



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par Université Toulouse II Jean Jaurès

Présentée et soutenue par

Damien LOVATO

Le 21 novembre 2022

Développement de critères d'optimisation pour la planification de tâches dans une chaîne d'assemblage aéronautique

École doctorale

EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications
de Toulouse

Spécialité

Mathématiques et Applications

Unité de recherche

IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par

Romain GUILLAUME et Olga BATAÏA

Jury

Mme Sylvie NORRE, Rapporteuse
Mme Hind EL HAOUZI, Rapporteuse
M. Evgeny GUREVSKY, Examineur
M. Jean-Charles BILLAUT, Examineur
M. Romain GUILLAUME, Directeur de thèse
Mme Caroline THIERRY, Co-directrice de thèse
Mme Olga BATAÏA, Co-directrice de thèse

Université Toulouse 2-Jean Jaurès
Laboratoire I R I T

THÈSE

Pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ
en Mathématiques et Applications

Développement de critères d'optimisation
pour la planification de tâches dans une
chaîne d'assemblage aéronautique

Damien LOVATO

Présentée et soutenue publiquement

Le 21 Novembre 2022

Thèse dirigée par

Romain GUILLAUME et Olga BATAÏA

Jury

Mme Sylvie NORRE, Rapporteur
Mme Hind EL HAOUZI, Rapporteur
M. Evgeny GUREVSKY, Examineur
M. Jean-Charles BILLAUT, Examineur
M. Romain GUILLAUME, Directeur de thèse
Mme Caroline THIERRY, Co-directrice de thèse
Mme Olga BATAÏA, Co-directrice de thèse

Remerciements

Je voudrais dans un premier temps remercier l'IRIT et l'ANR pour avoir financé et permis la réalisation de ce projet et de ma thèse.

Merci à Dassault Aviation de nous avoir accordé leur confiance lors de ce projet et pour les échanges constructifs que nous avons pu avoir tout au long de celui-ci. Merci à Philippe Walter pour avoir coordonné les nombreuses réunions qui ont permis ces échanges et pour avoir répondu à nos nombreuses questions.

Merci à Sylvie Norre et Hind El Haouzi pour avoir pris le temps de relire mon manuscrit de thèse ainsi qu'à Evgeny Gurevsky et Jean-Charles Billaut pour avoir fait partie des membres du jury de soutenance et leurs retours.

Merci à mes directeur.ice.s de recherche pour m'avoir constamment poussé vers le haut et pour avoir créé une dynamique de travail riche qui m'a permis d'avancer. Merci à Olga Battaïa pour son expertise et le temps qu'elle a passé jusque tard le soir à relire et corriger mes articles. Merci à Romain Guillaume pour son temps passé avec moi afin de réfléchir à des solutions et la manière de les mettre en place dans mes travaux. Merci à Caroline Thierry pour tous ses retours sur mes travaux, mes méthodes et pour m'avoir écouté lors des moments de doute. Leurs contributions humaine et technique ont été très précieuses tout au long de la réalisation de cette thèse et je suis heureux d'avoir pu réaliser ce projet avec eux.

Enfin, merci à ma famille pour son soutien et son amour infaillibles. Merci à mes parents pour leurs nombreux sacrifices qui m'ont permis d'arriver jusqu'ici.

Sommaire

INTRODUCTION	11
1. PROCESSUS D'ASSEMBLAGE AERONAUTIQUE	17
1.1. PLANIFICATION D'UNE CHAINE D'ASSEMBLAGE AERONAUTIQUE	19
1.2. CHAINE D'ASSEMBLAGE AERONAUTIQUE	19
1.3. RESSOURCES DE PRODUCTION	21
1.4. REPRESENTATION DU PROCESSUS.....	22
1.5. PROBLEMATIQUE INDUSTRIELLE.....	25
2. ETAT DE L'ART	33
2.1. RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM	35
2.1.1. Extension des tâches.....	37
2.1.2. Extension des contraintes temporelles.....	38
2.1.3. Généralisation des ressources.....	38
2.1.4. Variations de la fonction objectif	40
2.2. ASSEMBLY LINE BALANCING PROBLEM (ALBP)	43
2.2.1. L'équilibrage des charges	43
2.2.2. La planification dans les lignes d'assemblage.....	43
2.2.3. Replanification des lignes d'assemblage.....	44
2.3. LES APPROCHES DE PLANIFICATION SOUS INCERTITUDE.....	45
2.3.1. L'incertitude en planification	45
2.3.2. La gestion de l'incertitude	46
2.4. L'ASPECT HUMAIN DANS LES CHAINES D'ASSEMBLAGE	50
2.4.1. Les indices pour la mesure des impacts physiologiques	51
2.5. PROBLEMATIQUE ET QUESTIONS DE RECHERCHE	53
3. MODELISATION DU PROBLEME DE REPLANIFICATION DES LIGNES D'ASSEMBLAGE AERONAUTIQUES	55
3.1. MOTIVATIONS.....	57
3.2. MODELE DE REPLANIFICATION.....	58
3.2.1. Fonctions et méthodes CP Optimizer	58
3.2.2. Les données en entrée.....	60
3.2.3. Les critères de décision.....	62

3.2.4. Les contraintes du problème	67
3.3. PREPARATION DES TESTS EXPERIMENTAUX	69
3.3.1. Création des instances	70
3.3.2. Etude des modes de résolution du solver	70
3.3.3. Tests de replanification	71
3.4. APPROFONDISSEMENT SUR DES INSTANCES DIVERSIFIEES	80
3.4.1. Définition des instances.....	80
3.4.2. Résultats pour DS1.....	83
3.4.3. Résultats pour DS2.....	83
3.5. CONCLUSIONS	84
4. INTEGRATION DE CRITERES ERGONOMIQUES DANS LA PLANIFICATION DES LIGNES D'ASSEMBLAGE AERONAUTIQUES	87
4.1. MOTIVATIONS	89
4.2. CRITERES D'EVALUATION DE LA CHARGE ERGONOMIQUE	91
4.2.1. La charge ergonomique.....	91
4.2.2. Charge ergonomique maximale	91
4.2.3. Indice Moyen de Déséquilibre.....	92
4.2.4. Exemples applicatifs de ces critères.....	93
4.3. TESTS EXPERIMENTAUX	96
4.3.1. Instance utilisée.....	96
4.3.2. Evolution des critères.....	96
4.3.3. Double critère ergonomique	98
4.4. BILAN.....	99
4.4.1. Réponse aux questions	99
4.4.2. Conclusion	100
5. PRISE EN COMPTE DES PREFERENCES ORGANISATIONNELLES DANS LA PLANIFICATION DES LIGNES D'ASSEMBLAGE AERONAUTIQUES.....	103
5.1. CONTEXTE ET MOTIVATION	105
5.2. INTEGRATION DANS LA PLANIFICATION	106
5.2.1. Méthode choisie	106
5.2.2. Modélisation.....	110
5.3. EVALUATION DU SCORE ORGANISATIONNEL	113

5.3.1. Présentation de l'instance	114
5.3.2. Test du critère de Choquet	115
5.3.3. Discussion des résultats.....	116
CONCLUSION ET PERSPECTIVES.....	119
REFERENCES.....	123

Liste des tableaux

Tableau 1 - AMDEC présentant les risques les plus impactant pouvant survenir	29
Tableau 2 - Description des effets de chaque aléa et de l'action effectuée en réponse...	30
Tableau 3 - Efficacité de chaque stratégie de résolution du solver	71
Tableau 4 - Caractéristiques de chaque tâche	93
Tableau 5 - Evolution de l'écart à l'optimum pour le critère SUM_Max en fonction du temps écoulé	98
Tableau 6 - Charges ergonomiques des compagnons après une heure de calcul suivant chaque méthode	99
Tableau 7 - Valeurs possibles pour chaque variable de décision.....	111
Tableau 8 - Résultats via les deux approches de résolution pour calculer une solution avec le critère de Choquet.....	116

Table des illustrations

Figure 1 - Acteurs du projet Per4mance.....	13
Figure 2 - Organigramme des tâches à accomplir au cours du projet Per4mance	14
Figure 3 - Visualisation des stations d'assemblage d'une chaîne d'assemblage aéronautique	20
Figure 4 - Exemple de délimitation de zones de travail	21
Figure 5 - Illustration des plages horaires de chaque type	22
Figure 6 – Illustration des précédences	24
Figure 7 - Arbre de défaillance lorsqu'une tâche doit être recommencée	26
Figure 8 - Arbre de défaillance pour une tâche retardée	27
Figure 9 - Classification des extensions du RCPSP	36
Figure 10 - Schématisation d'une variable de type "intervalle"	59
Figure 11 - Représentation des valeurs que prend le makespan en escalier comparé au makespan	64
Figure 12 – Répartition des temps de résolution en seconde pour chaque fonction lexicographique.	72
Figure 13 - Valeurs de stabilité fournies par chaque fonction objectif pour le groupe "diversité".	73
Figure 14 - Répartition du temps en utilisant le makespan en escalier comme premier critère pour le groupe "diversité" (à gauche) et le groupe "intensité" (à droite).....	75
Figure 15 - Evolution du Makespan pour le groupe "intensité".	75
Figure 16 - Valeurs de stabilité obtenues avec (vii) pour le groupe "intensité"	76
Figure 17 - Pires cas des valeurs de stabilité obtenues avec (vii) et (vii) pour le groupe "intensité".	77
Figure 18 - Valeurs du critère de stabilité obtenues avec les fonctions (ix), (vii) et (viii) pour le groupe "intensité".	78
Figure 19 - Valeurs de stabilité obtenues avec les fonctions (ix), (vii) et (viii) pour quelques instances du groupe "diversité".....	79
Figure 20 - Diagrammes de Gantt pour les planifications issues de DS1 (en haut) et DS2 (en bas). Les tâches encadrées dans chaque diagramme représentent les tâches qui seront retardées d'un certain nombre d'heures et serviront à générer les instances de replanification.....	82
Figure 21 - Deux compagnons travaillant sur une ligne d'assemblage aéronautique.....	89
Figure 22 - Exemple de quelques étapes pour évaluer la dangerosité d'une tâche.....	90
Figure 23 - Ordonnancement en ne considérant que le makespan	94
Figure 24 - Ordonnancement en ne réduisant que la charge ergonomique maximale	94
Figure 25 - Ordonnancement en utilisant l'IMD pour équilibrer les charges ergonomiques	94

Figure 26 - Ordonnancement lorsque les scores de tâches se rapprochent les uns des autres	95
Figure 27 - Ordonnancements en utilisant des scores de pénibilités élevés pour toutes les tâches mais en conservant les écarts entre elles.	95
Figure 28 - Évolution des critères. Bleu = IMD (2) ; Rouge = SUM_Max (1). Les valeurs de l'IMD sont sur l'axe de droite, celles de SUM_Max sur l'axe de gauche. .	97
Figure 29 - Graphes de précédences pour une instance sans Contrainte Organisationnelle (à gauche) et avec une Contrainte Organisationnelle (à droite)...	105
Figure 30 - Ordonnancements résultants des instances sans Contrainte Organisationnelle (à gauche) et avec une Contrainte Organisationnelle (à droite)...	106
Figure 31 - Trois tâches (i, j et k) liées par deux CO de type FDE (i,j) et (j,k).....	107
Figure 32 - Ensemble de tâches liées par trois contraintes de précedence dont deux organisationnelles	107
Figure 33 - Aire de la fonction discrète de Choquet	113
Figure 34 - Graphe de précedence des tâches de l'instance de test du critère de Choquet	114
Figure 35 - Planning montrant la date de fin sans les préférences (droite de gauche) et avec les préférences (droite de droite).....	115

INTRODUCTION

Le secteur aéronautique évolue dans un contexte de concurrence croissante (globalisation des chaînes logistiques, volatilité des marchés, pression sur les indicateurs clés de performance coûts-délais-qualité, apparition de nouveaux acteurs). La complexité des produits, des processus et des systèmes de production rend la planification des ressources de plus en plus difficile tout en exigeant une meilleure réactivité face à des aléas fréquents.

Afin de développer une approche systémique innovante pour gérer les risques ergonomiques et économiques liés à la gestion des aléas dans les chaînes d'assemblage aéronautiques, une équipe a été réunie sous le projet Per4mance (« Planification Et Répartition Flexible du travail entre les OpérateuRs des chaînes d'asseMblage AéroNautiques : une approChe systémique pour gérer les risques Ergonomiques et économiques »). C'est un projet coordonné par l'ISAE Supaero et financé par l'Agence Nationale de la Recherche (ANR) avec un budget de 400 000€. Il compte parmi ses acteurs Dassault Aviation, Airbus, l'Institut de Recherche en Informatique de Toulouse (IRIT), l'ISAE Supaero, l'Université Polytechnique de Madrid, le LAAS CNRS et l'INU Champollion.



Figure 1 - Acteurs du projet Per4mance

Le projet se donne l'ambition d'améliorer les procédures de la planification des ressources afin de diminuer les retards de livraison, d'améliorer les conditions de travail pour les opérateurs et managers en diminuant le stress et en prenant en compte la pénibilité des tâches lors de leur répartition. Ces axes d'amélioration se trouvent être les tâches 3, 4 et 5 dans l'organigramme en figure 2. Leur réalisation parallèle permettra de valider une approche générale et combinée pour répondre au mieux aux nouveaux besoins de l'industrie de l'assemblage aéronautique.

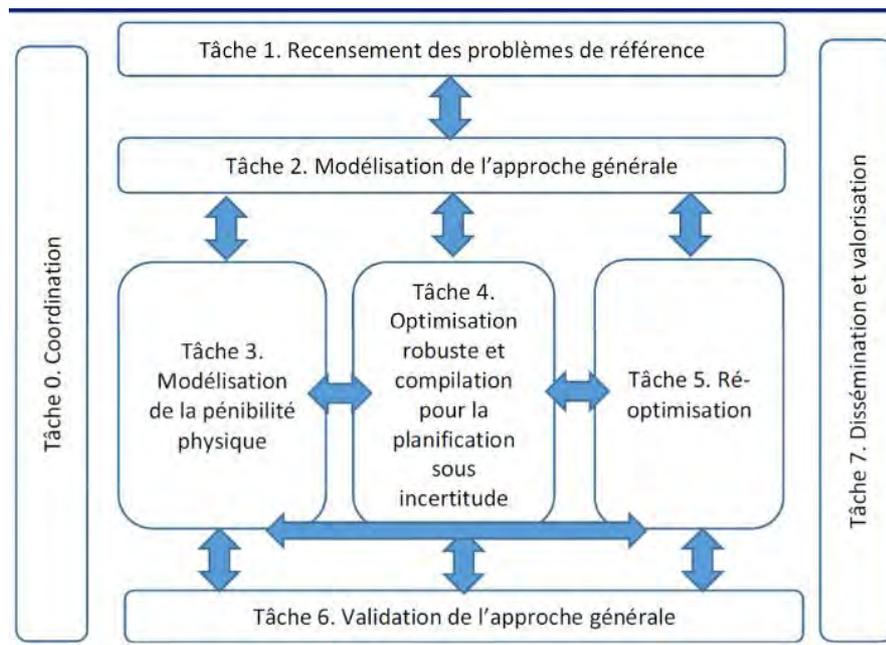


Figure 2 - Organigramme des tâches à accomplir au cours du projet Per4mance

Dans le cadre de ce projet, l'objectif de la thèse est de proposer des méthodes innovantes dans la réaction à des aléas peu connus et peu fréquents des partenaires industriels (tâche 5) contrairement à la tâche 4 qui sera concentrée sur des aléas couramment rencontrés et aura pour objectif de développer une approche robuste capable d'absorber ces aléas. Le pont entre ces deux aspects du projet Per4mance sera donc la frontière entre aléas courants et aléas rares, et nous nous plaçons dans une situation de réponse lorsque la méthode robuste développée n'est pas capable d'absorber l'aléa rencontré.

Nous préparerons également l'assimilation des travaux sur la pénibilité physique en intégrant l'ergonomie et les préférences organisationnelles des managers dans le modèle. Cela avec ceux de la tâche 3 avant la validation et mise en commun de la tâche 6. Nous voulons donc aborder les aspects techniques liés à la replanification tout en considérant fortement l'aspect humain dans la démarche du projet.

Afin de remplir ces objectifs, nous allons étudier les travaux déjà réalisés dans la littérature par rapport à la replanification, mais aussi à la planification et aux lignes d'assemblage aéronautiques. Comprendre ce qui a été fait nous a permis de déterminer une méthode efficace qui utilise des outils (langage de programmation) qui se rapprochent de ceux déjà utilisés par le partenaire industriel pour faciliter la communication, les échanges et leur permettre de développer un outil d'aide à la décision facilement intégrable. Nous espérons ainsi apporter une méthode novatrice qui exploite des axes de recherche peu rencontrés dans la littérature jusqu'à aujourd'hui.

La première section présente la ligne d'assemblage et le processus de planification. La section 2 présente l'étude de littérature et le positionnement par rapport à ces travaux. La section 3 présente la modélisation du problème ainsi que la définition des critères en escalier. La section 4 présente les travaux effectués pour intégrer des critères ergonomiques au modèle. La section 5 quant à elle va intégrer les préférences organisationnelles formulées par notre partenaire industriel. Enfin, une conclusion sera établie et une ouverture sur la poursuite de ces travaux sera présentée.

1.

PROCESSUS D'ASSEMBLAGE AERONAUTIQUE

1.1. PLANIFICATION D'UNE CHAÎNE D'ASSEMBLAGE AERONAUTIQUE

L'assemblage de véhicules est aujourd'hui un processus industriel omniprésent dans l'industrie. Cela concerne des véhicules nautiques (bateaux), routiers (voitures, camions) ou encore aéronautiques (avions) pour ne citer que les plus courants. Chaque véhicule répondant à des caractéristiques différentes, le processus d'assemblage qui lui est associé possède lui aussi ses particularités. Certains d'entre eux sont donc beaucoup plus mécanisés que d'autres, nous pensons par exemple à la ligne d'assemblage de Renault qui a mis en place une très forte automatisation pour ses chaînes de production et vise dans sa direction vers l'industrie 4.0 une forte dualité homme-machine avec ce qu'ils appellent des « cobot ». Ce sont des robots collaboratifs qui assistent les opérateurs dans leurs tâches.

Le niveau d'automatisation n'est pas le même pour les chaînes de montage aéronautique. De nombreuses tâches sont réalisées à l'intérieur du véhicule ou sur des portions où il n'est pas possible d'intégrer une machine complexe pour aider les opérateurs. L'automatisation intervient principalement dans des tâches qui se situent à l'extérieur de l'avion et qui demandent une force qu'un humain ne peut pas produire.

Construire un avion, dans le cas considéré dans ce manuscrit, représente également plusieurs jours de travail, de nombreuses étapes et un passage par des sites géographiques différents. Sur chaque site, l'assemblage de chaque avion sera avancé suivant une série de tâches. Un avion doit passer par plusieurs sites différents, et c'est seulement après être passé par chaque site qu'il est considéré comme terminé.

L'objectif de cette section est donc de présenter le processus d'assemblage d'un de ces sites géographiques et d'ensuite présenter les problèmes auxquels ces chaînes peuvent être confrontées.

1.2. CHAÎNE D'ASSEMBLAGE AERONAUTIQUE

Un premier aspect de l'organisation de la chaîne est l'acheminement des pièces nécessaires à l'assemblage d'un avion. Nous avons abordé dans la section 1.1. l'existence de différents sites géographiques. Dans le cas de cette thèse, ils ne

seront pas considérés directement, et nous nous concentrons sur le processus d’assemblage sur un unique site d’assemblage.

Temporairement, ce site est organisé en créneaux de huit jours ouvrés : c’est ce que nous appelons « Intervalle de Production » (IP). Au total, trois IP sont nécessaires à l’assemblage d’un avion sur le site, et ces trois IP (vingt-quatre jours ouvrés) représente la cadence de production d’un avion. Durant chaque IP, un ensemble de tâches réparties sur trois avions présents sur place, et c’est sur la base de l’IP que nous allons raisonner dans l’ensemble de ce manuscrit, et non pas sur le temps que représente la cadence. Il est également important de noter que, puisque l’assemblage se fait sur plusieurs jours, il y a des coupures du processus d’assemblage chaque nuit. Bien que les tâches soient mises en pause, celles-ci ne sont pas considérées comme préemptives pour autant, une tâche qui est démarrée doit obligatoirement être terminée avant d’en démarrer une autre. La mise en pause ne fait pas office de « coupure » de la tâche mais seulement du processus. Si l’ensemble des tâches n’a pas pu être terminé à la fin de cet IP, son rallongement se fera par l’ajout d’un ou plusieurs jours ouvrés complets. L’objectif est d’avoir un nombre de jours finis.

En ce qui concerne la logistique de la chaîne d’assemblage du site de Dassault à Biarritz, il faut visualiser ce site comme un ensemble de stations sur lesquelles se situent trois avions en cours d’assemblage. Ceux-ci sont positionnés côte à côte et déplacés d’une station à l’autre à la fin de chaque IP. Lors de son arrivée sur site, l’avion est positionné sur la station n°1 et changera pour passer à la seconde puis à la troisième afin de terminer son assemblage et être amené sur un autre site. L’ensemble du processus se décompose ainsi en trois phases d’assemblage, et chacune d’elle s’effectue sur l’une de ces stations, comme le montre la figure 3.

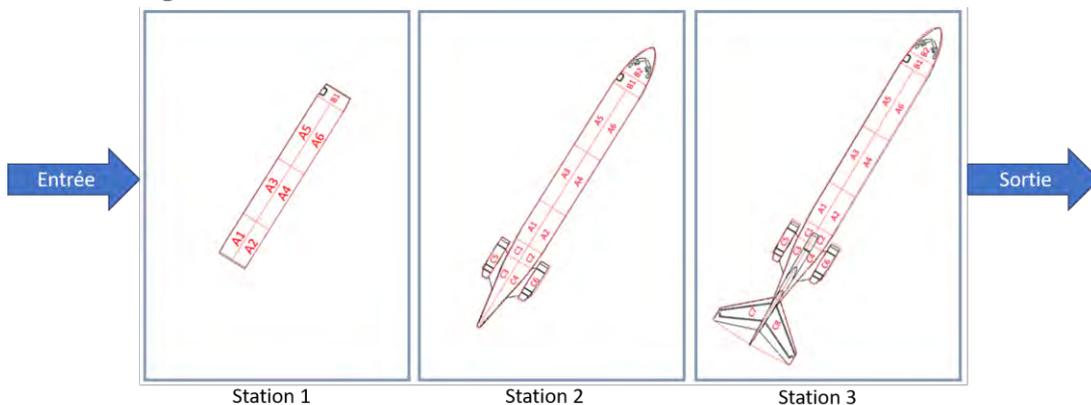


Figure 3 - Visualisation des stations d’assemblage d’une chaîne d’assemblage aéronautique

Un élément supplémentaire qu’il est possible de distinguer sur la figure 3 est la séparation en zones de travail pour chaque avion. Ces zones, dessinées en

rouge, sont présentes pour mieux délimiter l'espace disponible pour réaliser les tâches. En zoomant sur l'avion de la station 3 (cf. figure 4), nous pouvons mieux voir ces délimitations. Cette séparation implique que, si une tâche est réalisée sur une zone (admettons la zone A5), alors aucune autre tâche ne pourra être réalisée sur cette même zone de cet avion avant que celle-ci ne soit libérée (fin de la tâche). Il ne peut donc pas y avoir deux tâches différentes en cours sur une même zone. De plus, une tâche peut exclure des zones adjacentes pour des raisons de sécurité ou de manœuvre. Si par exemple une tâche i sur la zone A5 exclue les zones A3, A6 et B1, alors aucune autre tâche ne peut être réalisée sur l'ensemble de ces quatre zones tant que la tâche i de la zone A5 n'est pas terminée. Une même zone peut cependant être exclue par plusieurs tâches différentes. Par exemple la zone 13 peut être également exclue par une tâche réalisée sur la zone A1, même si i n'est pas terminée sur A5 et exclue toujours A3.

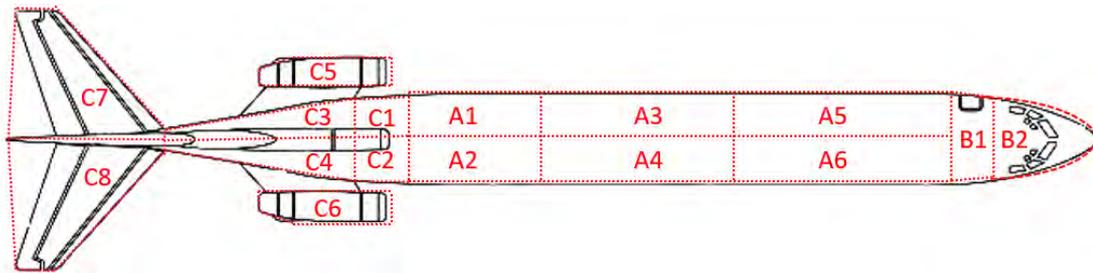


Figure 4 - Exemple de délimitation de zones de travail

Comme mentionné dans la section 1.1., la chaîne est loin d'être totalement mécanisée. Les opérateurs présents sur la chaîne sont couramment appelés « compagnon » dans le secteur de l'aéronautique, et nous utiliserons ce terme lorsque nous aborderons le sujet des lignes d'assemblage aéronautiques. Ces compagnons interviennent sur les avions de chaque station. Ceux-ci peuvent se déplacer librement que ce soit à l'extérieur comme à l'intérieur des avions. La section suivante s'intéresse plus amplement aux compagnons et leur organisation, mais aussi aux ressources matérielles utilisées.

1.3. RESSOURCES DE PRODUCTION

Les compagnons sont les personnes qui travaillent sur la chaîne d'assemblage. Chacun maîtrise un et un seul métier qui sera nécessaire pour la réalisation des tâches. Pour l'exécution des tâches, différents corps de métier sont nécessaires, et le nombre de compagnons par métier est variable en fonction du nombre de tâches nécessitant un ou des compagnon(s) maîtrisant le métier associé. Toute tâche sera donc réalisée par un compagnon maîtrisant le métier adéquat.

De plus, il faut savoir que chaque compagnon travaille selon un calendrier qui lui est propre, et qu’il ne peut donc réaliser des tâches que sur les heures définies par le type horaire (abrégié ‘TT’ pour « Temps de Travail »). Les types horaires sont au nombre de deux : TT1 et TT2. S’il est sous TT1, sa fenêtre de travail se situera exclusivement sur des horaires de jour, entre 8h et 15h30. En revanche, s’il est sous TT2, le compagnon pourra travailler sur des heures plus tardives, entre 6h30 et 21h30. Le type horaire définit donc la disponibilité de chaque compagnon et a son importance pour la planification. Les deux types horaires sont illustrés dans la figure 5 ci-dessous.

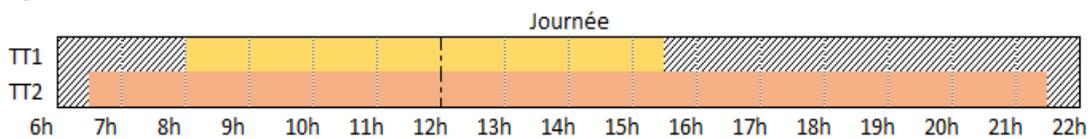


Figure 5 - Illustration des plages horaires de chaque type

Les compagnons sont considérés comme des ressources humaines, donc renouvelables et en quantité limitée, tout comme les ressources matérielles, mais avec un aspect ergonomique en plus ainsi qu’un aspect métier. Contrairement aux ressources matérielles, les compagnons sont sensibles aux conditions de travail et l’environnement dans lequel ils évoluent, et ces éléments sont importants à prendre en compte lors de l’élaboration d’un planning d’assemblage. L’affectation des tâches aux compagnons représente le cœur du problème et des informations nécessaires à la planification. Il est possible de représenter le processus grâce à elles et les données qu’elles représentent sont présentées dans la partie suivante.

Enfin, les compagnons utilisent différentes ressources matérielles pour réaliser leurs tâches. Ces ressources, en quantité limitée mais renouvelables, peuvent être variées en fonction de la tâche. D’un côté cela peut être un bras robotique, piloté par le compagnon. Celui-ci peut l’assister dans des tâches nécessitant une grande force impossible à assumer pour une personne. Ce type d’installation étant fixe, elle est souvent rattachée à une tâche en particulier et ne peut donc être déplacée. Par conséquent, elle ne peut pas non plus être remplacée en cas de panne et doit être réparée par les compagnons sur place. D’un autre côté, des ressources plus simples (semblables à des outils que le compagnon peut déplacer avec lui) peuvent être utilisées et sont également en quantité limitée.

1.4. REPRESENTATION DU PROCESSUS

Nous avons montré comment la chaîne d’assemblage est organisée ainsi que la place que prennent les compagnons et les ressources matérielles dans celle-ci.

Nous allons maintenant caractériser les tâches réalisées par les compagnons avec les ressources matérielles renouvelables décrites précédemment.

Le processus d'assemblage est éclaté en un nombre de tâches présentant de nombreuses caractéristiques. La granularité d'éclatement est la plus précise possible, ce qui veut dire que chaque tâche représente une action très précise plutôt qu'une étape plus globale de l'assemblage. Pour prendre un exemple illustratif, « fixer le nez de l'avion » serait divisé en deux tâches : 1) positionner le nez à l'avant ; 2) fixer le nez à l'aide du bras robot. Avec cette granularité précise, il est ainsi possible de créer des sous-groupes de tâches qui représenteront les différentes étapes de l'assemblage. Cette plus grande finesse permet d'avoir un plus grand levier d'action sur la répartition des tâches parmi les compagnons. Cela permet également d'avoir des tâches plus courtes, ce qui donne plus de possibilités de solution.

Par ailleurs, il faut considérer la séparation en stations d'assemblage pour travailler sur chaque avion au sein de la chaîne. Chaque avion étant assemblé en même temps, la planification du processus comprend les tâches qui seront réalisées sur l'ensemble de ces avions. Le volume de tâches sera donc réparti par rapport aux avions et chacune est rattachée à la station d'un avion de la chaîne. Si les compagnons sont répartis sur les trois stations, c'est aussi le cas des ressources matérielles. Celles-ci seront utilisées et déplacées par les compagnons par rapport aux tâches en cours, ce qui fait de chaque ressource une ressource globale pour la planification. Le même ensemble de tâche est donc planifié et réalisé à chaque IP, ce qui permet de ne calculer le planning qu'une seule fois, hors imprévu.

La première caractéristique d'une tâche est son temps d'exécution, le temps qui doit être respecté pour être considérée comme terminée. Ensuite, étant donné que chacune d'entre elles doit être réalisée par un compagnon, elle est rattachée à un unique métier qui figure parmi ceux maîtrisés par au moins un compagnon. Ceci est possible grâce à la granularité dont nous avons parlé : au lieu d'avoir une étape d'assemblage nécessitant plusieurs savoir-faire spécifiques, nous avons maintenant plusieurs petites tâches possédant un seul savoir-faire.

Ensuite, chaque tâche possède une à plusieurs relations de précédence avec d'autres tâches de l'ensemble. Considérant deux tâches a et b , ces relations de précédence sont de plusieurs types. Nous utiliserons les annotations « $start_x$ » et « end_x » pour représenter le début ou la fin d'une de ces deux tâches dans les exemples. De plus, un délai d est indiqué dans chaque expression et représente un temps d'attente.

- « FF » pour « FIN-FIN »

Pour deux tâches (a, b) du même ensemble, la tâche a doit être finie avant la fin de la tâche b en prenant en compte un délai d entre les deux fins :

$$end_a + d_{a,b} \leq end_b$$

- « DD » pour « DEBUT-DEBUT »

Pour deux tâches (a, b) du même ensemble, la tâche a doit avoir débuté avant le début de la tâche b , toujours en prenant en compte un potentiel délai d entre les deux débuts :

$$start_a + d_{a,b} \leq start_b$$

- « DDE » pour « DEBUT-DEBUT EXACT »

Pour deux tâches (a, b) du même ensemble, la tâche a doit avoir débuté en même temps que la tâche b , toujours en prenant en compte un potentiel délai d entre les deux débuts :

$$start_a + d_{a,b} = start_b$$

- « FD » pour « FIN-DEBUT »

Pour deux tâches (a, b) du même ensemble, la tâche a doit être terminée avant que la tâche b débute, toujours en prenant en compte un potentiel délai d entre les deux débuts :

$$end_a + d_{a,b} \leq start_b$$

- « FDE » pour « FIN-DEBUT EXACT »

Pour deux tâches (a, b) du même ensemble, la tâche a doit être terminée au moment où la tâche b débute, toujours en prenant en compte un potentiel délai d entre les deux débuts :

$$end_a + d_{a,b} = start_b$$

- « FDETT » pour « FIN-DEBUT EXACT TEMPS TRAVAIL »

Similaire à « FDE » mais cette fois-ci les horaires de pause ne sont pas pris en compte. Cela veut dire qu'une tâche peut se terminer à la dernière heure d'un jour pour que l'autre démarre le lendemain à la première heure, ce qui n'est pas le cas pour les relations « FDE ».

