République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ DE BATNA 2

Faculté de Technologie

Département de Génie Industriel

Laboratoire d'Automatique et Productique

THÈSE

Présentée par

TOUFIK BENTRCIA

En vue d'obtenir le grade de

DOCTEUR EN SCIENCES

Thème

Ordonnancement des systèmes de production sous contraintes technologiques ou contextuelles: Modélisation, étude de complexité et approches intégrées de résolution

Sous la direction de : **Prof. Leila Hayet MOUSS**

Jury

Prof. Kinza Nadia MOUSS,	Université de Batna 2, Algérie	Président
Prof. Daoud AIT-KADI,	Université Laval, Canada	Membre
Prof. Mohammed MOSTEFAI,	Université de Sétif 1, Algérie	Membre
Prof. Leila Hayet MOUSS,	Université de Batna 2, Algérie	Directrice de thèse
Prof. Zaki SARI,	Université de Tlemcen, Algérie	Membre
Prof. Farouk YALAOUI,	Université de Technologie de Troyes, France	Membre

Thèse soutenue publiquement le 22 Octobre 2017

THÈSE


بسم الله الرحمن الرحيم
وعلمك ما لم تكن تعلم
وكان فضل الله عليك عظيما

Remerciements

Louange à Allah, que la paix et la bénédiction soient sur son Prophète Mohammed, ainsi que sur sa famille et ses Compagnons.

En respectant la consigne du prophète (Paix et bénédiction sur lui) qui disait : « Celui qui ne remercie pas les gens n'aura pas remercié Allah.», je voudrais adresser mes plus vifs remerciements à certaines personnes. Franchement, la présente thèse n'aurait pas vu le jour sans leurs support et orientation.

Ce travail de thèse a été réalisé au sein du Laboratoire d'Automatique et Productique à l'Université de Mostefa Benboulaïd Batna 2.

Mes tous premiers remerciements vont naturellement à Madame Leila Hayet Mouss, Professeur à l'Université Batna 2 et directrice de cette thèse, pour ses conseils pertinents et sa disponibilité.

Je tiens à remercier Madame Nadia Kinza Mouss, Professeur à l'Université Batna 2, d'avoir accepté la lourde responsabilité de présider ce jury.

Je remercie vivement Monsieur Daoud Ait-Kadi, Professeur à l'Université Laval Canada, d'avoir accepté d'être parmi nous. Je suis vraiment honoré par sa présence dans mon jury.

Je tiens à exprimer ma gratitude à Monsieur Mohammed Mostefai, Professeur à l'Université Sétif 1, qui a bien voulu examiner ce travail malgré ses nombreuses occupations.

Je remercie également Monsieur Zaki Sari, Professeur à l'Université de Tlemcen, pour sa gentillesse et l'intérêt qu'il a porté à mon travail.

Je suis particulièrement reconnaissant à Monsieur Farouk Ya-

laoui, Professeur à l'Université de Technologie de Troyes France, pour les discussions fructueuses et pour l'acceptation de participer à ce jury.

Je garde une place spéciale dans mes remerciements à mes proches notament ma famille et mes amis pour leurs soutient et encouragement, comme d'habitude !!!. Peu importe l'ordre de mes remerciements, tous ceux que j'ai mentionné et sans oublier les personnes dans l'ombre m'ont apporté tout au long des années passées une aide précieuse et décisive pour l'aboutissement à cette thèse.

Dédicaces

À tous ceux que j'aime et qui m'aiment, je dédie ce modeste travail.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

« *Le début c'est la partie la plus importante du travail* »

Aristoclès Platon

Introduction générale

De nos jours, les tendances actuelles dans le domaine industriel sont orientées vers l'amélioration des performances des systèmes manufacturiers en termes d'adaptation rapide vu les fluctuations du marché et les perturbations internes. Une telle vision est étroitement sollicitée par la mondialisation exacerbant une concurrence acharnée désormais internationale.

En effet, la production devient de plus en plus diversifiée et les entreprises cherchent régulièrement à accroître leurs parts de marché ainsi que les marges de profit. Sous ces conditions, la fonction ordonnancement est considérée incontestablement dans la hiérarchie décisionnelle de l'entreprise comme un pilier solide pour prendre les décisions les plus appropriées tant pour les investisseurs et producteurs que pour les consommateurs.

Les problèmes d'ordonnancement à une machine constituent une cible très appréciée par la communauté des chercheurs malgré que cette structure connaisse peu de débauchées directes en pratique. Une telle focalisation sur ce type de problèmes peut être argumentée par la possibilité d'utiliser les modèles résultants pour implémenter des extensions valables à d'autres classes particulières de problèmes d'ordonnancement. Par ailleurs, la formalisation des problèmes d'ordonnancement à une machine est basée sur deux paradigmes de modèles largement utilisés : les approches déterministes et les approches stochastiques. Dans le premier cas, les valeurs associées aux paramètres des tâches sont des données précises permettant la simplification du cadre de description du problème d'ordonnancement. Cependant, cette vision déterministe est trop restrictive dans le sens où les valeurs de ces paramètres sont mal connues et même avec la consultation de l'avis des experts, il est fort probable que les informations obtenues soient sujettes à de nombreuses altérations. Pour les modèles élaborés sur une base stochastique, les paramètres des tâches sont exprimés par des distributions de probabilité. Ceci résulte souvent en des formulations très difficiles à manipuler. En outre, l'hypothèse stochastique peut être mise en doute car dans plusieurs situations les données statistiques sont manquantes. Aussi, la recherche de formalismes plus fiables devient nécessaire et l'idée la plus naturelle semble d'opter pour des outils mathématiques dédiés à la représentation d'incertitude dans le contexte de la théorie d'ordonnancement.

Dans cette thèse, nous nous intéressons à la résolution de problèmes d'ordonnancement à une machine avec la prise en compte des sources d'incertitude au niveau du système de production. Sous telle condition, la modélisation et l'optimisation de l'exécution des tâches par les approches d'ordonnancement conventionnelles sont remises en question. Notre objectif réside dans un premier temps de proposer des approches basées sur la théorie des nombres flous pour une représentation fidèle des problèmes d'ordonnancement avec paramètres incertains et par la suite d'implémenter des méthodes d'optimisation efficaces. Cette intégration entre la modélisation et l'optimisation se révèle une méthodologie intéressante pour remédier à la complexité des problèmes traités.

Contribution de la thèse

Les travaux de recherche présentés dans cette thèse portent particulièrement sur les problèmes d'ordonnancement à une machine tenant compte des paramètres incertains. Plusieurs approches de résolution sont implémentées notamment les métaheuristiques à base de solution unique et à base de population de solutions. Nos contributions portent sur plusieurs aspects à savoir la proposition de modèles d'ordonnancement, analyse de

la complexité de ces problèmes et finalement le développement d'approches efficaces de résolution. Tandis que les deux premières contributions sont d'ordre théorique alors que la troisième se situe dans un contexte pratique. Ces trois contributions sont illustrées en détail dans ce qui suit :

Nouveaux modèles d'ordonnancement :

Cette contribution concerne la généralisation des modèles conventionnels d'ordonnancement sur une machine dont l'objectif est d'élaborer une méthodologie cohérente pour supporter des données incertaines. En d'autres termes, nous proposons des modèles d'ordonnancement capables de refléter les différentes contraintes contextuelles et technologiques existantes dans les systèmes de production actuels. Il est à mentionner que ces modèles sont inspirés des modèles classiques à une machine disponibles dans la littérature auxquels s'ajoutent les propriétés suivantes :

- Paramètres d'ordonnancement incertains : Les paramètres correspondants aux tâches comme les dates échues ou les durées opératoires ne sont pas connus d'une façon exacte à cause des sources d'incertitude internes et externes au système ;
- Effet d'apprentissage : La position de la tâche dans la séquence d'ordonnancement affecte sa durée opératoire qui est réduite à cause de l'exécution répétée de tâches similaires ;
- Effet d'actualisation : Les coûts d'ordonnancement sont actualisés à un taux fixe qui est le cas des coûts de stockage ou de possession.

Analyse de la complexité des problèmes d'ordonnancement :

Comme l'analyse de la complexité permet de déceler la classe du problème d'ordonnancement notamment les classes dites facile et difficile, nous procédons à une étude chaque fois qu'un modèle est développé. À la base de cette analyse, le choix de la méthode de résolution est entamé.

Approches de résolution :

Selon la nature du modèle d'ordonnancement, des algorithmes polynômiaux ou des méthodes d'optimisation sont proposés pour résoudre les problèmes obtenus. Ces approches de résolution sont implémentées à la base des théorèmes de complexité prouvés. L'ensemble des outils développés peut fournir aux gestionnaires une plateforme performante destinée à l'ordonnancement sous incertitude afin de satisfaire les besoins des clients dans les plus brefs délais et ainsi améliorer la position concurrentielle de l'entreprise sur le marché.

Organisation de la thèse

Notre travail de thèse est structuré en quatre chapitres. Le premier chapitre est réservé à l'état de l'art des problèmes d'ordonnancement conventionnels et les trois chapitres suivants sont destinés à la présentation de nos contributions.

Dans le premier chapitre, nous essayons de positionner notre travail de recherche dans le cadre de l'ordonnancement des ateliers de production. Nous présentons d'abord l'importance de la fonction ordonnancement au sein du système décisionnel de l'entreprise. Nous rappelons ensuite les différentes composantes élémentaires constituant un problème d'ordonnancement. Nous dévoilons par la suite la classification des problèmes d'ordonnancement ainsi que les outils graphiques de base nécessaires à la représentation des solutions. Nous focalisons nos intérêts sur la criticité des contraintes lors du processus de modélisation. Nous donnons un aperçu global sur la théorie de complexité pour

bien cerner les classes des problèmes d'ordonnancement dépendamment du coût de résolution. Finalement, nous clarifions les principes sur lesquels sont fondées les grandes catégories des approches d'optimisation.

Nous nous intéressons dans le deuxième chapitre à la nécessité de la mise en oeuvre de critères adéquats dans le contexte de la théorie de possibilité pour l'évaluation de l'optimalité d'une séquence fixe dans l'ordonnancement à une machine avec des paramètres incertains et coûts actualisés. Aussi, nous dressons une présentation de la terminologie de la théorie de possibilité. Nous généralisons par la suite le problème conventionnel d'ordonnancement à une machine avec coûts actualisés au cas flou où plusieurs propriétés sont proposées et prouvées. Nous introduisons deux méthodes approchées : algorithme génétique et recherche par motifs, pour la résolution de la version floue du problème traité. Nous clôturons ce chapitre en exposant les résultats des simulations numériques où la robustesse des deux méthodes est validée statistiquement.

Dans le troisième chapitre, nous considérons le problème d'ordonnancement à une machine avec des paramètres incertains et effet d'apprentissage à base de position. L'objectif est de minimiser la somme pondérée des temps d'achèvement des tâches. Par la suite et en raison des données mal connues dans le modèle, nous proposons deux procédures de résolution pour le problème d'ordonnancement flou résultant. La première procédure est basée sur l'extension de la règle de Smith, ce qui donne un algorithme polynômial avec une complexité $\mathrm{O}\left(n \times \log(n)\right)$. Cependant, la contrainte d'agréabilité floue doit être satisfaite dans ce cas. La deuxième procédure basée sur des méthodes d'optimisation repose sur l'hypothèse d'existence des dates de disponibilité floues différentes, en plus de l'élimination de la contrainte d'agréabilité floue. Nous implémentons et appliquons trois métaheuristiques à base de trajectoire notamment le recuit simulé, la recherche tabou et la recherche kangourou pour résoudre le problème considéré. Pour les méthodes proposées tout au long du chapitre, plusieurs expérimentations numériques conjointement avec des déductions statistiques sont illustrées.

Dans le dernier chapitre, nous abordons un problème d'ordonnancement bi-objectif à une machine dans lequel la ressource subit l'effet d'apprentissage à base de position. Nous supposons que les durées opératoires et les dates d'échéance sont représentées par des nombres flous plats (intervalles flous) d'allure parabolique. Nous développons des formules compactes pour exprimer les fonctions objectifs élaborées à la base des concepts de classement et de possibilité, où notre objectif est de minimiser la magnitude de la somme pondérée des temps d'exécution, simultanémen avec la somme pondérée des possibilités de retard. Nous montrons ensuite que ce problème est NP-difficile au sens fort. Par conséquent, nous proposons un algorithme génétique de tri non dominé (NSGA II) et la recherche coucou (CS) comme outils de résolution. Nous intégrons dans ces métaheuristiques deux types de critères de dominance à savoir la règle de domination de Pareto et la règle de domination de Lorenz. Nous examinons les mesures de performance statistiques pour affirmer que l'approche d'optimisation coucou proposée est plus efficace et très compétitive en termes d'identification du front et diversité des solutions. Enfin, nous terminons cette thèse par une conclusion générale concernant les modèles proposés, les résultats de complexité associés et les performances des approches de résolutions implémentées suivies de quelques perspectives pour tracer nos directions futures de recherche.

Chapitre 1

Problèmes d'ordonnancement dans les systèmes de production

« *Details create the big picture* »

Sanford I. Weill

Sommaire

1.1 Introduction

Avec l'évolution accrue de l'environnement caractérisé par des marchés affectés fortement par la concurrence et la sévérité des attentes de la clientèle, la nécessité de reformuler les outils d'aide à la décision pour les gestionnaires devient de plus en plus une exigence urgente afin de remédier aux limites des disciplines traditionnelles [MARTINSONS et DAVISON, 2007]. L'ordonnancement occupant une position focale dans le système de gestion informatisée des flux de production constitue la liaison vitale entre deux niveaux de la décision hiérarchique au sein de l'entreprise : le niveau tactique et le niveau opérationnel. En effet, la fonction ordonnancement assure le pilotage de l'ensemble des ressources disponibles avec la résolution des conflits causés par la réalisation des gammes de produits sur l'horizon de temps estimé. La fonction ordonnancement est délimitée au sein d'une structure hiérarchisée par les fonctions planification en amont et lancement en aval. L'estimation des besoins en composants ainsi que la détermination des plans d'approvisionnement et de charge sont assurées par la fonction planification. La fonction lancement est responsable de la validation des ordres de fabrication à travers le regroupement et le dimensionnement de ces ordres en lots de production. Un mécanisme de contrôle vertical est adopté pour garantir la consistance des décisions entre différents niveaux. Par exemple, la non faisabilité du plan de production au niveau ordonnancement vis-à-vis les volumes demandés peut être justifiée comme due à une estimation erronée des capacités des ressources dans les ateliers [ESQUIROL et LOPEZ, 1999].

En comparaison avec les systèmes de production basés sur un paradigme de production mono produit ou bien gérée par la méthode kanban pour lesquels les flux de produit s'écoulent naturellement et doivent en principe s'autoréguler, la vision d'ordonnancement joue un rôle crucial dans tous les ateliers traitants des produits divers en flux poussé [KAABI-HARRATH, 2004]. Ceci peut être interprété par le fait que les gestionnaires dans de tels ateliers sont obligés de manipuler des informations relatives à différentes gammes de production simultanément avec l'objectif de maximiser le profit. De plus, comme la production dans un marché très compétitif est gouvernée par les ordres de fabrication issus de la fonction planification, la contrainte temps doit être impérativement considérée dans le contexte de la décision. Cependant, il est à noter que l'importance du concept d'ordonnancement n'est pas limitée uniquement à l'aspect industriel car un nombre impressionnant de problèmes d'ordonnancement existe dans d'autres domaines parmi lesquels nous pouvons citer :

- Le domaine informatique : le noyau d'un système d'exploitation contient la composante ordonnanceur pour l'utilisation maximale des capacités de l'environnement telles que des processeurs en parallèle ou bien le respect des contraintes temps réel dans les systèmes embarqués ;
- Le domaine hospitalier : l'ordonnancement des salles opératoires vise à la réalisation d'interventions sous des conditions de travail adéquates pour les ressources humaines (infirmiers, anesthésistes, chirurgiens...) et techniques (instruments médicaux-chirurgicaux, salles d'opération et de réveil...) ;
- Le domaine de transport : l'ordonnancement permet d'avoir les itinéraires d'une flotte de véhicules de transport pour satisfaire l'ensemble de requêtes des clients (problème de tournées de véhicule) ;
- Le domaine de l'éducation : avec la croissance rapide du nombre d'étudiants d'une année à l'autre, il est nécessaire d'exploiter d'une façon optimale les ressources pédagogiques. Des contraintes particulières sont souvent présentes dans ce cadre par exemple les préférences des enseignants.

Dans tous ces cas, la réalisation d'un travail exprimé par une série de tâches interdépendantes nécessite un mécanisme de coordination pour implémenter ces tâches (allocation des ressources adéquates).

Ce premier chapitre est réservé à la présentation des concepts de base concernant les problèmes d'ordonnancement. Dans ce contexte, nous faisons notre mieux pour aborder les différents aspects du thème d'une façon formelle sans lier ces notions à un domaine bien particulier. Ce chapitre est divisé en neuf sections. La Section 1.2 est destinée à la présentation du contexte de l'ordonnancement au sein de l'entreprise. La Section 1.3 décrit le problème d'ordonnancement notamment la définition, les notations et les objectifs d'optimisation. Les différentes typologies des problèmes d'ordonnancement ainsi que les outils de modélisation associés sont illustrés dans les Sections 1.4 et 1.5. L'importance des contraintes dans la modélisation des problèmes d'ordonnancement est accentuée dans la Section 1.6. La complexité des problèmes d'ordonnancement à une machine est traitée sous formulation standard et avec considération de quelques contraintes dans la Section 1.7. Les grandes familles des approches de résolution des problèmes d'ordonnancement sont exposées d'une façon succincte dans la Section 1.8. Quelques remarques et conclusions sont délivrées dans la Section 1.9.

