des produits et des files d'attentes des postes en aval afin d'empêcher l'apparition d'un problème. Il a été implémenté sur un modèle de simulation. La question de la synchronisation des différents optimisateurs a été amorcée. En effet, le plan de production généré par le système d'information donne une plage de passage acceptable pour les lots dans chaque atelier et les optimisateurs se doivent de la respecter. Ces travaux aideront l'entreprise à franchir sa transition vers l'ère de l'industrie 4.0.

ABSTRACT

This CIFRE thesis comes from a collaboration between Acta-Mobilier, manufacturer of high-end lacquered facades and CRAN. This thesis, which is a continuation of that of Mélanie Noyel, aims to achieve a hybrid control architecture based on control by the product. We chose to take inspiration from a meta-model developed within the CRAN team. This architecture is based on a VSM model, where each level is likely to make decisions on its own scale. The highest level oversees tactical decisions (for example, the production master plan), the lowest level of intelligence is distributed between the products endowed with means of communication and analysis (their intelligence is attributed to the responsiveness of the system because being closer to the need). In intermediate levels, we find on the one hand centralized optimizers to supervise the sub-workshops of the production chain, in order to reach the objectives of consumption, time of adjustments or productivity of the sub-workshops that they supervise. On the other hand, optimizers for specific workstations directly involving the products and information they possess in decision-making.

An optimizer of each of the two categories in accordance with the needs defined by the study of the workflow of the company have been realized. A centralized optimizer has been realized for the machining workshop in which the operations relating to the flow and shaping of the shapes of the products are carried out. It is built to make groupings in manufacturing batches and to schedule them to achieve a compromise between minimizing material consumption, machine setting times and minimizing WIP. Sequencing must ensure that clusters, when they are split, enable rapid customer order reform. This constraint is necessary before going to the application phase of the finish. It implements a genetic algorithm solving a multicriteria scheduling problem. To validate our choice of a meta-heuristic as a method of solving the problem, we tried to solve it by an analytical mathematical method and the results obtained confirmed that our decision was reasonable. This optimization was tested on a test platform and provided encouraging results. An implementation made in the company, is used every week for a specific planning

An optimizer of the second category has been studied to manage the case of the lacquer robot, it must provide two client stations having their own families of products but to be shipped on the same dates. In addition, the weak points of this post, namely the significant consumption of lacquer with each change of color and the long waiting time before being able to visualize the products and to know if a defect quality appeared, involving having to redo a cycle complete lacquering. The optimizer uses a non-quality prediction model to evaluate the risks associated with the passage of the next batch to be produced and if it is deemed too high, a process is triggered choosing among the lots present in the queue, the most suitable by considering several factors. This workstation optimizer gathers machine information, products, and queues from downstream workstations to prevent a problem from occurring. It has been implemented on a simulation model. The issue of synchronization of different optimizers has been initiated. Indeed, the production plan generated by the information system gives an acceptable range of passage for the batches in each workshop and the optimizers must respect it. This work will help the company make the transition to the Industry 4.0 era.