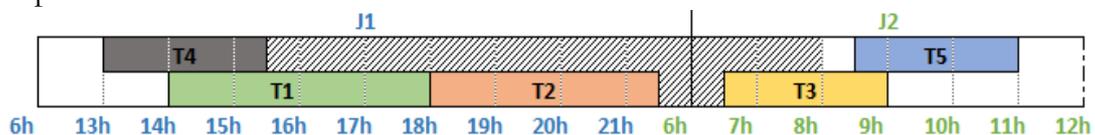


Figure 6 – Illustration des précédences

Dans la figure 6 nous trouvons la planification de cinq tâches, T1 à T5, qui respectent les relations de précédence suivantes :

- FF : (T3, T5) avec une marge de 2h
- DD : (T4, T1) avec une marge de 1h
- DDE : aucune

- FD : (T4, T5) avec une marge de 30min (sans compter la nuit, représentée par les sections hachurées)
- FDE : (T1, T2)
- FDETT : (T2, T3) (qui ne prend pas la nuit représentée par les sections hachurées en compte)

Cette planification permet de mieux représenter les différentes contraintes de précédence et de visualiser leur impact dans une solution de planification.

Etant donné que chaque relation de précédence associe deux tâches rattachées à un avion indépendamment de la station associée, certaines relations peuvent impliquer deux tâches de deux avions différents (i.e. deux avions sur deux stations d'assemblages différentes). Ces relations permettent la gestion de l'IP et de s'assurer que les tâches nécessaires sont terminées avant de débiter le prochain IP. Il faut donc les prendre en compte correctement lors de la planification.

Pour être réalisée, chaque tâche devra également se voir affecter un nombre spécifique de compagnons. D'une part, ces compagnons devront maîtriser le métier associé à la tâche pour pouvoir y être affecté. D'une autre part, comme pour les compagnons, les tâches sont elles aussi soumises à des types horaires : si une tâche est soumise à un type TT1, alors elle ne peut être effectuée que sur la tranche 8h – 15h30 tandis que si elle est du type TT2, elle pourra être effectuée n'importe quand sur la tranche 6h30 – 21h30. Cependant, cela ne restreint en rien l'affectation d'un compagnon à la tâche. En effet, il n'y a pas d'obligation que le régime de la tâche et du compagnon coïncident pour que ce dernier puisse être affecté à la tâche.

Maintenant que nous avons pu présenter comment fonctionne la chaîne d'assemblage, nous allons nous pencher sur le problème industriel.

1.5. PROBLEMATIQUE INDUSTRIELLE

Nous avons vu tout au long de cette section que le processus d'assemblage est complexe et qu'il est soumis à beaucoup de facteurs sujets à des perturbations. Ces perturbations, aussi appelées « aléas », peuvent être internes (absence de compagnon, panne mécanique) comme externes (délai de livraison). En fonction de son origine et son intensité, cet aléa a un impact différent sur la chaîne d'assemblage. L'action à prendre en réponse est déterminée par une équipe de managers en charge de la chaîne. La conséquence pour l'entreprise peut être économique (coûts supplémentaires pour répondre à la perturbation, perte de rentabilité) comme organisationnelle (délai dans l'assemblage, les livraisons,

planning perturbé). Une procédure pour synthétiser ces risques est donc importante pour avoir une meilleure vision des problèmes rencontrés.

Nous allons dans un premier temps établir un arbre de défaillance. Il s'agit d'une technique utilisée pour étudier la sécurité et la fiabilité d'un système comme le processus d'assemblage étudié ici. Pour réaliser cette étude, nous allons représenter graphiquement les différents événements possibles qui peuvent survenir pendant la planification et ainsi remonter jusqu'à leur(s) source(s) potentielle(s). Dans la figure 7 est présenté l'arbre de défaillance pour une tâche qui doit être recommencée.

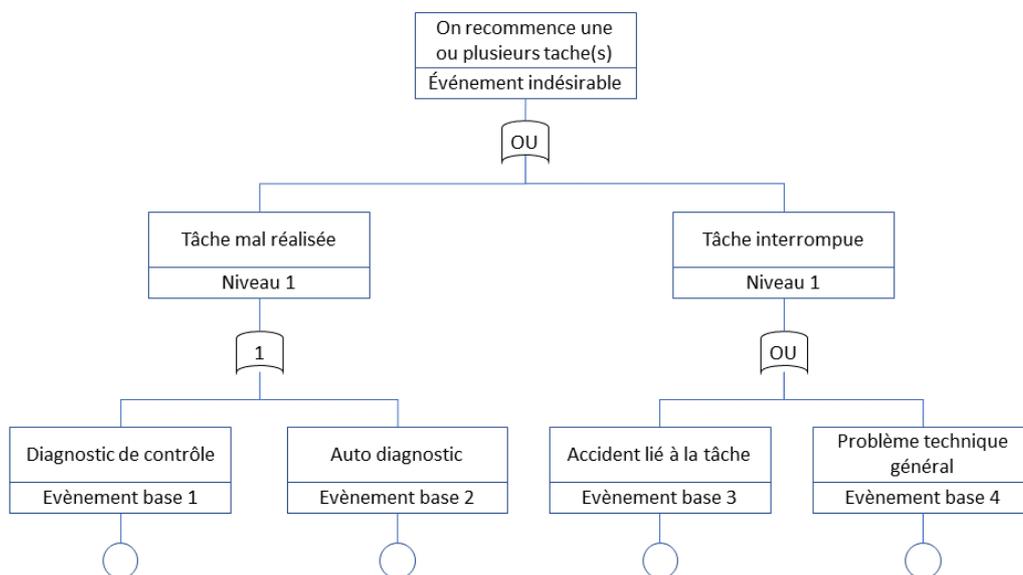


Figure 7 - Arbre de défaillance lorsqu'une tâche doit être recommencée

Ici nous identifions quatre sources potentielles à l'événement « une tâche est recommencée ». Parmi ces quatre événements, un seul des événements base 1 (EB1) et EB2 peut survenir, tandis que les EB3 et EB4 peuvent survenir en même temps. En effet le diagnostic de contrôle est effectué par une entité indépendante du compagnon en charge de la tâche et fait son propre diagnostic de qualité de réalisation. Dans la figure 8 nous trouvons un autre arbre de défaillance concernant cette fois-ci un retard de tâche.

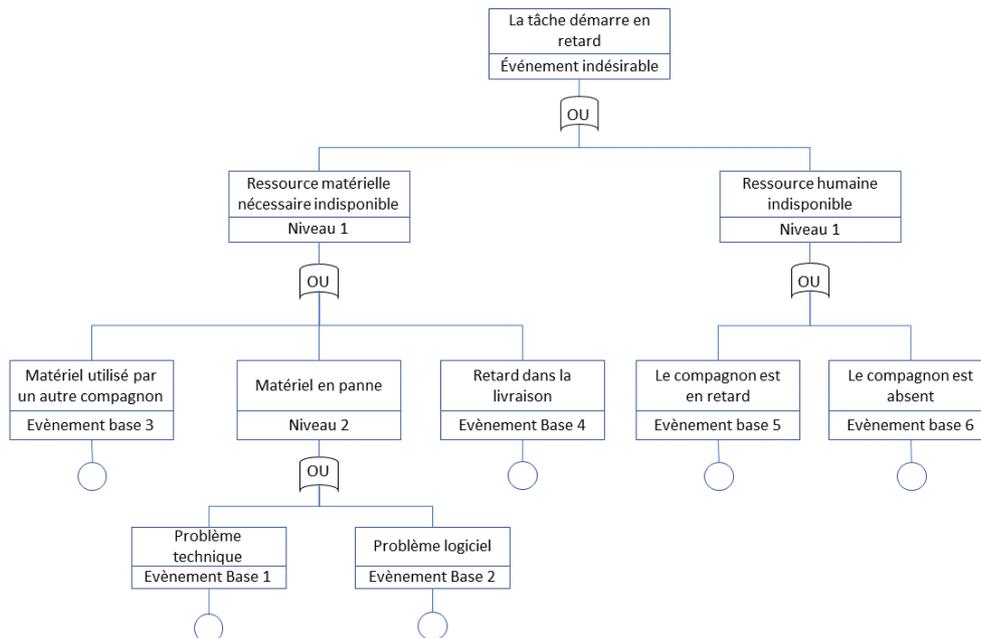


Figure 8 - Arbre de défaillance pour une tâche retardée

Dans ce cas, le nombre de sources est plus dense, et pour chaque EB une solution pour y pallier est possible. Pour déterminer comment répondre à ces différents événements, nous allons maintenant utiliser la méthode de l'AMDEC.

La méthode que nous allons utiliser est celle de l'AMDEC (Analyse des Modes de Défaillance, de leurs Effets, et de leur Criticité). Elle est intuitive et facile à mettre en place, elle est déjà répandue en entreprise et plus particulièrement dans le domaine aéronautique. La méthode de l'AMDEC est de type processus dans notre cas. Elle vise à analyser les causes de défaillance potentielles avant qu'elles ne surviennent pour définir une stratégie pour faire face à chacune d'elles dans le processus d'assemblage. C'est donc à travers l'AMDEC que nous allons présenter les potentiels aléas déjà connu ou anticipables qui peuvent survenir sur la chaîne.

L'AMDEC est séparée en deux parties dans les tableaux 1 et 2. Le premier de ces deux tableaux présente les différents risques identifiés et les aléas qui en découlent, avec leur type et occurrence. Et dans le second tableau sont décrits les effets sur la chaîne d'assemblage ainsi que les actions prises en réponse à ces aléas.

La liste de ces événements ne représente bien entendu pas l'étendue des possibilités et concerne surtout les aléas rares et ayant un impact lourd sur la chaîne d'assemblage. Un impact considéré comme lourd consiste en une interruption complète de la chaîne. Ces aléas étant rares, l'évaluation de leur probabilité d'occurrence est impossible car trop peu de données sont disponibles pour avoir une estimation fiable. Le jour où un tel aléa survient, l'entreprise doit y faire face sans stratégie pré établie et donc perdre du temps précieux sur son processus d'assemblage pour trouver une réponse. Il lui faut donc un moyen générique de

répondre à des aléas qu'elle n'a pas anticipés et qui demeure efficace peu importe le type d'aléa ou son intensité.

Enfin, tout au long de ce manuscrit nous parlerons de planning, planification ou replanification lorsqu'il s'agit de cas concernant l'instance issue du problème réel, et nous parlerons d'ordonnancement ou réordonnancement pour parler de méthodes de la littérature ou d'exemples illustratifs.

Risques	Type de risque	Aléa	Description	Type d'occurrence	Type d'effet
Problème logiciel ou technique	Ressources Matérielles	Panne ressource matérielle	La ressource est indisponible	Evènement rare	Arrêt (partiel) de la chaîne (1)
Livraisons retardées	Approvisionnement	Composant non disponible	Impossibilité de réaliser la tâche	Evènement rare	
Resource humaine non disponible	Ressources humaines	Absence compagnon	Une série de tâches ne peut pas être réalisée	Evènement rare	Réduction main d'œuvre (2)
Une tâche est mal réalisée	Chaîne d'assemblage	La tâche ne passe pas le contrôle qualité	La tâche concernée doit être recommencée, ainsi que toutes les tâches qui la précèdent / suivent et dont elle dépend.	Evènement rare	Ajout de nouvelles tâches (3a)
Une tâche est interrompue		L'arrêt d'une tâche qui ne pouvait pas être mise en pause met en péril la qualité du travail			
Un compagnon doit être remplacé	Chaîne d'assemblage	Rallongement de durée des tâches	Le compagnon qui doit faire les tâches a besoin de plus de temps que son collègue car il est moins habitué.	Possibilité	Durée tâche augmentée (3b)

Tableau 1 - AMDEC présentant les risques les plus impactant pouvant survenir

Type d'effet	Effet description	Effet quantification	Type d'action	Action en réponse
(1) Arrêt (partiel) de la chaîne	Une panne empêche les compagnons de continuer	Le temps que la panne soit réglée	Réaffectation	Les compagnons réalisent d'autres tâches en attendant la fin de la panne
(2) Réduction main d'œuvre	Un compagnon ne peut pas venir travailler	Ses tâches sont mises en pause le temps de trouver un remplaçant	Ajout de ressources	On affecte ses tâches à une équipe d'intervention en attendant l'arrivée d'un remplaçant
			Réaffectation	On affecte une partie des tâches à d'autres compagnons
(3a) Ajout de nouvelles tâches	Une ou plusieurs tâches doivent être réalisées en plus de ce qui était prévu	Augmentation de la charge de travail à réaliser	Ajout de ressources	En fonction de la quantité de tâches, un ou plusieurs compagnons supplémentaires peuvent être mobilisés
(3b) Durée tâche augmentée	Une tâche prend plus de temps que prévu	Cela peut casser certaines relations de précédence	Replanification	Il faut décaler certaines tâches voire replanifier les tâches restantes

Tableau 2 - Description des effets de chaque aléa et de l'action effectuée en réponse

Cette thèse s'inscrit dans cette démarche de réponse face à des aléas peu fréquents et à l'impact fort car le calcul d'un nouveau planning peut être lourd, et très coûteux sachant que l'ensemble de la chaîne est à l'arrêt. Déterminer une nouvelle solution rapidement est donc un objectif de taille pour améliorer la qualité de la ligne d'assemblage dans son global. Dans cette recherche de qualité, l'ergonomie sera aussi de mise et prise en compte pour améliorer le confort et la qualité de travail des compagnons. Prendre en compte la difficulté des tâches et leur potentiel impact sur la santé de ces derniers sera également un axe d'étude de cette thèse. La section suivante vise donc à faire une étude de la littérature pour étudier les problèmes similaires déjà abordés, voir les approches et méthodes abordées et nous positionner par rapport à ces travaux.

2. **ETAT DE L'ART**

Le problème auquel nous faisons face peut être vu différemment en fonction de l'angle dans lequel nous l'abordons. Nous pouvons associer ce problème comme étant de type « Assembly Line Balancing Problem » (ALBP) mais également comme de type « Resource-Constrained Project Scheduling Problem » (RCPSP). Les caractéristiques et méthodes de résolution de ces deux problèmes sont différentes et notre objectif est de présenter ce qui a été réalisé dans la littérature pour déterminer comment nous modéliserons le problème dans cette thèse. Les deux premières parties de cette section visent donc à exposer ces deux problèmes, puis nous parlerons des approches abordées pour gérer l'incertitude avant de terminer sur la place de l'ergonomie et la prise en compte de l'aspect humain dans des problèmes industriels.

2.1. RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM

Le RCPSP est défini de la manière suivante : considérons un ensemble de tâches I ayant chacune une durée fixe. Considérons aussi un ensemble de ressources renouvelables ayant chacune une quantité maximale. Chaque tâche nécessite une ou plusieurs ressources cumulatives en quantité fixe tout au long de sa réalisation et ne peut être interrompue lorsqu'elle a été commencée. De plus, il peut exister une relation de précédence entre deux tâches i et j de l'ensemble qui indique que, tant que i n'est pas terminée, j ne peut pas être commencée. En respectant cet ensemble de relations de précédence et en veillant à ne jamais dépasser la quantité maximale de chaque ressource utilisée par l'ensembles des tâches en cours, l'objectif du RCPSP est de minimiser le makespan, autrement dit la date de fin de la dernière tâche du projet.

Le RCPSP est un problème NP-difficile, mais sa définition est trop générique pour représenter fidèlement des problèmes issus de cas réels. C'est pourquoi de nombreux travaux ont été menés pour adapter ce modèle à différents problèmes, donnant naissance à de nombreuses extensions du RCPSP. Pour présenter ces extensions, nous nous basons sur les travaux de Habibi et al. (2018) et Hartmann et Briskorn (2022). La figure 9 représente ces différentes extensions et a été inspirée par Habibi et al. (2018).

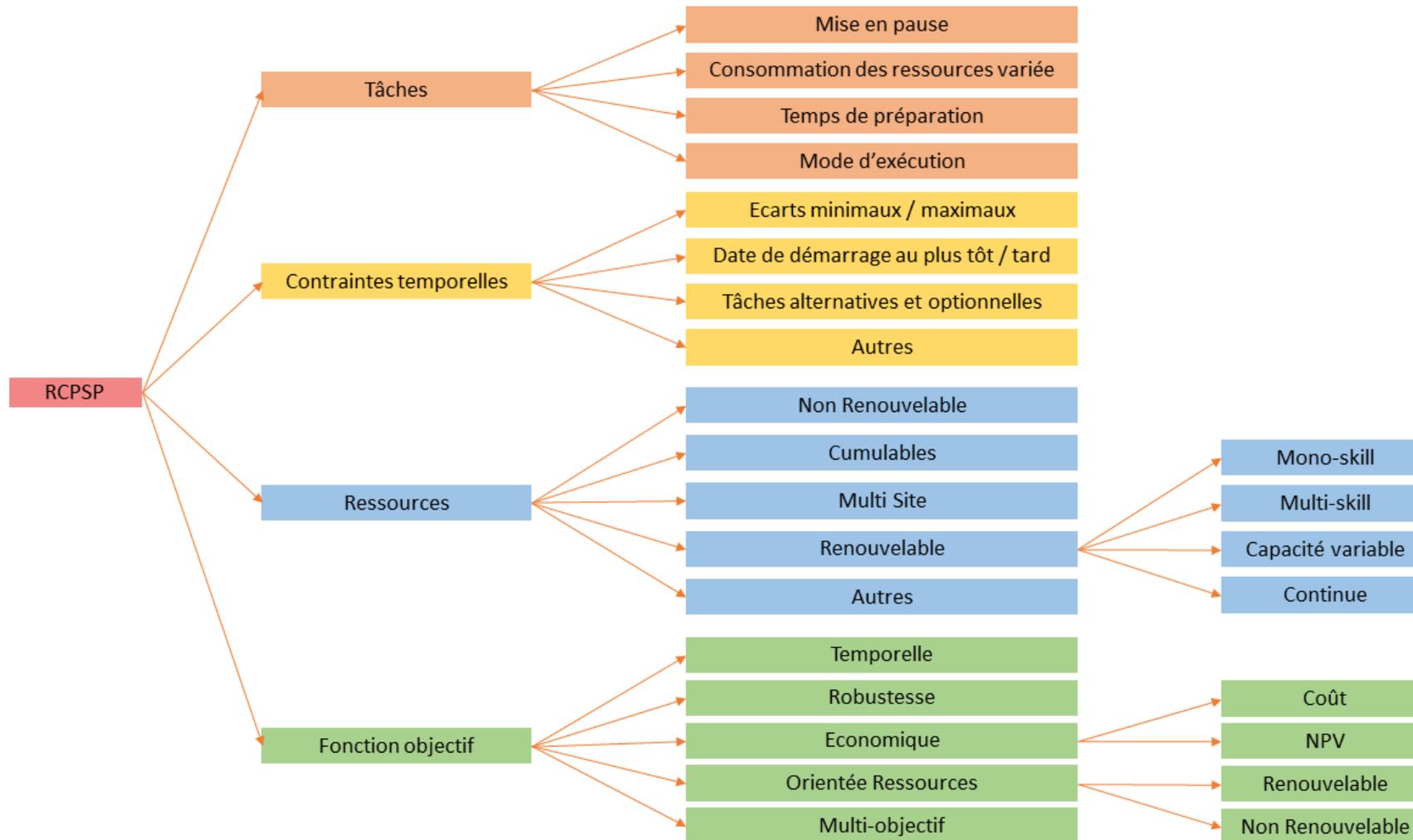


Figure 9 - Classification des extensions du RCSP

2.1.1. Extension des tâches

La définition des tâches dans le RCPSP implique que celles-ci soient réalisées jusqu'à ce qu'elles soient terminées, et ne peuvent donc pas être interrompues. Pour leur problème, Vanhoucke et Coelho (2019) autorisent l'interruption des tâches à certains points et imposent un temps de préparation pour reprendre les tâches interrompues. Ce type de RCPSP avec interruption est aussi appelé « Preemptive RCPSP (P-RCPSP) ». Une autre cause de l'interruption des tâches peut être la définition d'un calendrier de disponibilité pour les ressources. Kreter et al. (2016) appliquent ce principe où des tâches peuvent être mises en pause pendant les week-ends ou les vacances des compagnons.

La consommation des ressources par les tâches peut également différer du RCPSP classique où une tâche utilise une ou plusieurs ressources tout au long de sa réalisation. Dans leurs travaux, Hosseinian et al. (2019) considèrent un modèle avec des tâches dont les besoins en ressources dépendent du temps. Ce cas s'applique lorsque les ressources (ici humaines) ont des savoir-faire variés. Une approche plus flexible existe, le « Flexible RCPSP (FRCPS) », où une tâche peut modifier sa consommation de ressource au fil du temps. C'est donc lors de la résolution que les quantités de ressources affectées à chaque unité de temps sont définies. Tritschler et al. (2017) présentent ce type de modèle dans des cas de projets de recherche pharmaceutiques.

Dans certains cas, il peut être nécessaire de préparer l'exécution d'une tâche. Ce temps de préparation peut aussi être dû à l'enchaînement de deux tâches qui utilisent une même ressource (Hanzalek et Sucha (2017)) ou à des déplacements de ressources. Cela peut être le cas lorsque plusieurs sites sont pris en compte dans le problème, ou que la distance entre deux zones nécessitant une même ressource est assez grande pour ne pas considérer le transfert instantané. Cela nécessite des temps de préparation (par exemple installation de la ressource) pour les tâches à réaliser. Cette variation du RCPSP avec des temps de transfert est le RCPSP-TT (« RCPSP with Transfert Times ») et est notamment traité par Kadri et Boctor (2018).

Enfin, la version la plus connue de l'extension du RCPSP est le Multi Mode RCPSP (MRCPS). Ce modèle ajoute aux tâches différents modes d'exécution qui ont un impact sur différentes caractéristiques de ces tâches (durée, ressource(s) consommée(s)). Cette variante est très populaire et mixée avec de nombreuses autres variantes du RCPSP, comme l'utilisation des temps de préparation (Okubo et al. (2015)), la mise en pause des tâches (Van Peteghem and Vanhoucke (2010)) ou encore les dates de démarrage au plus tôt et deadlines (Cheng et al. (2015)).

2.1.2. Extension des contraintes temporelles

Alors que deux tâches liées par une contrainte de précédence doivent être réalisées seulement en respectant la précédence (qui peut être de type fin-début, fin-fin, début-début ou début-fin), il est possible d’étendre ces précédences pour les rendre plus précises. En indiquant un temps de décalage minimal ou maximal entre deux tâches liées par une de ces contraintes, il est possible de respecter un temps et respecter des écarts entre deux tâches. Cette variante du RCPSP est appelée RCPSP/max (ou RCPSP avec relations de précédence généralisées) et a notamment été abordée dans les travaux de De Azevedo, Pessoa, et Subramanian (2021).

Dans le modèle du RCPSP, les tâches peuvent aussi être commencées ou terminées sans restriction temporelle (indépendamment des contraintes de précédence et de ressource). Cependant, il est possible de définir des dates de démarrage au plus tôt ou au plus tard (créant des deadlines) pour ces tâches. Cela génère ainsi des fenêtres de réalisation pour les tâches et Cheng et al. (2015) utilisent des dates de démarrage et deadlines dans leurs travaux qui incluent également plusieurs modes et une mise en pause des tâches possible.

D’autres extensions des contraintes temporelles existent. Par exemple, il est possible de définir des tâches optionnelles qui ne sont réalisées que si une première tâche (qui les précède dans le graphe de précédences) est réalisée. Cette vision est abordée dans les travaux de Kellenbrink et Helber (2015). Cela peut être sous forme de chaînes, groupe de tâches et Servranckx et Vanhoucke (2019) proposent une approche dans laquelle une seule chaîne d’activités optionnelles doit être réalisée entre une tâche initiale et une tâche finale.

Il existe de nombreux autres cas particuliers et sont présentés dans Hartmann et Briskorn (2022) comme les tâches concourantes qui ne peuvent être réalisées durant la même fenêtre temporelle. Dans notre cas, une généralisation des contraintes de précédence est prise en compte. En plus des contraintes « Fin-Début », il sera donc aussi possible de déclarer des contraintes temporelles sans se restreindre au début de la deuxième tâche par rapport à la fin de la première.

2.1.3. Généralisation des ressources

Dans leur définition de base, les ressources sont considérées comme renouvelables et avec la même quantité pour la totalité du projet. Il existe cependant des cas de figure où les ressources sont consommées et en quantité limitée, ce qui peut induire des coûts supplémentaires de stockage ou d’approvisionnement supplémentaire. Les ressources partiellement renouvelables

sont notamment utilisées pour planifier des emplois du temps, organiser des tournois et peuvent être représentées par les terrains qui ne peuvent être utilisés qu'une fois par jour. Tao et Dong (2017) abordent un problème de ressources non renouvelables tandis que Watermeyer and Zimmermann (2020) sont dans un contexte de ressource partiellement renouvelable.

Les ressources cumulables sont des ressources qui peuvent être consommées mais également produites. Par conséquent, il est possible d'avoir une quantité croissante de la ressource au fil du temps ce qui peut avoir un impact sur la planification. La quantité peut être insuffisante pour réaliser certaines tâches à un instant t mais le devenir à un autre instant t' grâce à l'accumulation de la ressource concernée. Carlier et al. (2018) considèrent ce type de ressource dans leur problème de type « Event Scheduling Problem with Consumption and Production of Resources (ESPCPR) ».

Nous avons vu qu'il est possible qu'une ressource soit utilisée sur différents sites géographiques qui peuvent se trouver à plusieurs heures l'un de l'autre, ce qui implique de devoir rendre cette ressource commune à ces différents sites. Cela peut être le cas du RCMPSP (Resource-Constrained Multi-Project Scheduling Problem) où plusieurs projets doivent être réalisés en parallèle en utilisant des ressources communes et ayant des précédences en commun. Chen et al. (2009) ont travaillé sur ce type de problème multi-projet avec plusieurs contraintes sur les ressources. Laurent (2018) a également travaillé sur une version multi-site du RCPSP où les ressources sont partagées et déplacées sur les sites nécessaires dans le domaine médical et a proposé des méthodes exactes et méta-heuristiques pour résoudre le problème.

Lorsqu'elle est humaine, une ressource présente souvent un ou plusieurs savoir-faire (ou professions) qui lui permettent de réaliser chaque tâche. Cette extension est appelée Multi-Skill RCPSP (MS-RCPSP). Dans leurs travaux, Almeida et al. (2019) présentent et comparent différentes méthodes utilisant la PLNE pour résoudre le MS-RCPSP. De par les nombreux cas où une ressource présente différents savoir-faire, de nombreuses variations de ce problème ont également été étudiées. Par exemple, Maghsoudlou et al. (2016) considèrent différents modes d'exécution (cf. 2.2.1.4.) dans leur problème. Lin et al. (2020) de leur côté considèrent un problème où les savoir-faire sont maîtrisés à différents niveaux et un certain niveau est requis pour exécuter chaque tâche. Il est aussi possible de considérer l'apprentissage ou l'amélioration de ces savoir-faire pour améliorer l'efficacité des ressources comme dans les travaux de Zabihi et al. (2019).

Une ressource continue est une ressource qui peut être attribuée de manière fractionnée parmi les différentes tâches. Ces ressources continues sont impliquées notamment lors de la prise en compte de l'utilisation d'électricité ou quelque source hydraulique, pneumatique d'énergie. Le « Discrete-Continuous Resource-

Constrained Project Scheduling Problem (DCR-CPSP) considère donc à la fois les ressources discrètes, utilisées en quantités finies par les tâches, mais aussi une ressource continue. Dans les travaux de Waligóra (2011), une méthode heuristique est présentée pour résoudre ce problème et dans Bianco et al. (2016) une variation est étudiée avec des ressources flexibles.

Alors que chaque ressource est considérée en quantité constante tout au long du projet dans la version classique du RCPSP, il est possible que celle-ci voit sa quantité varier au cours du temps. En particulier, lorsqu'un projet nécessite des ressources humaines pendant plusieurs jours, voire semaines, alors celles-ci sont soumises à un calendrier de disponibilité. Si chaque unité de ressource humaine suit le même calendrier, alors cela peut être ramené à un problème avec disponibilité constante. En revanche, si chacun suit un calendrier différent, alors certains intervalles de temps ne considéreront pas certaines ressources pour la réalisation des tâches. Ce cas de figure où la quantité disponible est soit 0%, soit 100%, est considérée par Kreter et al. (2016, 2017). A noter que les ressources dont la capacité dépend du temps peut être considérées comme un cas spécial des ressources partiellement renouvelables.

D'autres extensions moins courantes dans la littérature ont été abordées comme la sélection du fournisseur de ressources matérielles (qui seront consommées par les tâches) lors de la commande (Tabrizi (2018)), les ressources matérielles à risque comme dans une centrale nucléaire (Li et al. (2018b)) ou encore les restrictions qui sont bloquées par une activité en cours (Poppenborg et Knust (2016)).

Dans notre cas nous aurons donc des ressources renouvelables en quantité limitées, qui pourront être des ressources humaines mais aussi matérielles. Les zones des avions peuvent quant à elles être vues comme des ressources disjonctives renouvelables. Ce sera donc une approche relativement classique des ressources, mais qui utilisera plusieurs types au sein du processus.

2.1.4. Variations de la fonction objectif

L'objectif classique du RCPSP est la minimisation du makespan. Cependant, de nombreux autres objectifs temporels ont été également utilisés comme la minimisation des retards par rapport aux dates de livraisons des tâches (Wang, Liu et Zheng (2019)). Dans leurs travaux, des dates de fin sont déterminées pour chaque tâche, et il est possible d'affecter un poids à chacune d'entre elles. De ce fait, si un retard est inévitable, les tâches au poids le plus haut seront terminées sans encaisser de retard, et ce dernier sera prévu sur les tâches moins impactantes en termes de qualité de solution. Minimiser le retard peut également être fait sur la

fin du projet dans le cas où celui-ci doit être terminé avant une certaine date (Aouni, d'Avignon et Gagnon (2015)).

Si certaines tâches peuvent prendre plus de temps que prévu, les managers peuvent être intéressés par des solutions capables d'absorber ce type de délai. Pour ce faire, ils passent par la génération d'une solution robuste face aux problèmes que peut rencontrer le projet. Palacio et Larrea (2017) par exemple maximisent la somme des écarts après chaque tâche pour absorber au mieux les augmentations de temps d'exécution. Il est aussi possible de considérer le pire scénario et, dans un cas où les tâches engendrent des coûts, mettre le coût maximal pour un nombre limité de tâches, (Hazır et al. (2015)).

Les coûts peuvent être de sources variées dans un projet, et de nombreux modèles ont donc eu pour objectif de minimiser les coûts induits par la réalisation du projet. Dans leurs travaux, Hariga et al. (2019) considèrent par exemple un MRCPSP avec mise en pause de tâches où les coûts sont liés aux changements de modes ou aux fluctuations des ressources. Un critère lié à ces coûts de production est le « Net Present Value (NPV) » qui représente la différence entre les profits et les coûts. Ce dernier est considéré lorsqu'en plus de ces coûts, des profits sont générés au fur et à mesure que certaines parties du projet sont remplies. Gu et al. (2015) étudient ce problème et proposent des méthodes exactes et de type heuristique pour le résoudre. Les coûts et profits peuvent être considérés de manière différente : Leyman et Vanhoucke (2016) considèrent par exemple les profits à des points spécifiques dans le temps tout comme les coûts (qu'ils appellent des « paiements ») à la fois dans le RCPSP et le MRCPSP.