1.2 Contexte de l'ordonnancement

De point de vue organisationnel de l'entreprise, l'ordonnancement forme une partie indispensable du système de contrôle et de planification de la production. Ce dernier régit la complexité des flux d'information dans le processus de prise de décision. Généralement, tels systèmes sont décomposés en des modules effectuant des fonctions différentes comme la planification des besoins en matière première [STORPER et HARRISON, 1991]. Ces structures hiérarchisées sont caractérisées par des décisions grossières à horizon temporel long au sommet. En revanche, le degré de détail des informations augmente en parallèle avec le raccourcissement des horizons temporels en allant vers les niveaux bas. Par conséquent, l'évolution de la prise de décision est concrétisée par un affinement successif lors du passage d'un niveau supérieur à un niveau inférieur dans l'hiérarchie donnant ainsi plus de réactivité au système décisionnel.

Cette approche de décomposition hiérarchique des systèmes de production est connue sous le nom de Planification de la Production Hiérarchique (en anglais HPP) et permet de déceler entre les niveaux stratégiques de décision dans une entreprise. Par exemple, Bertrand et ses collègues ont distingué entre le contrôle des flux de biens concernant les décisions de planification et de contrôle à l'échelle de l'entreprise et le contrôle de l'unité de production relatif aux décisions de planification et de contrôle à l'échelle des ateliers de fabrications [BERTRAND et collab., 1990]. Cependant, le contrôle des flux de biens supervise également la coordination entre les différents ateliers de fabrication.

Le paradigme de planification de la production hiérarchique a été largement utilisé et reconnu comme une méthodologie fiable de planification et de contrôle par plusieurs entreprises de moyenne et grande taille. L'adoption de l'objet de décision comme critère de classification conduit à distinguer : les décisions stratégiques concernant la manière dont l'entreprise gère sa position sur le marché, les décisions tactiques portant sur l'élaboration de plans pour parvenir aux objectifs prévus et les décisions opérationnelles s'appliquant dans le cadre du processus productif pratique au sein des ateliers [ANTHONY, 1965]. Ce type de décision est réversible via des actions correctives rapides et bénéfiques la plus part du temps.

Le type d'informations requis pour la prise de décision dans une organisation dépend directement du niveau décisionnel et de la structuration des situations de décision confrontées. Il est essentiel de comprendre que le contexte classique de gestion sous forme pyramidale reste valable pour les organisations d'aujourdui. Le processus de décision est toujours structuré à la base de niveaux hiérarchique mais avec modification de quelques attributs comme la forme et les participants qui continuent de changer avec l'évolution des structures organisationnelles. L'interdépendance entre les trois niveaux décisionnels et les types d'informations est illustrée sur la Figure 1.1 ([O'BRIEN et MARAKAS, 2011]).

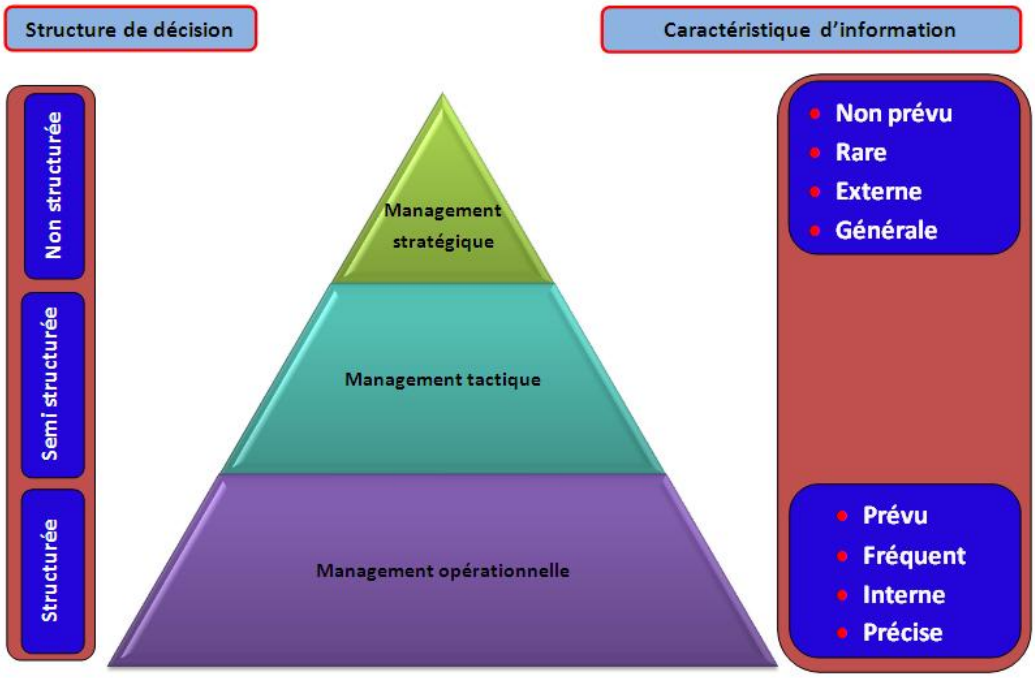

FIGURE 1.1 Dépendence entre le niveau de décision et le type d'information.

La décomposition basée sur la planification de la production hiérarchique permet d'associer aux tâches des attributs de privilège tels que la visibilité, l'autorité et la responsabilité. Malheureusement, l'adoption de cette décomposition a engendré dans certains cas des entreprises statiques et non flexibles face aux fluctuations de l'environnement. Ceci est dû au processus d'ordonnancement structuré d'une manière hiérarchique indépendamment de l'adaptabilité de ce choix au système de production [BITRAN et TIRUPATI, 1989].

Le gap entre la théorie et la pratique de l'ordonnancement a été signalé depuis les années soixante et a été discuté par plusieurs chercheurs. Plusieurs causes probables sont à l'origine de ce gap et contribuent aux difficultés rencontrées lors de l'amélioration des procédés de production. L'une de ces causes peut être due au manque de concordance entre le facteur humain et les facteurs formels intervenant dans les méthodes et les systèmes d'ordonnancement. En effet, les procédures de contrôle et de prise de décision dans une entreprise réelle contiennent nécessairement des actions humaines. Le facteur humain est présent dans plusieurs situations telles que l'affectation des ressources appropriées, la spécification des valeurs initiales aux paramètres de configuration d'un algorithme ou bien l'interprétation d'un plan issu d'un logiciel. Il est quasiment impossible de penser à des situations de nos jours où l'implémentation des approches de contrôle

et d'optimisation est auto-réglante/auto-adaptative aux contextes des marchés. Il est à mentionner que les efforts destinés à l'analyse de l'impact des facteurs humains sur les modèles d'ordonnancement par rapport aux aspects mathématiques et informatiques sont relativement faibles en termes des études publiées ([McKay et Wiers, 1999] et [McKay et Wiers, 2006]).

Dans le but d'analyser les types d'informations utilisés par l'ordonnanceur pendant le test de la faisabilité opérationnelle des ordonnancements, McKay a capturé lors de la période d'étude approximativement plus de 250 décisions non quotidiennes à prendre. Ces décisions concernent les considérations non traditionnelles liées aux tâches, ressources et temps. L'ordonnanceur dans l'entreprise utilise plusieurs types d'informations pour la prise de décision. En outre, l'information est manipulée sous plusieurs facettes : collection, augmentation, compression...etc. Dans ce contexte, certaines informations sont employées comme des signaux pour contrôler les activités de traitement des informations secondaires. Les différentes catégories des informations utilisées par l'ordonnanceur sont résumées sur le Tableau 1.1 ([McKay, 1992]).

Tableau 1.1 Informations utilisées dans le processus d'ordonnancement

Catégorie	Exemples d'information utilisée
Humain	Expérience/compétence, motivation, absence et autres caractéristiques individuelles des opérateurs, ingénieurs, vendeurs, fournisseurs, clients, transporteurs, techniciens, sous traiteurs
Organisation	Objectifs, procédures, responsabilité politique, potins
Resource	Capacité, flexibilité, sûreté, coût, location, état de maintenance, modes d'opération (manuel ou automatique), âge, et la sensibilité de : machines, outils, fixations, personnel, équipement de transport, tampons, palettes, sous traiteurs
Matériel	Date d'échéance, quantité requise, client, qualité, temps de traitement, âge, spécification, programme de commande numérique par calculateur, factures de matériel, routage, stabilité, taille du lot, niveau de stock, risque

La diversité des informations illustrées sur ce tableau montre clairement la difficulté des obstacles à contourner dans le cadre des hypothèses simplifiées de la théorie d'affectation classique. L'accent doit être mis sur le fait que la majorité du temps et d'effort de l'ordonnanceur est allouée au contrôle anticipé basé sur l'évaluation des risques et la reformulation des activités difficiles à assimiler. Pour pallier ces limites, la fonction ordonnancement dans un système de production doit interagir d'une façon dynamique avec les autres fonctions. Cette interaction dépend du système et s'adapte continûment avec le changement des circonstances. Un système de production moderne possède souvent un système d'information élaboré qui contient un calculateur central et une base de données. De plus, des réseaux locaux d'ordinateurs personnels, stations de travail et terminaux d'accès connectés au calculateur central sont utilisés pour la récupération des données existantes ou l'insertion des nouvelles informations dans la base de données. Le logiciel exploité pour la gestion du système d'information est nommé Système de Planification des Ressources de l'Entreprise (en anglais ERP). Ce système joue le rôle d'une passerelle d'informations traversant l'entreprise à tous les niveaux organisationnels pour relier les différentes composantes d'aide à la décision. Par conséquent, l'ordonnancement est

souvent achevé interactivement via un système d'aide à la décision installé sur un ordinateur personnel ou station de travail connectée au système de planification des ressources de l'entreprise. Les terminaux distribués à des points fixes donnent aux départements de l'entreprise un accès sécurisé en temps réel à toutes les informations d'ordonnancement. D'autre part, ces départements alimentent le système d'ordonnancement avec les informations récentes relatives aux états des tâches et des machines.

1.3 Description du problème d'ordonnancement

Pour illustrer les différents aspects de l'ordonnancement, nous abordons dans ce qui suit les concepts, définitions, notations et objectifs d'optimisation relatifs à la problématique d'ordonnancement.

1.3.1 Définition du terme ordonnancement

Due à la disponibilité d'un grand nombre de définitions de l'ordonnancement dans la littérature, nous présentons uniquement trois définitions brèves mais typiques comme illustré ci-dessous :

Définition 1.1. *Ordonnancer un ensemble de tâches, c'est programmer leur exécution en leur allouant les ressources requises et en fixant leur date de début [GOTHA, 1993].*

Définition 1.2. *Le problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchainement,...) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises [ESQUIROL et LOPEZ, 2001].*

Définition 1.3. *L'ordonnancement concerne l'allocation de ressources limitées aux activités avec l'objectif d'optimiser un ou plusieurs mesures de performance [LEUNG, 2004].*

En partant de ces trois définitions, les propriétés élémentaires d'un problème d'ordonnancement découlent directement. Un problème d'ordonnancement fait évoquer généralement les points suivants :

- Existence d'un nombre fini de tâches ;
- Partage de ressources limitées entre ces tâches ;
- Détermination de la date de début d'exécution pour chaque tâche ;
- Respect des contraintes imposées sur les tâches et les ressources.

Les concepts et les notations de base sont présentés avec plus de détails dans les sous sections suivantes.

1.3.2 Concepts et notations de base en ordonnancement

La terminologie et les notations suivantes constituent le vocabulaire de base dans l'élaboration des modèles d'ordonnancement [PINEDO, 2009]. Deux concepts fondamentaux interviennent dans la description d'un problème d'ordonnancement : les tâches et les ressources. Une tâche est une entité élémentaire de travail localisée dans le temps par une date de début et/ou une date de fin, dont l'exécution nécessite un nombre d'unités de temps et d'unités de ressources exploitées/consommées avec une intensité donnée. Une

ressource est un moyen, technique (machine, processeur, piste aéronautique) ou humain, destiné à l'exécution des tâches et dont l'état de disponibilité est connu d'avance. Chaque tâche est caractérisée, lorsqu'une seule ressource est considérée, par un ensemble de données notamment :

- **Durée opératoire** (p_i) : elle représente le temps nécessaire pour la réalisation de la tâche i sur une ressource unique ou lorsque cette durée ne dépend pas de la machine. Le taux de production de la machine peut être déduit comme $Q_i = \frac{1}{p_i}$ (nombre de pièces par unité de temps). Dans le cas où nous avons plusieurs ressources, une notation à deux indices est introduite ;
- **Date de disponibilité** (r_i) : elle est connue aussi sous le nom de la date de l'état prêt. C'est la date d'arrivé de la tâche i au système et pour laquelle le démarrage du traitement de la tâche est possible ;
- **Date échue** (d_i) : elle représente la date de livraison comme promis au client et la violation de ce délai engendre une pénalité de retard. En revanche, rien n'empêche la tâche d'être exécutée après sa date d'échéance. Lorsque la date échue est une contrainte forte à satisfaire, elle est appelée la date fin impérative ;
- **Poids** (ω_i) : c'est une sorte de facteur de priorité exprimant l'importance d'une tâche par rapport aux autres dans le système. Le poids peut refléter dans la pratique un coût de séjour de la tâche dans le système pour une unité de temps ou un coût de stockage.

Comme la liste de ces paramètres ne dépend pas de la séquence d'ordonnancement dans la formulation standard, les données associées sont considérées comme statiques. Par contre, les paramètres qui ne sont pas connus au préalable, à cause de leurs dépendances de la séquence d'ordonnancement, sont identifiés à des données dynamiques. Les principaux paramètres dans cette catégorie dynamique sont :

- **Date début** (S_i) : c'est le temps où la tâche i commence son exécution sur la machine. Dans un contexte multiressource, cette date désigne le temps du premier traitement de la tâche dans le système ;
- **Date fin** (C_i) : elle exprime le temps de la fin d'exécution de la tâche i sur la machine. Dans un contexte multimachine, cette date correspond au temps où la tâche quitte le système.

1.3.3 Objectifs d'optimisation dans l'ordonnancement

Le cardinal de l'ensemble des ordonnancements tenant compte des ressources disponibles et des contraintes imposées peut croître dans certains cas d'une façon exponentielle en fonction du nombre de tâches. Afin d'exploiter cet ensemble des ordonnancements admissibles, une fonction économique (critère d'optimalité) permettant le classement de ces ordonnancements doit être introduite. Alors, il devient possible de déterminer un ordonnancement optimal dans le sens de la meilleure satisfaction du critère d'optimalité considéré. Généralement, les critères d'optimalité les plus utilisés dans la littérature font intervenir les dates de fin d'exécution des tâches (C_i) comme paramètre de base en combinaison avec d'autres paramètres auxiliaires. Des opérateurs de maximum (critère goulet) ou de somme sont adoptés pour formuler ces critères où il est envisageable sous certaines situations d'introduire une pondération pour différencier entre les priorités des tâches. Dans les Tableaux 1.2 et 1.3, nous élucidons les critères élémentaires ainsi que la procédure de création des critères d'optimalités les plus rencontrés

dans le domaine de l'ordonnancement [BRUCKER, 2007]. Dans le contexte d'une optimisation multi-objectif, il est possible de proposer d'autres critères sophistiqués utilisant l'approche somme pondérée dont le but est d'élaborer un compromis entre deux ou plusieurs fonctions objectifs intégrées dans une seule formulation.

TABLEAU 1.2 Présentation des critères élémentaires utilisés dans le domaine d'ordonnancement

Critère	Notation	Expression
Date fin	C_i	$C_i = S_i + p_i$
Temps de séjour	F_i	$F_i = C_i - r_i$
Retard algébrique	L_i	$L_i = C_i - d_i$
Retard	T_i	$T_i = \max(C_i - d_i, 0)$
Indicateur de retard	U_i	$U_i = \begin{cases} 1 & si \ \ T_i > 0 \\ 0 & sinon \end{cases}$
Avance	E_i	$E_i = \max(d_i - C_i, 0)$

TABLEAU 1.3 Classification des critères de performances utilisés dans le domaine d'ordonnancement

Maximum	Maximum pondéré	Somme	Somme pondérée
$\max\limits_{1\le i\le n}(C_i)$	$\max\limits_{1\le i\le n}(\omega_i \times C_i)$	$\sum\limits_{i=1}^{i=n} C_i$	$\sum\limits_{i=1}^{i=n}(\omega_i \times C_i)$
$\max\limits_{1\le i\le n}(F_i)$	$\max\limits_{1\le i\le n}(\omega_i \times F_i)$	$\sum\limits_{i=1}^{i=n} F_i$	$\sum\limits_{i=1}^{i=n}(\omega_i \times F_i)$
$\max\limits_{1\le i\le n}(L_i)$	$\max\limits_{1\le i\le n}(\omega_i \times L_i)$	$\sum\limits_{i=1}^{i=n} L_i$	$\sum\limits_{i=1}^{i=n}(\omega_i \times L_i)$
$\max\limits_{1\le i\le n}(T_i)$	$\max\limits_{1\le i\le n}(\omega_i \times T_i)$	$\sum\limits_{i=1}^{i=n} T_i$	$\sum\limits_{i=1}^{i=n}(\omega_i \times T_i)$
$\max\limits_{1\le i\le n}(U_i)$	$\max\limits_{1\le i\le n}(\omega_i \times U_i)$	$\sum\limits_{i=1}^{i=n} U_i$	$\sum\limits_{i=1}^{i=n}(\omega_i \times U_i)$
$\max\limits_{1\le i\le n}(E_i)$	$\max\limits_{1\le i\le n}(\omega_i \times E_i)$	$\sum\limits_{i=1}^{i=n} E_i$	$\sum\limits_{i=1}^{i=n}(\omega_i \times E_i)$

Une propriété très utile dans l'obtention de l'optimum d'un problème d'ordonnancement est attachée à la notion de régularité du critère d'optimalité introduite par Conway et ses collègues [CONWAY et collab., 1967]. La définition suivante donne une interprétation rigoureuse de cette notion.