Les ressources sont une composante importante de chaque projet. Par conséquent, de nombreuses fonctions sont axées sur les investissements en ressources, on parle de « Resource Investment Problem (RIP) ». Dans ce problème, les dates de départ/fin des tâches sont à déterminer, comme dans le RCPSP, mais également la quantité totale de chaque ressource mobilisée (considérée constante tout au long du projet). Pour chaque unité de ressource, un coût est associé et l'objectif est donc de minimiser ce coût, considérant une deadline pour limiter le projet dans sa durée. Ce type de problème est considéré par Zhu et al. (2017). Des variantes existent comme dans les travaux de Qi et al. (2015) qui considèrent la version Multi Mode du problème avec uniquement des ressources renouvelables.

Le Resource Leveling Problem (RLP) est un autre type de problème étroitement lié au RIP. Son objectif est de lisser au mieux l'utilisation de la ressource concernée au fil du temps. La ressource doit donc être utilisée de la manière la plus régulière possible en respectant ici aussi une deadline. Parmi les objectifs les plus populaires de ce problème nous retrouvons la minimisation du carré des capacités pondérées par le coût. Ce problème a été abordé par Qiao et Li

(2018) ou encore Li et Dong (2018) qui considèrent le multi mode. Prenant en compte le lissage, la minimisation des changements de capacité entre deux périodes pour une ressource en valeur absolue est aussi abordée par Bianco et al (2017).

La composante multi-objectif d'un problème se manifeste notamment pour les objectifs conflictuels, où un compromis doit être trouvé pour avoir une solution de la meilleure qualité possible. Ballestín et Blanco (2015) donnent un aperçu des approches et modèles qui peuvent être considérés. Une de ces approches est de donner des poids différents à ces critères pour favoriser un type de solution. Par exemple, il est possible qu'augmenter les ressources permettent de terminer plus rapidement le problème, mais pour un coût supplémentaire. Dai et al. (2018) considèrent la somme pondérée des coûts d'utilisation des ressources d'un côté, et le makespan de l'autre. Similairement, Schnabel et al. (2018) considèrent la minimisation du coût d'utilisation de ressource supplémentaire et la minimisation du makespan. L'approche Pareto est également très utilisée pour déterminer des solutions efficaces au sens de Pareto et trouver le meilleur compromis entre deux critères. Shahsavari et al. (2015) considèrent le RCPSP/max suivant trois objectifs (makespan, investissement des ressources, changements d'affectation des ressources). Tirkolaee et al. (2019) utilisent cette approche dans le MRCPSPP en ayant d'un côté la minimisation du makespan et la maximisation du NPV et d'autres travaux comme Zabihy et al. (2019) appliquent cette méthode au MS-RCPSP.

L'approche lexicographique est aussi adoptée pour ordonner les critères dans la fonction objectif. L'ordre dans lequel chaque critère est déclaré dans la fonction détermine son importance lors de l'exploration de l'ensemble de solutions. Entre deux nouvelles solutions possibles, celle qui améliore le critère le plus prioritaire sera toujours choisie par rapport aux autres. Florez et al. (2013) utilisent cette approche considérant trois critères, dont un qui peut être exprimé de deux manières. Pour résoudre leur problème, ils utilisent une stratégie qui peut avoir une à trois phases. Le nombre de phases est déterminé au fur et à mesure du calcul de la solution : tant que la solution n'est pas suffisamment bonne, on passe à la phase suivante et boucle sur la 3^{ème} tant que nécessaire. Geiger (2017) utilise quant à lui deux critères dans son approche de recherche locale dans un contexte de RCPSP multi-mode multi-projet. D'autres travaux et modèles sur la planification multicritères sont disponibles dans le livre de T'kindt et Billaut (2006).

Dans l'ensemble des extensions citées, de nombreuses présentent des caractéristiques que nous rencontrons dans notre problème : MS-RCPSP, GPR, ressources renouvelables avec quantité variable, ressources disjonctives (zones définies sur chaque avion). Cependant, les lignes d'assemblage aéronautiques sont aussi concernées par une autre branche des problèmes d'optimisation

combinatoire connue dans la littérature sous le nom d' « Assembly Line Balancing Problem » ou le problème d'équilibrage des charges.

2.2. ASSEMBLY LINE BALANCING PROBLEM (ALBP)

2.2.1. L'équilibrage des charges

Considérant un ensemble de tâches ayant chacune une durée fixe et des stations ordonnées linéairement, le problème ALBP consiste à affecter ces différentes tâches à chaque station en respectant les contraintes de précédences qui existent entre ces tâches ainsi qu'une contrainte de temps de cycle. Ce cycle représente le temps maximal que ne doit pas excéder la somme des temps d'exécution des tâches affectées sur chaque station. L'objectif classique est de minimiser le nombre de stations utilisées pour réaliser l'ensemble des tâches du problème.

Depuis son introduction par Salveson (1955), de nombreuses nouvelles formulations sont apparues dans la littérature. L'étude de Battaïa et Dolgui (2013) propose une classification des formulations mathématiques en fonction des critères suivants : le nombre de lignes à équilibrer, les données d'entrée, les contraintes et la fonction objectif.

Avec les temps de cycle courts, le séquençement des tâches à l'intérieur d'une station en présence d'une seule ressource reste trivial. Ceci est différent si plusieurs ressources sont utilisées au sein de la même station en permettant une exécution parallèle ou nécessitant des temps de changement d'outils. Dans le cas des lignes d'assemblage aéronautique avec des temps de cycle longs, ce problème devient très complexe, car un problème de type RCPSP est à résoudre pour chaque ensemble des tâches affectables à une station pour vérifier si le makespan respecte la contrainte du temps de cycle.

2.2.2. La planification dans les lignes d'assemblage

Nous avons vu que le problème de l'ALBP, utilisé pour les lignes d'assemblage aéronautiques, consistait à répartir des tâches sur les stations. Cela implique qu'il faut également avoir recours à une méthode de planification par la suite pour affecter les ressources à chaque tâche au sein de chaque station lorsqu'elles sont multiples. Ce cas de figure étant de plus en plus courant, les travaux sur la planification des lignes d'assemblage deviennent aussi plus nombreux. Parmi ces

travaux nous pouvons citer notamment les travaux d'Arkhipov et al. (2018a, b) qui ont développé des modèles de type PPC et PLNE (Programme Linéaire en Nombres Entiers) pour effectuer la planification des tâches dans les stations considérant des contraintes économiques et ergonomiques. Shan et al. (2017) ont également présenté un Algorithme Génétique (AG) qui exploite une fonction composée de trois critères : la détermination du nombre de compagnons utilisés, de l'espace utilisé et de l'exploitation du temps de travail à disposition. La fonction aura pour objectif de trouver le meilleur équilibre entre ces trois critères. Ren et al. (2020) ont proposé un autre modèle basé sur le RCPSP tout en considérant le temps de transfert des ressources entre les différentes tâches. Biele et Mönch (2018) ont présentés deux approches (une linéaire et une non-linéaire) pour résoudre ce problème d'affectation des ressources aux tâches.

2.2.3. Replanification des lignes d'assemblage

En ce qui concerne la replanification des lignes d'assemblage aéronautiques, les travaux sont plus rares. Battaïa et al. (2019) ont proposé une approche PPC pour la replanification des lignes d'assemblage aéronautiques. Zhu et al. (2020) ont considéré un problème d'ordonnancement réactif dans une chaîne de montage d'avions en utilisant un modèle bi-objectif basé sur les coûts des ressources supplémentaires et des retards des tâches. Manceaux et al. (2014) ont quant à eux présenté un modèle de reconfiguration dynamique de leur ligne d'assemblage en utilisant une approche basée sur la « Reachability Analysis ». Cette méthode est utilisée dans différents contextes pour déterminer si un système peut atteindre un état particulier à partir d'un certain point de départ (dans leur cas une reconfiguration à la suite d'une interruption). Manceaux et al. (2016) ont également présenté une méthode basée sur la recherche locale itérative pour la reconfiguration dynamique des lignes d'assemblage. Dans ces deux derniers papiers, les méthodes sont testées sur des cas industriels.

Considérant la replanification des lignes d'assemblage, Grangeon et al. (2011) ont dans leurs travaux fait de la reconfiguration sur des lignes d'assemblages automobiles et proposent trois heuristiques différentes. Oliveira et al. (2012) ont également travaillé sur les lignes automobiles avec une mixité de modèles produits et présentent un PLNE utilisant des nombres binaires. Serin et al. (2019) ont quant à eux proposé une approche pour des lignes d'assemblage en U dans un cas où les durées de tâches sont considérées comme suivant une loi normale. Michels et al. (2021) ont orienté leur replanification vers la conservation des opérateurs déjà présents en utilisant un modèle de programmation linéaire mixte. Zhang et al.

(2018b) ont présenté un algorithme génétique multi-objectif pour les lignes d'assemblage à deux côtés (« two-sided assembly lines ») d'une entreprise chinoise.

Etant donné les dimensions du problème, dans la littérature, les études existantes ne considèrent que des aspects partiels de la problématique globale de l'équilibrage des charges dans les lignes d'assemblage. En ce qui concerne la replanification, la localisation sur une ou plusieurs stations est justifiée par le fait qu'un aléa dans un premier temps ne concerne qu'une seule station et l'objectif principal est de le contenir et d'éviter sa propagation sur les stations en aval. Dans cette thèse, nous nous concentrons sur cette problématique.

2.3. LES APPROCHES DE PLANIFICATION SOUS INCERTITUDE

2.3.1. **L'incertitude en** planification

2.3.1.1. **Le niveau d'information**

Le niveau d'information fait référence aux certitudes concernant les paramètres du projet. Malgré le fait qu'en planification, on considère des paramètres précis (comme les durées de tâche exactes, la présence sur des horaires précises des employés), une telle précision est rarement vérifiée dans la réalité. En effet, nous ne savons jamais quand est-ce qu'une information considérée comme vérifiée depuis longtemps peut changer et ne plus correspondre à la norme la concernant. De plus, certaines informations peuvent également être approximatives (ou floues). Par exemple, une tâche peut mettre « environ » 20 minutes pour être réalisée. Quelle que soit la raison pour cette estimation et le manque d'une durée plus assurée, cela souligne cet aspect important qu'est le niveau d'information dans la planification.

Lorsque l'ensemble des informations est considéré comme fixe et définitif, on parle de problème déterministe. On estime dans ce type de problème que toute information est vérifiée et pourra être vérifiée lors de la réalisation du projet. Cependant, cette supposition n'est que trop rarement juste. Cela nous amène au type de problème non déterministe, où nous savons que les paramètres du problème sont amenés à changer au cours de la réalisation du projet. Même s'il est possible d'anticiper que certains paramètres peuvent changer, il n'est pas toujours possible de déterminer lesquels seront concernés par les changements effectifs. Il

est donc nécessaire d'élaborer des méthodes de planification permettant de faire face aux incertitudes sur l'exactitude des données.

2.3.1.2. Identifier les aléas

Nous savons qu'il est difficile de savoir quel élément du projet est le plus incertain, mais il est possible de déterminer les sources possibles de ces incertitudes. En fonction de la nature du projet, ces sources peuvent être plus ou moins nombreuses. Miller et Lessard (2001) ont par exemple identifié trois sources d'incertitude : en fonction du marché (variation de la demande), l'état de complétion du projet (aspects techniques, organisationnels) et institutionnel. Leifer et al. (2001), de leur côté, ont identifié quatre sources : technique, liée au marché, organisationnelle et liée aux ressources. L'approche qui sera prise pour identifier ces sources dépendra donc du projet et de la vision des managers.

Les aléas, conséquences des incertitudes, peuvent avoir différentes intensités. Zhu et al. (2005) les classifient en deux catégories : perturbations et interruptions. Là où des perturbations sont issues généralement d'un changement dans la durée des tâches, elles sont relativement simples à gérer et peuvent même être naturellement absorbée par le projet. D'un autre côté, les interruptions sont plus rares, mais également plus impactantes étant donné que, comme leur nom l'indique, elles peuvent interrompre le projet et donc mettre en péril sa faisabilité. Ce genre d'événement peut résulter d'une livraison retardée de matières ou de composants ou un problème de personnel absent / machine défectueuse. Les aléas les plus lourds peuvent mettre en pause tout un processus pendant plusieurs dizaines d'heures.

La variété d'aléas possibles est très grande et spécifique à chaque problème. Nous avons vu dans l'AMDEC du chapitre 1 la variété d'aléas qui pouvaient survenir dans une ligne d'assemblage aéronautique. L'analyse des risques menée a également montré les forts impacts que pouvaient avoir ces aléas. Ceci explique l'importance qui a été donnée au développement des approches et méthodes de gestion de l'incertitude et des aléas dans la littérature.

2.3.2. La gestion de l'incertitude

Herroelen et Leus (2005) distinguent cinq grandes approches pour gérer l'incertitude : l'approche proactive (stochastique, floue, robuste), réactive et l'analyse de sensibilité. Les approches proactives considèrent les informations disponibles en amont d'un aléa, alors que les approches réactives sont utilisées dès

que l'incident sera survenu. L'analyse de sensibilité permet de mesurer la stabilité d'une solution face à des changements des valeurs des paramètres.

Dans la littérature, l'utilisation de ces différentes approches a été étudiée dans des contextes variés de planification. Hazir et Ulusoy (2020) ont proposé une classification des sources d'incertitudes pour ensuite aborder les différentes approches de gestion de ces incertitudes dans les projets de planification. Ivanov et al. (2017) ont recensé les méthodes de réparation pour la planification des chaînes d'approvisionnement. Hashemi-Petroodi et al. (2021) ont étudié les méthodes de replanification dans les systèmes de production reconfigurables.

Dans ce qui suit nous présentons chaque catégorie des méthodes utilisées dans la littérature.

2.3.2.1. Les méthodes stochastiques

Les méthodes stochastiques considèrent les durées de tâche comme des variables aléatoires ayant une distribution de probabilité connue. Dans la littérature RCPSP, une telle formulation est nommée « Stochastic RCPSP (SRCPSP) ». Creemers (2015) résout le SRCPSP avec une méthode exacte se basant sur les chaînes de Markov. De leur côté, Chen et al. (2018) étudient les performances de dix-sept heuristiques pour ce problème. Zhu et al. (2007) utilisent un modèle de programmation linéaire stochastique en nombres entiers et Davari et Demeulemeester (2018) développent un algorithme de branch-and-bound pour résoudre le problème. De nombreuses autres méthodes spécifiques à leur problème ont été présentées dans Hazir et Ulusoy (2020), comme l'utilisation de stratégies de planification qui permettent de prendre des décisions de manière dynamique tout au long de l'exécution du planning (Demeulemeester et Herroelen, 2002).

2.3.2.2. Les méthodes floues

Pour gérer l'incertitude, les méthodes floues (« fuzzy scheduling » en anglais) ont également été développées. Au lieu d'utiliser des probabilités comme dans les méthodes stochastiques, les paramètres incertains sont modélisés avec des nombres flous. Ceci est dû au fait que l'utilisation des méthodes floues se fait lorsqu'il n'est pas possible de déterminer la loi de probabilité pour la distribution des temps d'exécution des tâches. Wang (2004) a proposé d'utiliser la théorie des ensembles flous pour générer des plannings robustes, résistants aux imprévus.

Là où les méthodes floues peuvent représenter l'incertitude, elles peuvent aussi représenter les préférences des managers (Dubois et al. (2003)). Ces préférences sont modélisées via des contraintes définies par des ensembles flous et

fonctions d'appartenance. Liang (2010) a travaillé sur les préférences et utilisé la programmation floue pour modéliser des objectifs qui entrent en conflit. Les contributions sur les approches floues ont beaucoup progressé ces quinze dernières années, notamment dans le cadre de l'optimisation multi-objectif (Tavana et al. (2014), Xu et Feng (2014)).

2.3.2.3. Les méthodes robustes

Les méthodes robustes ont été élaborées pour anticiper les événements indésirables qui viendraient perturber le projet. Leur objectif est de minimiser les effets des aléas sur le projet en le rendant moins sensibles à ceux-ci (et donc robuste). Herroelen et Leus (2005) ont défini deux types de robustesse pour les solutions : robustesse de solution (ou stabilité) et robustesse de qualité. La stabilité consiste à maintenir les dates de départ des tâches par rapport aux variations observées. Ce type de robustesse vise à conserver la répartition des tâches et les écarts entre elles. À l'inverse, la robustesse de qualité vise à conserver la qualité de solution comme par exemple le makespan, au risque d'avoir un planning plus difficile à réaliser. Dans leurs travaux, Herroelen et Leus (2003) formulent un modèle mathématique pour générer une solution stable et développent un programme linéaire ainsi que quelques heuristiques. Rossi et al. (2016) ont également utilisé une méthode ayant pour objectif de maximiser la stabilité des solutions pour un problème de type ALBP en utilisant un PLNE. Chakraborty et al. (2017) ont travaillé sur un modèle robuste en termes de qualité et ont proposé six heuristiques différentes en fonction du problème rencontré.

Les approches robustes utilisent fréquemment le levier des « buffers » qui, lorsque basés sur le temps, consistent à ajouter du temps supplémentaire entre certaines tâches pour absorber les aléas (Van de Vonder et al. (2007), Zhang et al. (2018a)). D'un autre côté, il est possible de considérer des « buffers » pour les ressources et considérer des ressources supplémentaires pour anticiper des éventuels manques en ressource (Lambrechts et al. (2011)). Il est aussi possible d'associer ces deux types de buffers : Liang et al. (2020) ont intégré une solution utilisant à la fois les buffers temporels entre les tâches et l'affectation robuste des ressources dans une heuristique en deux phases. Gurevsky et al. (2013) ont élaboré l'approche robuste pour un problème de type ALBP. Leur objectif était de minimiser le nombre de stations de travail utilisées. Ils ont considéré les durées de tâches incertaines comme des intervalles de valeurs possibles et ils ont calculé le pire cas possible via une contrainte formulée comme un problème de type sac-à-dos (« knapsack problem »). Leur solution était jugée robuste lorsque celle-ci était optimale parmi l'ensemble des solutions faisables dans le pire cas. Enfin dans des

travaux récents, Chen et al. (2022) ont utilisé un algorithme d'Evolution Différentielle Discrete (ou « Discrete Differential Evolution » en anglais) pour gérer l'incertitude dans leur problème de type MRCPSP/max.

Les approches proactives sont donc utilisées pour les perturbations qui peuvent être anticipées, ce qui implique l'utilisation des méthodes réactives pour gérer les aléas qui n'ont pas pu être anticipés.

2.3.2.4. Les méthodes réactives

L'approche réactive permet de trouver un nouveau planning en effectuant une replanification qui prend en compte les nouvelles informations issues de l'aléa. Selon l'étude de Chaari et al. (2014), une telle replanification peut être réalisée de quatre manières différentes : à chaque fois qu'un aléa survient, à chaque fois qu'une nouvelle tâche est ajoutée à la liste des tâches à réaliser, à des intervalles de temps réguliers ou enfin quand le nombre d'aléas atteint un certain seuil.

Concernant les stratégies de replanification, Kuster et al. (2010) ont proposé une méthode de Local Rescheduling (LRS) qui consiste à replanifier seulement certaines parties de leur projet. Zhu et al. (2020) ont élaboré un modèle de replanification dans un contexte de ligne d'assemblage aéronautique lorsque les livraisons de certaines matières premières ne sont pas respectées par les fournisseurs. Leur méthode se base sur le machine learning et plus particulièrement sur le Support Vector Data Description (SVDD) pour déterminer la meilleure règle de replanification à utiliser. Elloumi et al. (2017) ont associé un algorithme génétique et une heuristique multi-objectif pour minimiser le makespan et une mesure d'interruption dans un MRCPSP. Zhu et al. (2005) ont de leur côté formulé un Programme Linéaire en Nombres Entiers (PLNE) pour réagir lorsque leur projet type RCPSP serait interrompu. Ce modèle PLNE a été conçu afin de trouver un nouveau planning qui redeviendrait identique au planning initial le plus vite possible.

Les méthodes de replanification sont nombreuses, et peuvent parfois même s'apparenter à de la simple planification en partant d'un point précis. Cependant, les méthodes de replanification ont un objectif qui diffère de la planification : celles-ci répondent à la nécessité de trouver une nouvelle solution en un temps raisonnable tout en gardant une certaine proximité avec la solution de départ. L'avantage de ce type d'approche est que l'on peut réagir rapidement à une interruption. La difficulté réside dans la nécessité de déterminer une méthode plus générique pour pouvoir pallier tout type d'aléa.

Il est également possible d’utiliser des méthodes hybrides associant le réactif et le proactif. De nombreux travaux abordent ces deux approches conjointement (Davari et Demeulemeester (2019), Lambrechts et al. 2008).

2.3.2.5. **L’analyse de sensibilité**

L’analyse de sensibilité est une approche qui peut commencer par la question « Et si... ? » à laquelle la réponse sera apportée en fonction des paramètres qui seront changés. Le type de questions auxquelles les chercheurs essaient généralement de répondre sont par exemple :

- Jusqu’à quand la solution reste optimale ?
- Quels sont les intervalles de variation des paramètres où la solution reste optimale ?
- Qu’est-ce que cela coûte de rester optimal dans tel cas de figure ?

Une telle étude permet de mieux évaluer la stabilité d’une solution et définir ses limites. Gurevsky et al. (2012) ont réalisé une étude de sensibilité pour leur problème de type SALBP (Simple Assembly Line Balancing Problem). D’autres méthodes et développements récents ont été analysés par Borgonovo et Plischke (2016).

2.4. L’ASPECT HUMAIN DANS LES CHAINES D’ASSEMBLAGE

La place de l’homme dans l’industrie est de plus en plus mise en avant en donnant de l’importance aux aspects physiologiques et mentaux du travail, trop longtemps délaissés (El Mouayni et al., 2019).

Les lignes d’assemblage faisant partie des processus industriels dans lesquels la main d’œuvre est encore très présente, la prise en compte de l’humain est ici aussi de plus en plus importante. A cause de la complexité du processus et des tâches lourdes qui le composent, prendre en compte l’aspect humain est devenu important pour les entreprises pour assurer un bon fonctionnement de leurs chaînes. Cette section présente comment, en partant de la notion d’équité, nous sommes arrivés à des indices de mesure qui permettent aujourd’hui de déterminer la difficulté d’un travail ou d’une tâche pour mieux mesurer l’impact sur la main d’œuvre et prévenir les blessures en mobilisant plus efficacement les forces.

La méthode de l'équilibre compétitif à partir de revenus égaux (CEEI) a été formalisée par Walras (1874) et Fisher (1892). Le CEEI repose sur le fait que les personnes d'un groupe ont un revenu égal (fictif) et peuvent acheter des biens provenant de ce qui doit être divisé. L'objectif est qu'ils disposent des biens qu'ils désirent pour maximiser leur satisfaction, et que personne n'envie un autre membre du groupe, sous réserve de son revenu limité. Une méthode basée sur ce concept a été élaborée par Budish (2011) qui l'a révisé pour partager les objets non divisibles entre les personnes.

Ces concepts d'équité ont amené des indices qui mesurent l'inégalité d'une allocation (d'une ressource, ou de tâches dans notre cas). L'indice de Gini, développé par Corrado Gini, est l'un d'entre eux. Lorsqu'il est utilisé en économie, son objectif est de mesurer la dispersion statistique, représentant l'inégalité de revenu ou de richesse entre des populations ou des groupes de taille différente. L'indice de Gini est défini mathématiquement sur la courbe de Lorenz. De plus amples informations sur l'indice de Gini et la courbe de Lorenz peuvent être trouvées dans Moulin (1988).

Bien que dans ces cas, les recherches et indices sont axées sur l'équilibre de la répartition de ressources ou de biens, l'équité peut aussi avoir un aspect protecteur en cherchant à réduire les impacts nocifs d'une activité sur la santé de la main d'œuvre d'un projet. Ces indices, ergonomiques, sont devenus très importants dans la compréhension d'un processus et de l'impact de celui-ci sur les hommes et femmes qui travaillent dessus.

2.4.1. Les indices pour la mesure des impacts physiologiques

Les indices pour mesurer la charge ergonomique d'une action ou d'une tâche sur un individu ou un groupe se sont développés avec le temps. Parmi eux nous pouvons retrouver :

- Rapid Upper Limb Assessment (RULA) : mesure les risques de troubles musculo-squelettiques (TMS) par rapport à de mauvaises postures ou lourdes charges sur le haut du corps.
- Occupational Repetitive Action (OCRA) : mesure les risques de TMS du haut du corps liés à des mouvements répétitifs.
- Ovako Work Posture Analysis System (OWAS) : mesure les risques de TMS par rapport à la posture de l'ensemble du corps lors d'une action.

- Rapid Entire Body Assessment (REBA) : mesure les risques de TMS de l'ensemble du corps sur des tâches spécifiques. C'est une méthode qui a pour objectif d'être plus globale et facile à mettre en place.

Ces indices sont parmi les plus représentés dans la littérature à notre connaissance. Kee (2021) compare la méthode du RULA à deux autres méthodes (OWAS et REBA) pour évaluer les TMS sur les données de différents sites industriels. Joshi et Deshpande (2021) ont mené une analyse approfondie sur la façon dont chaque facteur (position des bras, tronc, jambes) affecte le score donné par l'indice RULA. Ils ont identifié la position des bras comme le facteur le plus influent. L'indice RULA a été utilisé par Wilhelm et al. (2021) avec leur indicateur ErgoTakt pour capturer et évaluer l'ergonomie sur leur ligne d'assemblage en temps réel. Boenzi et al. (2016) utilisent l'indice Occupational Repetitive Action (OCRA) dans un cas industriel pour alléger la charge ergonomique sur un sous-groupe spécifique de travailleurs. Cet indice évalue les risques de manipulation répétitive à haute fréquence. L'algorithme associé évalue chaque rotation de poste pour sélectionner la plus adaptée.

Mokhtarzadeh et al. (2021) utilisent plusieurs indices. Ils ont développé un cadre en deux étapes pour équilibrer une ligne d'assemblage en U à modèle mixte. Dans la première partie, ils évaluent la difficulté des tâches à l'aide de différents indices : OCRA, JSI, NIOSH, EAWS, OWAS et COPSOQ. Leur objectif est d'obtenir une notation plus précise des tâches en association avec des méthodes comme ELECTRE TRI qui vise à assigner des éléments à des classes sur la base d'un certain nombre de critères. Dans une partie secondaire, un modèle mathématique est construit pour assigner les tâches à chaque station de travail, ce qui rend leur cadre assez complet et adapté à la fois aux problèmes de grande et de petite taille.

Katirae et al. (2021) ont créé leur propre matrice comprenant les caractéristiques individuelles et les perceptions des travailleurs pour chaque tâche en fonction de leur expérience et de leurs efforts physiques. Pour aider ces travailleurs à évaluer ces tâches, ils utilisent l'échelle de Borg, introduite par Borg (1990) et un modèle bi-objectif pour résoudre une instance de petite taille.

Arkhipov et al. (2018a) ont proposé une méthode d'affectation en prenant en compte l'ergonomie dans une ligne d'assemblage à faible volume avec des temps de cycle longs. Cette méthode utilise à la fois un modèle de PPC et un modèle PLNE et elle a été testée sur deux instances industrielles.

Dans le contexte des lignes d'assemblage, l'indice le plus approprié pour mesurer les risques ergonomiques semble être le RULA (McAtamney et Corlett (1993)), les tâches étant principalement exigeantes pour le haut du corps. Grâce à

cette méthode, l'évaluation de la difficulté des tâches peut être effectuée en résultant à un score ergonomique.

2.5. PROBLEMATIQUE ET QUESTIONS DE RECHERCHE

Dans cette thèse, nous nous intéressons à un problème de type Multi-Skilled Resource-Constrained Project Scheduling Problem avec General Precedence Relations (MS-RCPSP/GPR) ainsi que des ressources renouvelables avec des quantités variables. Nous considérons une configuration de lignes d'assemblage avec trois stations. Un contexte de replanification similaire a été précédemment considéré par Battaïa et al. (2018) où une approche de PPC a été présentée sur une situation restreinte avec une seule station d'assemblage où le critère conventionnel du makespan était utilisé. Zhu et al. (2020) ont également considéré un problème d'ordonnancement réactif d'une ligne d'assemblage aéronautique en ayant pour objectif la minimisation des coûts engendrés par la replanification.

Dans le contexte industriel considéré dans le cadre de cette thèse, la priorité des managers dans le cas d'une replanification est le respect du temps de cycle et du planning précédent en maintenant au plus les affectations initiales des compagnons à leurs tâches. Comme une interruption de la ligne représente des pertes financières importantes, une solution de replanification doit être trouvée le plus rapidement possible.

De plus, par l'importance apportée à l'aspect humain par notre partenaire et en association avec les autres axes de développement du projet Per4mance, les aspects ergonomiques et organisationnels sont intégrés dans le processus de planification. Considérer l'aspect humain dans les lignes d'assemblage aéronautiques consiste à prendre en compte les préférences des individus, les relations qu'ils ont pu développer avec leurs collègues en formant des équipes, leurs facilités techniques et physiques pour leur attribuer des tâches mais aussi à faire en sorte que leur sécurité soit garantie au mieux sur le court mais aussi sur le long terme en évitant les effets néfastes sur leur santé de certaines positions ou efforts physique conséquents qui peuvent résulter en troubles musculosquelettiques.

En résumé, la problématique suivante sera considérée dans la suite de ce manuscrit :

« Quels critères d'optimisation utiliser pour trouver une solution de replanification faisable tout en considérant l'ensemble des contraintes du problème en un laps de temps restreint ? Comment pouvons-nous incorporer les notions d'ergonomie et de préférences organisationnelles dans la planification ? »

L'objectif de ce manuscrit est donc d'apporter des réponses à ces questions et la première étape consiste à établir un modèle de replanification efficace pour

répondre aux aléas lourds qui surviennent lors de l'assemblage. La prochaine section va donc présenter les travaux publiés dans Lovato et al. (2020, 2022a) concernant cette replanification et la proposition de critères en escalier.

3.

**MODELISATION DU
PROBLEME DE
REPLANIFICATION DES
LIGNES D'ASSEMBLAGE
AERONAUTIQUES**

3.1. MOTIVATIONS

En raison du nombre important de tâches à réaliser et de la complexité du processus d'assemblage, de nombreux événements imprévus peuvent perturber le calendrier initial d'exécution des tâches et doivent être traités par les responsables de ligne. Ces perturbations peuvent être liées à un retard d'approvisionnement en pièces, à un problème de qualité ou au remplacement d'un compagnon. Elles peuvent survenir à tout moment et obliger certains compagnons en activité à s'arrêter et à attendre de nouvelles instructions. En pratique, actuellement, la replanification des tâches repose sur l'expérience des managers et elle est considérée comme une charge très stressante pour eux. Nous voulons donc développer une approche d'optimisation efficace et rapide qui permettra d'atteindre de meilleures solutions en moins de temps tout en respectant les contraintes du terrain. Il est important de préserver autant que possible le planning initial afin de limiter les changements dans les plans de travail des compagnons et de réduire le nombre d'erreurs humaines possibles.

Nous considérons une ligne avec plusieurs stations où l'objectif de makespan, habituellement considéré dans ce contexte, n'est pas pertinent. En effet, si le temps disponible pour l'exécution des tâches restantes affectées par la perturbation est suffisant pour respecter l'IP, le makespan est vu comme une contrainte et non comme un objectif. Dans ce cas, l'objectif est de garder le planning proche du planning initial. Dans certains cas, ce dernier est si important que les responsables préfèrent employer un compagnon supplémentaire afin de réaliser les tâches restantes dans les temps et éviter de modifier le planning existant. Cependant, si le retard créé par la perturbation est plus long que le temps disponible jusqu'à la fin de l'IP, un nouvel intervalle de temps égal à un jour ouvrable sera donné à la station perturbée puisqu'il n'est pas possible de déplacer l'avion à un autre moment. Cette observation a motivé le développement du critère de makespan en escalier proposé dans cette étude qui convient mieux à l'organisation réelle de la ligne.

Il est important de mentionner que plus de temps est passé à calculer nouveau planning, moins il en reste pour le mettre à l'exécution. C'est pourquoi il est essentiel de réduire au maximum ce temps de calcul. Cette étude vise à fournir aux gestionnaires un outil de décision puissant qui calculera rapidement une solution de replanification en fonction des objectifs des décideurs. Le temps gagné grâce à la génération rapide d'une solution de replanification permettra aux gestionnaires de contrôler les conséquences négatives de la perturbation survenue.

3.2. MODELE DE REPLANIFICATION

Nous avons vu dans l’étude de la littérature que les heuristiques et méta heuristiques étaient bien plus représentées que les modèles de PPC ou de Programmation Linéaire en Nombres Entiers (PLNE) pour les méthodes réactives. Comme le montrent Polo-Mejía et al. (2019), les modèles exactes peuvent être en difficulté lorsqu’il faut fournir une solution de bonne qualité pour une instance de grande taille.

Cependant, d’autres travaux montrent que la PPC peut fournir des solutions efficaces même pour les instances de grande taille, comme par exemple Arkhipov et al. (2018b) et Kizilay et Çil (2021). Ainsi nous faisons le choix de nous orienter vers un modèle de PPC en utilisant l’environnement CPLEX Studio d’IBM et la librairie CP Optimizer. Dans ce qui suit, nous présentons quelques fonctions qui seront utilisées dans notre modèle.

3.2.1. Fonctions et méthodes CP Optimizer

Pour mieux comprendre les notations utilisées et les éléments du modèle, nous allons présenter dans un premier temps comment sont définies certaines variables, fonctions et méthodes dans la librairie CP Optimizer. Concernant les variables de décision, il existe les variables classiques comme les nombres entiers, réels et nombres binaires, mais il existe aussi les intervalles, spécifiquement conçus pour modéliser des problèmes de planification.

- Variable de décision de type « intervalle »

Un intervalle est une variable de décision caractérisée par la date de début de la tâche qu’elle représente, sa date de fin, son intensité et sa durée (figure 10). Ses dates de début et de fin sont entières, inconnues et seront déterminées par la solution obtenue. Sa durée est considérée comme donnée en entrée mais peut être étendue ou raccourcie dans la solution. L’intensité d’un intervalle peut être représentée comme la « vitesse » à laquelle la tâche est réalisée. Une intensité de 100% signifie que pour un intervalle de durée 10 (unités de temps), il faudra 10 unités pour terminer cette tâche. En revanche, si l’intensité est de 50%, alors l’efficacité est divisée par deux et il faudra 20 unités pour terminer la tâche. A 0%, la tâche est considérée en pause. Nous considérons pour la totalité de nos travaux une intensité de 100%. Enfin un intervalle peut être considéré comme optionnel, c’est-à-dire que la tâche qu’il représente peut faire partie de la solution finale ou pas.

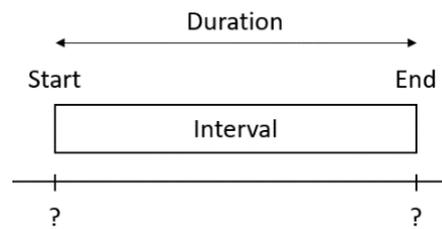


Figure 10 - Schématisation d'une variable de type "intervalle"

Certaines méthodes sont directement liées aux caractéristiques des intervalles, et seront utilisées pour la modélisation des contraintes :

- $startOf(i)$ Retourne la date de départ de l'intervalle i .
- $endOf(i)$ Retourne la date de fin de l'intervalle i .
- $presenceOf(i)$ Retourne 1 si l'intervalle i est présent dans la solution, 0 sinon.

- Fonction d'état

Une fonction d'état a pour objectif de montrer l'évolution de l'état d'un élément donné du problème. L'évolution de cette fonction est contrainte par les intervalles définis. Les fonctions d'état servent à définir des conditions pour l'exécution des tâches et donc la détermination des dates de début ou de fin des intervalles. Plusieurs méthodes servent à définir les états que peut prendre cette fonction.

- $alwaysEqual(f, i, n)$ Impose la valeur n à la fonction d'état f sur l'intervalle i .
- $alwaysNoState(f, i)$ Impose la valeur 0 à la fonction d'état f sur l'intervalle i .

- Fonction cumulative

Une fonction cumulative est une fonction utilisée pour représenter la somme des contributions individuelles de chaque variable intervalle. Ce type de fonction sert notamment pour déterminer la quantité utilisée d'une ressource donnée par l'ensemble des tâches en cours. Les méthodes associées sont les suivantes.

- $pulse(i, q)$ Impose la valeur q sur l'intervalle i à la fonction cumulative, et 0 le reste du temps.
- $step(t, q)$ La fonction prend la valeur q à partir de l'unité de temps t .

- Fonctions diverses

Pour alléger la présentation du modèle, nous allons présenter ici l'ensemble des fonctions issues de CP Optimizer qui seront utilisées.

- $endBeforeStart(i_1, i_2)$ Stipule que l'intervalle i_1 doit être terminé avant le début de l'intervalle i_2 dans la solution.

$endAtStart(i_1, i_2)$	Stipule que l’intervalle i_1 doit se terminer au même moment où l’intervalle i_2 démarre dans la solution.
$noOverlap(\{i_1, \dots, i_n\})$	Indique qu’aucun des intervalles de l’ensemble $\{i_1, \dots, i_n\}$ ne peut être réalisé au même moment dans la solution (aucune unité de temps ne doit comprendre plus d’un de ces intervalles sur l’ensemble des intervalles en cours d’exécution).
$alternative(i, \{j_1, \dots, j_n\}, n)$	Si l’intervalle i est actif dans la solution, alors exactement n intervalles de l’ensemble $\{j_1, \dots, j_n\}$ doivent également être présents.
$forbidStart(i, f)$	Interdit à l’intervalle i de démarrer sur une unité de temps où la fonction f est égale à 0.
$forbidEnd(i, f)$	Interdit à l’intervalle i de se finir sur une unité de temps où la valeur de la fonction f est égale à 0.

Maintenant que l’ensemble des fonctions utilisées sont présentées, nous pouvons présenter le modèle de replanification que nous avons élaboré.

3.2.2. Les données en entrée

Les données d’entrée du problème de replanification comprennent les informations du planning initial (heure de début de toutes les tâches et affectation compagnon-tâche) ainsi que les informations issues de l’aléa survenu (retard de matière première, décalage de tâche, absence de compagnon).

Ensembles :

I	Tâches ;
E	Relations de précédence ;
E_{FD}	Relations de précédence de type Fin-Début ($E_{FD} \in E$) ;
E_{FDE}	Relations de précédence de type Fin-Début Exact ($E_{FDE} \in E$) ;
E_{FF}	Relations de précédence de type Fin-Fin ($E_{FF} \in E$) ;
E_{DD}	Relations de précédence de type Début-Début ($E_{DD} \in E$) ;
E_{DDE}	Relations de précédence de type Début-Début Exact ($E_{DDE} \in E$) ;
P	Professions possibles ;
O	Compagnons (ressource humaine) ;
O_p	Compagnons maîtrisant la profession p ($p \in P$) ;
M	Ressources matérielles ;
A	Zones ;
A_i	Zones utilisées par la tâche i ($i \in I$) ;

\bar{A}_i	Zones exclues par la tâche i ($i \in I$) ;
W	Stations d'assemblage ;
I_w	Tâches à réaliser sur la station w ($w \in W$) ;
S	Tâches déjà terminées ou commencées et qui ne peuvent être réaffectées ;
D	Tâches à réaffecter.

Paramètres liés à la planification :

pt_i	Temps d'exécution de la tâche i ($i \in I$) ;
$skill_o$	Profession maîtrisée par le compagnon o ($o \in O$) ;
T	Date de fin du premier planning ;
cal_o	Type de calendrier pour le compagnon o ($o \in O$) ;
qte_m	Quantité disponible pour la ressource m ($m \in M$) ;
CT	Nombre d'heures minimal pour lequel un compagnon supplémentaire peut être engagé ;
day_i	Booléen qui prend la valeur 1 quand la tâche i ne peut être réalisée que durant la journée, 0 sinon ($i \in I$) ;
$earliest_d$	Date de démarrage au plus tôt pour la tâche d ($d \in D$) ;
$needed_{i,p}$	Nombre de compagnons maîtrisant la profession p nécessaires à la réalisation de la tâche i ($i \in I, p \in P$) ;
$needed_{i,m}$	Nombre d'unités de la ressource matérielle m nécessaire à la réalisation de la tâche i ($i \in I, m \in M$) ;
WH	Nombre d'heures de travail par jour.

Paramètres liés à la replanification :

$RStart$	Date à laquelle le processus d'assemblage doit redémarrer ;
old_end_i	Date de fin de la tâche i dans le planning initial ($i \in I$) ;
old_oper_s	ID du compagnon assigné à la tâche s dans le planning initial ($s \in S$) ;
old_start_s	Dates de départ de la tâche s ($s \in S$) ;
$old_x_{i,o}$	Booléen égal à 1 si la tâche i était assignée au compagnon o dans le planning initial, 0 sinon ($i \in I, o \in O$).

Fonctions

Pour représenter certaines contraintes, nous définissons des fonctions pour représenter des ressources ou horaires de travail. Nous les présentons ici.

Les chaînes d'assemblage d'avions, l'IP s'étendant sur plusieurs jours, le calendrier de travail personnel des compagnons doit être respecté. Ce calendrier définit leurs heures de travail ainsi que leurs heures de repos (nuit, week-end) et il est représenté par la fonction $Calendar_o$ déclarée pour chaque compagnon o qui est égale à 0 lorsque le compagnon est en repos. Le calendrier est défini grâce à un

de pauses définies en amont de la planification qui permettent de créer la fonction de disponibilité via la méthode « *step* ».

Enfin, la fonction « $use_{a,w}$ » représente l'utilisation de la zone « w » pour l'avion « a » et est une fonction d'état. Elle servira à contrôler l'exécution des tâches sur ces zones et à gérer les interdictions imposées.

3.2.3. Les critères de décision

Chaque critère est défini sur la base d'une ou plusieurs variables de décision. Dans notre problème de replanification, nous devons réordonnancer un ensemble de tâches $i \in I$. Les variables que nous avons définies sont les suivantes :

- $intvl_i$: intervalle d'exécution de la tâche i ($i \in I$).
- $x_{i,o}$: intervalle optionnel où le compagnon o réalise la tâche i . Sa taille est égale à la durée de i et l'intervalle est contraint par le calendrier de o ($i \in I, o \in O$) (c'est-à-dire l'agenda de travail du compagnon o).

Une fonction d'état est également utilisée pour les zones liées à chaque avion :

- $use_{a,w}$ est la fonction d'état permettant de contrôler la disponibilité de la zone a ($a \in A$) au poste de travail w ($w \in W$). Sa valeur est égale à 0 lorsque la zone a n'est utilisée par aucune tâche et à 1 lorsqu'une tâche est exécutée dans cette zone.

3.2.3.1. Critères génériques

- **Makespan**

Le premier critère exprimé ici est le makespan, que nous cherchons à minimiser.

$$\max_{i \in I, o \in O} endOf(x_{i,o}) \quad (1)$$

- **Retard**

Le retard des tâches est également un critère connu dans la littérature, utilisé avec le « démarrage au plus tôt » (« right shifting » en anglais). Ce critère est exprimé par (2) et calcule le retard total pour toutes les tâches par rapport au planning initial. Ce critère est à minimiser.

$$\sum_{i \in I} |endof(intvl_i) - old_end_i| \quad (2)$$

3.2.3.2. Critères spécifiques à notre cas industriel

- **Stabilité du planning**

La stabilité de la planification peut être évaluée en fonction du nombre de tâches restant affectées au même compagnon avant et après la perturbation. Le critère présenté dans l'équation (3) a donc été élaboré pour respecter le planning des tâches pour chaque compagnon. Comme il a été mentionné précédemment, la possibilité de conserver le même planning de travail après la perturbation évite aux compagnons des erreurs humaines liées au travail non planifié et donc du stress généré. Ce critère doit être maximisé.

$$\sum_{i \in I, o \in O} old_{x_{i,o}} * presenceof(x_{i,o}) \quad (3)$$

- **Charge du travailleur supplémentaire**

Dans la stratégie de replanification, un compagnon supplémentaire peut être employé pour réduire le makespan. Cette option a un coût qu'il faut minimiser. Ce critère est exprimé par l'équation (4) et représente le nombre d'heures que le compagnon supplémentaire doit travailler dans la solution de replanification. L'indice o^+ représente l'indice du compagnon supplémentaire. Nous cherchons à minimiser cette valeur.

$$\sum_{i \in I} presenceof(x_{i,o^+}) * pt_i \quad (4)$$

3.2.3.3. Critères escalier

Les critères escaliers ont été proposés dans cette thèse pour répondre au besoin des managers de se conformer à des règles particulières. Par exemple, gagner du temps et terminer un assemblage à 16 heures au lieu de 17 heures n'apporte aucun avantage car les compagnons sont payés à la journée et non à l'heure. La même philosophie peut être appliquée au compagnon supplémentaire qui est employé non pas à l'heure mais par jours de travail. En outre, si l'IP ne peut pas être respecté, indépendamment de la quantité de temps excessif, un jour de travail supplémentaire sera accordé afin de terminer l'assemblage. Pour calculer correctement ces critères en escalier, nous avons besoin de deux autres variables de décision. M^* représente la valeur du makespan escalier (voir le critère (5) et la contrainte (18) pour plus de détails). O^* représente la valeur de la charge de travail en escalier du compagnon supplémentaire (voir le critère (6) et la contrainte (19) pour plus de détails).

- **Makespan en escalier**

En cas de perturbation, selon la durée de la perturbation survenue, il n'est pas toujours possible de terminer avant la fin de l'IP. Cependant, le travail dans les chaînes d'assemblage aéronautiques et le transfert entre les stations d'assemblage sont organisés de telle sorte que l'avion ne peut pas être déplacé à tout moment mais seulement à la fin d'une journée. Cela signifie que si le makespan dépasse l'IP de quelques heures seulement, une journée entière sera consacrée à la fin de ce travail, perdant le reste des heures « vides ». Cela signifie que l'ensemble du processus, à tous les niveaux (assemblage, changement de station, transfert vers les sites suivants), est mis en pause durant le reste de la journée où s'est terminé l'assemblage. Cette « immobilisation » des avions durant le reste de la journée résulte indirectement en des coûts de stockage (main d'œuvre payée et autres charges impliquées dans l'assemblage en temps normal) que le partenaire industriel veut éviter. Car en effet, même si la minimisation des coûts ne sera pas un axe abordé dans ce manuscrit, cela ne signifie pas que nous ne nous soucions pas de l'optimisation de l'utilisation des ressources engagées dans le processus.

Cela change la priorité des fonctions objectives, puisque dans ce cas il est plus important de garder le planning aussi proche que possible du planning initial que de minimiser le makespan s'il y a un jour d'extension. Il en sera de même pour les jours suivants, c'est-à-dire 2 jours de retard sur 3 jours. Pour modéliser cet objectif des gestionnaires, nous proposons un nouveau critère : le “makespan en escalier” qui est présenté dans la figure 11.

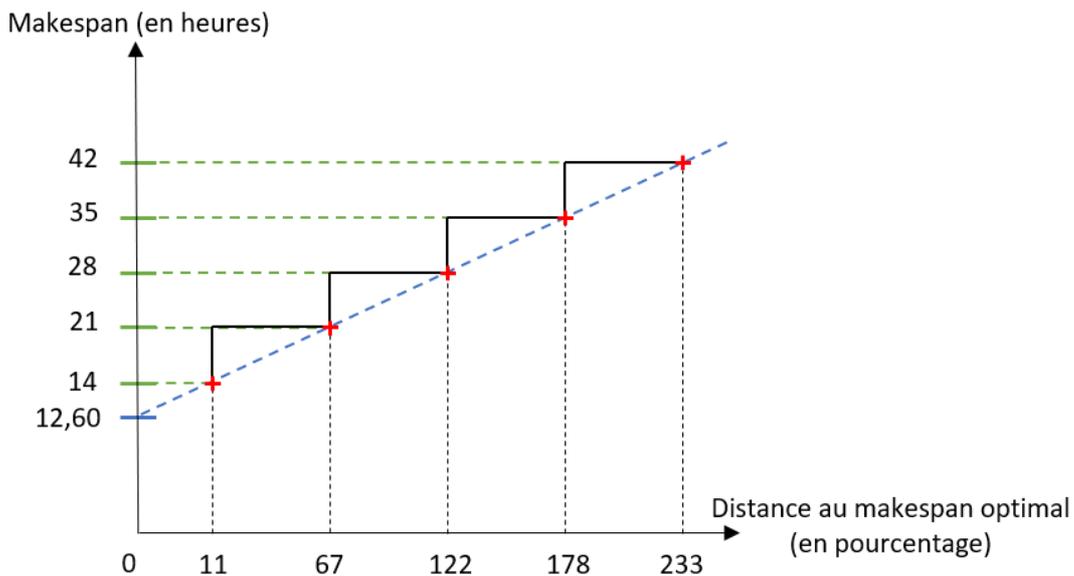


Figure 11 - Représentation des valeurs que prend le makespan en escalier comparé au makespan

Les valeurs du makespan en escalier sont marquées par les croix rouges tandis que les valeurs du makespan classique sont représentées par la ligne bleue pointillée. La ligne noire représente la valeur que prend le makespan en escalier. Chaque croix rouge représente la fin d'un jour. Ainsi, le makespan en escalier sera calculé comme la date de fin du jour actuel. La meilleure valeur pour le makespan en escalier est donc celle en bas à gauche et représente la fin de l'IP dans notre cas. Nous voulons donc minimiser ce critère.

$$M^* \quad (5)$$

$$s. t. \max_{i \in I, o \in O} \text{endof}(x_{i,o}) \leq T + M^* * WH \quad (18)$$

Pour calculer M^* , la contrainte (18) est nécessaire. Ici, T représente la date de fin de l'IP et WH donne la quantité de temps de travail par jour (qui représente également la granularité du temps, ici une journée). Ainsi, lorsque M^* est égal à 0, cela veut dire que notre makespan est inférieur à la date correspondant à la fin de l'IP.

- **Charge de travail du compagnon supplémentaire en escalier**

La philosophie appliquée au makespan peut aussi s'appliquer au compagnon qui vient en support sur l'assemblage. Nous voulons d'une part limiter la mobilisation de ce compagnon, car chaque heure avec de la main d'œuvre supplémentaire engendre obligatoirement des coûts (coûts qui ne sont pas pris en compte dans le modèle), et d'une autre part nous ne voulons pas faire déplacer un employé pour assumer une charge de travail négligeable comparé à la contrainte pour le compagnon de devoir modifier son planning personnel pour pouvoir répondre à la demande. C'est pourquoi nous pensons qu'une version escalier peut être utile pour ce critère afin de définir son temps de travail sur un multiple de 7h (une journée de travail, par exemple). Cela permet de déterminer des journées complètes afin de rentabiliser à la fois pour l'entreprise l'argent investi, et pour l'employé le temps investi. Cette granularité peut évidemment être ajustée au besoin, comme par exemple à la demi-journée. Nous cherchons donc à minimiser ce critère.

$$C^* \quad (6)$$

$$s. t. \sum_{i \in I} \text{presenceOf}(x_{i,o^+}) * pt_i \leq C^* * CT \quad (19)$$

Similairement à (5), la contrainte (19) est nécessaire pour calculer C^* . CT représente la granularité de la charge de travail que nous voulons attribuer au compagnon supplémentaire (ici sept heures, soit une journée de travail car nous voulons avoir un ensemble de journées pour maximiser la mobilisation du compagnon supplémentaire). Une valeur équivalente à 0 pour C^* signifie qu'il est possible de revenir à l'ancienne date de fin pour l'IP sans compagnon

supplémentaire, ce qui est un cas de figure normalement rare dans un cas où un aléa très lourd survient, ou alors qu'il est impossible de revenir à l'ancien IP et qu'il faut des jours supplémentaires (et qu'il est possible de terminer sans compagnon supplémentaire).

3.2.3.4. *Modèle multi-objectif*

Comme il a été mentionné plus haut, les critères que les managers doivent prioriser dépendent de la situation : si après replanification l'IP peut être respecté sans ressources supplémentaires, alors le planning original doit être préservé autant que possible. Sinon, la priorité est de respecter l'IP et de minimiser les ressources supplémentaires nécessaires. Cependant, si l'IP ne peut être respecté même avec des ressources supplémentaires, alors la priorité est de minimiser le nombre de jours de travail supplémentaires nécessaires et de préserver le planning initial autant que possible.

Nous proposons un ensemble de critères qui répondent à ces attentes et qui peuvent être utilisées dans un ordre lexicographique choisi par le manager grâce à la fonction CPO $staticLex(A, B, C)$, où A, B, C sont des critères de décision. Cet ordre (A, B, C) correspond au traitement lexicographique. Le premier critère est optimisé puis dans les cas d'égalité le deuxième puis le troisième. Ainsi, en changeant l'ordre des critères A, B et C, différentes solutions sont trouvées par le solveur.

Situons-nous dans un exemple où la replanification de la ligne d'assemblage est à effectuer. Sachant que l'ordre lexicographique adopte une priorité forte sur les critères (i.e. on ne valorise pas une amélioration importante du critère n°2 si on peut très légèrement améliorer le critère n°1), l'utilisation des critères escaliers est importante. Dans un cas où nous sommes en fin de planning, nous pourrions par exemple poser « A = makespan escalier » pour limiter le nombre de jours supplémentaires et « B = heures du compagnon supplémentaire » puis « C = Affectations » pour limiter au maximum l'utilisation de la main d'œuvre supplémentaire. Si la nécessité de faire recours à un jour supplémentaire est quasiment certaine, cela permet d'avoir une marge importante pour finir le travail restant sans engager de main d'œuvre supplémentaire.

3.2.4. Les contraintes du problème

3.2.4.1. Contraintes liées aux tâches

L'ensemble de contraintes (7) représente l'ensemble des relations de précédence généralisées telles que définies dans la section 1 et correspondant aux cinq catégories suivantes : fin-début, fin-début exact, début-début, début-début exact et fin-fin.

$$\left\{ \begin{array}{ll} \text{endBeforeStart}(\text{intvl}_i, \text{intvl}_j), & \forall (i, j) \in E_{FD} \\ \text{endAtStart}(\text{intvl}_i, \text{intvl}_j), & \forall (i, j) \in E_{FDE} \\ \text{startBeforeStart}(\text{intvl}_i, \text{intvl}_j), & \forall (i, j) \in E_{DD} \\ \text{startAtStart}(\text{intvl}_i, \text{intvl}_j), & \forall (i, j) \in E_{DDE} \\ \text{endBeforeEnd}(\text{intvl}_i, \text{intvl}_j), & \forall (i, j) \in E_{FF} \end{array} \right. \quad (7)$$

Le nombre de compagnons et de ressources matérielles étant limités, il est nécessaire de vérifier qu'à tout moment de l'exécution, ces limites ne sont pas dépassées. Pour contrôler la quantité de ressource nécessaire, nous utilisons la méthode *pulse(int, q)* pour modéliser l'utilisation de chaque ressource au cours du temps par l'ensemble des tâches. La contrainte (8) contrôle les ressources matérielles tandis que la contrainte (9) contrôle l'utilisation des ressources humaines. Dans ces contraintes, la valeur maximale de la fonction cumulative ne doit pas dépasser la capacité des ressources matérielles ou le nombre de compagnons disponibles.

$$\sum_{i \in I} \text{pulse}(\text{intvl}_i, \text{needed}_{i,m}) \leq \text{capacity}_m, \quad \forall m \in M \quad (8)$$

$$\sum_{i \in I} \text{pulse}(x_{i,o}, \text{needed}_{i,p}) \leq \text{sk_oper}_p, \quad \forall p \in P, \forall o \in O \quad (9)$$

Comme chaque compagnon ne peut effectuer qu'une seule tâche à un moment donné, nous utilisons la méthode « *noOverlap* » pour s'assurer qu'un compagnon n'ait qu'une tâche à réaliser à la fois (contrainte (10)).

$$\text{noOverlap}(x_{i,o} : \forall i \in I \mid \text{needed}_{i,p} > 0), \quad \forall p \in P, \forall o \in O \quad (10)$$

Il est également nécessaire de vérifier que le bon nombre de compagnons est utilisé pour chaque tâche. Pour cela, nous utilisons la méthode « *alternative* » dans la contrainte (11).

$$\begin{aligned} & \text{alternative}(\text{intvl}_i, x_{i,o} : \forall o \in O \mid p_o = p, \text{needed}_{i,p}), \\ & \forall i \in I, \forall p \in P \mid \text{needed}_{i,p} > 0 \quad (11) \end{aligned}$$

Afin d'assurer la compatibilité entre les tâches et les compétences des compagnons, nous utilisons la négation de la fonction « *presenceOf* » avec « ! » pour forcer un intervalle représentant une association entre un compagnon et une tâche impossible à être inactif. Nous accédons à la compétence du compagnon o à travers p_o et la comparons avec les compétences requises pour la tâche avec $\text{needed}_{i,p}$. Si cette première condition est égale à 0, alors la tâche ne peut pas être affectée au compagnon. La deuxième condition " $\text{day}_i \ \&\& \ \text{cal}_o \neq 1$ " permet d'interdire l'affectation des tâches uniquement disponibles en journée à des compagnons susceptibles de travailler de nuit, ce qui entraînerait une incompatibilité.

$$\begin{aligned} & \text{!presenceOf}(x_{i,o}), \\ & \forall i \in I, \forall o \in O \mid (\text{needed}_{i,p_o} = 0 \vee (\text{day}_i \ \&\& \ \text{cal}_o \neq 1)) \quad (12) \end{aligned}$$

3.2.4.2. **Contraintes liées à la chaîne d'assemblage aéronautique**

Nous utilisons les fonctions « *forbidStart* » et « *forbidEnd* » pour limiter l'exécution des intervalles en fonction des calendriers de disponibilité des compagnons. Si toutefois l'intervalle $x_{i,o}$ est actif lorsque Calendar_o est égal à 0, alors la longueur de $x_{i,o}$ est étendue, ce qui signifie que la durée de la tâche est étendue de la même durée que la pause du compagnon.

$$\text{forbidStart}(x_{i,o}, \text{Calendar}_o), \quad \forall i \in I, \forall o \in O \quad (13)$$

$$\text{forbidEnd}(x_{i,o}, \text{Calendar}_o), \quad \forall i \in I, \forall o \in O \quad (14)$$

Bien que les zones d'avion puissent être considérées comme des ressources disjonctives comme les ressources matérielles, l'aspect d'exclusion les rend légèrement différentes. Afin de mieux modéliser ces contraintes de zone, nous utilisons les méthodes « *alwaysEqual* » et « *alwaysNoState* » pour contrôler l'utilisation des zones via les fonctions « $\text{use}_{a,w}$ ». Les contraintes (15) et (16) vérifient que chaque fois qu'une zone est exclue, aucune autre tâche utilisant cette zone n'est exécutée au même moment.

$$alwaysEqual(use_{a,w}, intvl_i, 1), \quad \forall a \in A_i, \forall i \in I_w, \forall w \in W, \quad (15)$$

$$alwaysNoState(use_{a,w}, intvl_i), \quad \forall a \in \bar{A}_i, \forall i \in I_w, \forall w \in W, \quad (16)$$

En plus des contraintes (15) et (16), la contrainte suivante permet de vérifier que chaque zone est utilisée par au plus une tâche en même temps :

$$\sum_{i \in I \mid a \in A_i \& i \in I_w} pulse(intvl_i, 1) \leq 1, \quad \forall a \in A, \forall w \in W \quad (17)$$

3.2.4.3. Contraintes liées à la replanification

Comme il a déjà été mentionné, il est important pour les managers de ne pas trop modifier le planning initial. Il est également nécessaire de prendre en compte les informations sur les tâches qui ont déjà été terminées ou commencées au moment de la prise en compte de l'aléa. Afin de fixer leur heure de début, nous utilisons la méthode « *startOf(int)* » qui force le début de l'intervalle *int*. Ainsi, la contrainte (20) fixe la date de début et l'affectation de compagnon pour chaque tâche déjà commencée avant la perturbation afin de les garder identiques au planning précédent. La contrainte (21) assure que toutes les tâches non commencées avant la perturbation ne peuvent être commencées qu'après *RDate*, le moment du redémarrage de la ligne. La contrainte (22) modélise le retard des tâches perturbées en les forçant à commencer après la date qui a été établie par les gestionnaires comme la date à laquelle le travail peut être redémarré.

$$startOf(x_{s,old_o_s}) = date_s, \quad \forall s \in S \quad (20)$$

$$startOf(intvl_i) \geq RDate, \quad \forall i \in I \setminus S \quad (21)$$

$$startOf(intvl_d) \geq earliest_d, \quad \forall d \in D \quad (22)$$

3.3. PREPARATION DES TESTS EXPERIMENTAUX

Les expériences de calcul ont été réalisées sous l'environnement IBM ILOG CPLEX Optimization Studio version 12.10 et la bibliothèque CP Optimizer (CPO) incorporée. Le processeur utilisé était un Intel(R) Core (TM) i7-8665U CPU @ 1.90GHz en plus de 16Go de mémoire.

3.3.1. Création des instances

Les instances de test ont été construites sur la base des informations fournies par notre partenaire industriel. La ligne d'assemblage étudiée contient 3 stations d'assemblage, 472 tâches, 879 relations générales de précédence, 409 zones d'avion, 3 ressources matérielles (chacune limitée à une unité), 11 professions différentes qui sont maîtrisées par 20 compagnons différents (un compagnon ne maîtrisant qu'une seule profession). Un compagnon supplémentaire est disponible pour la replanification maîtrisant la profession associée à la tâche retardée.

Deux groupes d'instances ont été créés pour tester notre modèle et sont disponibles en téléchargement à l'adresse suivante :

<https://cloud.irit.fr/index.php/s/uZ0YglcwqZSoflU>.

Le premier groupe d'instances sera le groupe "diversité". Ce nom est dû au fait que dans ces instances, nous cherchons à diversifier les tâches qui seront affectées par un aléa. Toujours au nombre d'une par instance et n'affectant qu'une tâche, chaque perturbation provoquera un report de 1, 10, 20, 25 ou 30 heures d'une des 6 tâches sélectionnées aléatoirement, ce qui donne trente instances différentes.

Le second groupe sera le groupe « intensité ». Contrairement au premier, nous faisons varier plus intensément le nombre d'heures de report mais pour une seule tâche. Ce groupe comprend trente-deux instances, chacune correspondant au report de 1 à 32 heures de la tâche sélectionnée (qui est donc la même pour toutes les instances).

Pour étiqueter les différentes instances, nous utilisons le même schéma : l'étiquette "OTX+Y" signifie que la perturbation se produit sur la tâche X reportée de Y heures.

3.3.2. Etude des modes de résolution du solver

Le premier objectif de l'étude expérimentale était de trouver la meilleure stratégie de recherche de CP pour le problème. CP Optimizer fournit différentes stratégies de recherche comme "Depth-First", "Restart", "Multi-Point" et "Iterative Diving". En se basant sur la documentation de CP Optimizer, les stratégies sont présentées comme suit : "Depth-First" est un algorithme de recherche arborescente tel que chaque fixation (ou instanciation) d'une variable de décision peut être considérée comme une branche dans un arbre de recherche. "Restart" vise à redémarrer de temps en temps la recherche "Depth-First". La recherche "Multi-Point" crée un ensemble de solutions et les combine pour produire de meilleures solutions. La recherche "Multi-point" est plus diversifiée

que la recherche "Depth-First" ou "Restart". Comme elle s'exécute jusqu'à un point où elle considère que la meilleure solution trouvée ne peut pas être améliorée, une limite de recherche est recommandée. Enfin, la recherche "Iterative Diving" est conçue pour les problèmes de planification comportant au moins une variable d'intervalle, et elle utilise une technique de plongée plus agressive pour trouver plus rapidement des solutions aux problèmes de grande taille.

"Iterative Diving" semble être la stratégie de recherche la plus appropriée pour notre problème étant donné que notre modèle repose essentiellement sur les intervalles et s'avère être un problème de planification, et nous allons essayer de vérifier cette hypothèse via des tests expérimentaux.

Ces tests consistent donc à calculer un nouveau planning en prenant en compte un retard de vingt-six heures sur une tâche et dont le critère de la fonction objectif est le makespan. Un test sera lancé pour chaque stratégie avec une limite de temps de calcul de dix minutes, car il correspond à la barrière de temps psychologique acceptée par les managers dans une telle situation de stress.