Définition 1.4. *Soient* $C = [C_1, C_2, ..., C_n]$ *le vecteur des dates de fin des tâches et* $F_C^{opt} : C \to \mathbb{R}$ *un critère d'optimalité. Le critère est dit régulier si et seulement si pour tout autre vecteur des dates de fin* $C' = \left[C'_1, C'_2, ..., C'_n\right]$, *nous avons* $F_C^{opt}\left(C'\right) \ge F_C^{opt}(C)$ *est vérifié si et seulement si* $\exists k \in \{1, 2, ..., n\}$ *tel que* $C'_k \ge C_k$.

D'une façon équivalente, un critère est régulier si et seulement s'il est une fonction non-décroissente des dates de fin d'exécution des tâches.

Théorème 1.1. *Dans le problème standard d'ordonnancement à une machine avec critère d'optimalité régulier, les ordonnancements sans temps d'oisiveté constituent un ensemble dominant.*

Démonstration.

Soit A l'ensemble des ordonnancements sans temps d'oisiveté. Supposons que l'ordonnancement optimal n'appartient pas à cet ensemble, donc il contient au moins un temps d'oisiveté. Par hypothèse, le critère d'optimalité est régulier d'où la possibilité de réduire sa valeur pour le cas de l'ordonnancement optimal en éliminant les temps d'oisiveté par décalage des tâches localisées en aval. Mais ceci contredit l'optimalité de la solution d'où la solution optimale appartient forcement à l'ensemble A. Ce qui signifie que l'ensemble des ordonnancements sans temps d'oisiveté constitue un ensemble dominant lorsque le critère d'optimalité est régulier. □

1.4 Classification des problèmes d'ordonnancement

À cause de l'existence d'une très grande variété de problèmes d'ordonnancement, la notation à trois champs $\alpha|\beta|\gamma$ proposée initialement par Graham et ses collègues [Graham et collab., 1979] et reprise par la suite par plusieurs auteurs ([Chen et collab., 1998]) s'est rapidement émergée en tant que cadre de référence pour donner une classification claire tenant compte des caractéristiques de la plateforme et de l'environnement du problème d'ordonnancement étudié comme indiqué sur le Tableau 1.4.

Tableau 1.4 Significations et valeurs des champs dans la notation de Graham

Champ	Sous champs	Valeurs
Environnement (α)	Type de machine (α_1)	$\{\phi, 1, P, Q, R, O, F, J, FH\}$
	Nombre de machines (α_2)	$\{\phi, k\}$
Contrainte (β)	Préemption (β_1)	$\{\phi, pmtn\}$
	Ressources additionnelles (β_2)	$\{\phi, res\}$
	Précédence (β_3)	$\{\phi, prec, uan, tree, chains\}$
	Dates de disponibilité (β_4)	$\{\phi, r_i\}$
	Durées opératoires (β_5)	$\{\phi, p_i = p, p^- \leq p_i \leq p^+\}$
	Dates de fin impérative (β_6)	$\{\phi, d_i^{++}\}$
	Nombre maximum d'opérations par tâche (β_7)	$\{\phi, n_i \leq n_{\max}\}$
	Propriété d'attente (β_8)	$\{\phi, no-wait\}$
Objectif (γ)	/	Un des critères d'optimalité du Tableau 1.2

Les valeurs prises par le sous champ α_1 sont décrites dans le Tableau 1.5. Pour le champ β, uniquement les valeurs de la propriété de précédence (sous champ β_3) sont interprétées dans le Tableau 1.6 comme les valeurs affectées aux autres sous champs signifient la presence ou l'absence de la propriété considérée.

Bien que les valeurs affectées aux champs permettent de modéliser un très grand nombre de problèmes d'ordonnancement, multitudes extensions ont été proposées en

TABLEAU 1.5 Interprétation des valeurs du champ α_1

Valeur	Description
ϕ ou 1	Environement à une seule machine
P	Environement à machines identiques parallèles
Q	Environement à machines parallèles uniformes
R	Environement à machines parallèles quelconques
O	Système open shop
F	Système Flow shop
J	Système Job shop
FH	Système flow shop hybride

TABLEAU 1.6 Interprétation des valeurs du champ β_3

Valeur	Description
ϕ	Tâches indépendentes
$prec$	Tâches avec contraintes de précédence générales
uan	Tâches formant un réseau d'activités uniconnexe
$tree$	Tâches formant un arbre
$chains$	Tâches formant union de chaînes

particulier au niveau des champs β et γ dans le but de supporter d'autres catégories de problèmes particulières telles que la classe des problèmes stochastiques ou multicritères. Ce processus d'extension est un axe évolutif où il est très probable dans le futur que d'autres contraintes et extensions soient encore créées [T'KINDT et BILLAUT, 2006].

1.5 Outils de modélisation pour l'ordonnancement

Dans le domaine d'ordonnancement, plusieurs outils sont disponibles non seulement pour la représentation graphique des solutions mais également pour la spécification sans ambiguité des données et des contraintes du problème. Dans cette section, nous présontons quelques outils de base largement employés dans la littérature des problèmes d'ordonnancement.

1.5.1 Diagramme de Gantt

Malgré que les diagrammes de Gantt datent de plus d'un siècle, ils sont considérés comme outil de modélisation très populaire en gestion pendant la phase de planification, puis comme un procédé de contrôle pour valider les résultats [CLARK, 1923]. Le diagramme de Gantt est un modèle graphique simple et flexible pour représenter l'exécution des tâches sur une période de temps. Un segment horizontal de longueur proportionnelle à la durée opératoire est associé à chaque tâche avec l'hypothèse que ces durées sont connues avec certitude. Il existe deux visions pour l'élaboration de ce genre de diagrammes [WILSON, 2003] :

- Le diagramme à base de ressource où la ligne horizontale correspond à la ressource, ce qui rend possible l'illustration de ses périodes d'opération et d'oisiveté ainsi que l'ordre de passage des opérations ;

- Le diagramme à base de produit où une ligne est affectée à chaque tâche pour permettre de déterminer le chaînage de ses opérations et le temps d'attente entre deux opérations consécutives.

Il est possible également de considérer des diagrammes de Gantt dans un espace à trois dimensions avec les axes orthogonaux représentant les machines, les tâches et le temps. Ce genre de diagrammes est devenu ces dernières années plus pratique en particulier avec l'évolution d'outils dédiés de traitement graphique de données. Dans la figure 1.2, l'exécution de tâches sur trois machines est visualisée ([JONES, 1988]).

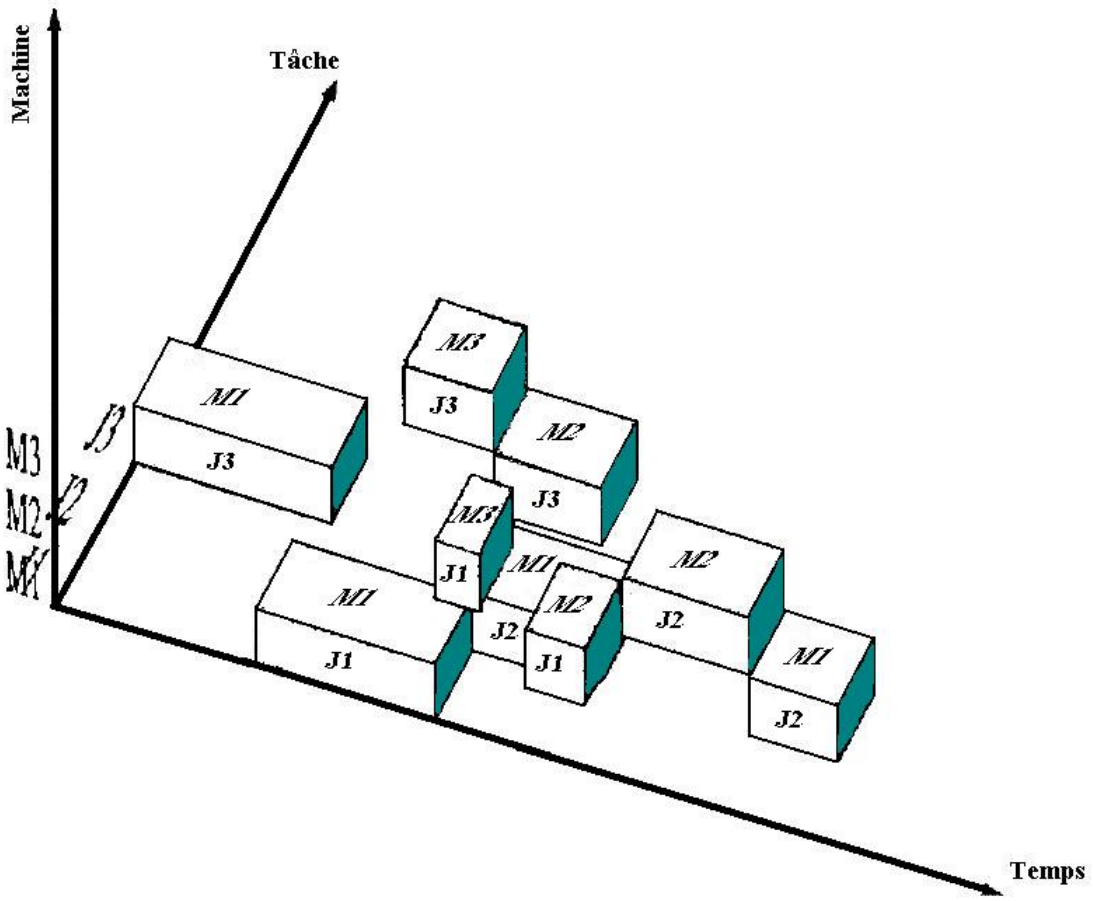

FIGURE 1.2 Visualisation d'un ordonnancement sur trois machines utilisant un diagramme de Gantt à trois dimensions.

L'utilisation des diagrammes de Gantt est généralement basée sur l'analyse des relations de précédence entre tâches dans un ordonnancement et la réorganisation des opérations pour comparer les performances de différentes configurations. Dans un environnement distribué, plusieurs usines d'attributs différents peuvent être sélectionnées pour le traitement des tâches d'où des plans de production multiples. La génération d'une séquence adéquate nécessite une intégration efficace du diagramme de Gantt dans le contexte d'une méthode d'optimisation telle que les algorithmes génétiques [JIA et collab., 2007].

1.5.2 Approche géométrique

L'approche géométrique constitue un outil très puissant pour la résolution optimale des problèmes d'ordonnancements du type job shop à deux tâches et avec n'importe quel critère régulier. Les fondements de cette approche ont été élaborés par Akers et Friedman et étudiés par la suite plus en détail par Brucker ([AKERS et FRIEDMAN, 1955] et [BRUCKER, 1988]). L'idée consiste à représenter le problème d'ordonnancement dans un plan à deux dimensions avec des obstacles reflétant le partage des ressources entre les opérations des tâches. Le plan de représentation est construit ainsi : Chaque axe correspondant à une tâche est divisé en un nombre de sous intervalles égale au nombre d'opérations de la tâche avec la longueur de chaque sous intervalle est proportionnelle à la durée de l'opération associée. Le rectangle généré par l'intersection de deux sous intervalles forme un obstacle si les deux opérations associées partagent la même ressource. Le chemin faisable le plus court entre le point de l'origine et le point final (déterminé par les sommes des durées opératoires des opérations des deux tâches) est composé de chemins élémentaires horizontal, vertical ou diagonale qui évitent le passage à travers un obstacle. Une illustration typique de cette approche est donnée sur la figure 1.3, (voir [MATI et XIE, 2008]).

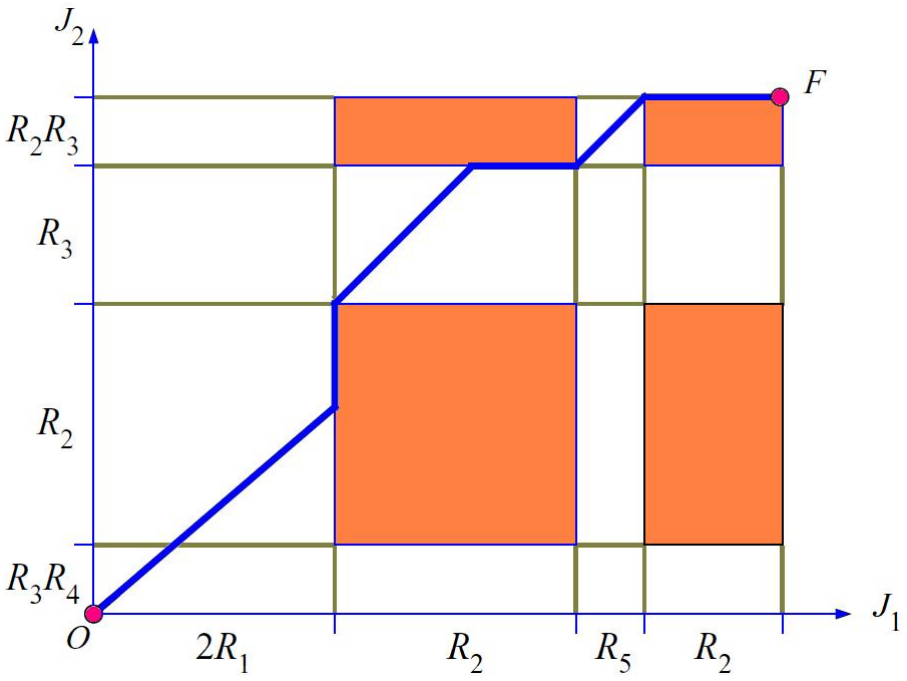

FIGURE 1.3 Exemple générique illustrant les éléments de base de l'approche géométrique.

Par conséquent, la recherche d'un ordonnancement qui minimise le makespan est équivalente à la recherche du plus court chemin faisable dans ce plan [MATI et collab., 2001]. Plusieurs extensions de l'approche géométrique ont été proposées pour résoudre des problèmes d'ordonnancement plus compliqués à deux tâches et parmi lesquelles nous pouvons citer l'hybridation avec des métaheuristiques pour supporter des périodes de non disponibilité des machines dans un environnement flow shop [AGGOUNE et PORTMANN, 2006].

1.5.3 Graphe disjonctif

L'utilisation des graphes disjonctifs pour la modélisation des problèmes d'ordonnancements d'ateliers est sans doute parmi les techniques les plus exploitées. Introduite initialement par Roy et Sussman en 1964, elle fut reprise par la suite par plusieurs chercheurs où elle est à l'origine des premières contributions pour le traitement des problèmes d'ordonnancement [JAIN et MEERAN, 1999]. Cette tendance peut être expliquée par la simplicité de construction de ces graphes et le support de différentes contraintes entre les dates de début et de fin des opérations. Un graphe disjonctif est composé des éléments suivants [TRABELSI, 2012] :

- Les noeuds : Ces noeuds représentent soit des débuts des opérations, ou bien des opérations fictives correspondantes à la fin de chaque travail, ou bien les dates du début et de la fin de l'ordonnancement ;
- Les arcs : Nous distinguons deux types d'arcs. Les arcs conjonctifs sont associés aux contraintes de précédence entre deux opérations consécutives et valués par la durée opératoire de la première opération. Les arcs disjonctifs exprimant les conflits d'utilisation de ressources où deux opérations ne peuvent pas s'exécuter en même temps si elles sont reliées par un tel arc.

Un ordonnancement admissible sur le graphe est obtenu en arbitrant chacune des disjonctions. Cela revient donc à préserver uniquement un arc de chaque paire d'arcs dans le but de fixer l'ordre de passage sur la machine. Donc, le graphe disjonctif contient toutes les informations necessaires pour décrire une solution partielle ou complète du problème d'ordonnancement comme indiqué par la figure 1.4 associée à un atelier du type job shop avec trois machines ([BŁAŻEWICZ et collab., 2000]).

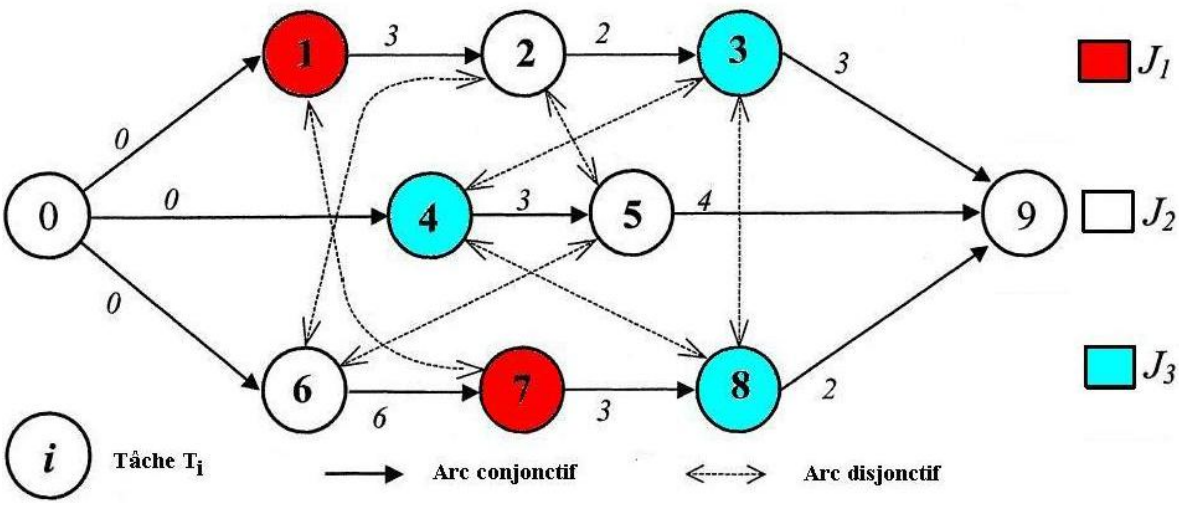

FIGURE 1.4 Graphe disjonctif d'un problème d'ordonnancement de type job-shop.

On dit que l'ordonnancement est parfaitement défini si toutes les disjonctions sont arbitrées sans génération de circuits dans le graphe [PENZ, 1994]. Le développement d'approches d'optimisation (par exemple recherche locale ou recuit simulé) à base de graphe disjonctif peut conduire à des procédures plus performantes en termes de qualité de solutions et de temps de calcul en particulier en tenant compte des contraintes complexes

et/ou objectifs multiples comme c'est le cas de l'industrie des composants à base de semiconducteur [YUGMA et collab., 2012].

1.5.4 Réseau de Petri

Les réseaux de Petri proposés pour la première fois par Carl Adam Petri sont devenus actuellement un outil de modélisation graphique et mathématique très efficace pour la représentation du comportement dynamique des systèmes de production [LEE et DICESARE, 1994]. Ceci est dû à l'adaptabilité du formalisme des réseaux de Petri temporisés aux phénomènes de synchronisation et de concurrence caractérisant l'interaction entre les tâches. Un réseau de Petri temporisé est constitué de trois composantes : les places représentées par des cercles, les transitions représentées par des bars et les arcs qui sont orientés et marqués par des poids. La notion du temps est introduite dans les réseaux de Petri en associant à chaque place un temps minimal de séjour des jetons [MURATA, 1989]. Il est possible de suivre l'évolution d'un système en observant le changement du marquage (vecteur du nombre de jetons dans chaque place) du modèle réseau de Petri équivalent où le marquage est gouverné par les règles de franchissement suivantes :

- La validité d'une transition dépend du marquage de toutes les places en amont avec des nombres de jetons dépassent ou égalent aux poids des arcs connectant les places à cette transition ;
- Lors du franchissement d'une transition, des jetons sont enlevés des places en amont et rajoutés aux places en aval selon les capacités des arcs connectés à ces places.

Par exemple, sur la figure 1.5 est representée la modélisation à base de réseau de Petri pour un atelier fonctionnant sans tampons avec deux machines et deux travaux. Il est claire dans ce cas qu'il est possible de tomber dans une situation de blockage (voir [MEJÍA et MONTOYA, 2009]).

En profitant de la grande analogie entre le régime stationnaire des réseaux de Petri temporisés et le comportement des problèmes d'ordonnancement cyclique, plusieurs travaux ont été proposés dans cette direction. Cependant, le passage d'un problème d'ordonnancement quelconque au modèle réseau de Petri correspondant est rendu possible en adoptant la procédure présentée dans [VAN DER AALST, 1996]. Les métaheuristiques ont été combinées avec les capacités de modélisation des réseaux de Petri afin de résoudre les problèmes d'ordonnancement complexes. Par conséquent, des problèmes d'ordonnancement ardus peuvent être modélisés par réseau de Petri simultanément avec l'obtention d'une solution proche de l'optimale dans un temps de calcul raisonnable [TUNCEL et BAYHAN, 2007]. Il est à mentionner que l'aspect incertain est introduit dans les réseaux de Petri à travers l'intégration avec la théorie des ensembles flous où la pertinence de cette vision a été validée par plusieurs applications industrielles [ZHOU et ZAIN, 2016].

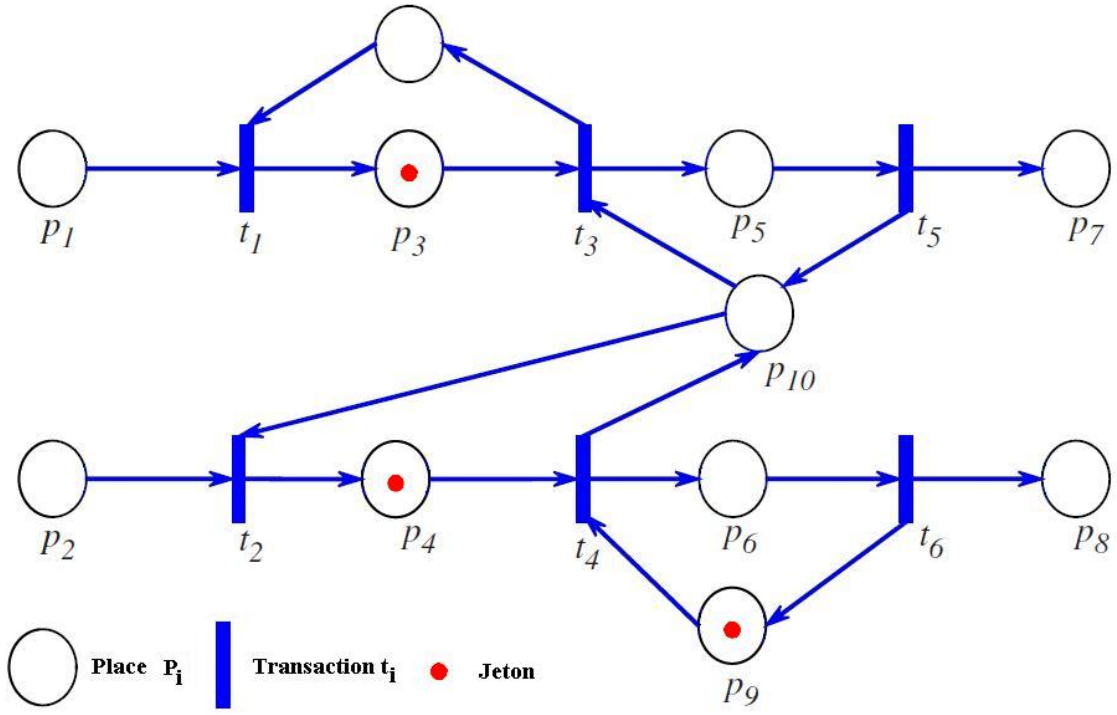

FIGURE 1.5 Modélisation par réseau de Petri de l'exécution de deux travaux sur un atelier à deux machines.

1.6 Importance des contraintes dans les problèmes d'ordonnancement

Le terme contrainte relative à un ensemble de variables de décision désigne une restriction imposée sur les domaines de valeurs associées à ces variables. Dans l'ordonnancement, nous distinguons deux types de variables de décision selon qu'elles portent sur des décisions liées au temps ou aux ressources [TRUNG, 2005] :

- Les contraintes temporelles sont utilisées pour exprimer les interdépendances temporelles lors de l'exécution des tâches. Ces contraintes modélisent en pratique les aspects technologiques ou logistiques qui conditionnent la réalisation d'un produit. Le formalisme des inégalités de potentiel constitue l'outil de base pour représenter les contraintes temporelles en particulier pour illustrer des écarts minimaux ou maximaux entre deux événements relatifs aux tâches ;
- Les contraintes de ressources sont utilisées pour représenter à la fois, les propriétés de consommation des tâches sur les ressources ainsi que les attributs de disponibilité des ressources. Dans ce cas, l'ordonnancement résultant doit garantir la cohérence de la consommation des tâches sur les ressources avec leurs disponibilités au cours du temps. Le respect des contraintes de ressources est équivalent au non chevauchement temporel des tâches.

Le Tableau 1.7 expose quelques contraintes temporelles et de ressources habituellement existantes dans la littérature de l'ordonnancement [BENHMIDA, 2009].

Les données associées à un problème d'ordonnancement telles que les durées opératoires ou les dates d'occurrence de certains événements sont souvent altérées par plusieurs sources d'incertitude internes et externes [BILLAUT et collab., 2008]. Pour les

TABLEAU 1.7 Présentation de quelques contraintes rencontrées en ordonnancement

Contrainte	Type	Description
Contrainte temporelle	Contrainte de temps absolu	Limitation des valeurs possible pour les dates des tâches.
	Contrainte de temps relatif	Satisfaction de la cohérence technologique.
Contrainte de ressources	Contrainte de capacité	Imposition de la réalisation disjointe des tâches différentes
		Interdiction de la violation de la capacité maximale d'une ressource donnée
	Contrainte d'affectation	Fixation de l'ensemble des ressources candidates pour l'exécution d'une tâche
		Imposition de l'utilisation de ressources différentes pour la réalisation d'un certain nombre de tâches

sources d'incertitudes internes nous trouvons :

- La durée d'une tâche dépend des conditions de réalisation en particulier les ressources matérielles et humaines ;
- La communication entre deux tâches est conditionnée par l'état de réseau de communication, les conflits et la disponibilité des liens ;
- Les temps de transport des composants entre opérations séparées dans un processus de production sont gouvernés par les propriétés des moyens de transport ;
- Quelques ressources comme les machines polyvalentes nécessitent un temps de reconfiguration dépendant du type et localisation des outils dans l'atelier.

Pour les sources d'incertitude externes au système de production nous pouvons citer :

- Les dates de début dans un ordonnancement peuvent dépendre des événements occurrents en dehors du système tels que la chaîne logistique ou les demandes de la clientèle ;
- Les périodes de disponibilité des ressources humaines ou matérielles sont très difficiles à prédire avec précision à cause par exemple d'absence des opérateurs ou manque de la matière première ;
- Les coûts affectés à une tâche peuvent changer sans notification en particulier si le système fait partie d'un grand système hiérarchique.

La flexibilité des contraintes temporelles sous paramètres incertains est un aspect très intéressant dans les problèmes d'ordonnancement car il rend compte de la possibilité de s'éloigner des instances qui satisfont parfaitement ces contraintes. La notion de flexibilité des contraintes décrit des préférences entre les combinaisons des valeurs d'ordonnancement. En effet, la considération des contraintes temporelles de disponibilité et de délai

comme impératives peut conduire au rejet d'un ordonnancement acceptable même si la violation de ces contraintes est insignifiante vis-à-vis des limites réalistes de prédiction de ces dates [FARGIER, 1994]. La théorie des possibilités permet de prendre en compte les problèmes d'ordonnancement intégrant des contraintes flexibles ou des paramètres incertains. L'adoption de ce cadre de modélisation permet également de généraliser la notion de graphe disjonctif utile pour l'évaluation d'inconsistances partielles ([DUBOIS et collab., 1995] et [DUBOIS et collab., 2003]).

1.7 Complexité des problèmes d'ordonnancement à une machine

La théorie de la complexité est destinée à l'analyse formelle de la difficulté théorique intrinsèque en particulier estimée en termes de temps de calcul ou taille de mémoire associés à la résolution d'un problème d'optimisation combinatoire. Une telle étude permet la distinction entre les problèmes dits "facile" et les problèmes dits "difficiles", d'où la possibilité de choisir la méthodologie de résolution la plus appropriée [HARTMANIS, 1988]. Comme les problèmes d'ordonnancements appartiennent à la classe des problèmes d'optimisation combinatoire, les concepts de la théorie de complexité sont applicables dont le but est la mise en oeuvre d'une approche de résolution efficace. Avant d'aborder ceci, nous présentons quelques définitions nécessaires pour l'assimilation de cet aspect [ZIMAND, 2004].

Définition 1.5. *Une machine de Turing est un modèle abstrait simulant l'exécution d'une procédure sur une machine de calcul. Elle se compose de deux éléments de base :*

- *Unité de commande : responsable de la mise à jour et les transitions entre les états possibles supposés finis ;*
- *Unité de mémoire : exprimée par un ruban de stockage constitué de nombre infini de cases consécutives.*

La communication entre ces deux éléments se fait par une tête de lecture/écriture qui balaye les symboles sur le ruban, et a la possibilité de se déplacer aux deux sens du ruban.

Une machine de Turing est appelée déterministe si à chaque état est associée au plus une seule transition (le prochain état est unique). Si plusieurs transitions sont permises pour certains états, on dit que la machine de Turing est non déterministe (existence de plusieurs états prochains candidats).

La machine de Turing universelle représente un outil plus avancé car elle est capable de simuler le comportement de n'importe quelle autre machine de Turing. Il en résulte que la machine de Turing universelle est dotée de la capacité de calculer des expressions complexes (fonction récursive, langage récursif, langage partiellement décidable).

La thèse de Church-Turing affirme l'existence d'une équivalence entre les problèmes résolubles par une machine de Turing universelle et les problèmes résolubles par un algorithme ou par une méthode concrète de calcul. Un problème de décision est un prédicat logique admettant deux solutions (Oui ou Non). La structure standard d'un problème de décision contient deux parties élémentaires : l'instance générique du problème en termes des différentes composantes et la question posée sur cette instance. La déduction du format décisionnel associé à un problème de minimisation peut être achevée en considérant une borne supérieure dans la partie question afin d'examiner l'existence d'une instance

ayant un coût qui ne dépasse pas cette borne. En se basant sur le concept de la machine de Turing avec considération de contrainte de temps, les deux classes suivantes (P et NP) sont les plus usuelles [GOLDREICH, 2008].

Définition 1.6. *La classe P est l'ensemble des langages reconnus par une machine de Turing deterministe en temps polynômial.*

Les problèmes appartenant à cette classe sont souvent considérés comme des problèmes faciles pour lesquels il est possible de développer un algorithme efficace de résolution optimale.

Définition 1.7. *La classe NP est l'ensemble des langages reconnus par une machine non déterministe en temps polynômial.*

Les problèmes appartenant à cette classe peuvent être également reconnus par une machine de Turing deterministe mais avec un temps d'ordre exponentiel.

Dans la théorie de la complexité, nous nous intéressons à la reconnaissance des problèmes les plus durs de la classe NP. Deux notions sont d'une importance cruciale dans la théorie de la complexité notamment celle de la NP-Complétude et celle de La NP-difficulté [ARORA et BARAK, 2009].

Définition 1.8. *Un problème de décision X est dit NP-Complet s'il appartient à la classe NP et il est possible de réduire tout autre problème Y de la classe NP à X en temps polynômial.*

Définition 1.9. *Un problème X est dit NP-difficile s'il est au moins aussi difficile que tous les problèmes appartenant à la classe NP.*

Il en découle de cette définition qu'un problème NP-difficile n'est pas forcément un élément de la classe NP ni même un problème de décision.

Pour permettre une classification assez précise des problèmes d'ordonnancement, la réduction polynômiale est utilisée pour prouver d'une façon formelle qu'un problème est plus difficile à résoudre qu'un autre. Ceci permet l'élaboration d'une hiérarchie partielle d'où la possibilité de réutilisation des approches de résolutions développées pour un problème après quelques modifications mineures pour d'autres problèmes. La Figure 1.6 présente un arbre de réduction des critères d'optimisation en ordonnancement.

Ce graphe de réduction s'interprète de la manière suivante : Pour un environnement d'atelier donné et pour des contraintes et un critère donnés, si un problème est NP-difficile, tous ses successeurs dans l'arbre le sont également. Afin d'alléger la mission de classification des problèmes combinatoires selon leur complexité, un logiciel nommé MSPCLASS a été développé avec une bibliothèque de 4536 problèmes parmi lesquels une classe de 120 problèmes d'ordonnancement à une machine [LAGEWEG et collab., 1982]. L'utilisateur fournit une liste de problèmes de complexité connue (P ou NP) et en se basant sur les relations d'ordre partiel dans la classe considérée, le logiciel génère automatiquement la nature de chaque problème (facile, ouvert ou difficile). L'ensemble des ordres partiels entre les classes des problemes d'ordonnancement dans un graphe de réduction permet de bien cerner les critères d'optimalités pouvant être simultanément employés dans le cadre d'une optimisation multi-objectif.

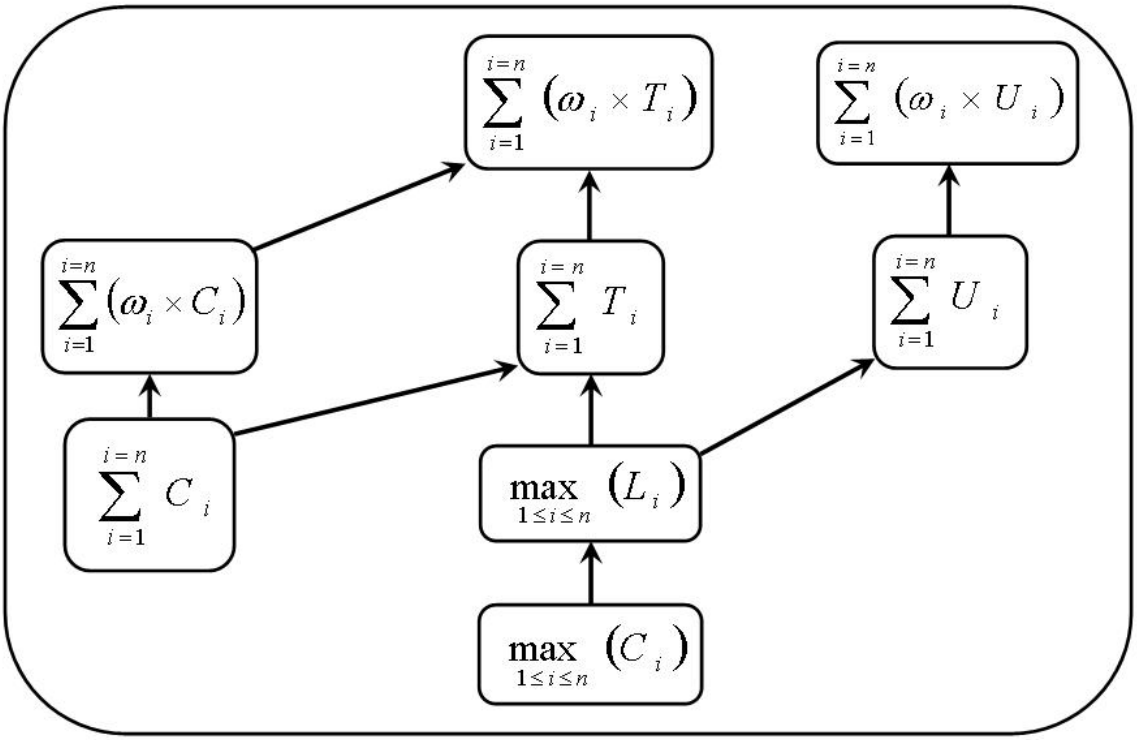

FIGURE 1.6 Présentation de quelques réductions élémentaires entre les critères de performance d'ordonnancement.

1.7.1 Complexité du problème d'ordonnancement standard

Quelques problèmes classiques d'ordonnancement à une machine ainsi que leurs classes de complexité sont illustrés sur le Tableau 1.8, où les paramètres des modèles sont représentés par des valeurs entières [LENSTRA et collab., 1977].