	<i>Best Value</i>	<i>Depth-First</i>	<i>Restart</i>	<i>Multi-Point</i>	<i>Iterative Diving</i>
<i>Makespan</i>	18801	20436	18801	19855	18801
<i>Temps (s)</i>	15.33	600	15.33	600	18.66

Tableau 3 - Efficacité de chaque stratégie de résolution du solver

Dans le tableau 3, nous pouvons voir que seules les stratégies "Restart" et "Iterative Diving" ont réussi à obtenir la valeur optimale, et ce assez rapidement. "Iterative Diving" semble être la stratégie la plus appropriée car elle fait partie des deux stratégies les plus rapides, et a été présentée comme adaptée aux problèmes avec des variables de type intervalle. Nous retenons donc cette stratégie pour la résolution des instances plus complexes.

3.3.3. Tests de replanification

3.3.3.1. Replanification sans critères en escalier

Pour les tests, nous avons utilisé l'ordre lexicographique pour différencier les différents critères et leur poids dans la fonction objectif. Les critères sont ceux présentés plus tôt, hors critères escaliers : le makespan (1), le retard (2) et la stabilité de la planification (3). Sachant que nous minimisons la fonction lexicographique, pour maximiser le critère de stabilité, nous minimisons son opposé. Les six combinaisons lexicographiques sont les suivantes :

- staticLex(makespan, stabilité, retard) (i) ;
- staticLex(makespan, retard, stabilité) (ii) ;
- staticLex(stabilité, makespan, retard) (iii) ;
- staticLex(stabilité, retard, makespan) (iv) ;
- staticLex(retard, stabilité, makespan) (v) ;
- staticLex(retard, makespan, stabilité) (vi).

La figure 12 montre la répartition du temps de calcul d'une solution exprimé en secondes obtenu pour le groupe "diversité" pour chacune des 6 combinaisons lexicographiques de ces fonctions sous la limite de temps de calcul de dix minutes. Elle montre que seules les fonctions (v) et (vi) ont fourni quelques solutions optimales (cinq pour chacune, représentées par les points) dans un délai de dix minutes. Les autres fonctions objectives ont fourni quelques solutions réalisables, mais jamais une solution optimale.

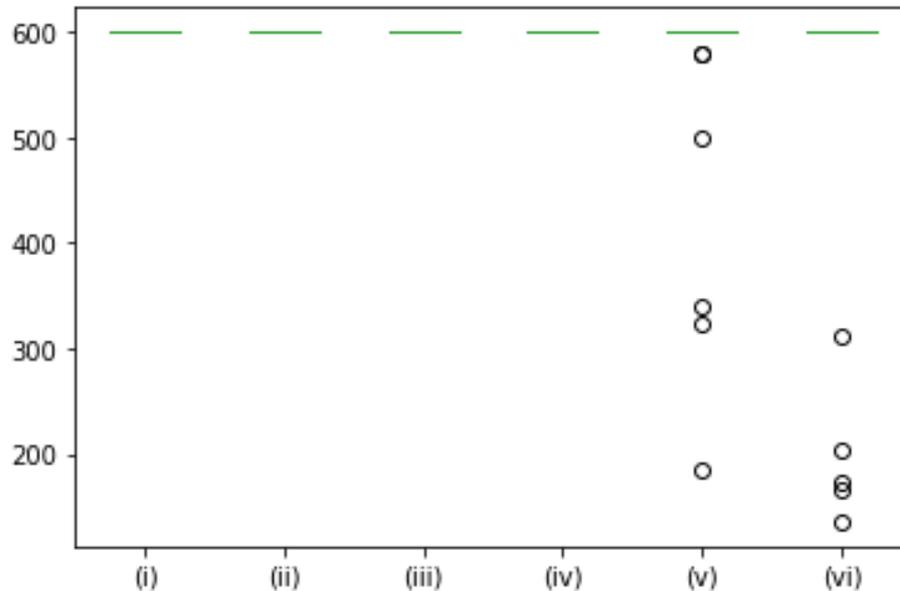


Figure 12 – Répartition des temps de résolution en seconde pour chaque fonction lexicographique.

(i) : MSR; (ii) : MRS; (iii) : SMR; (iv) : SRM; (v) : RSM; (vi) : RMS. (M pour Makespan, S pour Stabilité, R pour Retard)

Lorsque nous regardons le makespan obtenu pour chaque solution et le comparons à la date qui représente la fin de l'IP (que nous ne voulons pas dépasser), le makespan dépasse cette date dans 20 cas pour la fonction (v) et dans 18 cas pour la fonction (vi) contre seulement 8 fois pour les fonctions (i) et (ii). La fin de l'IP est dépassée dans 16 cas pour la fonction (iii) et dans 23 cas pour la fonction (iv). Par définition, les fonctions (i) et (ii) sont les plus efficaces pour trouver un meilleur makespan, elles sont donc les plus pertinentes dans le cas où la solution de replanification peut encore respecter l'IP initialement déterminé.

Le choix du makespan comme premier critère a un impact important sur les autres critères. Nous présentons dans la figure 13 le score de stabilité parmi toutes les instances pour chaque fonction objectif (i.e. le nombre de tâches assignées à un même compagnon) dans le groupe "diversité". Ce critère doit être maximisé et sa meilleure valeur possible est 595 car il représente le nombre total de tâches assignées à un compagnon (cette valeur est calculée en additionnant le nombre de compagnons nécessaires pour toutes les tâches). Les fonctions (i), (v) et (vi) fournissent de bons scores avec une valeur médiane supérieure à 550. En effet, une détérioration de 45 signifie une perte de 7,5%. La fonction (ii) fournit les plus mauvais résultats avec une valeur médiane très basse. En tenant compte de la limite du temps de calcul, on peut supposer qu'après l'optimisation des critères de makespan et de retard, il ne reste plus de temps pour le dernier critère.

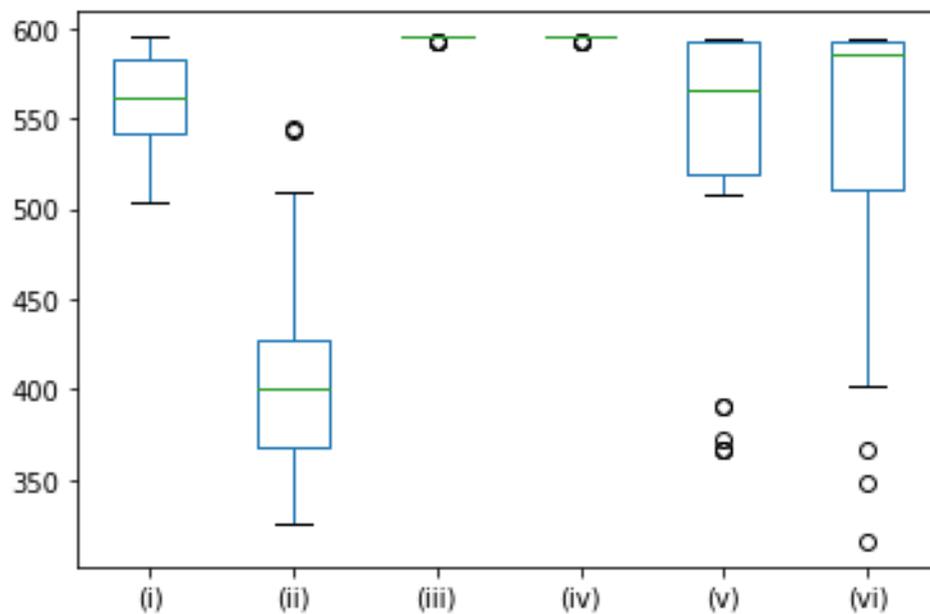


Figure 13 - Valeurs de stabilité fournies par chaque fonction objectif pour le groupe "diversité".

Les résultats obtenus permettent de tirer les conclusions suivantes. La minimisation du makespan et du retard est plus coûteuse en temps, mais ne pas considérer le makespan comme critère prioritaire comporte un risque de perdre une solution respectant l'IP. Afin d'obtenir des solutions de bonne qualité dans un temps de résolution réduit, nous proposons un nouveau critère, le "makespan en escalier" défini par l'équation (5) dont les avantages sont évalués par la deuxième série de tests.

3.3.3.2. Replanification avec makespan en escalier

Afin de comparer le critère makespan avec un des critères proposés dans cette thèse, le makespan escalier (M^*)¹, nous avons lancé les tests avec un seul critère : makespan VS makespan escalier. Dans cette expérience, le domaine pour le makespan en escalier était le suivant : $M^* \in \llbracket 0,8 \rrbracket$, ce qui correspond à un retard maximal possible de huit jours. Les résultats obtenus montrent que pour toutes les instances où l'IP était respecté avec le critère du makespan, il l'était aussi avec la version escalier. Cependant, les plannings sont différents si un jour ouvrable supplémentaire doit être utilisé. Les solutions obtenues avec le critère makespan ont un makespan plus court ce jour-là, ce qui est moins important pour les managers. Le temps moyen de solution pour atteindre l'optimum était de 28,46 secondes pour le critère makespan et de 7,78 secondes pour le critère makespan en escalier. De plus, le temps de solution le plus élevé était de 117,63 secondes pour le critère de makespan et seulement 17,81 pour le critère de makespan en escalier. D'après ces résultats, nous pouvons affirmer que l'utilisation du modèle en escalier consomme moins de temps en général et en particulier lorsque quelques jours de travail supplémentaires sont nécessaires pour terminer toutes les tâches. De ce fait, plus de temps pourra être utilisé pour l'optimisation d'autres critères dans les modèles multicritères.

Pour les tests suivants, nous avons considéré les combinaisons de critères suivantes, où le critère C représente la charge de travail du compagnon supplémentaire :

- staticLex(M^* , S, C) (vii);
- staticLex(M^* , C, S) (viii).

La figure 14 présente la répartition du temps de résolution pour les fonctions (vii) et (viii) sous la limite de dix minutes de calcul. Des solutions réalisables ont été trouvées pour toutes les instances du problème. En comparaison avec la figure 12, nous pouvons voir que beaucoup plus de solutions ont été trouvées avant la limite de temps et donc des solutions optimales ont été fournies, précisément, pour 15(/32) instances du groupe "intensité" pour la fonction (vii) et 12(/30) instances pour la fonction (viii).

¹ s. t. makespan $\leq T + M^* * WH$

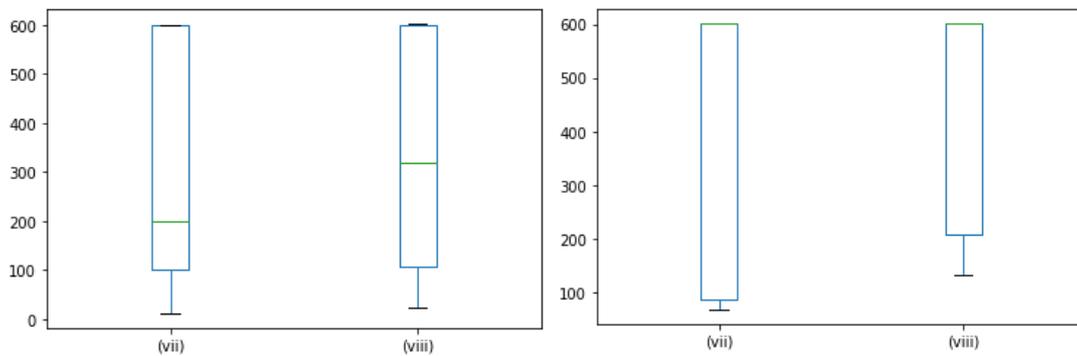


Figure 14 - Répartition du temps en utilisant le makespan en escalier comme premier critère pour le groupe "diversité" (à gauche) et le groupe "intensité" (à droite).

La figure 15 montre l'évolution du makespan pour les instances pour lesquelles le temps de solution a atteint le délai limite. Nous analysons les instances du groupe "intensité" pour identifier le point de rupture (c'est-à-dire le délai à partir duquel il était impossible de respecter l'IP). Les résultats montrent que lorsqu'une perturbation est supérieure à 26 heures, il n'est plus possible de respecter l'IP et donc un jour supplémentaire est nécessaire pour compléter le planning.

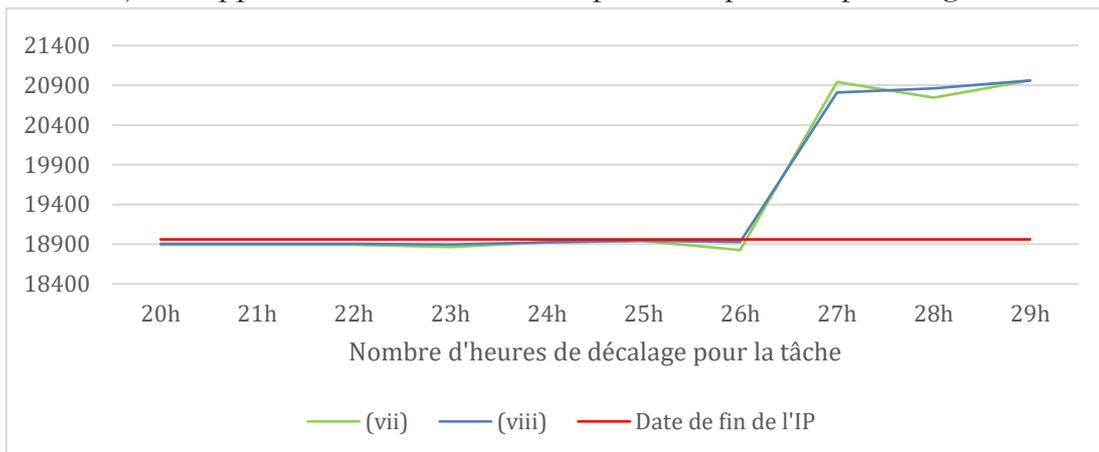


Figure 15 - Evolution du Makespan pour le groupe "intensité".

La figure 16 montre les valeurs du critère de stabilité pour les instances du groupe "intensité". On peut voir que la valeur de stabilité a commencé à se détériorer à partir d'une perturbation de 12 heures. La meilleure valeur de stabilité possible est 595 mais n'est pas nécessairement atteignable tout en étant sous la valeur de l'IP. La pire valeur est atteinte pour un retard de 26 heures.

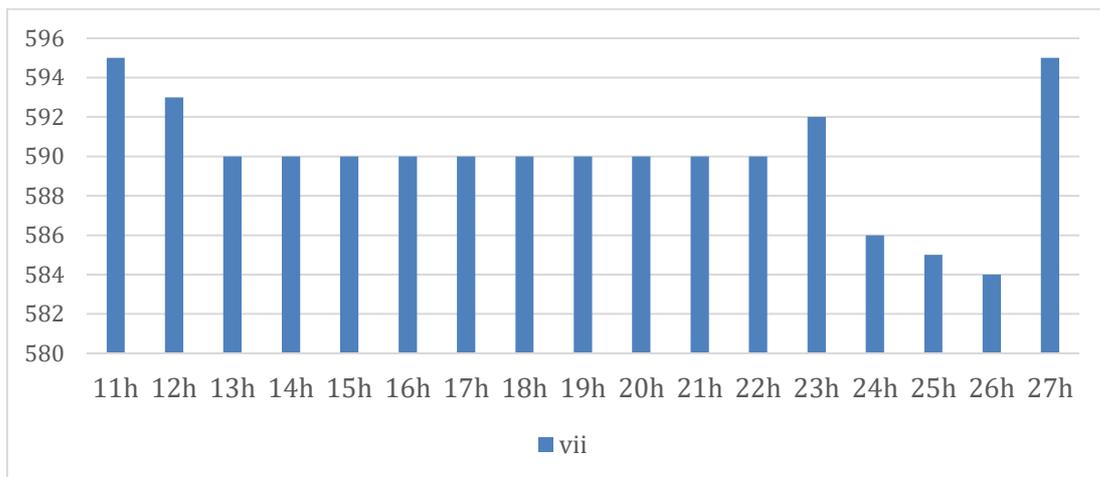


Figure 16 - Valeurs de stabilité obtenues avec (vii) pour le groupe "intensité".

La figure 17 quant à elle compare les pires valeurs du critère de stabilité obtenues avec (vii)² et (viii)³. Nous observons que dans tous les cas, la fonction (vii) obtient au minimum un score de 583/595 (98%), ce qui indique le nombre d'affectations qui demeurent identiques à la solution initiale de planification avoisine les 98% pour chaque compagnon. Le maintien du planning est donc respecté et la solution très satisfaisante même dans les pires cas. De plus, considérant qu'il y a un compagnon additionnel, celui-ci n'avait pas de tâches affectées dans le planning initial, donc une partie des tâches qui ne sont plus affectées au même compagnon correspondent en fait à un délestage des compagnons déjà présents dans le premier planning. Le planning individuel est donc beaucoup moins perturbé lorsque nous utilisons la fonction (vii), faisant d'elle la meilleure fonction. Il faut cependant faire attention à la quantité de travail du compagnon supplémentaire, et s'assurer que celle-ci soit cohérente avec le reste de la solution (i.e. un compagnon qui ne doit pas revenir 3 fois pour 20min de travail par exemple).

² staticLex(M*, S, C)

³ staticLex(M*, C, S)

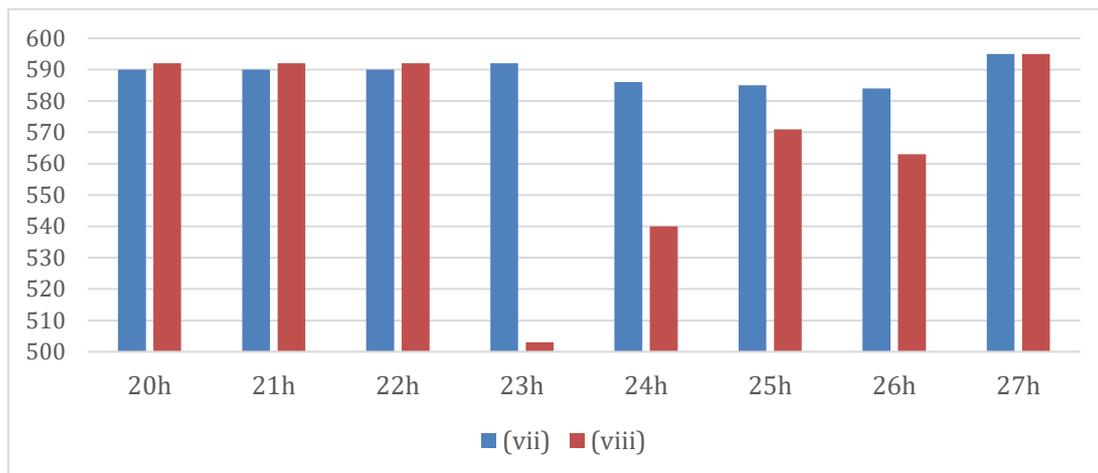


Figure 17 - Pires cas des valeurs de stabilité obtenues avec (vii) et (viii) pour le groupe "intensité".

3.3.3.3. Replanification avec makespan et charge escalier

Sur la base de la même idée que nous utilisons pour le makespan en escalier, nous formulons une version en escalier pour les heures de travail du compagnon supplémentaire. Cette formulation vise à améliorer la solution lorsque la stabilité est le critère le moins important dans l'ordre lexicographique car nous cherchons à gagner du temps de calcul en réduisant les heures de travail à un certain nombre de jours. Dans nos tests, nous utilisons le domaine suivant pour la charge d'escalier : $O^* \in \llbracket 0,16 \rrbracket$, 16 représentant le nombre d'heures de mobilisation maximale pour le compagnon supplémentaire (i.e. deux jours de travail). La nouvelle fonction objectif où le mode escalier est utilisé à la fois pour le makespan et pour la charge de travail du compagnon supplémentaire (représentée par " O^* ") est la suivante :

$$\text{staticLex}(M^*, C^*, S) \quad (\text{ix})$$

Nous supposons qu'en raison de la diminution du temps consacré à l'optimisation de la charge, davantage de temps de calcul peut être utilisé pour optimiser le dernier critère de la liste des priorités : la stabilité de la planification. La figure 18 compare les valeurs de ce critère de stabilité obtenues avec les fonctions (ix) et (viii)⁴. Ici (vii)⁵ est utilisé comme référence de la meilleure valeur possible. On peut voir que de meilleures valeurs ont été obtenues avec (ix) sauf pour la perturbation de 13 heures et en particulier les valeurs ont été substantiellement améliorées pour les pires cas (23h à 26h). La raison à cette

⁴ staticLex(M^* , C, S)

⁵ staticLex(M^* , S, C)

amélioration résulte du fait qu'on a probablement affecté plus de tâches au compagnon supplémentaire que lors des résolutions précédentes étant donné qu'on s'autorisait un pallier de 8h de travail. De ce fait, cela donnait plus d'espace aux autres compagnons pour récupérer les tâches qui leur étaient affectées auparavant (dans le cas où on cherche à minimiser au maximum les heures de travail avec (vii) et (viii), cela forçait à compresser l'ensemble des tâches sans possibilité de respecter les anciennes affectations). Donc en plus de donner plus de liberté, cela a également permis d'améliorer le 3^e critère.



Figure 18 - Valeurs du critère de stabilité obtenues avec les fonctions (ix), (vii) et (viii) pour le groupe "intensité".

La figure 19 montre la comparaison pour certaines instances du groupe diversité, précisément les sous-groupes "OT69", "OT115" et "OT244". Pour les deux premiers sous-groupes, (ix) semble suivre la même évolution que (viii) tout en ayant de meilleures valeurs, notamment pour le sous-groupe "OT115". Cependant, pour deux instances du sous-groupe "OT244" la fonction (viii) a montré des scores de stabilité bien meilleurs (lorsque les décalages étaient de 10h et 20h) là où l'écart était totalement en faveur de la fonction (ix) pour les décalages de 1h et 25h. L'instance de 30h, quant à elle, n'aide pas à départager étant donné que les deux fonctions donnent le même score. En conclusion, l'efficacité de (ix) pour le critère de stabilité semble globalement meilleur, mais le fait que nous ayons trouvé aussi des cas où il est clairement en dessous sur un si petit lot d'instances laisse penser que les cas particuliers peuvent survenir assez régulièrement. Prévoir une fonction alternative pour ces cas de figure peut donc s'avérer intéressant pour éviter de tomber sur une solution peu satisfaisante.

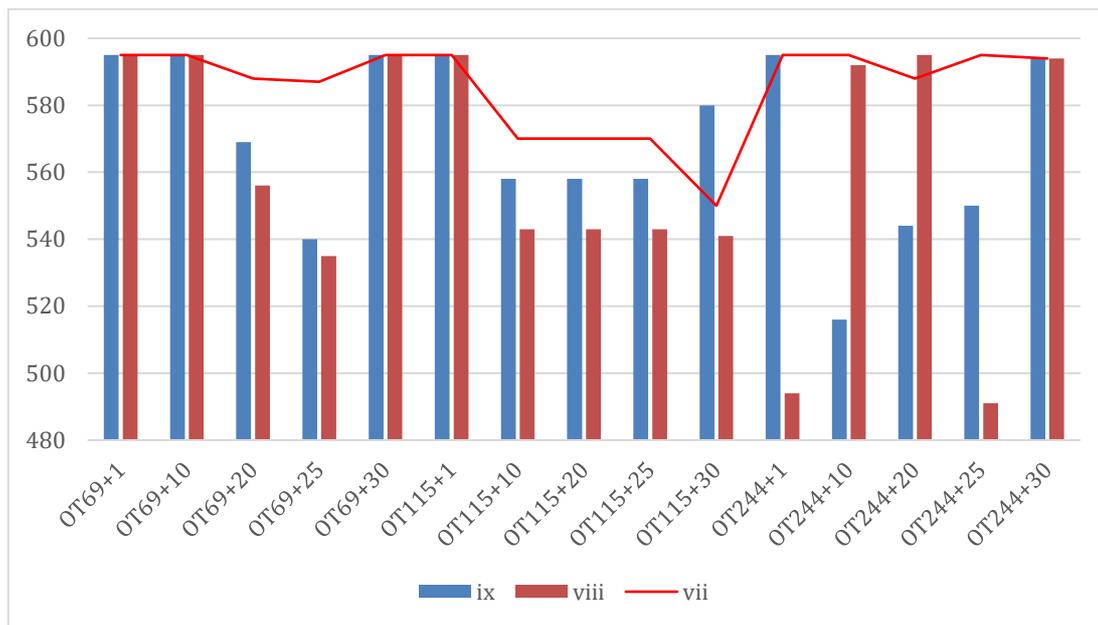


Figure 19 - Valeurs de stabilité obtenues avec les fonctions (ix), (vii) et (viii) pour quelques instances du groupe "diversité".

3.3.3.4. Conclusion sur les nouveaux critères

L'utilisation du makespan en escalier comme premier critère objectif correspond bien aux objectifs des gestionnaires : avec ce critère, ils peuvent à la fois minimiser le makespan si l'IP peut être respecté et donner plus de liberté et de temps de calcul pour d'autres critères comme la stabilité de la planification et la charge du compagnon supplémentaire. L'ordre lexicographique de ces deux derniers critères peut quant à lui être choisi par les gestionnaires en fonction de leurs priorités, mais aussi en fonction de la première solution de replanification trouvée. Par exemple, si la priorité est de maintenir le planning, mettre la stabilité en deuxième critère permet de limiter les risques de tomber sur une solution qui présente beaucoup de changements d'affectation. En revanche, si cette stabilité a moins d'importance, et que l'entreprise veut minimiser la mobilisation de ressources supplémentaires, mettre la charge de travail en deuxième peut être une solution. Et même en faisant ainsi, il est possible de mettre soit le critère de manière stricte, soit sa version en escalier pour laisser de l'espace pour le critère de stabilité.

Ce système de critère en escalier donne donc plus de flexibilité aux managers pour recouvrer d'un aléa rapidement, et leur permet de relancer la chaîne en attendant de trouver une solution sur un plus long terme en calculant par exemple un autre planning en mobilisant plus de ressources de calcul et de temps pour déterminer une solution répondant à des attentes plus exigeantes (en termes de makespan par exemple).

3.4. APPROFONDISSEMENT SUR DES INSTANCES DIVERSIFIÉES

Afin de confirmer la conclusion tirée de l'étude sur les grandes instances industrielles, nous avons conçu des ensembles de données supplémentaires et plus petits dont les données sont disponibles en accès libre (<https://cloud.irit.fr/index.php/s/uZ0YglcwqZSoflU>) pour les comparer avec d'autres méthodes.

Trois fonctions objectif sont comparées dans cette expérience. Nous reprenons les fonctions (viii) et (ix) de la section 3.3. (x) est une fonction utilisée pour confirmer les conclusions sur l'utilisation du makespan par rapport au makespan en escalier. M représente le makespan, C la charge de travail du compagnon supplémentaire et S la stabilité par rapport à l'ancien planning (i.e. maintien des affectations).

$$\text{staticLex}(M^*, C, S) \quad (\text{viii})$$

$$\text{staticLex}(M^*, C^*, S) \quad (\text{ix})$$

$$\text{staticLex}(M, C, S) \quad (\text{x})$$

3.4.1. Définition des instances

Deux ensembles de données de nouvelles instances de problèmes ont été créés sur la base d'une structure réelle de contraintes de précédence et de durées de tâches : DS1 et DS2 (DS pour « Data Set »). Ensuite, cinq tâches différentes ont été sélectionnées pour chaque ensemble de données, et pour chacune d'elles, vingt instances de replanification ont été créées. On obtient ainsi deux cents instances de replanification (chaque instance ne perturbe qu'une seule tâche qui est retardée d'un certain temps).

Les instances du problème dans DS1 ont 192 tâches. Il y a 236 zones par avion. Il y a un total de 10 compétences et huit de ces compétences ne sont maîtrisées que par un seul compagnon. Les deux autres compétences sont maîtrisées par deux compagnons, ce qui donne un total de 12 compagnons. Nous ne considérons pas les contraintes de disponibilité des ressources matérielles. Il existe 360 relations de précédence : 251 d'entre elles sont des relations fin début (ES), et 98 des relations fin début exactes (ESE). Enfin, les compagnons ne travaillent que pendant la journée.

Les instances du problème dans DS2 ont 265 tâches. Il y a 409 zones par avion. Il y a un total de 2 compétences avec six compagnons pour l'une (S1), et quatre pour l'autre (S2), ce qui donne 10 compagnons au total. Il y a une unité de

3 ressources matérielles différentes. Il y a 518 relations de précédence : 320 d'entre elles sont des relations ES et 178 des relations ESE. Les autres sont uniquement SS ou SSE. Cinq des compagnons maîtrisant S1 travaillent de jour ou de nuit (ou les deux) et 2 des quatre maîtrisant S2 travaillent de jour ou de nuit (ou les deux). Tous les autres compagnons travaillent uniquement de jour. La figure 20 montre les horaires initiaux de toutes les instances de DS1 et DS2, respectivement. Les tâches qui sont sélectionnées pour être perturbées sont indiquées par une couleur différente. Pour chaque tâche perturbée, le retard est compris entre 3 et 22 heures.

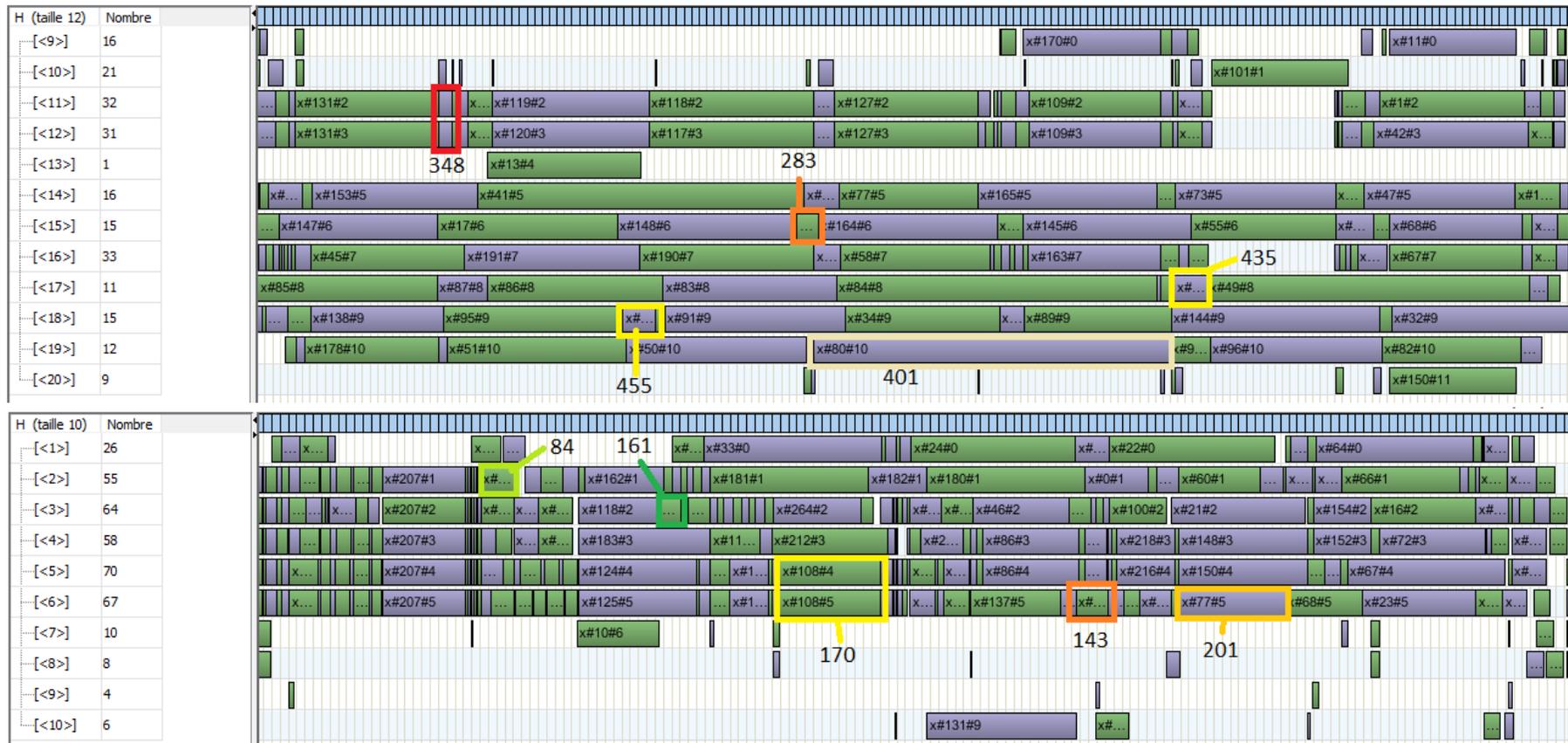


Figure 20 - Diagrammes de Gantt pour les planifications issues de DS1 (en haut) et DS2 (en bas). Les tâches encadrées dans chaque diagramme représentent les tâches qui seront retardées d’un certain nombre d’heures et serviront à générer les instances de replanification.