Il est à mentionner dans ce cas l'existence de plusieurs problèmes pouvant être résolus à l'optimalité, ce qui justifie la concentration des efforts des chercheurs sur les problèmes d'ordonnancement mono ressource.

1.7.2 Complexité du problème d'ordonnancement à une machine avec considération de contraintes

L'effet d'apprentissage dans le contexte industriel se réfère au phénomène de réduction du temps de traitement unitaire lorsque le produit associé est généré en grandes quantités. Cet effet peut être expliqué par le fait que les différentes stations de travail améliorent continuellement leurs performances comme résultat de la répétitions des mêmes tâches ou tâches similaires. L'effet d'apprentissage a été prouvé empiriquement dans le secteur manufacturier et secteur service comme un indice des entreprises bien gérées.

Actuellement, la technologie de groupe et la fabrication cellulaire sont très répandues dans les petits systèmes de production par lots où uniquement un nombre limité mais de tâches similaires est traité dans un cycle de production. Par conséquent, l'importance pratique de la prise en compte d'effets d'apprentissage dans le processus de planification des tâches devient une contrainte très forte à respecter [CHENG et WANG, 2000]. Comme plusieurs modèles d'apprentissage ont été introduits dans le domaine d'ordonnancement, nous présentons sur le Tableau 1.9 quelques problèmes d'ordonnancement

TABLEAU 1.8 Classification de quelques problèmes classiques d'ordonnancement à une machine

Notation	Complexité	Ordre
$1\left\|prec, r_i\right\|\mathrm{C_{max}}$	Polynômiale	$\mathrm{O}\left(n^2\right)$
$1\left\|tree\right\|\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|\mathrm{C}_i\leq\bar{d}_i\right\|\sum_{i=1}^{i=n}\mathrm{C}_i$	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|prec\right\|\mathrm{L_{max}}$	Polynômiale	$\mathrm{O}\left(n^2\right)$
$1\left\|prec, r_i, p_i=1\right\|\mathrm{L_{max}}$	Polynômiale	$\mathrm{O}\left(n^2\right)$
$1\left\|r_i, p_i=1\right\|\sum_{i=1}^{i=n}\omega_i\mathrm{T}_i$	Polynômiale	$\mathrm{O}\left(n^3\right)$
$1\left\|r_i, p_i=1\right\|\sum_{i=1}^{i=n}\omega_i\mathrm{U}_i$	Polynômiale	$\mathrm{O}\left(n^2\right)$
$1\,\|\|\sum_{i=1}^{i=n}\mathrm{U}_i$	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|prec, p_i=1\right\|\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$	NP-difficile au sens fort	/
$1\left\|prec\right\|\sum_{i=1}^{i=n}\mathrm{C}_i$	NP-difficile au sens fort	/
$1\left\|r_i\right\|\sum_{i=1}^{i=n}\mathrm{C}_i$	NP-difficile au sens fort	/
$1\left\|\mathrm{C}_i\leq\bar{d}_i\right\|\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$	NP-difficile au sens fort	/
$1\left\|r_i\right\|\mathrm{L_{max}}$	NP-difficile au sens fort	/
$1\,\|\|\sum_{i=1}^{i=n}\omega_i\mathrm{T}_i$	NP-difficile au sens fort	/
$1\left\|prec, p_i=1\right\|\sum_{i=1}^{i=n}\mathrm{T}_i$	NP-difficile au sens fort	/
$1\left\|r_i\right\|\sum_{i=1}^{i=n}\mathrm{T}_i$	NP-difficile au sens fort	/
$1\left\|prec, p_i=1\right\|\sum_{i=1}^{i=n}\mathrm{U}_i$	NP-difficile au sens fort	/
$1\left\|r_i\right\|\sum_{i=1}^{i=n}\mathrm{U}_i$	NP-difficile au sens fort	/
$1\,\|\|\sum_{i=1}^{i=n}\mathrm{T}_i$	NP-difficile au sens ordinaire	/
$1\,\|\|\sum_{i=1}^{i=n}\omega_i\mathrm{U}_i$	NP-difficile au sens ordinaire	/

supportant cet aspect [BISKUP, 2008].

TABLEAU 1.9 Classification de quelques problèmes d'ordonnancement à une machine avec considération d'effet d'apprentissage

Notation	Complexité	Ordre
$1\left\|p_{ir}=p_i r^a\right\|\mathrm{C_{max}}$	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i r^a\right\|\sum_{i=1}^{i=n}\mathrm{C}_i$	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i r^a, p_i=p\right\|\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i r^a\right\|\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$ avec $\{\omega_i\}$ agréables	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i r^a\right\|\mathrm{T_{max}}$ avec $\{d_i\}$ agréables	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i r^a\right\|\mathrm{L_{max}}$ avec $\{d_i\}$ agréables	Polynômiale	$\mathrm{O}\left(n\times\log(n)\right)$
$1\left\|r_i, p_{ir}=p_i r^a\right\|\mathrm{C_{max}}$	NP-difficile au sens fort	/
$1\left\|p_{ir}=p_i r^a\right\|\sum_{i=1}^{i=n}\mathrm{U}_i$	NP-difficile au sens fort	/
$1\left\|p_{ir}=p_i r^a\right\|\sum_{i=1}^{i=n}\mathrm{T}_i$	NP-difficile au sens fort	/

Le processus de réglage comprend les travaux liés à une multitude d'opérations telles que l'obtention des outils, le positionnement des pièces, le nettoyage, l'inspection et la configuration des machines ...etc. Le temps/coût associé à une opération de réglage a été considéré pour longtemps comme négligeable ou faisant partie de la durée opératoire. Bien que ceci est justifié pour certains problèmes d'ordonnancement, de nombreuses autres situations nécessitent un traitement séparé pour les temps (coûts) de réglage [ALLAHVERDI et collab., 1999]. Dans ce cas, nous distinguons entre le réglage dépendant uniquement de la tâche en cours et le réglage dépendant de la tâche en cours ainsi que la tâche précédente. Le temps de réglage pour le premier cas est dit indépendant de la séquence (s_{si}), cependant le temps dans le deuxième cas est dit dépendant de la séquence (s_{sd}) avec la possibilité de dépendance à plusieurs tâches précédentes (s_{psd}).

L'importance et les avantages de prise en compte des temps/coûts de réglage dans les axes de recherches d'ordonnancement ont été abordés par plusieurs chercheurs depuis les années 1960s où une moyenne de 40 articles publiés par an est ajoutée à la littérature. Le Tableau 1.10 résume la complexité de quelques problèmes d'ordonnancement avec considération du temps de réglage et le lecteur intéressé pour une description plus detaillée peut consulter les références [ALLAHVERDI et collab., 2008] et [ALLAHVERDI, 2015].

TABLEAU 1.10 Classification de quelques problèmes d'ordonnancement à une machine avec considération des temps de règlage

Notation	Complexité	Ordre
$1\left\|p_{ir}=p_i\left(\alpha a\sum_{j=1}^{r-1}\omega_j p_{[j]}+\beta\right)b^{r-1}, s_{psd}\right\|\max_{1\le i\le n}(C_i)$	Polynômiale	$O\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i\left(\alpha a\sum_{j=1}^{r-1}\omega_j p_{[j]}+\beta\right)b^{r-1}, s_{psd}\right\|\sum_{i=1}^{i=n}\omega_i C_i$ avec $\{\omega_i\}$ agréables	Polynômiale	$O\left(n\times\log(n)\right)$
$1\left\|s_{psd}\right\|\max_{1\le i\le n}(C_i)$	Polynômiale	$O\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i\left(1+\sum_{j=1}^{r-1}\log\left(p_{[j]}\right)\right)^a, s_{psd}\right\|\sum_{i=1}^{i=n}C_i^2$	Polynômiale	$O\left(n\times\log(n)\right)$
$1\left\|p_{ir}=p_i\times f\left(\sum_{j=1}^{r-1}\beta_j p_{[j]}, r\right), s_{psd}\right\|\max_{1\le i\le n}(L_i)$	Polynômiale	$O\left(n\times\log(n)\right)$
$1\left\|r_i, s_{si}=s, \text{pmtn}\right\|\sum_{i=1}^{i=n}C_i$	NP-difficile au sens fort	/
$1\left\|r_i, s_{si}, \text{pmtn}\right\|\max_{1\le i\le n}(C_i+d_i)$ avec $\{d_i\}$ agréables	NP-difficile au sens fort	/

1.8 Approches de résolution des problèmes d'ordonnancement

Lors de la résolution d'un problème d'ordonnancement, l'analyse au préalable de la complexité de calcul du problème décisionnel associé est requise. Par la suite, les données relatives à une instance de test sont générées numériquement afin de valider la performance d'une approche de résolution. Il n'existe pas une approche systématique pour la génération des instances de test mais il est possible d'adopter quelques règles d'ordre général pour l'obtention de données fiables [HALL et POSNER, 2001]. La procédure de résolution diffère selon la classe de complexité déduite car si le problème d'ordonnancement appartient à la classe P alors il est possible de développer un algorithme efficace pour l'obtention d'une solution exacte. Autrement, d'autres alternatives sont adoptées pour construire une solution approchée. Dans ce qui suit, nous décrivons les approches de résolutions les plus connues dans le domaine de l'ordonnancement en enrichissant le partitionnement présenté dans [MACCARTHY et LIU, 1993]. Nous commençons par les méthodes polynômiales éfficaces, nous passons ensuite aux approches énumératives éxactes et nous terminons avec le paradigme des techniques approchées.

1.8.1 Méthodes polynômiales éfficaces

Pour cette classe de méthodes, le temps en nombre d'instructions élémentaires nécessaire pour l'aboutissement à un optimum global augmente d'une façon polynômiale avec la valeur prise par une variable d'ordonnancement (généralement c'est le nombre de tâches ou de machines). L'avantage majeur dans ce cas est la faisabilité de résolution de certains problèmes d'ordonnancement de grande taille en un temps raisonnable (souvent borné par un polynôme d'ordre deux). Par exemple, la règle de Smith permet de

résoudre à l'optimalité quelques problèmes d'ordonnancement à une machine ($1||\sum_{i=1}^{i=n} C_i$ et $1||\sum_{i=1}^{i=n}(\omega_i \times C_i)$) [SMITH, 1956]. La règle de Moore est une autre méthode polynômiale efficace permettant de minimiser le nombre de tâches en retard pour le problème d'ordonnancement à une machine ($1||\sum_{i=1}^{i=n} U_i$) [MOORE, 1968]. Mais il est à mentionner qu'un tel formalisme n'est disponible que pour un nombre très réduit de problèmes d'ordonnancement (machine unique, flow shop et job shop) et de critères de performance.

1.8.2 Méthodes énumératives éxactes

Les méthodes dites exactes connues également sous le nom de méthodes optimales énumératives garantissent l'obtention d'une solution optimale pour le problème considéré. Généralement, ces outils sont caractérisés par un parcourt partiel de l'espace de solutions où les régions non prometteuses ne sont pas exploitées. Comme la complexité théorique des méthodes exactes est de comportement exponentiel, il en découle que leur application est limitée uniquement à des instances de taille moyenne pour lesquelles le temps d'exécution est acceptable. Nous détaillons ci-dessous deux méthodes de cette classe.

1.8.2.1 Procédures par Séparation et évaluation

La procédure par séparation et évaluation (en anglais Branch and Bound) est considérée parmi les méthodes exactes les plus reconnues dans la résolution optimale des problèmes combinatoires. Cette méthode repose comme son nom l'indique sur deux notions clés : la séparation et l'évaluation. Lors de l'exécution de la procédure, une arborescence avec la racine correspond à l'espace de solutions du problème initial est créée d'une façon incrémentale [LAWLER et WOOD, 1966]. Ceci est achevé à travers la dissociation d'un sommet exprimant une sous classe en une partition de sous-sous classe. Cette opération de division est combinée avec la comparaison du sommet traité avant l'exploration à un majorant dans le cas des problèmes de minimisation. La stratégie de calcul de ces bornes doit être le plus simple possible pour ne pas introduire des coûts supplémentaires au temps global de calcul. Si la valeur de la fonction objectif correspondante au sommet dépasse la borne alors l'arbre est élagué. Sinon, un autre tour de la décomposition du sommet est relancé. Il est à noter que les bornes sont améliorées chaque fois qu'une nouvelle meilleure solution est trouvée [MORRISON et collab., 2016].

1.8.2.2 Programmation dynamique

La programmation dynamique est considérée comme une approche générique permettant de concevoir des méthodologies robustes de résolution exacte des problèmes d'optimisation. Le fondement de cette approche s'appuit sur la décomposition du problème original en sous problèmes résolus localement d'une manière optimale [LEW et MAUCH, 2007]. Le processus de résolution est conduit d'une façon ascendante où les solutions des sous problèmes les plus petits sont utilisées en combinaison avec les informations intermédiaires pour construire progressivement la solution optimale. Cette vision est basée sur le principe d'optimalité de Belman qui stipule qu'une stratégie optimale a des sous stratégies optimales. La programmation dynamique possède plusieurs avantages en particulier [COOPER et COOPER, 1981] :

- La transformation d'un problème d'optimisation à plusieurs dimensions en un ensemble de problèmes d'optimisation unidimensionnels ;
- La solution obtenue est un optimum global en comparaison avec d'autres techniques pour lesquelles l'optimalité globale n'est pas assurée ;
- L'imposition des contraintes d'intégrité sur les valeurs des variables mène à une simplification significative du processus de calcul contrairement à plusieurs autres techniques d'optimisation.

1.8.3 Techniques approchées

Les méthodes approchées constituent l'alternative la plus adaptée aux problèmes combinatoire lorsque la taille est de l'ordre de ceux rencontrés dans la pratique. Ces méthodes permettent de trouver des solutions proches de l'optimum dans un temps de calcul raisonnable. Il existe deux classes de méthodes approchées : les heuristiques et les métaheuristiques.

Le terme "Heuristique" veut dire l'acte de découvrir et désigne une méthode d'optimisation qui profite des connaissances disponibles sur un problème spécifique dans le but de guider le processus de recherche de solutions. Les heuristiques sont généralement définies à la base de solutions de haute qualité en combinaison avec des règles empiriques. En outre, la conception efficace exige d'avoir des informations pertinentes sur la structure du problème à résoudre ainsi que les attributs différenciant les solutions de bonne qualité et celles de mauvaise qualité. Un certain degré de compromis doit être établi entre l'efficacité et la portée de la méthode heuristique [ROTHLAUF, 2011]. En effet, l'augmentation du nombre des propriétés spécifiques au problème conduit à une limitation de l'adaptabilité de la méthode développée pour d'autres problèmes. Dans cette situation, l'heuristique obtenue est très fiable en termes de qualité des solutions pour un type particulier de problèmes mais une fois la structure est modifiée même légèrement, les performances se dégradent rapidement. Les heuristiques peuvent être divisées en deux catégories selon la stratégie de génération des solutions [H. A. EISELT et SANDBIOM, 2004] :

- Les heuristiques de construction produisent une solution de telle sorte qu'à chaque itération, un nouvel élément est sélectionné pour créer progressivement la solution du problème ;
- Les heuristiques d'amélioration qui effectuent des améliorations à une solution initiale complète entre itérations jusqu'à atteindre un critère d'arrêt. La solution initiale dans cette catégorie peut être générée aléatoirement ou bien à travers une heuristique de construction plus efficace.

En ce qui concerne la signification du terme "Métaheuristique" dans le domaine d'optimisation, c'est un processus de guidage d'une heuristique subordonnée afin de générer itérativement des solutions quasi-optimales. Ceci est achevé en combinant plusieurs concepts pour la gestion intelligente de l'espace de recherche [OSMAN et KELLY, 1996]. Généralement, les approches métaheuristiques sont des outils génériques indépendants du gradient de la fonction objectif et pouvant être utilisés pour une multitude de problèmes sans beaucoup de modifications.

Toutes les approches métaheuristiques sont gouvernées par deux principes : la diversification (exploitation) et l'intensification (exploration). Dans la diversification, l'approche génère différentes solutions pour explorer l'espace de recherche d'une manière globale, tandis que dans l'intensification, l'approche concentre la recherche dans une région locale sachant qu'une bonne solution est trouvée dans cette région. Par conséquent,

un compromis entre ces deux principes doit être établi dans la sélection des meilleures solutions afin d'améliorer le taux de convergence et assurer l'obtention d'un optimum global.

Les approches métaheuristiques sont généralement inspirées de la nature et utilisent une solution unique ou une population de solution pour l'exploration de l'espace de recherche. Les méthodes utilisant une solution unique englobent les techniques de recherche locale comme la recherche tabou et le recuit simulé. Les méthodes à base d'une population explorent l'espace à travers l'évolution d'un ensemble de solutions telles que les algorithmes génétiques [SHELOKAR et collab., 2014]. De plus, ces approches peuvent être caractérisées par l'inclusion d'une mémoire pour le stockage et l'échange d'informations entre les individus de la même population ou bien entre plusieurs populations dans le contexte d'une métaheuristique parallèle.

1.9 Conclusion

Tout au long de ce chapitre, nous avons présenté les concepts de base nécessaires à mieux cerner les problèmes d'ordonnancement en particulier dans les ateliers de production. En premier, nous avons mis en évidence la criticité de la fonction ordonnancement au sein de l'hiérarchie de l'entreprise. Puis, nous nous somme intéressés par les caractéristiques générales des problèmes d'ordonnancement comprenant les formalismes mathématiques adoptés pour la description et la classification des problèmes d'ordonnancement sans oublier quelques outils de modélisation graphique largement utilisés pour la représentation des solutions ou bien la résolution proprement dite du problème considéré. Nous avons ensuite donné un bref aperçu sur l'importance des contraintes dans le contexte d'ordonnancement, ceci ne permet pas uniquement d'appréhender les contraintes technologiques liées au processus de fabrication mais également les contraintes imposées par des sources de perturbations externes à l'entreprise.

Dans un second temps, nous avons élaboré un survol de la complexité des problèmes d'ordonnancement à une machine les plus abondants dans la littérature ainsi que quelques problèmes considérant l'effet d'apprentissage et/ou les temps de réglage. L'étude de la complexité constitue le fil conducteur vers le choix de la méthodologie de resolution adéquate et pour cela nous avons exposé une taxinomie des approches de resolution (exactes et approchées) en dernière partie de ce chapitre. En se basant sur la littérature abordée, nous pouvons observer que la majorité des modèles les plus couramment utilisés pour formaliser un problème d'ordonnancement sont de nature déterministe. Dès lors des données incertaines sont présentes comme propriété intrinsèque de l'environnement, ces modèles déterministes deviennent rigides et incapables de supporter tel aspect d'imprécision. Par conséquent, nous nous intéressons dans les trois chapitres suivants aux implications de la prise en compte des paramètres incertains dans quelques problèmes d'ordonnancement. Nous justifions aussi l'adaptation de plusieurs outils de modélisation et d'optimisation pour des problèmes incluant des contraintes d'incertitude au niveau des paramètres d'ordonnancement.

1.10 Références

VAN DER AALST, W. M. P. 1996, «Petri net based scheduling», *OR Spektrum*, vol. 18, nº 4, p. 219–229. 21

AGGOUNE, R. et M.-C. PORTMANN. 2006, «Flow shop scheduling problem with limited machine availability : A heuristic approach», *International Journal of Production Economics*, vol. 99, n° 1-2, p. 4–15. 19

AKERS, S. B. et J. FRIEDMAN. 1955, «A non-numerical approach to production scheduling problems», *Journal of the Operations Research Society of America*, vol. 3, n° 4, p. 429–442. 19

ALLAHVERDI, A. 2015, «Third comprehensive survey on scheduling problems with setup times/costs», *European Journal of Operational Research*, vol. 246, n° 2, p. 345–378. 28

ALLAHVERDI, A., J. N. D. GUPTA et T. ALDOWAISAN. 1999, «A review of scheduling research involving setup considerations», *Omega The International Journal of Management Science*, vol. 27, n° 2, p. 219–293. 28

ALLAHVERDI, A., C. T. NG, T. C. E. CHENG et M. Y. KOVALYOV. 2008, «A survey of scheduling problems with setup times or costs», *European Journal of Operational Research*, vol. 187, n° 3, p. 985–1032. 28

ANTHONY, R. N. 1965, *Planning and control systems : a framework for analysis*, Division of Research, Graduate School of Business Administration, Harvard University. 10

ARORA, S. et B. BARAK. 2009, *Computational complexity a modern approach*, Cambridge University Press. 25

BENHMIDA, A. 2009, *Méthodes arborescentes pour la résolution de problèmes d'ordonnancement flexible*, thèse de doctorat, INSA de Toulouse. 22

BERTRAND, J. W. M., J. C. WORTMANN et J. WYNGAARD. 1990, *Production Control : A Structural and Design-Oriented Approach*, Elsevier Press. 10

BILLAUT, J.-C., A. MOUKRIM et E. SANLAVILLE. 2008, *Introduction to flexibility and robustness in scheduling*, chap. 1, in J-C. Billaut, and A. Moukrim, E. Sanlaville. (eds.), Flexibility and robustness in scheduling, ISTE et John Wiley. 22

BISKUP, D. 2008, «A state-of-the-art review on scheduling with learning effects», *European Journal of Operational Research*, vol. 188, n° 2, p. 315–329. 28

BITRAN, G. R. et D. TIRUPATI. 1989, «Hierarchical production planning», cahier de recherche 3017-89-MS, MIT Sloan School. Working Paper. 11

BŁAŻEWICZ, J., E. PESCH et M. STERNA. 2000, «Deterministic job-shop scheduling : Past, present and future», *European Journal of Operational Research*, vol. 127, n° 2, p. 317–331. 20

BRUCKER, P. 1988, «An efficient algorithm for the job-shop problem with two jobs», *Computing*, vol. 40, n° 4, p. 353–359. 19

BRUCKER, P. 2007, *Scheduling Algorithms*, Springer Press. 15

CHEN, B., C. N. POTTS et G. J. WOEGINGER. 1998, *A review of machine scheduling : Complexity, algorithms and approximability*, in D. Z. Du and P. Pardalos. (eds.), Handbook of Combinatorial Optimization, Kluwer. 16

CHENG, T. C. E. et G. WANG. 2000, «Single machine scheduling with learning effect considerations», *Annals of Operations Research*, vol. 98, nº 1, p. 273–290. 26

CLARK, W. 1923, *The Gantt chart a working tool of management*, Ronald Press. 17

CONWAY, R. W., W. L. MAXWELL et L. W. MILLER. 1967, *Theory of scheduling*, Addison-Wesley Press. 15

COOPER, L. et M. W. COOPER. 1981, *Introduction to Dynamic Programming*, Pergamon Press. 30

DUBOIS, D., H. FARGIER et P. FORTEMPS. 2003, «Fuzzy scheduling : Modelling flexible constraints vs. coping with incomplete knowledge», *European Journal of Operational Research*, vol. 147, nº 2, p. 231–252. 24

DUBOIS, D., H. FARGIER et H. PRADE. 1995, «Fuzzy constraints in job-shop scheduling», *Journal of Intelligent Manufacturing*, vol. 6, nº 4, p. 215–234. 24

ESQUIROL, P. et P. LOPEZ. 1999, *L'ordonnancement*, Edition Economica. 9

ESQUIROL, P. et P. LOPEZ. 2001, *Concepts et methodes de base en ordonnancement de la production*, chap. 12, in P. Esquirol and P. Lopez. (eds.), Ordonnancement de la production, Hermes. 13

FARGIER, H. 1994, *Problème de satisfaction de contraintes flexibles application à l'ordonnancement de production*, thèse de doctorat, Université Paul Sabatier de Toulouse. 24

GOLDREICH, O. 2008, *Computational complexity a conceptual perspective*, Cambridge University Press. 25

GOTHA. 1993, «Les problèmes d'ordonnancements», *RAIRO-Recherche Opérationnelle*, vol. 27, nº 1, p. 77–150. 13

GRAHAM, R. L., E. L. LAWLER, J. K. LENSTRA et A. H. G. RINNOOY KAN. 1979, «Optimization and approximation in deterministic sequencing and scheduling : a survey», *Annals of Discrete Mathematics*, vol. 5, p. 287–326. 16

H. A. EISELT, H. A. et C. L. SANDBIOM. 2004, *Decision Analysis, Location Models, and Scheduling Problems*, Springer Press. 31

HALL, N. G. et M. E. POSNER. 2001, «Generating experimental data for computational testing with machine scheduling applications», *Operations Research*, vol. 49, nº 6, p. 854–865. 29

HARTMANIS, J. 1988, *Overview of computational complexity theory*, chap. 1, in J. Hartmanis. (eds.), Computational complexity theory, American mathematical society. 24

JAIN, A. S. et S. MEERAN. 1999, «Deterministic job-shop scheduling : Past, present and future», *European Journal of Operational Research*, vol. 113, nº 2, p. 390–434. 20

JIA, H. Z., J. Y. H. FUH, A. Y. C. NEE et Y. F. ZHANG. 2007, «Integration of genetic algorithm and gantt chart for job shop scheduling in distributed manufacturing systems», *Computers and Industrial Engineering*, vol. 53, nº 7, p. 313–320. 18

JONES, C. V. 1988, «The three-dimensional gantt chart», *Operations Research*, vol. 36, nº 6, p. 891–903. 18

KAABI-HARRATH, J. 2004, *Contribution à l'ordonnancement des activités de maintenance dans les systèmes de production*, thèse de doctorat, Université de Franche-Comté. 9

LAGEWEG, B. J., J. K. LENSTRA, E. L. LAWLER et A. H. G. RINNOOY KAN. 1982, «Computer-aided complexity classification of combinatorial problems», *Communications of the ACM*, vol. 25, nº 11, p. 817–822. 25

LAWLER, E. L. et D. E. WOOD. 1966, «Branch and bound methods : a survey», *Operations Research*, vol. 14, nº 4, p. 699–719. 30

LEE, D. L. et F. DICESARE. 1994, «Scheduling flexible manufacturing systems using petri nets and heuristic search», *IEEE Transactions on Robotics and Automation*, vol. 10, nº 2, p. 123–132. 21

LENSTRA, J. K., A. H. G. RINNOOY KAN et P. BRUCKER. 1977, «Complexity of machine scheduling problems», *Annals of Discrete Mathematics*, vol. 1, p. 343–362. 26

LEUNG, J. Y.-T. 2004, *Introduction and Notation*, chap. 1, in J. Y-T. Leung. (eds.), Handbook of scheduling algorithms, models, and performance analysis, CRC. 13

LEW, A. et H. MAUCH. 2007, *Dynamic programming a computational tool*, Springer Press. 30

MACCARTHY, B. L. et J. LIU. 1993, «Addressing the gap in scheduling research : a review of optimization and heuristic methods in production scheduling», *International Journal of Production Research*, vol. 31, nº 1, p. 59–79. 29

MARTINSONS, M. G. et R. M. DAVISON. 2007, «Strategic decision making and support systems : Comparing american, japanese and chinese management», *Decision Support Systems*, vol. 43, nº 1, p. 284–300. 9

MATI, Y., N. REZG et X. XIE. 2001, «A taboo search appraoch for deadlock-free scheduling of automated manufacturing systems», *Journal of Intelligent Manufacturing*, vol. 12, nº 5, p. 535–552. 19

MATI, Y. et X. XIE. 2008, «A genetic-search-guided greedy algorithm for multiresource shop scheduling with resource flexibility», *IIE Transactions*, vol. 40, nº 12, p. 1228–1240. 19

MCKAY, K. N. 1992, *Production Planning and Scheduling : A Model for Manufacturing Decisions Requiring Judgement*, thèse de doctorat, University of Waterloo. 12

MCKAY, K. N. et V. C. S. WIERS. 1999, «Unifying the theory and practice of production scheduling», *Journal of Manufacturing Systems*, vol. 18, nº 4, p. 241–255. 12

MCKAY, K. N. et V. C. S. WIERS. 2006, *The human factor in planning and scheduling*, chap. 2, in J. W. Herrmann. (eds.), Handbook of production scheduling, Springer. 12

MEJÍA, G. et T. MONTOYA, C. MURATA. 2009, «Scheduling manufacturing systems with blocking : a petri net approach», *International Journal of Production Research*, vol. 47, nº 22, p. 6261–6277. 21

MOORE, L. J. 1968, «An n-job, one-machine sequence algorithm for minimizing the number of late jobs», *Management Science*, vol. 15, n° 1, p. 102–109. 30

MORRISON, D. R., S. H. JACOBSON, J. J. SAUPPE et E. C. SEWELL. 2016, «Branch-and-bound algorithms : A survey of recent advances in searching, branching, and pruning», *Discrete Optimization*, vol. 19, n° 1, p. 79–102. 30

MURATA, T. 1989, «Petri nets : properties, analysis and applications», *Proceedings of the IEEE*, vol. 77, n° 4, p. 541–580. 21

O'BRIEN, J. A. et G. M. MARAKAS. 2011, *Management Information Systems*, McGraw-Hill/Irwin Press. 11

OSMAN, I. H. et J. P. KELLY. 1996, *Meta-heuristics :Theory and Applications*, Kluwer Press. 31

PENZ, B. 1994, *Constructions agrégatives d'ordonnancements pour des jobs-shops statiques, dynamiques et réactifs*, thèse de doctorat, Université Joseph-Fourier - Grenoble I. 20

PINEDO, M. L. 2009, *Planning and Scheduling in Manufacturing and Services*, Springer Press. 13

ROTHLAUF, F. 2011, *Design of modern heuristics principles and application*, Springer Press. 31

SHELOKAR, P., A. KULKARNI, V. K. JAYARAMAN et P. SIARRY. 2014, *Metaheuristics in process engineering : A historical perspective*, chap. 1, in J. Valadi and P. Siarry. (eds.), Applications of metaheuristics in process engineering. 32

SMITH, W. E. 1956, «Various optimizers for single stage production», *Naval Research Logistics Quarterly*, vol. 3, n° 1-2, p. 59–66. 30

STORPER, M. et B. HARRISON. 1991, «Flexibility, hierarchy and regional development : The changing structure of industrial production systems and their forms of governance in the 1990s», *Research Policy*, vol. 20, n° 5, p. 407–422. 10

T'KINDT, V. et J.-C. BILLAUT. 2006, *Multicriteria scheduling*, Springer Press. 17

TRABELSI, W. 2012, *Ordonnancement des systèmes de production flexibles soumis à différents types de contraintes de blocage*, thèse de doctorat, Universite de Lorraine. 20

TRUNG, L. H. 2005, *Utilisation d'ordres partiels pour la caractérisation de solutions robustes en ordonnancement*, thèse de doctorat, INSA de Toulouse. 22

TUNCEL, G. et G. M. BAYHAN. 2007, «Applications of petri nets in production scheduling : a review», *International Journal of Advanced Manufacturing Technology*, vol. 34, n° 7, p. 762–773. 21

WILSON, J. M. 2003, «Gantt charts : A centenary appreciation», *European Journal of Operational Research*, vol. 149, n° 2, p. 430–437. 17

YUGMA, C., S. DAUZÈRE-PÉRÈS, C. ARTIGUES, D. A. et O. SIBILLE. 2012, «A batching and scheduling algorithm for the diffusion area in semiconductor manufacturing», *International Journal of Production Research*, vol. 50, n° 8, p. 2118–2132. 21

ZHOU, K.-Q. et A. M. ZAIN. 2016, «Fuzzy petri nets and industrial applications : a review», *Artificial Intelligence Review*, vol. 45, nº 4, p. 405–446. 21

ZIMAND, M. 2004, *Computational complexity : a quantitative perspective*, Elsevier Press. 24

Chapitre 2

Problèmes d'ordonnancement à une machine avec coûts actualisés et paramètres incertains

« *Pour croire avec certitude, il faut commencer par douter* »

Proverbe polonais

Sommaire

2.1 Introduction et état de l'art

Dans un environnement réel, la perception générale d'ordonnancement consiste en la planification d'un ensemble de tâches sur un groupe de ressources disponibles sujettes à des contraintes imposées. Cette classe de problèmes remonte aux années cinquante avec les travaux de Johnson, Smith et Jackson [GUPTA et KYPARISIS, 1987]. Malgré la simplicité apparente du problème de base à une machine, il a reçu une attention considérable et a trouvé des applications diverses dans plusieurs domaines. Ceci peut être expliqué par le fait que l'optimisation de plusieurs critères est possible à achever dans un temps polynômial. En outre, le problème d'ordonnancement à une machine dans la pratique est la pierre de base des grands systèmes de production. Donc, il est recommandé de résoudre initialement le problème à une machine avant d'entamer le problème original qui est le cas de machine goulot d'étranglement. Dans d'autres cas, la totalité de la chaîne de production peut être modélisée comme étant un problème à une machine [BAKER et TRIETSCH, 2009].

De point de vue complexité de calcul, plusieurs problèmes d'ordonnancement sont NP-difficiles dans le sens que le temps nécessaire pour la résolution croît d'une façon exponentielle en fonction de la taille des instances du problème. Les efforts sont accentués sur l'élaboration d'approches efficaces qui fournissent des solutions proches de l'optimale dans un temps acceptable [DAHAL et collab., 2007]. Au vu de la complexité intrinsèque des systèmes de production actuels caractérisés par une forte intégration de composants hétérogènes, fluctuations aléatoires et facteurs humains [CHRYSSOLOURIS, 2006], la vision classique des paramètres d'ordonnancement comme des paramètres à valeurs déterministes est mise en doute et doit être reformulée.

L'inclusion d'aspects incertains dans les modèles d'ordonnancement doit être parmi les enjeux primordiaux des gestionnaires pour gagner plus de stabilité envers les variations de la demande sur le marché. En considérant cet aspect, les problèmes d'ordonnancement sous incertitude sont traités en utilisant deux grandes familles d'approches : les approches stochastiques et les approches floues. La première est développée sous l'hypothèse qu'une partie ou la totalité des paramètres d'ordonnancement suivent des distributions de probabilité connues à priori ou bien estimées à partir d'un historique de données [LEUNG, 2004]. Tandis que la seconde est basée sur le paradigme de la théorie des ensembles flous élaboré par Lotfi Zadeh en 1965 [ZADEH, 1965]. La modélisation floue est généralement recommandée dans la manipulation des informations incertaines, vagues ou subjectives dans les modèles mathématiques pour les raisons suivantes [DONG, 2003] :

- Les approches basées sur la théorie des ensembles flous sont plus flexibles pour la représentation de l'imprécision des paramètres d'entrée en raison d'informations non quantifiables, d'informations incomplètes ou d'une ignorance partielle de données ;
- Difficulté des problèmes stochastiques en comparaison aux problèmes flous car la conversion des formulations probabilistes en des modèles déterministes génère souvent des problèmes de programmation non linéaires même si le problème stochastique original est linéaire. Alors que la majorité des problèmes flous linéaires sont équivalents à des problèmes de programmation linéaires d'où un coût de calcul réduit ;
- Les approches floues sont meilleures que les approches stochastiques en termes de simplicité de manipulation et adaptabilité pour les problèmes à grand échelle.