3.4.2. Résultats pour DS1

Lors de la résolution des instances de replanification, notre modèle s'est avéré très efficace. Sur les 100 instances du groupe DS1, (viii) a atteint l'optimalité pour 80 instances en 1 minute et pour 82 instances en 10 minutes. (ix) a atteint l'optimalité pour 80 instances en 1 minute et pour 84 instances en 10 minutes. (x) a atteint l'optimalité pour 78 instances en 1 minute et 81 instances en 10 minutes.

Cependant, parmi celles-ci, 38 solutions optimales obtenues avec (x) nécessitaient un compagnon supplémentaire alors que cela n'était pas nécessaire avec deux autres fonctions.

En ce qui concerne les valeurs du makespan et de la stabilité, elles étaient comparables partout, avec des valeurs de stabilité légèrement meilleures obtenues avec (ix). Des différences sont observées sur la charge de travail du compagnon supplémentaire, où la fonction (viii) a utilisé moins de temps du compagnon supplémentaire.

Nous avons ensuite étendu la limite de temps à 20 minutes et résolu les instances sans optimalité après 10 minutes à nouveau. Ce temps de calcul supplémentaire n'a pas amélioré de manière significative la solution déjà trouvée sous la limite de 10 minutes.

Nous avons considéré ici des cas faciles car la tension entre l'utilisation des ressources est faible puisque chaque compagnon maîtrise une compétence unique. Les résultats montrent que sur ce type de cas, aucune valeur ajoutée réelle n'est apportée par l'utilisation du makespan en escalier.

3.4.3. Résultats pour DS2

Les instances de DS2 semblent plus difficiles à résoudre que celles de DS1. Sur 100 instances, (viii) a atteint l'optimalité pour 16 instances en 1 minute et 57 en 10 minutes, avec un temps moyen 350 secondes pour arriver à l'optimalité. (ix) a atteint l'optimalité pour 17 instances en 1 minute et 51 en 10 minutes, avec un temps moyen 376 secondes pour arriver à l'optimalité. (x) a atteint l'optimalité pour 3 instances en 1 minute et 15 en 10 minutes, avec un temps moyen de 554 secondes pour arriver à l'optimalité.

Pour ces instances, l'utilisation du makespan en escalier a apporté une réelle différence dans le temps de résolution, confirmant que son utilisation aide à réduire le temps de résolution.

Les instances les plus faciles sont celles générées par l'interruption de la tâche 201, qui est celle dont l'heure de début est la plus tardive dans le planning

initial. En raison de son heure de début tardive, l'objectif du makespan en escalier peut facilement identifier qu'un jour de travail supplémentaire est obligatoire. Dans le même temps, les instances qui n'ont pas été résolues de manière optimale concernent les tâches situées au milieu du programme.

Toutes les solutions obtenues étaient comparables en termes de stabilité. Cependant, de grandes différences ont été observées dans la valeur de la charge supplémentaire des compagnons, (viii) fournissant la charge la plus faible et (x) utilisant le plus de compagnons supplémentaires. Par exemple, pour les instances de la "tâche moyenne" (perturbations des tâches 143, 161, 170), en utilisant (x), la limite de temps était toujours atteinte et la charge de travail du compagnon supplémentaire était très mauvaise (de quatre heures à plus de vingt-deux heures de travail). Le temps de calcul et la charge de travail ont été considérablement améliorés en utilisant la fonction objectif (viii) au lieu de (x). Certaines instances sont passées de vingt-deux heures de travail supplémentaire à seulement quatre heures, certaines sont même tombées à zéro. Mais alors que ces valeurs sont assez stables parmi toutes les instances en utilisant (viii), il en est différemment en utilisant la fonction (ix). Les temps de solution pour (ix) étaient équivalents ou pires que pour (viii). La charge de travail du compagnon supplémentaire a également augmenté en raison du calcul en mode escalier. Ces résultats suggèrent que pour les cas difficiles en pratique, il est préférable d'inclure la charge de travail sans mode escalier dans la fonction objectif.

Avec la limite de temps de calcul de vingt minutes, pratiquement aucune amélioration n'a été observée, ni en termes de nombre d'instances résolues de manière optimale, ni en termes de valeurs de la fonction objectif pour les solutions non optimales. La barrière psychologique de 10 minutes a été une limite de temps pertinente pour les instances testées du point de vue de l'optimisation également.

3.5. CONCLUSIONS

Nous avons développé un nouveau modèle de programmation par contraintes pour résoudre un problème de replanification de manière réactive et proposé des critères d'optimisation nouveaux et pertinents. Ces critères utilisent un modèle en escalier, qui correspond parfaitement aux objectifs des gestionnaires qui gèrent le processus de replanification. Afin de prendre en compte les différents objectifs que les managers ont en pratique, nous avons proposé plusieurs combinaisons de critères ordonnés lexicographiquement et testé leurs performances sur de grandes instances de problèmes industriels. Pour confirmer les phénomènes observés, un jeu de données supplémentaire de 200 instances plus petites a été créé et testé.

Les expériences menées ont apporté plusieurs informations utiles. Premièrement, la complexité du problème de replanification dépend du moment où la perturbation se produit. Il est possible de distinguer les cas de replanification difficiles des cas faciles à résoudre en examinant l'emplacement de la tâche perturbée dans le calendrier d'assemblage. Si la perturbation se produit au début du programme ou vers la fin du temps de cycle, les cas semblent être faciles à résoudre : dans le premier cas, il est encore possible de respecter le temps de cycle en raison de la répartition initiale du temps et dans le second cas, il n'y a aucune chance de respecter le temps de cycle sans qu'un jour de travail supplémentaire soit nécessaire. Lorsque la perturbation se produit au milieu du planning, il n'est pas toujours facile pour le solveur de déterminer rapidement si un jour supplémentaire est nécessaire ou non.

Les résultats montrent qu'il n'y a pas de réelle différence dans la performance des critères objectifs testés lorsque l'instance est facile à résoudre. Si le problème de replanification est difficile, l'utilisation du makespan en escalier permet de réduire le temps de résolution et de consacrer plus de temps à l'optimisation d'autres critères inclus dans l'ordre lexicographique tels que la stabilité de la planification dont les valeurs peuvent être améliorées. L'utilisation d'un modèle d'escalier pour la charge de travail du compagnon supplémentaire peut être plus sensible à la difficulté de l'instance : pour une instance facile, le modèle d'escalier peut aboutir à de meilleures solutions ou à l'obtention de bonnes solutions plus rapidement. Toutefois il peut détériorer la solution pour une instance difficile. Enfin, le compagnon supplémentaire a un impact plus important lors de l'intégration d'un petit groupe de compagnons ayant la même compétence que pour un grand groupe.

D'un point de vue managérial, les résultats de cette étude aident les managers à comprendre l'importance et l'impact de chaque critère sur le temps et la qualité de la solution. En particulier, nous montrons que l'utilisation de l'approche CP peut être très efficace pour les problèmes de replanification dans le contexte considéré de lignes d'assemblage cadencées dans l'industrie aéronautique.

La suite de notre travail a pour objectif de proposer une approche permettant de réduire la charge ergonomique des compagnons et de distribuer le travail de manière plus équitable. Pour ce faire, je présente dans la section 4 les travaux de Lovato et al. (2022b).

4.

**INTEGRATION DE CRITERES
ERGONOMIQUES DANS LA
PLANIFICATION DES
LIGNES D'ASSEMBLAGE
AERONAUTIQUES**

4.1. MOTIVATIONS

Dans ce chapitre nous nous intéressons à l'aspect humain de la chaîne et aux compagnons présents sur place. Le travail dans la ligne d'assemblage est majoritairement manuel et certaines tâches pénibles peuvent avoir un impact sur la santé des compagnons, par exemple, celles qui nécessitent une position inconfortable prolongée comme le montre la figure 21.



Figure 21 - Deux compagnons travaillant sur une ligne d'assemblage aéronautique

Plusieurs méthodes ergonomiques peuvent être utilisées pour évaluer la pénibilité et la contrainte des tâches, par exemple l'indice Rapid Upper Limb Assessment (RULA), qui évalue la façon dont le haut du corps est affecté par les tâches. Cet indice prend en compte différents paramètres ergonomiques (cf. figure 22) afin d'évaluer le degré d'inconfort de la position de travail et permet de noter le risque de développer des troubles musculo-squelettiques sur le long terme.

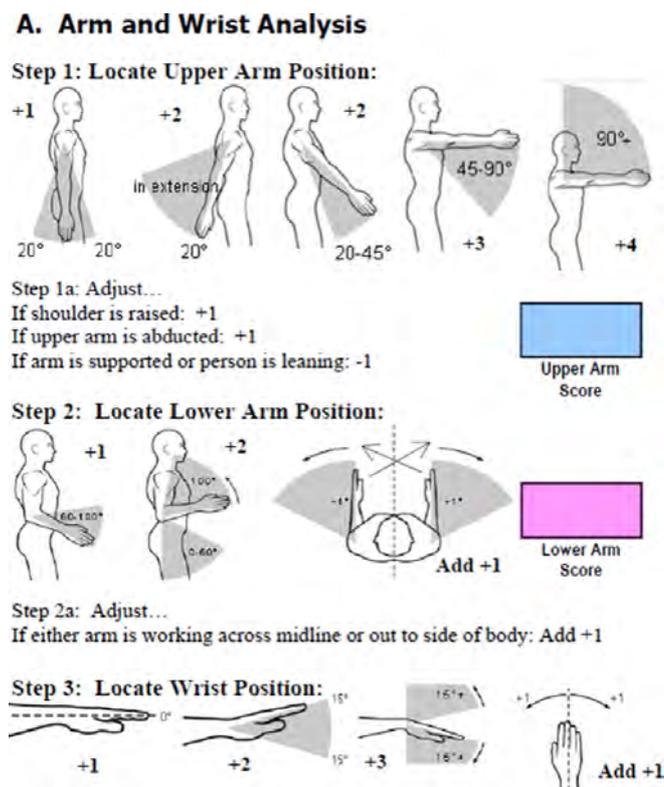


Figure 22 - Exemple de quelques étapes pour évaluer la dangerosité d'une tâche

Source: https://www.researchgate.net/figure/RULA-employee-assessment-worksheet-16_fig1_326232801

En pratique, une telle évaluation est généralement effectuée lorsque les tâches sont déjà assignées aux stations d'assemblage et aux compagnons. Une telle évaluation a posteriori ne peut pas empêcher une distribution injuste des tâches où certains compagnons doivent gérer un plus grand nombre de tâches difficiles que n'importe lequel de leurs collègues.

Afin de fournir aux managers des outils de planification efficaces qui prennent en compte les impacts ergonomiques de l'attribution des tâches, nous améliorons notre modèle mathématique avec un nouveau critère de décision.

Notre objectif est donc de fournir aux managers une procédure capable d'évaluer l'équité dans la répartition des tâches entre les compagnons et de proposer la solution la plus équitable. Une meilleure équité réduit les risques d'accidents et favorise l'unicité du groupe puisqu'aucun compagnon n'enviera le planning plus facile d'un autre. Cet objectif est rempli grâce au modèle dont nous présentons les modifications dans la section suivante.

4.2. CRITERES D'EVALUATION DE LA CHARGE ERGONOMIQUE

4.2.1. La charge ergonomique

Dans l'analyse de la littérature, nous avons vu que l'aspect ergonomique est abordé en évaluant les tâches via une ou plusieurs méthodes utilisant un indice comme RULA ou OCRA par exemple. Cependant, cette méthode vise seulement à évaluer la difficulté ou l'impact d'une position sur la personne réalisant ses tâches, et aucune notion de temps passé dans cette position n'est mis en avant. Estimant qu'il est nécessaire de faire la différence entre une tâche courte et une tâche longue, indépendamment de la position, nous décidons également de prendre en compte la durée de la tâche pour calculer ce qu'on appellera le « score de pénibilité ». Pour une tâche $i \in I$ donnée, le score de pénibilité d_i est calculé à partir du score obtenu via la méthode RULA r_i et de la durée de la tâche pt_i :

$$d_i = pt_i * \left(\frac{r_i}{100}\right)^2$$

L'indice de RULA est divisé par 100 pour le normaliser sur l'unité utilisée dans le planning (ici le centième d'heure). Ainsi, avec la variable de décision $x_{i,o}$ qui vaut 1 lorsque la tâche i est attribuée au compagnon o et 0 sinon, la charge ergonomique endurée par un compagnon $o \in O$ est calculée telle que :

$$e_o = \sum_{i \in I} x_{i,o} * d_i$$

Cela conduit au premier critère de décision : la somme des charges ergonomiques maximales parmi tous les compagnons de chaque corps de métier.

4.2.2. Charge ergonomique maximale

Nous avons défini la charge ergonomique d'un compagnon comme étant une somme de scores de pénibilité des tâches qui lui sont affectées. La minimisation de la plus importante charge parmi les compagnons d'un même corps de métier est un premier critère potentiel d'optimisation pour arriver à une situation d'équité :

$$SUM_Max = \sum_{p \in P} \max_{h \in O_p} e_h$$

« *SUM_Max* » représente la somme des plus importantes charges ergonomiques de chaque corps de métier différent. L'objectif sera donc de

minimiser cette somme pour forcer le modèle à redistribuer une potentielle chaîne de tâches difficiles originellement attribuées au même compagnon. Cependant, ce type de critère produit un effet non désiré : l’effet de noyade. Cet effet signifie que, si on cherche à minimiser seulement la valeur la plus importante, alors toutes les autres ne seront pas prises en compte. Dans un contexte d’équité, où l’équilibrage est de mise, cela veut dire qu’ hormis pour le compagnon à la plus importante charge, les autres ne verront pas leur situation s’améliorer, alors qu’il serait possible de mieux rééquilibrer le travail pénible affecté aux autres compagnons. Par conséquent, en se concentrant sur la réduction de la charge ergonomique maximale, ce critère omet l’équilibrage des charges des autres compagnons. C’est pour palier à ce type de problème que nous introduisons notre deuxième critère ergonomique.

4.2.3. Indice Moyen de Déséquilibre

Le problème de noyade peut rapidement aboutir à de forts déséquilibres entre les compagnons n’ayant pas la plus importante charge ergonomique. Le second critère, que nous appelons « Indice Moyen de Déséquilibre » (IMD), a donc pour objectif de minimiser les écarts entre chaque paire de compagnon d’un même corps de métier et de ne pas se cantonner à la minimisation du pire cas. Nous avons eu l’idée de ce critère en étudiant l’indice de Gini, un indice souvent employé en économie pour analyser les différences de revenu (par exemple) entre les individus d’une nation ou d’un groupe. La différence avec Gini réside dans le fait que l’on ne compare pas des groupes mais uniquement des individus, mais aussi que la somme des difficultés des tâches est constante, et que par conséquent le dénominateur ne change pas, contrairement à l’indice.

L’IMD utilisera ainsi les charges ergonomiques de chaque compagnon comme donnée pour équilibrer les différences entre chacun. Son dénominateur a également été modifié pour que la valeur de l’IMD se situe entre 0 et 1. 0 représente l’équité totale (i.e. chaque compagnon a une charge ergonomique parfaitement identique) et 1 le déséquilibre total (un seul compagnon endosse la totalité de la charge). Considérant $|O|$ le nombre total de compagnons, l’IMD s’exprime :

$$IMD = \frac{\sum_{p \in P} \sum_{o, o' \in O_p} |e_o - e_{o'}|}{2 * (|O| - 1) * \sum_{i \in I} d_i}$$

avec $\sum_{i \in I} d_i$ la constante dont nous parlions. Lorsqu’utilisé comme critère d’optimisation, nous cherchons également à minimiser l’IMD pour l’avoir la valeur la plus proche de 0. Lorsque nous atteignons 0, cela correspond à une équité parfaite et chaque compagnon qui travaille doit endosser exactement la même

charge ergonomique. Cette valeur correspond également à la meilleure valeur atteignable pour « *SUM_Max* ».

4.2.4. Exemples applicatifs de ces critères

Pour mettre en avant la différence des deux critères, nous prenons un exemple applicatif. Considérons une instance de dix tâches numérotées de 101 à 110. Les difficultés des tâches varient en fonction de l'instance et une seule contrainte de précédence existe entre 104 et 109. Les tâches présentent les mêmes caractéristiques que dans une instance réelle (TT, nombre de compagnons, ressource matérielle nécessaire, durée et zones utilisées ou exclues) et nous estimons des scores de difficulté pour chaque tâche. Les tâches 101 à 104 sont considérées comme « faciles », 105 à 109 comme « moyennes » et la tâche 110 comme « difficile ». 3 compagnons et une ressource matérielle sont disponibles dans l'instance. La plupart de ces informations sont présentées dans le tableau 4. Le score de difficulté, qui est numérique, sera précisé avant chaque exemple.

<i>Tâche</i>	<i>Durée</i>	<i>Compagnons</i>	<i>R. Matérielle</i>	<i>Score de pénibilité</i>
101	3	1	0	F
102	6	1	0	F
103	6	1	1	F
104	9	1	0	F
105	7	1	0	M
106	6	1	0	M
107	8	1	1	M
108	4	1	0	M
109	14	1	0	M
110	10	1	0	D

Tableau 4 - Caractéristiques de chaque tâche

Le diagramme de Gantt qui représente la solution en ne considérant que le makespan comme critère de décision est présenté dans la figure 23. Dans un contexte ne considérant pas l'ergonomie, nous aurions quasiment toutes les tâches faciles (vert) regroupées sur un seul compagnon (C1) là où le compagnon C2 doit assumer des tâches moyennes (orange) et difficiles (rouge). Les couleurs ne représentent que le score de pénibilité attribué. Nous allons donc observer comment nos critères affectent la solution.



Figure 23 - Ordonnancement en ne considérant que le makespan

Prenons un premier exemple en posant « $F = 1$ », « $M = 5$ » et « $D = 10$ » pour les scores de pénibilité des tâches. L’ordonnancement de la figure 24 a été généré en utilisant une fonction lexicographique cherchant à minimiser le makespan en priorité et « *SUM_Max* » en deuxième.

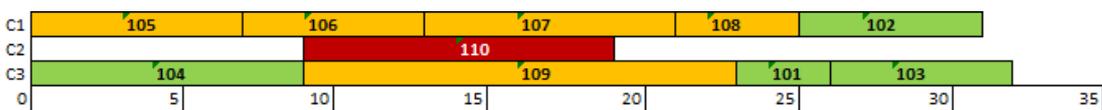


Figure 24 - Ordonnancement en ne réduisant que la charge ergonomique maximale

Le premier élément qui saute aux yeux est l’isolement de la tâche 110 sur un seul compagnon. Cette tâche est celle ayant le plus haut score de pénibilité en plus d’une durée qui semble un peu au-dessus de la moyenne. Les autres tâches sont donc réparties sur les deux autres compagnons sans se soucier d’équilibrer les charges entre eux. Cela se voit par la présence plus importante de tâches orange chez le compagnon C1 et la présence plus importante de tâches vertes chez le compagnon C3. Ceci est l’effet de noyade dont nous avons parlé plus tôt. Nous voulons donc voir si l’IMD permet de supprimer cet effet.

Si nous reprenons l’exemple illustratif vu plus tôt, et en remplaçant « *SUM_Max* » par l’IMD, nous observons un rééquilibrage entre les compagnons C1 et C3 (cf. figure 25).

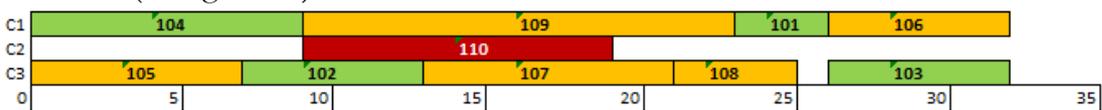


Figure 25 - Ordonnancement en utilisant l’IMD pour équilibrer les charges ergonomiques

Les tâches 101 et 104 ont été réaffectées à C1 tandis que 102 a été réaffectée à C3 pour garder un bon équilibre. L’utilisation de l’IMD a donc permis d’obtenir une meilleure répartition entre ces deux compagnons malgré le compagnon C2 qui possède toujours la charge maximale.

Nous allons maintenant changer les scores de pénibilité. Là où nous avons des scores très éloignés les uns des autres (1,5,10), nous utilisons un ensemble différent, avec des scores proches les uns des autres : « $F = 1$ », « $M = 2$ » et « $D = 3$ ». Dans la figure 26 nous montrons l’ordonnancement obtenu en utilisant « *SUM_Max* », mais qui est aussi celui obtenu en utilisant « *IMD* », toujours dans la même configuration que pour l’exemple précédent.

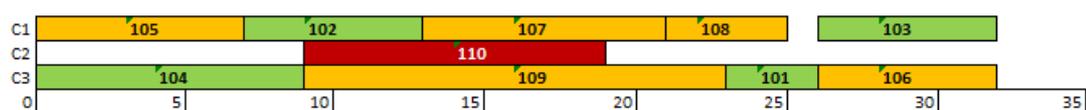


Figure 26 - Ordonnancement lorsque les scores de tâches se rapprochent les uns des autres

Comme nous l'avons laissé entendre en présentant la figure, cet ordonnancement est le même indépendamment du critère ergonomique utilisé pour déterminer la solution. Cela met déjà en avant que, selon le « profil » de pénibilité des tâches de l'instance, le choix du critère utilisé aura une importance dans la définition de la solution.

Pour le dernier test, nous voulons conserver le même écart entre chaque score de pénibilité, mais augmenter celui-ci pour avoir « F = 8 », « M = 9 » et « D = 10 ». Les ordonnancements de chaque instance (en utilisant « *SUM_Max* » en haut et « *IMD* » en bas) sont présentés dans la figure 27.

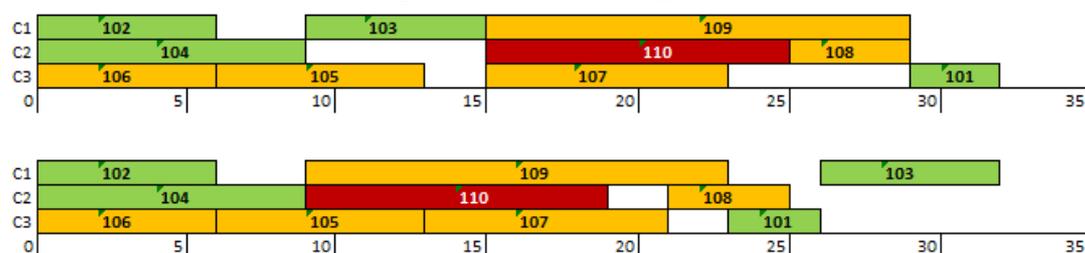


Figure 27 - Ordonnements en utilisant des scores de pénibilités élevés pour toutes les tâches mais en conservant les écarts entre elles.

Malgré l'ordonnancement différent des deux solutions, elles restent équivalentes sur un point de vue ergonomique en termes de score. La répartition des tâches étant la même, alors nous avons forcément la même solution, mais la différence faite sur le compagnon C1, par exemple, aura un impact différent en alternant entre tâche facile et moyenne en réalité. Nous avons également un plus grand nombre de tâches sur le compagnon C2. Cela est dû au fait qu'en augmentant les scores ergonomiques, l'écart de 1 unité comparé aux scores min/max a eu moins d'impact, et que donc la tâche 110 était moins pénalisante dans l'environnement de pénibilité « 8/9/10 ».

Ces exemples ont permis de voir qu'en plus du critère, la solution dépend aussi du profil de difficulté des tâches. Nous avons essayé de considérer une répartition des difficultés cohérentes par rapport à notre cas industriel, et nous allons maintenant faire des tests sur une instance issue d'un cas réel pour tenter de déterminer le critère le plus adapté.

4.3. TESTS EXPERIMENTAUX

Tous les tests ont été effectués sous l'environnement IBM ILOG CPLEX Optimization Studio version 12.10 avec la bibliothèque CP Optimizer (CPO). Le processeur utilisé était un Intel(R) Core (TM) i7-8665U CPU @ 1.90GHz en plus de 16Go de mémoire.

Les tests seront réalisés sur des instances de planification et non pas de replanification. Nous pensons en effet que le critère de maintien des affectations viendra en conflit avec les critères ergonomiques, et nous voulons donc concentrer les tests sur l’ergonomie. Les objectifs des tests sont donc de répondre aux questions suivantes :

- o Comment ces deux critères peuvent influencer le planning ?
- o Comment les incorporer dans le calcul d’un planning ?
- o Quel critère est le mieux adapté et suivant quels paramètres ?

4.3.1. Instance utilisée

Nous considérons une instance de test avec un seul avion en cours d'assemblage, 10 compagnons (6 maîtrisant la profession P1, 4 maîtrisant la profession P2) et 200 zones différentes définies sur l'ensemble de l'avion. Il y a 265 tâches et 3 ressources matérielles disjonctives différentes R1, R2 et R3 ainsi que 518 contraintes de précédence.

4.3.2. Evolution des critères

Les solutions ont été générées via une procédure de démarrage à chaud : une première résolution cherche la meilleure valeur possible pour le makespan escalier. Ensuite, la solution est sauvegardée comme point de départ et une contrainte est ajoutée pour ne pas dépasser la fin de cette journée, et l'objectif est modifié pour le remplacer par l’un des critères ergonomiques : *SUM_Max* (1) ou *IMD* (2).

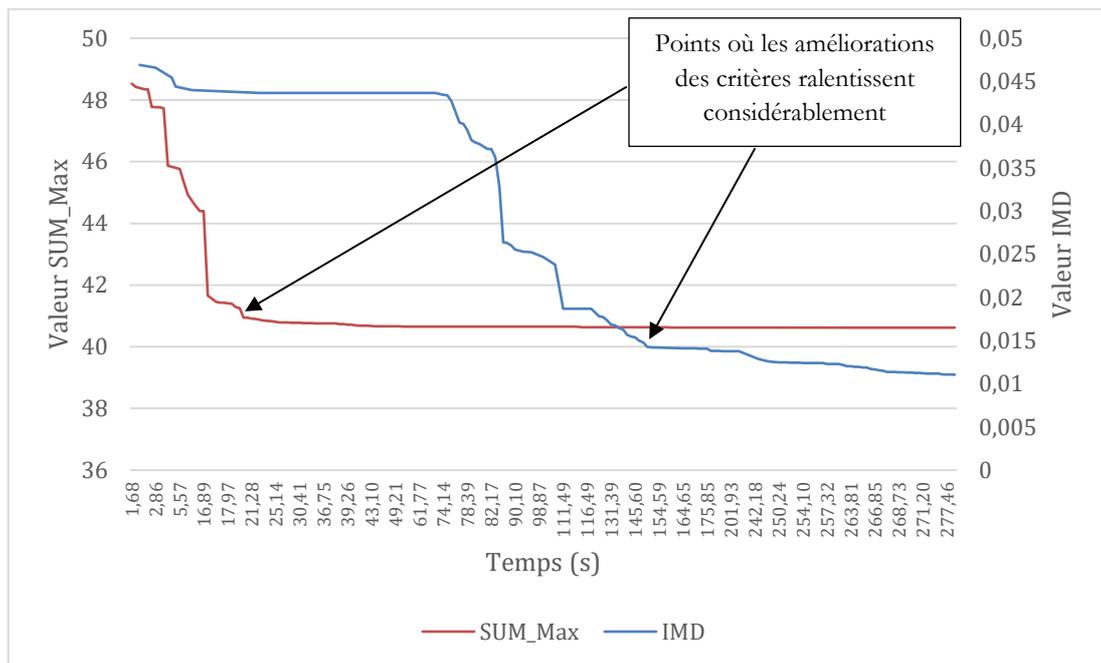


Figure 28 - Évolution des critères. Bleu = IMD (2) ; Rouge = SUM_Max (1). Les valeurs de l'IMD sont sur l'axe de droite, celles de SUM_Max sur l'axe de gauche.

La première observation que nous faisons concerne l'efficacité de chaque critère pour trouver la meilleure solution possible. La figure 28 montre l'évolution de chaque critère dans le temps. L'axe du temps (axe des x) a été coupé pour une meilleure lisibilité. Le temps limite de calcul était d'une heure mais la figure ne représente que les 300 premières secondes pour une meilleure lisibilité. En effet chaque résolution a atteint la limite des 3600 secondes, mais l'évolution des courbes ne change rien pour l'analyse par rapport à leurs positions à 300 secondes.

SUM_Max est plus rapide à réduire l'écart avec sa valeur optimale que IMD. Lorsqu'elle s'aplatit, les améliorations de SUM_Max sont très faibles et assez proches de la valeur optimale (moins de 2% de proximité). SUM_Max atteint sa valeur proche de l'optimum en 20 secondes, tandis qu'il a fallu 150 secondes à l'IMD pour ralentir sa progression de manière significative, et 250 secondes pour atteindre un écart avec sa valeur optimale qui est au moins aussi bon que celui de SUM_Max en 20 secondes. En comparaison, SUM_Max est plus rapide en convergence que IMD.

Sur l'ensemble de l'exécution, SUM_Max a réduit sa valeur de 16,3% (de 48,5273 à 40,6176) tandis que IMD a réduit sa valeur de 76,8% (de 0,0469 à 0,0109). Alors que SUM_Max semble être plus rapide, IMD semble avoir un meilleur potentiel d'amélioration dans le temps.

Le SUM_Max semble être un critère plus adapté pour trouver rapidement une bonne solution. Cette différence pourrait être exploitée par un solveur. Lors de la première étape, le SUM_Max pourrait être utilisé pour atteindre rapidement un point avancé pour que l'IMD prenne la solution comme point de départ et

l'améliore plus efficacement que le SUM_Max qui ralentit fortement. Cette résolution en deux étapes pourrait permettre de trouver une meilleure solution dans le même temps de calcul.

4.3.3. Double critère ergonomique

Dans la continuité de notre conclusion de la section 4.3.2., nous effectuons une résolution en deux étapes en utilisant nos deux critères ergonomiques (après la première étape du makespan escalier). Tout d'abord, SUM_Max est optimisé jusqu'à ce qu'il atteigne un écart par rapport à sa valeur optimale inférieur à 2 % (écart défini par le solveur par rapport à la valeur optimale calculée). Ensuite, nous prenons la solution actuelle et définissons un nouveau point de démarrage à chaud où l'IMD devient le nouveau critère d'optimisation.

Premièrement, nous présentons dans le tableau 5 l'évolution de l'écart à l'optimum pour le critère SUM_Max par rapport au temps de calcul pour voir en combien de temps cet écart est atteint.

Valeur obtenue	Temps écoulé (secondes)	Gap avec l'optimum
44.06	16.08	8.66 %
43.69	25.29	7.89 %
40.94	43.11	1.71 %
40.87	49.36	1.54 %
40.82	53.74	1.41 %

Tableau 5 - Evolution de l'écart à l'optimum pour le critère SUM_Max en fonction du temps écoulé

Nous observons donc qu'il faut seulement 43 secondes à SUM_Max pour atteindre une valeur avec une distance inférieure à 2% de la valeur optimale calculée. Si nous considérons qu'il reste du temps disponible de calcul, il est alors possible de changer pour le critère de l'IMD et ainsi réparer un potentiel effet de noyade tout en améliorant encore la solution.

Ensuite, nous comparons les valeurs de SUM_Max et de l'IMD lorsque, d'un côté, seul le critère SUM_Max est optimisé, tandis que de l'autre côté nous procédons à l'optimisation en deux étapes (via un démarrage à chaud) où le critère SUM_Max dans la première partie puis remplacé par l'IMD dans la deuxième. Nous ne faisons pas de résolution via l'IMD seul car celui-ci est trop long pour converger vers son optimum et le temps de réaction étant important, nous considérons seulement une instance qui utilise uniquement le SUM_Max. La limite de temps de calcul pour chaque solution est d'une heure.