Dans la pratique, plusieurs problèmes d'ordonnancement réels ont été traités avec des formulations floues induites par l'ambigüité au niveau des valeurs de paramètres du

système. Par exemple, dans la production des transistors à couches minces pour affichage à cristaux liquides, il est difficile d'identifier l'état courant des ressources ou attributs d'un produit car cet état est dépendant du temps [LIU et YIH, 2013]. Dans la spécialité de production de peinture, le temps de configuration entre les séquences des tâches est d'une importance particulière dans l'obtention des spécificités d'une couleur. Les conditions de l'environnement comme la température peuvent aussi générer des incertitudes ayant un impact sur le processus de production en particulier la durée opératoire totale [SCHULTMANN et collab., 2006]. Dans les raffineries de pétrole situées au bord de la mer, le problème d'ordonnancement du pétrole brut contient plusieurs tâches telles que le vidange du pétrole brut à partir de navires pétroliers aux citernes de stockages, transfert des citernes de stockages vers les citernes de chargements et finalement le passage aux unités de distillation. Sous les contraintes des temps d'arrivée des navires pétroliers, capacités limitées, climat et demande imprévue, l'objectif final devient une mission fortement perturbée [CAO et collab., 2009]. Dans d'autres situations où le processus de modélisation est influencé par l'aspect flou des paramètres, le lecteur peut consulter ([NIKULIN et DREXL, 2010], [OLARU et SMITH, 2005] et [PETROVIC et collab., 2011]).

Dans la littérature, le travail de Prade est parmi les premiers papiers publiés dans le cadre de l'utilisation des concepts flous pour un problème d'ordonnancement réel [PRADE, 1979]. Par la suite, l'aspect flou est diffusé sur plusieurs niveaux de décision stimulé par les contraintes sévères dans le monde réel. Han et ses collègues ont exploité le problème d'ordonnancement à une machine avec des dates échues floues. Les auteurs ont considéré une fonction d'appartenance linéaire ainsi que des hypothèses simplificatrices pour permettre la résolution du problème généralisé du maximum de retard en un temps polynômial [HAN et collab., 1994]. Stanfield et ses collègues ont considéré un problème plus général où les temps de services et les dates échues sont des quantités vagues. Un algorithme par séparation et évaluation a été appliqué au modèle flou et une comparaison avec des modèles probabilistes a été réalisée [STANFIELD et collab., 1996]. Un problème d'ordonnancement par lots à une machine avec des temps de configuration, durées opératoires et dates échues floues a été étudié par Harakrishnan et Ishii [HARIKRISHNAN et ISHII, 2005]. Une approche polynômiale a été établie, elle est basée sur des formulations de programmation linéaire avec l'objectif de maximiser le degré minimal de satisfaction. Duenas et Petrovic ont développé une approche d'optimisation multiobjectif pour le problème d'ordonnancement à une machine avec des dates échues incertaines. La minimisation du maximum ainsi que la minimisation de la moyenne des temps de retard des tâches sont définies comme des fonctions objectifs. Une approche hybride basée sur les algorithmes génétiques et la recherche locale a été proposée et validée pour une société de fabrication de poterie [DUENAS et PETROVIC, 2008]. Liao et Liao ont utilisé des ensembles flous triangulaires et trapézoïdaux pour modéliser l'incertitude des dates échues et durées opératoires des tâches. Ils ont démontré la possibilité d'obtenir une séquence optimale des tâches maximisant le degré minimum de satisfaction en utilisant un algorithme polynômial de complexité $O\left(n^2 \times \log(G)\right)$ [LIAO et LIAO, 1998]. Wang et ses collègues ont appliqué la programmation par contraintes stochastiques pour la construction du profil de vraisemblance de la fin d'une tâche, ce qui permet un classement adéquat des solutions obtenues. Ils ont présenté également les conditions nécessaires d'optimalité des solutions, ce qui permet de réduire la taille de l'espace de recherche [WANG et collab., 2002]. Wu et ses collègues ont traité le problème d'ordonnancement de groupe à une machine avec les hypothèses de l'existence des temps de configuration entre les groupes et la nature floue des durées opératoires. Une méthodologie flexible de classement basée sur la méthode des valeurs centrales a été adoptée pour résoudre ce

problème. Ces auteurs ont prouvé que les résultats obtenus restent valables pour d'autres fonctions de deffuzification ayant une allure lineaire [Wu et Lee, 2005]. Mahrabadi et Pahlavani ont manipulé une extension du problème d'ordonnancement à une machine avec la somme pondérée des temps de retard qui est un problème NP-difficile. Ils ont supposé que les durées opératoires et les dates échues des tâches sont données par des nombres flous triangulaires. Les auteurs ont proposé aussi trois métaheuristiques pour conduire des expérimentations sur le problème [Saidi Mehrabad et Pahlavani, 2009]. Les concepts d'avance et de retard absolus flous ont été decrits en détail dans [Wu, 2010], ces critères sont basés sur les opérateurs de soustraction et maximum appliqués aux durées opératoires et dates échues floues. La somme pondérée de l'avance et de retard absolus flous a été minimimisée par un algorithme génétique où l'effet de l'allure des nombres flous a été également discuté par les auteurs. Ahmadizar et Hosseini ont proposé un algorithme polynômial pour le problème d'ordonnancement à une machine avec effet d'apprentissage de position et durées opératoires floues. La procédure de minimisation du temps total opératoire est basée sur l'application de la règle Délai de Fabrication le plus Court (en anglais SPT rule) pour les durées opératoires floues triangulaires [Ahmadizar et Hosseini, 2011]. Les mêmes auteurs ont présenté deux algorithmes polynômiaux pour la résolution du problème d'ordonnancement à une machine avec effet d'apprentissage de position et durées opératoires floues où l'objectif est de minimiser la durée totale de l'ordonnancement i.e. le makespan. La première approche est basée sur la programmation par contraintes stochastiques floues tandis que la deuxième est basée sur une méthode de classement de nombres flous [Ahmadizar et Hosseini, 2013]. Il est à souligner que l'aspect flou est souvent introduit dans les modèles d'ordonnancement sans justifier l'optimalité des fonctions objectifs choisies lorsque les versions floues des règles d'ordonnancement classiques sont utilisées. Özelkan et ses collègues ont étudié l'optimalité dans le cas flou de quelques règles d'ordonnancement telles que la règle Délai de livraison le plus proche (en anglais EDD). Ils ont illustré que la procédure de démonstration basée sur la permutation des tâches reste valable sous certaines conditions. Les auteurs ont conclu que la fuzzification des règles classiques et l'utilisation d'une fonction de classement arbitraire ne donnent pas toujours une règle optimale dans le contexte d'ordonnancement flou [Özelkan et Duckstein, 1999]. En suivant le même chemin, Chung a proposé une procédure sous l'acronyme "LARGE" pour trouver le plus grand ensemble de sous ensembles flous discrétisés. Cette procédure a été utilisée pour développer un algorithme basé sur la règle Délai de livraison le plus proche dans le cas de durée opératoire, dates de disponibilité et dates échues floues [Chuang, 2004]. Malgré le nombre élevé des travaux consacrés aux modèles d'ordonnancement flous, seulement une partie limitée de la littérature aborde la signification des coupes gamma (d'une façon equivalente alpha dans plusieurs références) des paramètres flous dans le modèle d'ordonnancement en particulier que l'information fournie par les différents niveaux des coupes peut être différente. Ainsi il sera intéressant d'exploiter au maximum les données introduites comme paramètres flous dans le modèle d'ordonnancement. Chanas et Kasperski ont introduit la notion de fonction floue monotone pour généraliser l'algorithme de Lawler. Les auteurs ont appliqué cette extension pour la maximisation du degré de possibilité que le maximum de la somme pondérée des retards algébriques ne dépasse pas la valeur seuil d'un certain objectif flou [Chanas et Kasperski, 2001]. Par la suite, les mêmes auteurs ont élaboré les concepts d'optimalité possible et d'optimalité nécessaire des solutions et ont présenté un calcul détaillé des paramètres dans le problème d'ordonnancement à une machine avec la somme pondérée des temps d'exécution et durées opératoires floues. La complexité de leur approche est $O\left(n \times \log(\varepsilon)\right)$ [Chanas et Kasperski, 2004]. Kasperski

a modélisé cinq problèmes d'ordonnancement avec la théorie de possibilité et a montré que quelques uns de ces problèmes sont NP-difficiles au sens fort [KASPERSKI, 2005]. Li et ses collègues ont construit des modèles d'ordonnancement pour le problème d'affectation des dates échues avec durées opératoires floues et contraintes de précédence afin de minimiser la somme pondérée [LI et collab., 2011]. De plus, Kasperski et Zieliński ont proposé une approche minmax où les paramètres flous induisent une distribution de possibilité sur l'ensemble des scenarii. Ils ont démontré que la détermination d'une séquence optimale dans le problème flou n'est plus difficile que dans le cas où les valeurs des paramètres sont localisées dans des intervalles [KASPERSKI et ZIEŁIŃSKI, 2011].

À la lumière de ces contributions, nous considérons une version floue du problème d'ordonnancement à une machine avec l'objectif de minimiser la somme pondérée des dates de fin d'exécution actualisées où tous les paramètres du modèle (durées opératoires, coefficients de pondération et taux d'actualisation) sont représentés par des nombres flous trapézoïdaux. Au lieu d'utiliser le principe d'extension pour l'évaluation de la fonction objectif dans le contexte d'un problème d'optimisation combinatoire, la théorie de possibilité est adoptée pour quantifier le degré de satisfaction de l'événement "Une séquence de tâches donnée est optimale en matière de la vérification d'un ensemble de règles". Donc, la notion d'optimalité devient elle-même un concept incertain qui nécessite une estimation dans l'intervalle [0, 1] en utilisant des modèles mathématiques compacts et la simulation numérique. Notre objectif dans ce chapitre est de proposer des critères basés sur les notions d'optimalités possible et nécessaire pour évaluer l'optimalité d'une séquence de tâches bien déterminée. Nous montrons aussi que dans certains cas particuliers, les formulations proposées se réduisent à des règles d'ordonnancement déjà disponibles dans la littérature. En se basant sur une formulation sous contraintes non linéaires de l'optimalité possible, deux approches de résolution (algorithme génétique et recherche par motifs) sont développées. Les contraintes non linéaires sont incluses dans le modèle à travers une fonction objectif augmentée. Dans le but d'évaluer et de comparer les performances des deux approches, plusieurs instances sont générées et utilisées comme des benchmarks pour mener les tests. Les méthodologies de modélisation proposées dans ce chapitre peuvent être considérées comme de bonnes alternatives pour l'évaluation efficace de la robustesse des ordonnancements dans un environnement incertain.

L'organisation du chapitre est comme suit. Les notions fondamentales sur la theorie de possibilité sont introduites dans la Section 2.2. Les formulations théoriques sont présentées dans la Section 2.3, suivie par les composantes de base de notre simulation dans la Section 2.4. Des exemples numériques ainsi que les résultats d'expérimentation sont traités dans la Section 2.5. Finalement, nous concluons avec quelques remarques et perspectives futures dans la dernière Section.

2.2 Terminologie de base sur la théorie de possibilité

Avant d'aborder les théorèmes principaux relatifs à la théorie de possibilité, les définitions et propriétés de base attachées aux nombres flous sont illustrées ([KLIR et YUAN, 1995], [SIVANANDAM et collab., 2007] et [ZIMMERMANN, 1985]). Les nombres flous ou plus généralement les ensembles flous de la ligne réelle sont un schéma convenable pour la représentation et la manipulation arithmétique des quantités mal connues.

Définition 2.1. *Une fonction* $\tilde{u} : \mathbb{R} \to [0, 1]$ *est dite un nombre flou si elle satisfait :*

i) $\tilde{u}$ *est normale, i.e.* $\exists x_0 \in \mathbb{R}$ *avec* $\tilde{u}(x_0) = 1$;

ii) $\tilde{u}$ *est un ensemble flou convexe (i.e.* $\tilde{u}\left(\lambda x+(1-\lambda)\, y\right) \geq \min\left\{\tilde{u}\left(x\right), \tilde{u}\left(y\right)\right\}\ \forall x, y \in \mathbb{R}, \forall \lambda \in [0,1]$*);*

iii) $\tilde{u}$ *est semi continue supérieure sur* $\mathbb{R}$*;*

iv) $\overline{\{x_0 \in \mathbb{R}:\ \tilde{u}\left(x\right) > 0\}}$ *est un ensemble compact, avec* $\overline{\{\}}$ *est la fermeture de l'ensemble.*

Il est plus convenable de représenter les nombres flous en utilisant un ensemble d'intervalles fermés générés selon la définition suivante.

Définition 2.2. *Soit* γ *un nombre réel tel que* $\gamma \in [0,\ 1]$*. L'ensemble des coupes (niveaux)* γ *de* $\tilde{u}$ *est l'ensemble réel* $[\tilde{u}]^{\gamma}$ *défini par :*

$$[\tilde{u}]^{\gamma} = \begin{cases} \left\{x \in \mathbb{R}:\ \tilde{u}\left(x\right) \geq \gamma\right\} & si \quad \gamma > 0 \\ \\ \overline{\{x \in \mathbb{R}:\ \tilde{u}\left(x\right) > 0\}} & si \quad \gamma = 0 \end{cases} \tag{2.1}$$

Par conséquent, l'intervalle fermé et borné $[\tilde{u}]^{\gamma}$ peut s'écrire sous forme paramétrique $[\tilde{u}]^{\gamma} = \left[\underline{u}\left(\gamma\right), \overline{u}\left(\gamma\right)\right]$ avec $\bar{u}\left(\underline{u}\right)$ est une fonction bornée continue à gauche croissante (décroissante) sur l'intervalle $[0,1]$.

Définition 2.3. *Un nombre flou trapézoïdal noté par* $\tilde{u} = \left(a, b, \alpha, \beta\right)$*, avec* $a, b \in \mathbb{R}$ *et* $\alpha, \beta \geq 0$*, est défini par la fonction :*

$$\tilde{u}\left(x\right) = \begin{cases} 1 - \left(\frac{a-x}{\alpha}\right) & si\ a - \alpha \leq x < a \\ \\ 1 \quad si\ a \leq x \leq b \\ \\ 1 - \left(\frac{x-b}{\beta}\right) & si\ b < x \leq b + \beta \\ \\ 0 \quad autrement \end{cases} \tag{2.2}$$

On dit que $\tilde{u}$ est un nombre flou trapezoidal avec intervalle de tolérance $[a, b]$, de largeur à gauche α et largeur à droite β. Pour le nombre flou trapezoidal $\tilde{u} = \left(a, b, \alpha, \beta\right)$, les fonctions $\underline{u} : [0,1] \rightarrow \mathbb{R}$ et $\overline{u} : [0,1] \rightarrow \mathbb{R}$ définissant les bornes des intervalles $[\tilde{u}]^{\gamma}$, $\gamma \in [0,1]$, ont la forme $\left[a - \left(1-\gamma\right)\alpha, b + \left(1-\gamma\right)\beta\right]$. Le support de $\tilde{u}$ est $\left(a - \alpha, b + \beta\right)$.

Dans le but de généraliser les opérations arithmétiques ordinaires ainsi que les fonctions composées, le théorème suivant est utilisé pour calculer les ensembles de niveau γ des nombres flous obtenus comme résultat de l'application de ces opérations sur des opérandes de même type [NGUYEN, 1978].

Théorème 2.1. *Soient* $f : \mathbb{R}^d \rightarrow \mathbb{R}$ *une fonction continue et* $\tilde{\mathrm{A}}_1, \ldots, \tilde{\mathrm{A}}_d$ *un ensemble de d nombres flous. Alors :*

i) La fonction $f\left(\tilde{\mathrm{A}}_1, \ldots, \tilde{\mathrm{A}}_d\right) : \mathbb{R}^d \rightarrow [0,1]$ *construite par application du principe d'extension de Zadeh est un nombre flou;*

ii) Pour tout $\gamma \in [0,1]$*, on a* $\left[f\left(\tilde{\mathrm{A}}_1, \ldots, \tilde{\mathrm{A}}_d\right)\right]^{\gamma} = f\left(\left[\tilde{\mathrm{A}}_1\right]^{\gamma}, \ldots, \left[\tilde{\mathrm{A}}_d\right]^{\gamma}\right)$ *avec* $\left[f\left(\tilde{\mathrm{A}}_1, \ldots, \tilde{\mathrm{A}}_d\right)\right]^{\gamma} = \left\{f\left(x_1, \ldots, x_d\right) \middle| x_1 \in \left[\tilde{\mathrm{A}}_1\right]^{\gamma}, \ldots, x_2 \in \left[\tilde{\mathrm{A}}_d\right]^{\gamma}\right\}$.

En se basant sur ce Théorème, les opérations généralisées suivantes peuvent être déduites pour tous nombres flous positifs ($\tilde{\mathrm{A}}$ et $\tilde{\mathrm{B}}$) et scalaire λ, respectivement :

i) $\left[\tilde{A} \oplus \tilde{B}\right]^{\gamma} = \left[\tilde{A}\right]^{\gamma} + \left[\tilde{B}\right]^{\gamma} = \left[\underline{a}(\gamma) + \underline{b}(\gamma), \overline{a}(\gamma) + \overline{b}(\gamma)\right]$;

ii) $\left[\lambda \times \tilde{A}\right]^{\gamma} = \begin{cases} \left[|\lambda| \times \underline{a}(\gamma),\ |\lambda| \times \bar{a}(\gamma)\right] & si\ \lambda \geq 0 \\ \left[|\lambda| \times \bar{a}(\gamma),\ |\lambda| \times \underline{a}(\gamma)\right] & si\ \lambda < 0 \end{cases}$;

iii) $\left[\tilde{A} \otimes \tilde{B}\right]^{\gamma} = \left[\underline{a}(\gamma) \times \underline{b}(\gamma), \overline{a}(\gamma) \times \overline{b}(\gamma)\right]$ *pour* $\gamma \in [0,\ 1]$.

La théorie de possibilité évoluée à partir des propriétés algébriques riches des nombres flous a pour objectif de remplacer la théorie de probabilité dans la modélisation des situations vagues [ZADEH, 1978]. Pour cela, les mesures de possibilité et de nécessité substituent les distributions de probabilité. Dans ce qui suit, nous présentons d'une façon succincte quelques résultats principaux de la théorie de possibilité ([DUBOIS et PRADE, 1988], [GEORGESCU, 2012], [JAMISON, 1998] et [LODWICK et KACPRZYK, 2010]). Soit $P(\Omega)$ l'ensemble des parties d'un ensemble non vide noté Ω. Pour un élément $D \in P(\Omega)$, on note son complément par $D^C = \Omega - D$.