	SUM_MAX	SUM_MAX + IMD
IMD	4.3247 E ⁻⁵	1.4194 E ⁻⁵
OP 1	36.9052	36.9018
OP 2	36.9039	36.9023
OP 3	36.905	36.9019
OP 4	36.9006	36.9024
OP 5	36.8968	36.9021
OP 6	36.9035	36.9028
OP 7	3.3856	3.3831
OP 8	3.384	3.3825
OP 9	3.3818	3.388
OP 10	3.381	3.3828

Tableau 6 - Charges ergonomiques des compagnons après une heure de calcul suivant chaque méthode

Le tableau 6 présente les charges ergonomiques de chaque compagnon obtenues par deux approches : l'optimisation de SUM_Max et l'optimisation en deux étapes. Les compagnons 1 à 6 appartiennent au même sous-groupe, et les compagnons 7 à 10 à un autre. La valeur de IMD a également été calculée dans les deux cas pour une meilleure comparaison.

La première observation est que malgré le critère l'IMD qui est trois fois meilleur pour l'optimisation en deux étapes, les scores des compagnons dans les deux méthodes sont extrêmement proches les uns des autres. Ensuite, la différence entre les deux solutions est relativement faible avec un temps de calcul long. D'un point de vue industriel, ces petites différences peuvent être négligeables dans l'application. Compte tenu de cette conclusion, l'utilisation de l'IMD peut nécessiter un temps de calcul sensiblement plus important, en particulier pour les grands groupes compagnons. L'utilisation de SUM_Max semble être plus appropriée pour le contexte industriel étudié.

4.4. BILAN

4.4.1. Réponse aux questions

- **Comment les deux critères peuvent-ils influencer le planning ?**

D'une part, l'utilisation des deux critères ralentit le calcul d'un planning, mais l'utilisation du critère IMD nécessite un temps plus important pour obtenir

une solution acceptable. D'autre part, l'utilisation des deux critères résulte en des scores ergonomiques égaux parmi les compagnons, ce qui était l'objectif principal. Nous n'avons donc pas observé d'effet de noyade avec SUM_Max, et une hypothèse peut être émise quant à un nombre important de tâches qui permet d'éviter plus facilement cet effet. Cependant, ces deux critères améliorent la solution ergonomiquement et apportent un meilleur équilibre de la charge de travail entre les compagnons.

- **Comment pourrait-on les ajouter dans le calcul d'un planning ?**

La meilleure utilisation de ces critères serait lors de la création du planning initial. A cette étape, le temps de calcul est moins critique et permet d'obtenir des solutions de bonne qualité en termes de répartition des charges.

- **Quel critère est le mieux adapté et pour quels paramètres ?**

D'après les résultats obtenus, les deux critères semblent équivalents pour améliorer une solution ergonomiquement si on se réfère aux scores de chaque compagnon. Malgré la rapidité du critère SUM_Max pour converger, la différence en termes de score à la fin de la résolution est totalement négligeable. Par conséquent, si le critère de l'IMD permet d'éviter les effets de noyade, alors celui-ci est meilleur que le SUM_Max et permet d'avoir une meilleure stabilité dans la résolution et pour la solution finale.

4.4.2. Conclusion

Dans cette étude, nous considérons le contexte d'une ligne d'assemblage aéronautique où les conditions ergonomiques doivent être améliorées. Le critère de l'indice de moyen de déséquilibre (IMD) a été développé. Les impacts ergonomiques des tâches sont calculés via la méthode RULA. L'objectif du critère IMD est de trouver une affectation des tâches aux compagnons telle que la charge ergonomique soit équivalente pour tous les membres du groupe et ainsi éviter un effet de noyade que pourrait provoquer la minimisation de la pire charge ergonomique. Ce critère a été testé sur un cas réel et son efficacité a été comparée à une minimisation classique de la valeur maximale de la charge ergonomique entre tous les compagnons d'un même groupe.

Les résultats des tests ont montré que l'IMD est plus lent à converger vers sa solution optimale, mais que sa solution optimale semble être meilleure avec un temps de calcul plus long. Cette observation a conduit à une optimisation en deux étapes avec, dans une première partie, une optimisation par la minimisation de la charge ergonomique maximale pour atteindre rapidement une bonne solution, puis le passage à l'IMD pour améliorer la solution. Cependant, le temps nécessaire pour

atteindre la valeur optimale semble être long dans le contexte de l'application industrielle et l'utilisation de SUM_Max semble être suffisante.

Au vu de ces observations, l'utilisation de ces critères est une bonne solution pour obtenir rapidement un planning qui respecte un équilibre ergonomique entre les compagnons. Il permet d'éviter des troubles musculosquelettiques à long terme. Par rapport à d'autres modèles, il associe l'évaluation de RULA et la durée des tâches pour évaluer leur difficulté au lieu de seulement considérer l'évaluation qui tiens compte que des gestes (et du coup ignore les expositions longues ou courtes). Par conséquent, notre modèle est une alternative intéressante pour obtenir une solution de bonne qualité pour une redistribution des charges équilibrée.

5.

**PRISE EN COMPTE DES
PREFERENCES
ORGANISATIONNELLES
DANS LA PLANIFICATION
DES LIGNES D'ASSEMBLAGE
AERONAUTIQUES**

5.1. CONTEXTE ET MOTIVATION

Le nombre de tâches à exécuter implique une grande quantité de mouvements, de transmissions de ressource matérielle, changements de position qui sont réalisés par les compagnons. Il est nécessaire de gérer l'enchaînement de tâches de façon efficace afin de réduire les gestes et les déplacements inutiles et réduire ainsi la probabilité d'erreur. Ceci peut être réalisé grâce à l'utilisation des contraintes organisationnelles (CO) de précédence. Pour cela, le planificateur utilise des contraintes un type « fin-début exact » (ESE) en plus des contraintes « fin-début » (ES). Ces contraintes ne traduisent pas une interdiction stricte de process, elles sont donc considérées comme optionnelles, on peut aussi parler des préférences.

Grâce à ces contraintes optionnelles, deux tâches qui auraient été séparées dans le temps voire même qui auraient été affectées à deux compagnons différents se retrouveront collées l'une à l'autre, ce qui favorise le fait qu'un compagnon unique enchaîne les deux tâches, limitant ainsi les changements et réduisant la probabilité d'erreur. Dans la figure 29 nous pouvons trouver deux graphes de précédence : un sans CO (à gauche) et un avec une CO (à droite, la CO étant représentée par la flèche orange).

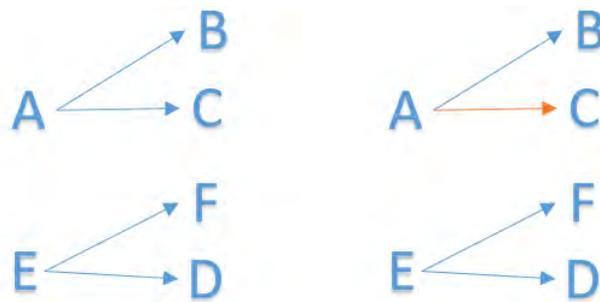


Figure 29 - Graphes de précédences pour une instance sans Contrainte Organisationnelle (à gauche) et avec une Contrainte Organisationnelle (à droite)

Les ordonnancements des tâches résultants de ces graphes sont présentés dans la figure 30. A gauche nous avons l'ordonnancement correspondant à l'instance sans CO et à droite l'instance avec une CO. La CO reliant A et C implique que la tâche C est déplacée du compagnon C3 au compagnon C1 pour que les deux tâches s'enchaînent sans altérer le makespan de la solution. De plus, comme la tâche A se termine alors que les compagnons C2 et C3 n'ont pas fini les leurs, cela force la tâche C à rester affectée au même compagnon que A. S'il faut la même ressource matérielle pour A et C, du temps est économisé à la fois pour le changement de compagnon, mais aussi pour le transfert de la ressource matérielle.

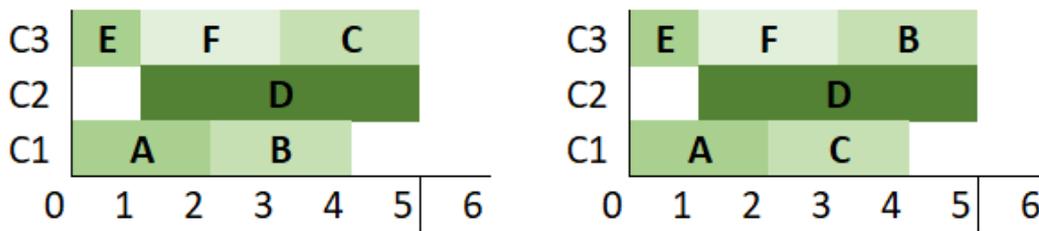


Figure 30 - Ordonnements résultants des instances sans Contrainte Organisationnelle (à gauche) et avec une Contrainte Organisationnelle (à droite)

Dans un contexte industriel comme le nôtre, les tâches sont nombreuses, ce qui implique que les compagnons n’ont pas ou très peu d’intervalles de temps libre. Par conséquent, il y a de fortes chances que pousser deux tâches à s’enchaîner les affecte au même compagnon étant donné que tous les autres auront déjà une tâche affectée (à noter que le faire sur un même compagnon n’est pas obligatoire mais préféré). Cela simplifie ainsi l’implémentation de la solution.

5.2. INTEGRATION DANS LA PLANIFICATION

5.2.1. Méthode choisie

Les contraintes organisationnelles sont plutôt considérées comme des préférences. A ce jour, elles ne sont pas forcément respectées dans le planning final et c’est au planificateur de réarranger les tâches manuellement lorsqu’une solution est calculée. Notre objectif est donc d’automatiser ce processus en intégrant directement les préférences organisationnelles dans le calcul du planning sans qu’elles ne dégradent la qualité de la solution et les critères déjà développés.

5.2.1.1. Représentation des préférences organisationnelles

La modélisation des préférences organisationnelles implique de définir des contraintes qui peuvent ou non être respectées dans la solution. Cela implique de trouver dans un premier temps une manière de vérifier si une contrainte de précédence organisationnelle est respectée, et dans un second temps il faut élaborer un critère capable d’évaluer, en fonction des CO respectées, la qualité organisationnelle de la solution.

Nous proposons un modèle qui permet de représenter les CO de type FDE de deux manières différentes. La première représente le cas où cette contrainte FDE auraient une tâche en commun avec une autre (cf. figure 31).



Figure 31 - Trois tâches (i , j et k) liées par deux CO de type FDE ((i,j) et (j,k))

Ces deux CO forment ce qu'on peut appeler une « chaîne organisationnelle », sous-entendu une chaîne de CO avec en son centre la tâche j . De plus, chaque chaîne comporte trois tâches. En effet, les chaînes ont pour objectif d'évaluer si toutes les CO d'une tâche (ici j) sont respectées pour un enchaînement donné (ici $i \rightarrow j \rightarrow k$). Pour faire ceci, une valorisation plus grande sera attribuée si ces deux CO sont respectées plutôt que deux CO non reliées par une même tâche (ce qui veut dire que le cas de la figure 31 sera valorisé par rapport au cas de la figure 32). Tout ce travail de valorisation sera réalisé grâce au critère que nous avons élaboré et que nous présentons dans la section suivante.



Figure 32 - Ensemble de tâches liées par trois contraintes de précédence dont deux organisationnelles

La deuxième manière de représenter une contrainte FDE est comme une CO dite « solitaire ». Une CO solitaire est une CO qui relie deux tâches qui ne comprennent pas d'autre CO. Une illustration serait la contrainte ($k \rightarrow l$) de la figure 32. La tâche k possède une relation de précédence classique (bleue) et une relation organisationnelle (orange) et la tâche l ne possède qu'une relation organisationnelle avec k .

5.2.1.2. Intégrale de Choquet

L'intégrale de Choquet a vu son utilisation croître au fil des années dans les problèmes d'optimisation, notamment grâce à la popularisation des approches multicritères. La multiplicité des critères et le besoin de les réunir dans un objectif fait de l'opérateur d'agrégation non additif qui est l'intégrale de Choquet un outil efficace dans les modèles récents. Plus particulièrement populaire au Japon, un acteur de sa démocratisation en Europe fût Grabisch (1996) qui introduit l'intégrale en prenant comme exemple les applications déjà présentes en Asie.

Pour mieux comprendre comment elle est utilisée, prenons pour exemple un ensemble d'attributs $X_1, \dots, X_n, n \geq 2$ avec $N := \{1, \dots, n\}$. Ces attributs peuvent être de n'importe quelle nature (une valeur ou une couleur par exemple) et l'objectif est de modéliser les préférences d'un manager entre les éléments de $X = X_1 \times \dots \times X_n$. Pour tout $x \in X$, $x = (x_1, \dots, x_n)$ avec $x_i \in X_i, \forall i \in \{1, \dots, n\}$. La manière la plus simple de modéliser cette préférence \geq est via l'utilisation d'une fonction d'utilité $u: X \rightarrow \mathbb{R}$:

$$\forall x, y \in X, \quad x \geq y \Leftrightarrow u(x) \geq u(y)$$

La fonction d’utilité globale qui en découle est la suivante :

$$U(x) = F(u_1(x_1), \dots, u_n(x_n))$$

où les $u_i(x_i)$ sont les fonctions d’utilité et F la fonction d’agrégation. Cette formulation représentant la base du problème de décision multicritère est présentée dans les travaux de Labreuche et Grabisch (2003). La difficulté réside ensuite dans la définition des fonctions d’utilité u_i et de la fonction d’agrégation F qui dépend des paramètres du problème. Pour la suite de la présentation, l’association d’un élément i de N à sa fonction d’utilité u_i sera désigné par le terme « critère ».

Debreu (1960) montre que dans un cas où les préférences entre les critères sont indépendantes avec aucune interaction entre elles, l’expression de F prend la forme d’une somme pondérée. Cependant, Grabisch et al. (2008) montrent que ce cas de figure est rarement présent dans la réalité. Pour être capable de prendre en compte les interactions entre chaque critère, il est possible de remplacer le vecteur des poids par une fonction d’ensemble sur N dans le calcul de F . Cette approche, au-delà de prendre en compte l’importance de chaque critère mais aussi l’importance de chaque ensemble de critères possible. C’est dans ce contexte que l’intégrale de Choquet est utilisée. L’utilisation de cette intégrale nécessite donc d’être capable de pondérer, chaque ensemble de critères pour obtenir une évaluation d’une solution et pas seulement chaque critère.

Pour présenter la définition de l’intégrale, nous reprenons le travail de Grabisch et al. (2008) et utiliserons $\mathcal{P}(N)$ pour représenter la totalité des ensembles possibles en utilisant les éléments de N .

Définition 5.1. La mesure sur N est la fonction d’ensemble $\mu: \mathcal{P}(N) \rightarrow [0,1]$ satisfaisant les conditions suivantes :

- (i) $\mu(\emptyset) = 0$ et $\mu(N) = 1$
- (ii) Pour tout $S \subseteq T \subseteq N \Rightarrow \mu(S) \leq \mu(T)$

De plus, pour tout ensemble disjoint $S, T \subseteq N$, la mesure μ sur N est dite :

- *Sous-additive* si $\mu(S \cup T) \leq \mu(S) + \mu(T)$
- *Additive* si $\mu(S \cup T) = \mu(S) + \mu(T)$
- *Super-additive* si $\mu(S \cup T) \geq \mu(S) + \mu(T)$
- *Basée sur la cardinalité* si, pour tout $T \subseteq N$, $\mu(T)$ ne dépend que de la cardinalité de T .

Pour chaque ensemble $S \subseteq N$, la mesure $\mu(S)$ peut être interprétée comme le poids de l’ensemble S . De plus, la monotonie de μ implique que le poids d’un

ensemble ne peut pas décroître si on y ajoute un élément de N , cela ne peut que résulter en un poids plus fort ou égal. En utilisant μ pour modéliser le poids d'un ensemble, l'opérateur d'agrégation est l'intégrale de Choquet.

Définition 5.2. Considérons la fonction de mesure μ dont les éléments sont notés x_1, \dots, x_n et une fonction $f: X \rightarrow \mathbb{R}$ qui représente la satisfaction d'un objet sur n critères. L'intégrale discrète de Choquet de la fonction f est définie par :

$$C_\mu(f) := \sum_{i=1}^n [f(\sigma_i) - f(\sigma_{i-1})] * \mu(A_i)$$

Avec $A_i := \{\sigma_i, \dots, \sigma_n\}$, $\forall i \in \{1, \dots, n\}$ et $A_{n+1} = \emptyset$ et où σ_i est une permutation des x_i telle que $f(\sigma_1) \leq f(\sigma_2) \leq \dots \leq f(\sigma_n)$.

En termes de décision multicritère, la fonction de mesure (ou de capacité) μ et l'intégrale de Choquet sont des outils très efficaces qui permettent de valoriser à la fois les critères individuellement mais aussi les ensembles de critères (notamment grâce à la permutation effectuée avec les σ_i). Guillaume et al. (2014) utilisent par exemple cette formulation dans un problème d'affectation de personnel à des tâches en considérant des compétences mais dont la maîtrise n'est pas exacte pour chaque personne. L'intégrale de Choquet est par la suite employée pour l'interaction possible entre certaines compétences et prendre une approche robuste du problème.

Cette expression et ces caractéristiques sont très proches de notre cas de figure où nous avons à la fois des CO solitaires et des chaînes de CO reliant trois tâches différentes. Nous avons choisi de ne considérer que trois tâches car il est possible de représenter l'ensemble des chaînes possibles peu importe leur longueur avec des associations de triplets et de CO solitaires. Cela permet ainsi de réduire le nombre de variables qui aurait dû considérer toutes les possibilités. Par exemple, pour considérer une suite de 10 tâches liées par des contraintes organisationnelles, il aurait été nécessaire de créer toutes les chaînes de trois tâches, de quatre tâches et ainsi de suite jusqu'à 10. Éviter cette explosion de variables était donc un aspect important de la modélisation. L'utilisation de Choquet nous permettra donc de valoriser les ensembles de chaînes et CO solitaires validées tout en considérant les chaînes de CO qui ne sont pas complètes (i.e. qui n'ont qu'une précedence sur les deux vérifiée) tout en respectant les chaînes plus longues.

5.2.2. Modélisation

5.2.2.1. Les contraintes de précedence organisationnelles

Nous avons vu que l’intégrale de Choquet permet de prendre en compte l’importance de chaque critère tout comme celles sur les ensembles de critères. Dans notre cas, nous voulons évaluer les tâches en fonction du nombre de CO vérifiées qui leur sont associées. Nous avons donc quatre cas de figure pour une tâche, compte tenu de la représentation vue en 5.2.1. :

- (i) Une tâche n’est associée à aucune CO ;
- (ii) Une tâche possède une unique CO, créant une CO solitaire ;
- (iii) Une tâche possède exactement deux CO, une avec une tâche précédente, et une avec une tâche suivante (cf. figure 31), ce qui aboutit à une chaîne organisationnelle ;
- (iv) Une tâche possède plus de deux CO, et plusieurs variables sont définies pour représenter l’ensemble de ces précédences.

Nous proposons une modélisation qui permet de représenter chacun de ces quatre cas.

Comme nous ne pouvons pas définir les précédences de manière classique (i.e. avec les fonctions de la librairie CPO), nous utilisons les expressions de précedence respectives à chaque type de contrainte e :

$$e_i + d_{i,j} = s_j, \quad \forall i, j \in \text{pour les FDE}$$

$$s_i + d_{i,j} = s_j, \quad \forall i, j \in \text{pour les DDE}$$

Pour rendre optionnel le respect de ces contraintes nous proposons l’utilisation de variables de décision supplémentaires. Les chaînes étant différentes des CO solitaires, leur implémentation des CO sera différente. Les CO solitaires et les chaînes seront dissociées dans le modèle et n’auront donc pas de variable de décision en commun.

En donnée d’entrée, nous avons l’ensemble \mathcal{C} qui représente l’ensemble des chaînes organisationnelles. Pour tout $c \in \mathcal{C}$, trois tâches sont définies et nommées suivant la figure 31 (i.e. l’ordre sera $i \rightarrow j \rightarrow k$) ainsi que les potentiels délais associés à chaque contrainte FDE. i représente donc la première tâche de la chaîne, j la deuxième et k la troisième. Les ensembles E_{FDE} et E_{DDE} sont également définis pour représenter les CO solitaires de chaque type de précedence. Les variables de décision supplémentaires définies pour le modèle sont les suivantes :

- g_c : intervalle optionnel qui est présent dans la solution si les deux CO de la chaîne sont respectées, $\forall c \in \mathcal{C}$.

- ij_c : variable booléenne qui indique si la CO entre i et j est respectée, $\forall c \in \mathcal{C}$.
- jk_c : variable booléenne qui indique si la CO entre j et k est respectée, $\forall c \in \mathcal{C}$.
- $half_c$: variable booléenne qui indique si exactement 1 CO sur les deux est respectée, $\forall c \in \mathcal{C}$.
- b_e : variable booléenne qui indique si la CO solitaire $e \in E_{FDE} \cup E_{DDE}$ est respectée.

Les contraintes suivantes sont déclarées pour définir la présence ou non des intervalles de décision g_c . De plus, même lorsque qu'ils sont présents, ces intervalles ne seront pas représentés dans la solution finale. Leur présence sert au décompte des chaînes organisationnelles validées et permettent d'utiliser les méthodes associées aux intervalles de décision de CP Optimizer qui sont très efficaces.

$$\forall c \in \mathcal{C} \begin{cases} ij_c * (endOf(intvl_{c,i}) - startOf(intvl_{c,j})) = 0 & [1b] \\ jk_c * (endOf(intvl_{c,j}) - startOf(intvl_{c,k})) = 0 & [2b] \\ ij_c * jk_c \geq presenceOf(g_c) & [3b] \\ ij_c + jk_c - 2 * presenceOf(g_c) \geq half_c & [4b] \end{cases}$$

Les contraintes [1b] à [4b] servent à déterminer si pour chaque chaîne zéro, une ou deux CO sont respectées. Les deux premières contraintes sont utilisées pour affecter les valeurs aux variables booléennes ij_c et jk_c . Ensuite, via la contrainte [3b], si ces deux variables sont à 1, alors g_c est défini comme présent dans la solution. A l'inverse, si une seule des deux variables est à 1, alors « $presenceOf(g_c)$ » renvoie la valeur 0 (vu que g_c est absent) et via la contrainte [4b], $half_c$ peut prendre ou non la valeur 1 (en fonction de si au moins une des deux variables ij_c et jk_c est à 1). L'ensemble des valeurs que peuvent prendre chaque variable est défini dans le tableau 7.

ij_c	jk_c	$ij_c * jk_c$	$ij_c + jk_c$	g_c	$half_c$
0	0	0	0	0	0
	1	0	1	0	1
1	0	0	1	0	1
	1	1	2	1	0

Tableau 7 - Valeurs possibles pour chaque variable de décision

Pour modéliser les CO solitaires les contraintes [5b] et [6b] seront utilisées.

$$b_e * (endOf(intvl_{e,i}) - startOf(intvl_{e,j}) + delay) = 0, \forall e \in E_{ESE} [5b]$$

$$b_e * (startOf(intvl_{e,i}) - startOf(intvl_{e,j}) + delay) = 0, \forall e \in E_{SSE} [6b]$$

Elles ont la même structure que les contraintes [1b] et [2b] et la validation des CO se fera uniquement via les variables booléennes déclarées. L’évaluation de la qualité organisationnelle d’une solution se fera ensuite via l’utilisation d’un critère basé sur l’intégrale de Choquet que nous présentons dans la partie suivante.

5.2.2.2. Evaluation via Choquet

Maintenant que les contraintes organisationnelles sont modélisées pour les précédences, il faut pouvoir évaluer la solution en termes de qualité organisationnelle. La seule information par rapport aux préférences des décideurs étant les précédences déclarées et que nous pouvons vérifier, nous évaluons la qualité organisationnelle de la solution par le calcul du nombre de CO respectées.

L’objectif est donc de déterminer l’apport de chaque variable g_c et co_e qui représentent ces CO dans la solution. La fonction d’évaluation a pour objectif de valoriser, dans cet ordre, les chaînes organisationnelles qui verront leurs deux CO respectées ainsi que les contraintes solitaires, puis avec une importance moins forte nous retrouvons les chaînes incomplètes et au dernier plan les chaînes dont aucune CO n’est vérifiée ainsi que les contraintes solitaires non respectées.

Pour ce faire, considérons les ensembles $G := \{g_c | c \in C\}$, $B := \{b_e | e \in E_{FDE} \cup E_{DDE}\}$ et $GB := G \cup B$. La fonction d’évaluation $f: GB \rightarrow [0,1]$ est définie comme tel :

$$f(gb) = \begin{cases} 1 & \text{si TOUTES les CO sont satisfaites} \\ 0.5 & \text{si AU MOINS une CO est satisfaite,} \\ 0 & \text{sinon} \end{cases} \quad \forall gb \in GB$$

A noter que pour tout élément $b_e \in B$, f ne peut prendre que deux valeurs possibles étant donné qu’une seule CO peut être satisfaite.

La définition de cette fonction permet de déterminer trois ensembles :

- $A_1 := \{gb | f(gb) \geq 1\}$
- $A_{0.5} := \{gb | f(gb) \geq 0.5\}$
- $A_0 := \{gb | f(gb) \geq 0\}$

Pour évaluer ces ensembles, considérant un ensemble quelconque $A \in P(GB)$, nous définissons la mesure suivante :

$$\mu(A) = |A|^2$$

Nous avons donc une mesure qui se base sur la cardinalité de chaque ensemble. Par la définition de la mesure et la nature discrète de ω qui ne peut prendre que trois valeurs, nous définissons l’intégrale de Choquet comme suit :

$$C_{\mu}(f) := (1 - 0.5) * \mu(A_1) + (0.5 - 0) * \mu(A_{0.5})$$

Contrairement à la définition 5.2., la fonction de mesure μ proposée n'évalue qu'un seul élément puisqu'on se base sur la cardinalité des ensembles. De plus, ces ensembles A_{ω} sont formés par les évaluations via la fonction f . Notre objectif est donc de valoriser les ensembles d'éléments vérifiant le plus de CO, et ce calcul est semblable au calcul d'aire de la figure 33.

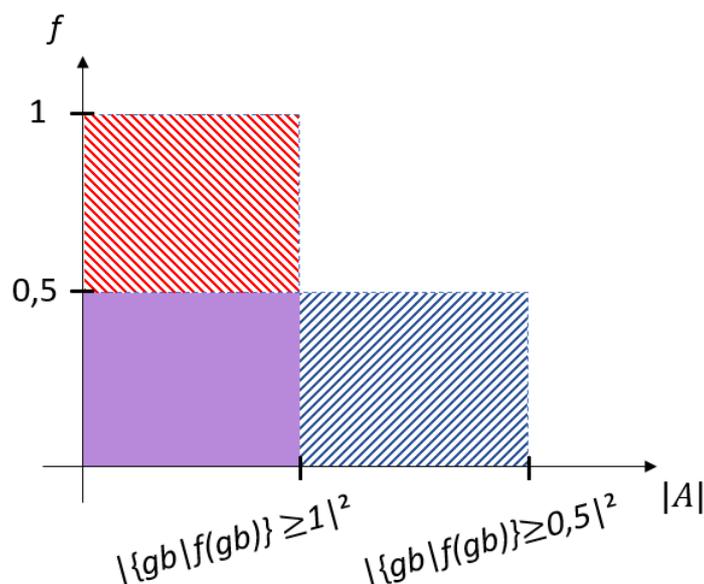


Figure 33 - Aire de la fonction discrète de Choquet

Le calcul de l'intégrale de Choquet revient donc à calculer l'aire représentée par les rectangles rayés, tout en soustrayant l'intersection des deux (partie pleine) qui est par conséquent comptabilisée deux fois. Cela donne ainsi l'intégrale de Choquet suivante :

$$C_{\mu}(f) = 0.5 * (|A_1|^2 + |A_{0.5}|^2)$$

Il est maintenant possible d'évaluer la qualité de la solution par rapport aux contraintes solitaires et aux chaînes organisationnelles. Grâce à la mesure μ , l'ensemble A_1 sera valorisé et nous espérons obtenir une solution qui exploite au maximum les contraintes organisationnelles complètes. L'objectif est donc maintenant de tester ce critère sur une instance issue d'un cas réel.

5.3. EVALUATION DU SCORE ORGANISATIONNEL

Grâce au critère de Choquet (terme que nous utiliserons désormais pour parler du critère de décision basé sur l'intégrale de Choquet présentée dans le 5.2.), nous voulons améliorer l'organisation d'une solution à travers les préférences organisationnelles. Ces préférences, sous la forme de précédences optionnelles, ont

pour objectif dans la solution de fluidifier l’enchaînement des tâches effectuées par les compagnons. Cela comprend l’enchaînement des tâches d’un même groupe les unes à la suite des autres en évitant les changements de compagnon entre deux tâches.

Nous avons donc choisi une instance qui, lorsqu’on considère aucune préférence organisationnelle, donne une planification de n jours, et lorsqu’on rajoute toutes les précédentes optionnelles comme si elles étaient obligatoires, donne une planification de minimum $n + 1$ jours. Nous espérons ainsi observer le comportement de la solution mais aussi l’impact sur la vitesse de résolution dans un cadre de planification.

Nous ne nous plaçons pas ici dans un contexte de replanification car nous estimons que si l’instance de départ est optimisée d’un point de vue organisationnel, le critère des affectations suffira à maintenir une bonne qualité organisationnelle. Le critère de Choquet pourra à tout moment être réintégré dans un contexte de replanification si les décideurs ont le temps nécessaire.

5.3.1. Présentation de l’instance

L’instance est une version simplifiée d’un cas réel. Elle présente 20 tâches nécessitant chacune un seul compagnon pour être réalisée et il y a deux compagnons disponibles qui travaillent suivant les mêmes horaires (T2). Le graphe de précedence des tâches est présenté dans la figure 34.

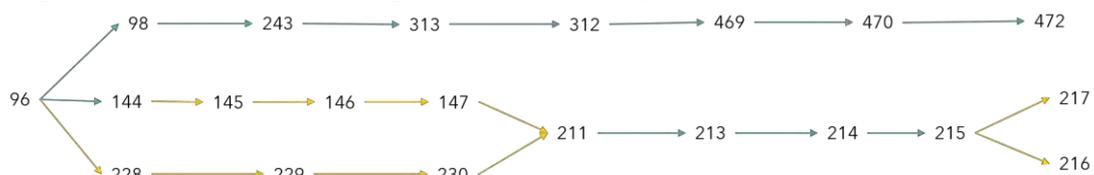


Figure 34 - Graphe de précedence des tâches de l'instance de test du critère de Choquet

Dans ce graphe, les flèches bleues représentent les précédences obligatoires de type « Fin-Début » tandis que les flèches jaunes représentent les précédences organisationnelles. Ces dernières sont considérées comme des précédences « Fin-Début » obligatoires et l’objectif sera de faire en sorte qu’il n’y ait pas de pause entre les deux tâches concernées dans la solution finale. Partant de ce graphe, les chaînes de précedence seront les suivantes :

- (144, 145, 146)
- (145, 146, 147)
- (146, 147, 211)
- (96, 228, 229)
- (228, 229, 230)

(229, 230, 211)

Quant aux contraintes organisationnelles solitaires, il y en a deux :

(215, 216)

(215, 217)

Pour vérifier que l'instance correspond au cas de figure auquel nous voulons répondre (i.e. la différence de jours nécessaires pour réaliser le projet sans préférences organisationnelles et avec préférences organisationnelles), nous avons lancé deux résolutions : une sans les préférences, et une avec les préférences comme des contraintes de précédence obligatoirement vérifiées. Les deux solutions ont un makespan différent avec 2003 sans les préférences et 3353 avec les préférences (cf. figure 35).

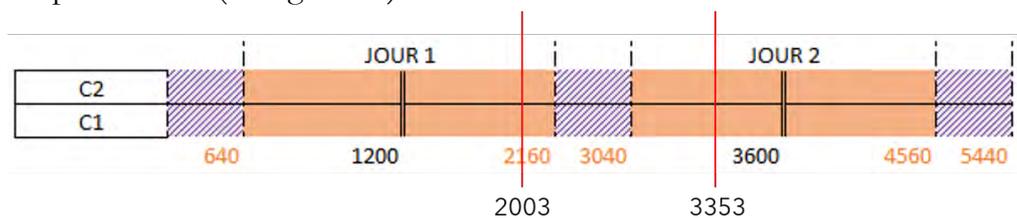


Figure 35 - Planning montrant la date de fin sans les préférences (droite de gauche) et avec les préférences (droite de droite)

Sur cette figure, les plages horaires de travail des compagnons C1 et C2 sont représentées par les rectangles pleins. Les doubles segments verticaux représentent la pause de midi de chaque jour et les zones hachurées les nuits (période de non travail). Le makespan sans les contraintes organisationnelles se termine donc à la date « 2003 » du planning, soit un peu avant la fin du jour 1. Lorsque les contraintes organisationnelles sont considérées (et donc que l'ensemble de précédences considérées comme « Fin-Debut » deviennent des précédences « Fin-Début Exact »), le makespan obtenu devient 3353 unités de temps. Nous avons donc une dégradation de la qualité du makespan mais surtout un décalage de la fin du projet sur le jour 2, ce qui correspond au cas de figure que nous recherchons.

5.3.2. Test du critère de Choquet

L'objectif est de déterminer comment le critère améliore la solution d'un point de vue organisationnel, et de voir s'il permet d'accéder rapidement à une planification qui facilite la vie de travail des compagnons sur place. Pour ceci, nous allons confronter une résolution en démarrage à chaud et une résolution lexicographique.

Dans le démarrage à chaud, nous calculons une première solution sans les contraintes organisationnelles en utilisant le critère du makespan, puis dans la deuxième phase nous ajoutons une contrainte interdisant à la solution de dépasser

la fin du jour 1 et utilisons le critère de Choquet comme critère de décision. A noter également qu’il y a un critère d’arrêt dans la première phase de résolution suivant le makespan. Ce critère implique que la résolution passe à la deuxième phase si on obtient un makespan possédant un gap inférieur à 2% avec la solution optimale calculée par le solver. Ce pourcentage est estimé pour que la planification se termine dans le jour 1 et ne dépasse pas au jour 2 et est calculé en prenant le plus petit makespan possible et en regardant sa proximité avec la fin de l’IP.

Dans la résolution lexicographique, le makespan escalier est le critère prioritaire tandis que Choquet est le critère secondaire. Nous utilisons le makespan en escalier et non pas le makespan car, comme pour le démarrage à chaud et pour chaque test que nous avons effectué, nous voulons obtenir un planning dont la fin se situe avant la fin d’une journée et non pas forcément la meilleure solution possible en termes d’unités de temps.

Nous pourrions ainsi comparer l’efficacité de ces deux méthodes et les solutions fournies, le tout dans une limite de temps de 10 minutes maximum.

5.3.3. Discussion des résultats

Dans le tableau 8 figurent les résultats des deux approches. Nous présentons pour chacune le makespan de la solution finale et le temps qu’il a fallu pour déterminer cette solution. La valeur du critère de Choquet n’est pas présentée car elle ne fournit pas d’information supplémentaire par rapport à celles que nous possédons déjà. Dans la deuxième section du tableau, nous présentons pour chaque chaîne organisationnelle et contrainte solitaire le nombre de CO validées dans la solution.

	Démarrage à chaud	Lexicographique
<i>Makespan</i>	2137	2159
<i>Temps</i>	0,23 + 0,21 = 0,44 s	1,06 s
(144, 145, 146)	2/2	2/2
(145, 146, 147)	2/2	2/2
(146, 147, 211)	2/2	2/2
(96, 228, 229)	2/2	1/2
(228, 229, 230)	2/2	2/2
(229, 230, 211)	1/2	2/2
(215, 216)	1/1	1/1
(215, 217)	1/1	1/1

Tableau 8 - Résultats via les deux approches de résolution pour calculer une solution avec le critère de Choquet

La première observation concerne le makespan, qui est relativement similaire et correspond à nos attentes de finir dans la première journée. Le critère de Choquet quant à lui est le même pour les deux solutions. Cependant, le temps pour déterminer ces deux solutions double pour l'approche lexicographique comparé au démarrage à chaud, avec 1,06 secondes contre seulement 0,44 secondes. Dans les 0,44 secondes, il a fallu 0,23 secondes pour calculer la première solution via le makespan, puis 0,21 secondes pour améliorer la solution finale via le critère de Choquet. Le démarrage à chaud est plus adapté pour des situations où le temps de calcul est très contraint car cette méthode est plus rapide pour converger vers la bonne solution. Cette différence est due à plusieurs facteurs :

- (i) Le critère d'arrêt sur le makespan dans la phase 1 du démarrage à chaud permet de ne pas dépenser des ressources inutilement pour améliorer une solution qui sera dégradée (suivant le critère du makespan) dans la phase 2.
- (ii) Le critère du makespan a une convergence plus rapide que le makespan escalier.
- (iii) L'aspect lexicographique qui fait que les deux critères qui sont optimisés en même temps implique plus de calculs pour définir si la solution explorée est meilleure que la solution actuelle ou non.

En ce qui concerne les préférences organisationnelles, nous pouvons voir en gras les deux chaînes qui ne sont pas complètement vérifiées, avec chacune une contrainte organisationnelle sur deux de validée.

D'un point de vue métier, nous avons potentiellement une excellente solution qui favorise les préférences déterminées, et techniquement la meilleure solution possible compte tenu que valider la totalité de ces précédences aboutirait à la solution qui se termine au jour 2. La seule différence réside désormais dans la contrainte organisationnelle qui n'est pas respectée, et dans les deux cas se situe sur la suite $96 \rightarrow 228 \rightarrow 229 \rightarrow 230 \rightarrow 211$. Même s'il est possible de déterminer quelle précédence n'est pas vérifiée dans chaque cas, déterminer quelle est la meilleure solution devient la tâche des décideurs selon leurs préférences. Nous considérons donc que pour cette instance, le critère remplit pleinement son rôle dans l'élaboration d'une meilleure solution organisationnelle et dans la fluidification du travail des compagnons avec les deux types de résolution.

CONCLUSION ET PERSPECTIVES

Dans le cadre de cette thèse, nous avons étudié le problème de replanification d'une ligne d'assemblage aéronautique à la suite d'une perturbation de son fonctionnement. Les lignes d'assemblage aéronautiques sont des systèmes de production très complexes où les sources d'aléa sont multiples. Dans ce contexte, les managers ont besoin des méthodes efficaces pour chercher rapidement des solutions de repli en cas d'une perturbation. Selon la situation, ils peuvent mettre en priorité des critères d'optimisation différents parmi lesquels se trouvent la nécessité de respecter le temps de cycle, de minimiser les ressources complémentaires nécessaires ou de préserver au maximum l'affectation initiale des tâches aux compagnons. Nous développons une approche réactive permettant d'intégrer tous ces critères.

Le modèle mathématique développé tire parti des informations du planning initial pour déterminer un point de départ lors du calcul de la solution et se sert de critères originaux pour guider plus spécifiquement la recherche des bonnes solutions de replanification. Parmi ces critères nous retrouvons le makespan escalier, qui permet de faire coïncider la fin du nouveau planning avec la fin de l'ancien (au mieux) ou au pire avec la fin de journée la plus proche de l'ancienne date de fin du projet. Couplé à ce critère, nous avons également développé un critère visant à maintenir la stabilité des affectations des tâches aux compagnons. Cette stabilité est primordiale pour les managers et l'entreprise car elle permet de réduire les changements induits pour les compagnons présents sur la ligne d'assemblage et donc réduire le risque d'une erreur ainsi que de faciliter la reprise du processus plus rapidement étant donné que cela limite les changements à effectuer. Nous avons étudié l'impact de l'utilisation de plusieurs combinaisons lexicographique des critères élaborés en fournissant une aide à la décision en ce qui concerne le choix des meilleurs critères à utiliser dans chaque situation pour les managers. Grâce aux tests réalisés sur des instances de taille réelle, nous avons obtenu des résultats encourageants quant à l'applicabilité de ce modèle en pratique. En effet, notre approche permet d'obtenir une nouvelle solution en moins de dix minutes, évitant ainsi un arrêt trop long de la chaîne. Ceci permettrait aux managers de pouvoir décider si la solution obtenue est suffisamment efficace et avoir du temps pour faire des calculs complémentaires si nécessaire.

Nous avons également étudié l'intégration des risques ergonomiques dans le modèle de planification afin d'améliorer les conditions de travail et le bien être des compagnons. Nous avons élaboré une évaluation de la pénibilité des tâches en prenant en compte le score fourni par la méthode RULA (donc une note allant de 1 à 10) et le temps d'exécution de chaque tâche. Nous avons introduit un nouveau critère appelé « Indice Moyen de Déséquilibre » pour évaluer l'équité de la distribution des tâches pénibles parmi les compagnons. Nous avons testé ce critère sur les instances de taille réelle en le comparant avec le critère de la minimisation

de la charge ergonomique maximale en fournissant une aide à la décision aux managers souhaitant améliorer les risques ergonomiques dans les lignes d'assemblage aéronautiques.

Enfin nous avons étudié l'utilisation des préférences organisationnelles pour l'enchaînement des tâches. Ces contraintes organisationnelles permettent de fluidifier l'assemblage en évitant les changements de compagnon pour les tâches qui nécessitent une suite logique d'exécution, mais n'exigent pas une précedence stricte du point de vue du process d'assemblage. Ces préférences se caractérisent par des relations de précedence modifiées pour forcer deux tâches ou plus à se suivre sans pause dans le planning. Pour intégrer ces contraintes dans notre modèle PPC, nous avons utilisé l'intégrale de Choquet pour évaluer la solution en termes de sa qualité organisationnelle. Nous avons testé ce critère sur des instances de taille réelle et les résultats obtenus ont montré son utilité et efficacité.

En résumé, cette thèse a permis d'élaborer une méthode de replanification efficace en exploitant des critères de décision innovants dans le contexte des lignes d'assemblage aéronautiques. Le premier modèle avec des critères économiques basé sur la modélisation PPC a été enrichi par des critères complémentaires portant sur l'évaluation des risques ergonomiques et la satisfaction des préférences organisationnelles. L'ensemble représente un outil d'aide à la décision performant dont l'efficacité a été attestée par de nombreux tests sur des instances de taille réelle.

Parmi les extensions possibles de ces travaux, il est possible de coupler notre approche avec une méthode proactive robuste afin de bénéficier des avantages des deux stratégies. Ceci permettra de développer une méthode plus complète et plus polyvalente face aux différentes sources de l'incertitude. Il est également envisageable d'ajouter une hybridation avec des méthodes d'apprentissage automatique en capitalisant sur l'information observée lors des perturbations précédentes.

En ce qui concerne les critères ergonomiques, il sera intéressant d'enrichir le modèle avec d'autres indicateurs par exemple des scores subjectifs représentant le ressenti des compagnons de la pénibilité de leurs tâches qui peut être personnalisé. Ainsi l'équité de la distribution du travail sera confrontée à la satisfaction des préférences personnelles des compagnons. Il sera intéressant d'étudier l'interaction de deux critères potentiellement contradictoires.

En ce qui concerne les contraintes organisationnelles, le lien avec la qualité de l'exécution des tâches peut être étudié afin de vérifier le vrai impact de ces choix optionnels. Il est possible de modéliser les changements entre les tâches avec une fonction de coût et comparer l'utilisation d'un tel critère avec le modèle que nous avons élaboré dans cette thèse.

De manière plus générale, il sera intéressant de tester le modèle que nous avons élaboré dans un contexte industriel différent pour étudier sa généralité et explorer les performances des critères et des contraintes proposés sur des instances ayant des caractéristiques différentes en termes du nombre de tâches, opérateurs et la densité du graphe de précédence.

REFERENCES

Almeida B. F., Correia I. & Saldanha-da-Gama F. (2019). Modeling frameworks for the multi-skill resource-constrained project scheduling problem: a theoretical and empirical comparison. *International Transactions in Operational Research*, 26(3), 946–967.

Aouni, B., d'Avignon, G., & Gagnon, M. (2015). Goal programming for multi-objective resource-constrained project scheduling. In C. Schwindt, & J. Zimmermann (Eds.), *Handbook on project management and scheduling*, 1, 429–442.

Arkhipov D., Battaïa O., Cegarra J. & Lazarev A. (2018a). Operator assignment problem in aircraft assembly lines: a new planning approach taking into account economic and ergonomic constraints. *Procedia CIRP*, 76, 63-66.

Arkhipov D., Battaïa O., Cegarra J. & Lazarev A. (2018b). Work planning in low-volume assembly lines under ergonomic constraints. *Procedia CIRP*, 72, 786-789.

Ballestín, F., & Blanco, R. (2015). Theoretical and practical fundamentals. In C. Schwindt, & J. Zimmermann (Eds.), *Handbook on project management and scheduling*, 1, 411–427.

Battaïa O., Sanmartin L. & Pralet C. (2019). Dealing with disruptions in low-volume manufacturing: A constraint programming approach. *Procedia CIRP*, 81, 1372-1375.

Bianco, L., Caramia, M. & Giordani, S. (2016). Resource levelling in project scheduling with generalized precedence relationships and variable execution intensities. *OR Spectrum*, 38, 405–425.

Bianco, L., Caramia, M., & Giordani, S. (2017). The total adjustment cost problem with variable activity durations and intensities. *European Journal of Industrial Engineering*, 11(6), 708–724.

Biele A. & Mönch L. (2018). Hybrid approaches to optimize mixed-model assembly lines in low-volume manufacturing. *Journal of Heuristics*, 24(1), 49-81.

Borgonovo E. & Plischke E. (2016). Sensitivity analysis: a review of recent advances. *European Journal of Operational Research*, 248, 869–887.

Carlier J., Moukrim A. & Sahli A. (2018). Lower bounds for the event scheduling problem with consumption and production of resources. *Discrete Applied Mathematics*, 234, 178–194.

Chaari T., Chaabane S., Aissani N. & Trentesaux D. (2014). Scheduling under uncertainty: Survey and research directions. *2014 International Conference on Advanced Logistics and Transport*, Hammamet, Tunisia.

Chakraborty R.K., Sarker R.A. & Essam D.L. (2017). Resource constrained project scheduling with uncertain activity durations. *Computers & Industrial Engineering*, 112, 537–550.

Chen A. H.-L., Liang Y.-C. & Padilla J. D. (2022). Dealing with Uncertainty in the MRCPSP/Max Using Discrete Differential Evolution and Entropy. *Applied Sciences*, 12(6):3049.

Chen P. H. & Shahandashti S. M. (2009). Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. *Automation in Construction*, 18(4), 434-443.

Chen Z., Demeulemeester E., Sijun B. & Yuntao G. (2018). Efficient priority rules for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, 270(3), 957–967.

Cheng J., Fowler J., Kempf K. & Mason S. (2015). Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting. *Computers & Operations Research*, 53, 275–287.

Creemers S. (2015). Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. *Journal of Scheduling*, 18(3), 263–273.

Dai H., Cheng W., & Guo P. (2018). An improved tabu search for multi-skill resource-constrained project scheduling problems under step-deterioration. *Arabian Journal for Science and Engineering*, 43, 3279–3290.

Davari M. & Demeulemeester E. (2018). A novel branch-and-bound algorithm for the chance-constrained resource-constrained project scheduling problem. *International Journal of Production Research*, 57(4), 1265-1282.

Davari M. & Demeulemeester E. (2019). The proactive and reactive resource-constrained project scheduling problem. *Journal of Scheduling*, 22(2), 211-237.

De Azevedo G. H. I., Pessoa A. A. & Subramanian A. (2021). A satisfiability and workload-based exact method for the resource constrained project scheduling problem with generalized precedence constraints. *European Journal of Operational Research*, 289(3), 809–824.

Demeulemeester E.L. & Herroelen W.S. (2002). Project Scheduling: A Research Handbook, vol. 49. Kluwer Academic Publishers, Boston.

Dubois D., Fargier H. & Fortemps P. (2003). Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research*, 147(2), 231-252.

El Mouayni I., Demesure G., Bril El Haouzi H., Charpentier P. & Siadat A. (2019). Jobs scheduling within Industry 4.0 with consideration of worker's fatigue and reliability using Greedy Randomized Adaptive Search Procedure. *IFAC-PapersOnLine*, 52(19), 85-90.

Elloumi S., Fortemps P. & Loukil T. (2017). Multi-objective algorithms to multi-mode resource-constrained projects under mode change disruption. *Computers & Industrial Engineering*, 106, 161-173.

Florez L., Castro-Lacouture D. & Medaglia A. L. (2013). Sustainable workforce scheduling in construction program management. *Journal of the Operational Research Society*, 64(8), 1169–1181.

Geiger M. J. (2017). A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem. *European Journal of Operational Research*, 256(3), 729–741.

Grabisch M. (1996). The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research*, 89(3), 445-456.

Grangeon N., Leclaire P. & Norre S. (2011). Heuristics for the re-balancing of a vehicle assembly line, *International Journal of Production Research*, 49(22), 6609-6628.

Gu H., Schutt A., Stuckey P. J., Wallace M. G. & Chu G. (2015). Exact and heuristic methods for the resource-constrained net present value problem. *Handbook on project management and scheduling*, 1, 299–318.

Gurevsky E., Battaïa O. & Dolgui A. (2012). Balancing of simple assembly lines under variations of task processing times. *Annals of Operations Research*, 201(1), 265-286.

Gurevsky E., Hazir Ö., Battaïa, O. & Dolgui A. (2013). Robust balancing of straight assembly lines with interval task times. *Journal of the Operational Research Society*, 64(11), 1607-1613.

Guillaume R., Houé R. & Grabot B. (2014). Robust competence assessment for job assignment. *European Journal of Operational Research*, 238(2), 630-644.

Habibi F., Barzinpour F. & Sadjadi S.J. (2018). Resource-constrained project scheduling problem: review of past and recent developments. *Journal of Project Management* 3, 55-88.

Hanzalek Z. & Sucha P. (2017). Time symmetry of resource constrained project scheduling with general temporal constraints and take-give resources. *Annals of Operations Research*, 248, 209–237.

Hariga M., Shamayleh A. & El-Wehedi F. (2019). Integrated time–cost tradeoff and resources leveling problems with allowed activity splitting. *International Transactions in Operational Research*, 26(1), 80–99.

Hartmann S. et Briskorn D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 297(1), 1-14.

Hashemi-Petroodi S.E., Dolgui A., Kovalev S. & Kovalyov M.Y. (2021). Workforce reconfiguration strategies in manufacturing systems: a state of the art. *International Journal of Production Research*, 59(22), 6721-6744.

Hazir Ö., Haouari M. & Erel E. (2015). Robust optimization for the discrete timecost tradeoff problem with cost uncertainty. *Handbook on project management and scheduling*, 2, 865–874.

Hazir Ö. & Ulusoy G. (2020). A classification and review of approaches and methods for modelling uncertainty in projects. *International Journal of Production Economics*, 223, 107522.

Herroelen W. & Leus R. (2003). The construction of stable project baseline schedules. *European Journal of Operational Research*, 156, 550–565.

Herroelen W. & Leus R. (2005). Project scheduling under uncertainty – survey and research potentials. *European Journal of Operational Research*, 165, 289–306.

Hosseinian A., Baradaran V. & Bashiri M. (2019). Modeling of the time-dependent multi-skilled rcpsp considering learning effect: An evolutionary solution approach. *Journal of Modelling in Management*, 14(2), 521–558.

Ivanov D., Dolgui A., Sokolov B. & Ivanova M. (2017). Literature review on disruption recovery in the supply chain. *International Journal of Production Research*, 55(20), 6158-6174.

Kadri R. L. & Boctor F. F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, 265(2), 454–462.

Kellenbrink C. & Helber S. (2015). Scheduling resource-constrained projects with a flexible project structure. *European Journal of Operational Research*, 246(2), 379–391.

Kizilay D. & Çil Z. A. (2021). Constraint programming approach for multi-objective two-sided assembly line balancing problem with multi-operator stations. *Engineering Optimization*, 53(8), 1315-1330.

Kreter S., Rieck J. & Zimmermann J. (2016). Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars. *European Journal of Operational Research*, 251(2), 387–403.

Kreter S., Schutt A. & Stuckey P. J. (2017). Using constraint programming for solving RCPSP/max-cal. *Constraints*, 22, 432–462.

Kuster J., Jannach D. & Friedrich G. (2010). Applying Local Rescheduling in response to schedule disruptions. *Annals of Operations Research*, 180(1), 265-282.

Labreuche C. & Grabisch M. (2003). The Choquet integral for the aggregation of interval scales in multicriteria decision making. *Fuzzy Sets and Systems*, 137(1), 11-26.

Lambrechts O., Demeulemeester E. & Herroelen W. (2008). Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *Journal of Scheduling*, 11(2), 121-136.

Lambrechts O., Demeulemeester E. & Herroelen W. (2011). Time slack-based techniques for robust project scheduling subject to resource uncertainty. *Annals of Operations Research*, 186(1), 443-464.

Laurent A. (2018). Proposition et étude d'une extension du RCPSP pour la mutualisation entre plusieurs sites : définition, formalisation, méthodes exactes et métaheuristiques. Thèse, hal-id : tel-01886901.

Leifer R., O'connor G.C. & Rice M. (2001). Implementing radical innovation in mature firms: the role of hubs. *Academy of Management Perspectives*, 15(3), 102–113.

Li S., Zhang Z., Yan X., Hu K. & Zhao S. (2018). Research on a resource-constrained project scheduling problem in a hazardous environment and its staffing strategies based on PSO algorithm. *Systems Science & Control Engineering*, 6(1), 304–318.

Liang T.F. (2010). Applying fuzzy goal programming to project management decisions with multiple goals in uncertain environments. *Expert Systems with Applications*, 37(12), 8499–8507.

Liang Y., Cui N., Hu X. & Demeulemeester E. (2020). The integration of resource allocation and time buffering for bi-objective robust project scheduling. *International Journal of Production Research*, 58(13), 3839-3854.

Lin J., Zhu L. & Gao K. (2020). A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Expert Systems with Applications*, 140, 112915.

Lovato D., Guillaume R., Thierry C. & Battaïa O. (2020). Etude des critères d'optimisation pour la replanification des chaînes d'assemblage aéronautiques. *International Conference on Modeling, Optimization and Simulation - MOSIM 2020*. hal-03015008.

Lovato D., Guillaume R., Thierry C. & Battaïa O. (2022a). Managing disruptions in aircraft assembly lines with staircase criteria. *International Journal of Production Research*.

Lovato D., Guillaume R., Thierry C. & Battaïa O. (2022b). Aircraft final assembly line planning with staircase criteria and equity criteria. *IFAC-PapersOnLine*, Elsevier, 55(10), 860-865.

Manceaux A., Bril El-Haouzi H., Bentaha Mohand L. & Wahiba Ramdane C.-K. (2016). Iterated Local Search for dynamic assembly line rebalancing problem. *IFAC-PapersOnLine*, Elsevier, 49(12), 515-519.

Manceaux A., Bril El-Haouzi H., Thomas A. & Pétin J.-F. (2014). Dynamic Rebalancing of an Assembly Line with a Reachability Analysis of Communicating Automata. *IFIP International Conference on Advances in Production Management Systems (APMS)*, Sep 2014, Ajaccio, France. 597-604.

McAtamney L. & Corlett E. N. (1993). RULA: a survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, 24(2), 91-99.

Maghsoudlou H., Afshar-Nadjafi B. & Niaki S. T. A. (2016). A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem. *Computers & Chemical Engineering*, 88, 157–169.

Michels A. S., & Costa A. M. (2021). Conserving workforce while temporarily rebalancing assembly lines under demand disruption. *International Journal of Production Research*, 60(21), 6616-6636.

Miller R. & Lessard D. (2001). Understanding and managing risks in large engineering projects. *International Journal of Project Management*, 19(8), 437–443.

Okubo H., Miyamoto T., Yoshida S., Mori K., Kitamura S. & Izui Y. (2015). Project scheduling under partially renewable resources and resource consumption during setup operations. *Computers & Industrial Engineering*, 83, 91–99.

Oliveira F.S., Vittori K., Russel R.M.O. & Travassos X. L. (2012) Mixed assembly line rebalancing: a binary integer approach applied to real world problems in the automotive industry. *International Journal of Automotive Technology*, 13(6), 933–940.

Palacio J. D. & Larrea O. L. (2017). A lexicographic approach to the robust resourceconstrained project scheduling problem. *International Transactions in Operational Research*, 24(1-2), 143–157.

Polo-Mejía O., Artigues C., Lopez P. & Basini V (2020). Mixed integer/linear and constraint programming approaches for activity scheduling in a nuclear research facility. *International Journal of Production Research*, 58(23), 7149-7166.

Poppenborg J. & Knust S. (2016). Modeling and optimizing the evacuation of hospitals based on the MRCPSPP with resource transfers. *EURO Journal on Computational Optimization*, 4, 349–380.

Qi J.-J., Liu Y.-J., Jiang P. & Guo B. (2015). Schedule generation scheme for solving multi-mode resource availability cost problem by modified particle swarm optimization. *Journal of Scheduling*, 198, 285–298.

Ren Y., Lu Z. & Liu X. (2020). A branch-and-bound embedded genetic algorithm for resource-constrained project scheduling problem with resource transfer time of aircraft moving assembly line. *Optimization Letters* 14, 2161–2195.

Rossi A., Gurevsky E., Battaia O. & Dolgui A. (2016). Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations. *Discrete Applied Mathematics*, 208, 123-136.

Serin F., Mete S. & Çelik E. (2019), An efficient algorithm for U-type assembly line re-balancing problem with stochastic task times, *Assembly Automation*, 39(4), 581-595.

Servranckx T. & Vanhoucke M. (2019). A tabu search procedure for the resourceconstrained project scheduling problem with alternative subgraphs. *European Journal of Operational Research*, 273(3), 841–860.

Shahsavari A., Najafi A. A. & Niaki S. T. A. (2015). Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem. *Computers & Industrial Engineering*, 87, 4–15.

Shan S., Hu Z., Liu Z., Shi J., Wang L. & Bi Z. (2017). An adaptive genetic algorithm for demand-driven and resource-constrained project scheduling in aircraft assembly. *Information Technology and Management*, 18(1), 41-53.

Tabrizi, B. H. (2018). Integrated planning of project scheduling and material procurement considering the environmental impacts. *Computers & Industrial Engineering*, 120, 103–115.

Tao S., & Dong Z. S. (2017). Scheduling resource-constrained project problem with alternative activity chains. *Computers & Industrial Engineering*, 114, 288–296.

Tavana M., Abtahi A.R. & Khalili-Damghani K. (2014). A new multi-objective multi-mode model for solving preemptive time–cost–quality trade-off project scheduling problems. *Expert Systems with Applications*, 41(4), 1830–1846

Tirkolaee E. B., Goli A., Hematian M., Sangaiah A. K. & Han T. (2019). Multi-objective multi-mode resource constrained project scheduling problem using pareto-based algorithms. *Computing*, 101, 547–570.

Tritschler M., Naber A. & Kolisch R. (2017). A hybrid metaheuristic for resourceconstrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, 262(1), 262–273.

T'kindt V. & Billaut J.-C. (2006). Multicriteria Scheduling: Theory, Models and Algorithms. *Springer Science & Business Media*.

Van de Vonder S., Demeulemeester E. & Herroelen W. (2007). A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling*, 10(3), 195-207.

Vanhoucke M. & Coelho J. (2019). Resource-constrained project scheduling with activity splitting and setup times. *Computers & Operations Research*, 109, 230–249.

Van Peteghem V. & Vanhoucke M. (2010). A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 201(2), 409–418.

Waligóra G. (2011). Heuristic approaches to discrete-continuous project scheduling problems to minimize the makespan. *Computational Optimization and Applications*, 48, 399–421.

Wang J. (2004). A fuzzy robust scheduling approach for product development projects. *European Journal of Operational Research*. 152, 180–194.

Wang Q., Liu C. & Zheng L. (2019). A column-generation-based algorithm for a resource-constrained project scheduling problem with a fractional shared resource. *Engineering Optimization*, 52(5), 798-816.

Watermeyer K. & Zimmermann J. (2020). A branch-and-bound procedure for the resource-constrained project scheduling problem with partially renewable resources and general temporal constraints. *OR Spectrum*, 42, 427–460.

Xu J. & Feng C. (2014). Multimode resource-constrained multiple project scheduling problem under fuzzy random environment and its application to a large-scale hydropower construction project. *The Scientific World Journal*.

Zabihi S., Kahag M. R., Maghsoudlou H. & Afshar-Nadjafi B. (2019). Multi-objective teaching-learning-based meta-heuristic algorithms to solve multi-skilled project scheduling problem. *Computers & Industrial Engineering*, 136, 195–211.

Zhang J., Jia S. & Diaz E. (2018a). Dynamic monitoring and control of a critical chain project based on phase buffer allocation. *Journal of the Operational Research Society*, 69(12), 1966-1977.

Zhang Y., Hu X. & Wu C. (2018b). A modified multi-objective genetic algorithm for two-sided assembly line re-balancing problem of a shovel loader, *International Journal of Production Research*, 56(9), 3043-3063.

Zhu G., Bard J. & Yu G. (2005). Disruption management for resource-constrained project scheduling. *Journal of the Operational Research Society*, 56(4), 365-381

Zhu G., Bard J.F. & Yu G. (2007). A two-stage stochastic programming approach for project planning with uncertain activity durations. *Journal of Scheduling*, 10, 167–180.

Zhu H., Lu Z., Lu C. & Ren Y. (2020). A reactive scheduling method for disturbances in aircraft moving assembly line. *International Journal of Production Research*, 59(15), 4756-4772.

Zhu X., Ruiz R., Li S. & Li X. (2017). An effective heuristic for project scheduling with resource availability cost. *European Journal of Operational Research*, 257(3), 746–762.

Résumé

Le secteur aéronautique évolue dans un contexte de concurrence croissante (globalisation des chaînes logistiques, volatilité des marchés, pression sur les indicateurs clés de performance coûts-délais-qualité, apparition de nouveaux acteurs). La complexité des produits, des processus et des systèmes de production rend la planification des ressources dans les chaînes d'assemblage aéronautiques de plus en plus difficile tout en exigeant une meilleure réactivité face aux aléas. Cette thèse, réalisée dans le cadre du projet Per4mance, financé par l'ANR, présente des approches de planification face aux aléas lourds.

Ces approches font appel à la programmation par contraintes afin d'exploiter au mieux les caractéristiques du problème industriel et permettre sa résolution dans un temps acceptable. Des outils issus de l'optimisation multicritère sont mis à contribution afin de rechercher des solutions correspondant aux priorités des décideurs.

Etant donné la forte composante humaine dans les chaînes d'assemblage aéronautiques, des critères issus des méthodes ergonomiques ont été intégrés dans le processus décisionnel afin d'améliorer les conditions de travail des compagnons. Un critère exploitant la méthode RULA et prenant en compte l'équité de la distribution des tâches entre les compagnons a été développé.

Enfin, les contraintes organisationnelles qui représentent les préférences des planificateurs ont été intégrées dans le modèle en utilisant l'intégrale de Choquet. Ainsi, ce modèle permet de gérer le compromis entre satisfaction de ces contraintes et la qualité de la fonction objectif.

L'ensemble des approches développées ont été validées sur des instances industrielles. Ainsi cette thèse fournit un ensemble d'outils d'aide à la décision efficaces et opérationnels pour la gestion des risques économiques et ergonomiques dans les chaînes d'assemblage aéronautiques.

Mots clefs : *Incertitude, Replanification, Ligne d'Assemblage Aéronautique, Ergonomie, Programmation Par Contraintes*

Developing new criteria for tasks' scheduling in an aircraft assembly line

The aeronautical sector is evolving in a context of increasing competition (globalisation of supply chains, market volatility, pressure on key performance indicators cost-delay-quality, appearance of new players). The complexity of products, processes and production systems makes resource planning in aeronautical assembly lines more and more difficult while requiring a better reactivity to hazards. This thesis, carried out within the framework of the Per4mance project, financed by the French National Research Agency (ANR), presents approaches for planning in the face of heavy hazards.

These approaches use constraint programming in order to make the best use of the characteristics of the industrial problem and to allow its resolution in an acceptable time. Tools from multicriteria optimisation are used to find solutions corresponding to the priorities of decision-makers.

Given the strong human component in aeronautical assembly lines, criteria derived from ergonomic methods have been integrated into the decision-making process in order to improve the working conditions of operators. A criterion exploiting the RULA method and taking into account the fairness of the distribution of tasks between operators was developed.

Finally, the organisational constraints representing the planners' preferences were integrated into the model using the Choquet integral. Thus, this model makes it possible to manage the trade-off between satisfaction of these constraints and the quality of the objective function.

All the approaches developed have been validated on industrial instances. Thus, this thesis provides a set of efficient and operational decision support tools for the management of economic and ergonomic risks in aeronautical assembly lines.

Keywords : *Uncertainty, Rescheduling, Aeronautic Assembly Line, Ergonomy, Constraint Programming*