Définition 2.4. *Une mesure floue sur Ω est une fonction $m : P(\Omega) \to [0,1]$ telle que :*

i) $m(\phi) = 0$;

ii) $m(\Omega) = 1$;

iii) $\forall D_1, D_2 \in P(\Omega) : (D_1 \subseteq D_2) \Rightarrow (m(D_1) \leq m(D_2))$.

À la base de cette Définition, les mesures de possibilité et de nécessité sont illustrées par les Définitions 2.5 et 2.6, respectivement.

Définition 2.5. *Une mesure de possibilité sur Ω est une fonction $\Pi : P(\Omega) \to [0,1]$ satisfaisant les conditions suivantes :*

i) $\Pi(\phi) = 0$;

ii) $\Pi(\Omega) = 1$;

iii) $\Pi\left(\underset{i \in I}{\cup} D_i\right) = \sup\limits_{i \in I} \Pi(D_i)$, *pour chaque famille* $\{D_i\}_{i \in I}$ *de sous ensembles de* Ω.

Définition 2.6. *Une mesure de nécessité sur Ω est une fonction $N : P(\Omega) \to [0,1]$ telle que les propriétés suivantes sont vérifiées :*

i) $N(\phi) = 0$;

ii) $N(\Omega) = 1$;

iii) $N\left(\underset{i \in I}{\cap} D_i\right) = \inf\limits_{i \in I} \Pi(D_i)$ *pour chaque famille* $\{D_i\}_{i \in I}$ *de sous ensembles de* Ω.

La notion de distribution de possibilité est étroitement liée à celle de la mesure de possibilité. Une distribution de possibilité sur Ω est une fonction $\mu : \Omega \to [0,1]$ telle que $\sup\limits_{x \in \Omega} \mu(x) = 1$.

Définition 2.7. *Soit Π une mesure de possibilité sur Ω et $\mu : \Omega \to [0,1]$ une distribution de possibilité. Les applications $\mu_{\Pi} : \Omega \to [0,1]$ et $Pos_{\mu} : P(\Omega) \to [0,1]$ sont définies par $\mu_{\Pi}(x) = \Pi(\{x\})\ \forall x \in \Omega$ et $Pos_{\mu}(D) = \sup\limits_{x \in D} \mu(x)\ \ \forall D \in P(\Omega)$ respectivement.*

2.3 Formulation du problème d'ordonnancement

Dans ce chapitre, le problème d'ordonnancement à étudier est formulé comme suit. Un ensemble de n tâches à exécuter sans préemption sur une machine supposée sans interruption (défaillance, maintenance,...). Seulement une tâche est servie à la fois et les autres tâches prêtes sont en attentes. Les dates de disponibilité sont fixées à l'origine ce qui signifie que la machine ainsi que les tâches sont prêtes au temps zéro. En outre, les durées de configurations de la machine sont indépendantes de la séquence des tâches et sont incluses dans les durées opératoires. Pour chaque tâche i, $1 \leq i \leq n$, un ensemble de paramètres : durée opératoire p_i, poids ω_i et taux d'actualisation r est défini. Alors, les coûts sont actualisés à un taux fixe (souvent proche de 0 i.e. au voisinage de 10%), indiquant que si la tâche i n'est pas achevée à la date t, un coût supplémentaire $\omega_i r e^{-rt} dt$ sera considéré sur l'intervalle $[t,\ t+dt]$. Soit C_i la date fin de la tâche i, si elle est complétée à $t = C_i$, le coût total sur la période $[0, C_i]$ est $\omega_i \left(1 - e^{-rC_i}\right)$. L'objectif final est d'ordonnancer l'ensemble des tâches de telle sorte que la somme actualisée pondérée des dates de fin d'exécution donnée par $\sum_{i=1}^{i=n} \omega_i \left(1 - e^{-rC_i}\right)$ soit minimale ([ROTHKOPF, 1966]). L'introduction du taux d'actualisation reflète en général le coût de possession où le coût de stockage qui est supposé actualisé à un taux $0 < r < 1$ par unité de temps comme indiqué dans plusieurs problèmes pratiques ([BŁAŻEWICZ et collab., 2007], [ARTIGUES et collab., 2008] et [GAWIEJNOWICZ, 2008]). Dans la première sous section, nous présentons les propriétés du modèle classique associé au problème d'ordonnancement $1 \,||\, \sum_{i=1}^{i=n} \omega_i \left(1 - e^{-rC_i}\right)$. Dans la deuxième sous section, nous proposons une extension pour ce problème basée sur la théorie de possibilité. Cette extension est donnée utilisant la notation de Graham par $1 \left| \tilde{p}_i, \tilde{\omega}_i, \tilde{r} \right| \sum_{i=1}^{i=n} \omega_i \left(1 - e^{-rC_i}\right)$ avec $\tilde{p}_i$, $\tilde{\omega}_i$ et $\tilde{r}$ sont des paramètres flous.

2.3.1 Optimalité ordinaire dans le problème d'ordonnancement à une machine avec coûts actualisés

Lorsque les paramètres d'ordonnancement prendront des valeurs réelles ce qui reflète une vision déterministe du problème, la fonction objectif peut être considérée comme une généralisation de la somme pondérée non actualisée des dates de fin. Dans ce cas, la minimisation de la fonction objectif est possible par la règle de priorité suivante [PINEDO, 2012].

Théorème 2.2. *Dans le problème d'ordonnancement* $1 || \sum_{i=1}^{i=n} \omega_i \left(1 - e^{-rC_i}\right)$, *un ordonnancement* π *est optimal si et seulement si la condition suivante est satisfaite pour chaque* $i = \overline{1, n-1}$ *:*

$$\frac{e^{r p_{\pi(i)}} - 1}{\omega_{\pi(i)}} \leq \frac{e^{r p_{\pi(i+1)}} - 1}{\omega_{\pi(i+1)}} \tag{2.3}$$

Démonstration.
Necessité des conditions ($\Rightarrow$)
Supposons que nous avons un ordonnancement optimal θ et les conditions ne sont pas satisfaites. Alors, il existe au moins une tâche $k \in \{1, ..., n-1\}$ telle que $\frac{e^{r p_{\pi(i)}} - 1}{\omega_{\pi(i)}} > \frac{e^{r p_{\pi(i+1)}} - 1}{\omega_{\pi(i+1)}}$. Il est possible de générer un nouvel ordonnancement δ par permutation des tâches $\theta(k)$ et $\theta(k+1)$ dans la séquence θ, i.e. $\delta = (\theta(1), ..., \theta(k-1), \theta(k+1), \theta(k), \theta(k+2), ..., \theta(n))$ comme indiqué sur la Figure 2.1.

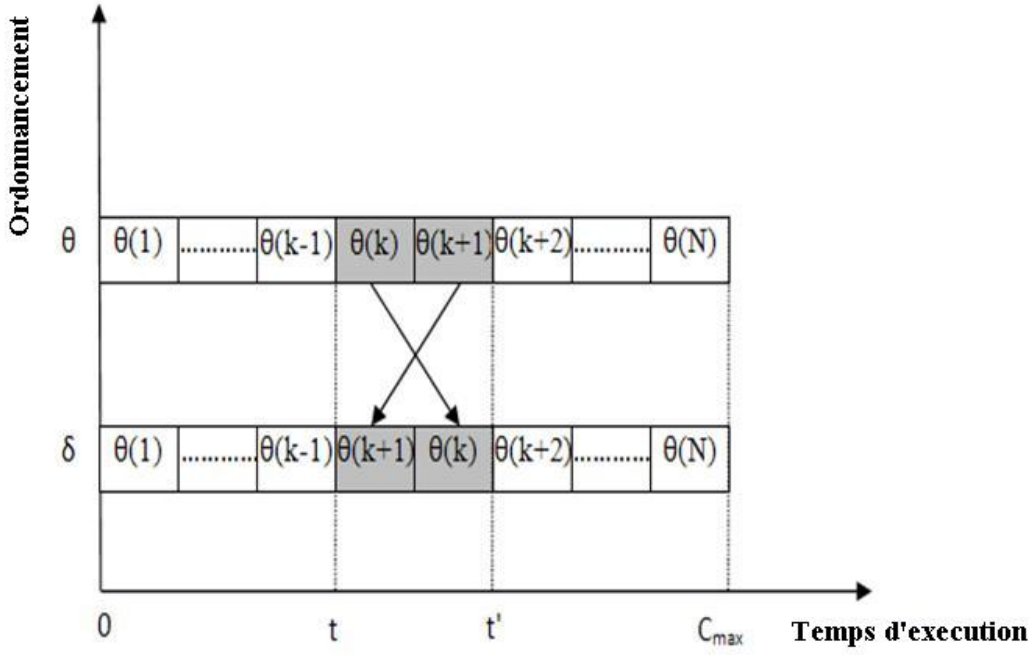

FIGURE 2.1 Application de l'argument de permutation pour les tâches aux positions k et $k+1$.

Nous pouvons vérifier que la différence suivante $\sum_{i=1}^{i=n} \omega_i \left(1-e^{-rC_i(\theta)}\right) - \sum_{i=1}^{i=n} \omega_i \left(1-e^{-rC_i(\sigma)}\right)$ peut être simplifiée pour obtenir $e^{-rt}\left\{\omega_{\theta(k+1)} e^{-rp_{\theta(k+1)}} \left(1-e^{-rp_{\theta(k)}}\right) - \omega_{\theta(k)} e^{-rp_{\theta(k)}} \left(1-e^{-rp_{\theta(k+1)}}\right)\right\}$. Comme cette quantité est strictement positive ce qui implique que $\sum_{i=1}^{i=n} \omega_i \left(1-e^{-rC_i(\theta)}\right)$ est supérieure à $\sum_{i=1}^{i=n} \omega_i \left(1-e^{-rC_i(\sigma)}\right)$, d'où une contradiction avec notre hypothèse sur l'optimalité de θ. Donc, l'ensemble des conditions sont nécessaires pour l'optimalité d'un ordonnancement pour le problème $1||\sum_{i=1}^{i=n} \omega_i \left(1-e^{-rC_i}\right)$.

Suffisance des conditions (⇐)
Supposons maintenant que nous avons un ordonnancement κ qui viole ces conditions. Donc, il existe au moins deux tâches voisines dans la séquence pour laquelle la règle de priorité n'est pas vérifiée. Si nous effectuons une permutation locale de ces tâches, les dates de fin des autres tâches ne seront pas affectées mais nous aurons une diminution de la somme actualisée pondérée des dates de fin d'ordonnancement original κ. Ce processus est répété itérativement jusqu'à l'obtention d'un ordre total des tâches selon le rapport $\frac{e^{rp_{\kappa(i)}}-1}{\omega_{\kappa(i)}}$. À ce niveau, aucune amelioration n'est possible et l'ordonnancement final est optimal. La procédure de classement est de complexité $O\left(n \times \log(n)\right)$. Donc, nous déduisons que cette règle constitue une condition suffisante d'optimalité d'un ordonnancement dans le problème $1||\sum_{i=1}^{i=n} \omega_i \left(1-e^{-rC_i}\right)$. □

Le lien entre le problème d'ordonnancement à une machine avec coûts actualisés

$1||\sum_{i=1}^{i=n}\omega_i\left(1-e^{-r\mathrm{C}_i}\right)$ et la version non actualisée $1||\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$ est donnée par le Lemme 2.1.

Lemme 2.1. *Pour des valeurs très petites du taux d'actualisation* $(r\to 0)$*, le problème* $1||\sum_{i=1}^{i=n}\omega_i\left(1-e^{-r\mathrm{C}_i}\right)$ *se réduit au problème* $1||\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$.

Démonstration.
En considérant la fonction objectif comme fonction du taux d'actualisation $f(r)=\sum_{i=1}^{i=n}\omega_i\left(1-e^{-r\mathrm{C}_i}\right)$, le développement limité de Taylor de la fonction f au voisinage du zéro donne $f(r)=f(0)+f'(0)\,r+\ldots+\frac{f^{(k)}(0)r^k}{k!}+\mathrm{R}_k(r)$. Nous pouvons limiter la série de Taylor au deux premiers termes car r est très proche de zéro et nous obtenons $f(r)\cong f(0)+f'(0)\,r=r\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$. Le résultat obtenu diffère de la somme pondérée des dates de fin uniquement par une constante positive de multiplication qui n'influe pas sur l'optimalité de la fonction. D'où le problème $1||\sum_{i=1}^{i=n}\omega_i\left(1-e^{-r\mathrm{C}_i}\right)$ se réduit au problème $1||\sum_{i=1}^{i=n}\omega_i\mathrm{C}_i$ pour des valeurs faibles de r. □

2.3.2 Mesures possibilistes d'optimalité pour le problème d'ordonnancement à une machine avec coûts actualisés et paramètres incertains

Comme les paramètres associés à une tâche i sont supposés incertains, nous utilisons des nombres flous positifs pour représenter les durées opératoires, les coefficients de pondération et le taux d'actualisation dans le problème d'ordonnancement $1\left|\tilde{p}_i,\tilde{\omega}_i,\tilde{r}\right|\sum_{i=1}^{i=n}\omega_i\left(1-e^{-r\mathrm{C}_i}\right)$. Nous obtenons pour $\gamma\in[0,1]$ fixe l'intervalle compact généré par le produit cartésien des coupes γ des différents paramètres flous $\Gamma(\gamma)=\left(\bigotimes_{i=1}^{i=n}\left[\underline{p}_i^{\gamma},\overline{p}_i^{\gamma}\right]\right)\times\left(\bigotimes_{i=1}^{i=n}\left[\underline{\omega}_i^{\gamma},\overline{\omega}_i^{\gamma}\right]\right)\times\left[\underline{r}^{\gamma},\overline{r}^{\gamma}\right]$. En se basant sur cette représentation, les mesures d'optimalité possible et nécessaire peuvent être développées. En premier, le concept d'optimalité possible pour une séquence de tâches fixe est présenté dans la Définition suivante.

Définition 2.8. *On dit qu'un ordonnancement fixe* π *est possiblement optimal si et seulement s'il existe une configuration* τ *pour laquelle l'ordonnancement* π *est optimal au sens ordinaire. Cette définition peut être formulé mathématiquement par :*
$\left(\pi\ est\ possiblement\ optimal\right)\Leftrightarrow\left(\exists\tau\in\Gamma(\gamma)/\pi\ est\ optimal\ pour\ \tau\right)$

L'optimalité possible est notée par $\mathrm{P}os_{\Gamma(\gamma)}$ et nous écrivons $\mathrm{P}os_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right)=1$ dans le cas où π est possiblement optimal pour $\Gamma(\gamma)$. Autrement, nous avons $\mathrm{P}os_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right)=0$. La propriété suivante est vérifiée pour la mesure d'optimalité possible.

Propriété 2.1. *En considérant un ordonnancement donné* π *et un réel* $\gamma\in[0,1]$*, nous avons* $\mathrm{P}os_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right)=1$ *si et seulement si le système d'inégalités suivant est faisable :*

$$\begin{cases} \underline{p}^{\gamma}_{\pi(i)} \le p_{\pi(i)} \le \bar{p}^{\gamma}_{\pi(i)} & i = \overline{1,n} \\ \underline{\omega}^{\gamma}_{\pi(i)} \le \omega_{\pi(i)} \le \bar{\omega}^{\gamma}_{\pi(i)} & i = \overline{1,n} \\ \underline{r}^{\gamma} \le r \le \bar{r}^{\gamma} & \\ \frac{e^{r p_{\pi(i)}} - 1}{\omega_{\pi(i)}} \le \frac{e^{r p_{\pi(i+1)}} - 1}{\omega_{\pi(i+1)}} & i = \overline{1,n-1} \end{cases} \tag{2.4}$$

Démonstration.
Necessité des conditions ($\Rightarrow$)
Supposons que $Pos_{\Gamma(\gamma)}(\pi \; est \; optimal) = 1$ pour un certain $\gamma \in [0,1]$. Alors, de la Définition 2.8, il existe au moins une configuration $\tau = (p_1,...,p_n,\omega_1,...,\omega_n,r) \in \Gamma(\gamma)$ pour laquelle π est optimal. Comme $\tau \in \Gamma(\gamma)$, nous déduisons que $\underline{p}^{\gamma}_i \le p_i \le \overline{p}^{\gamma}_i$, $\underline{\omega}^{\gamma}_i \le \omega_i \le \overline{\omega}^{\gamma}_i$ et $\underline{r}^{\gamma} \le r \le \overline{r}^{\gamma}$ pour $i = \overline{1,n-1}$. En outre, π est optimal pour τ d'où $\frac{e^{r p_{\pi(i)}} - 1}{\omega_{\pi(i)}} \le \frac{e^{r p_{\pi(i+1)}} - 1}{\omega_{\pi(i+1)}}$ pour $i = \overline{1,n-1}$. Ceci signifie que τ vérifie le système des inégalités et le système est faisable.
Suffisance des conditions ($\Leftarrow$)
Maintenant, nous supposons qu'il existe une configuration $\tau = (p_1,...,p_n,\omega_1,...,\omega_n,r)$ satisfaisant le système. Comme $p_i \in \left[\underline{p}^{\gamma}_i, \overline{p}^{\gamma}_i\right]$, $\omega_i \in [\underline{\omega}^{\gamma}_i, \overline{\omega}^{\gamma}_i]$, $r \in [\underline{r}^{\gamma}, \overline{r}^{\gamma}]$, nous avons donc $\tau \in \Gamma(\gamma)$. De plus, τ vérifie les conditions suffisantes d'optimalité de π (voir Théorème 2.2). Alors, π est optimal pour $\tau \in \Gamma(\gamma)$. De la Définition 2.8, nous obtenons $Pos_{\Gamma(\gamma)}(\pi \; est \; optimal) = 1$ □

Définition 2.9. *Pour un ordonnancement fixe* π*, le degré d'optimalité possible* Pos *est défini comme suit :*

$$Pos(\pi \; est \; optimal)$$
$$= \begin{cases} 0 \quad si \quad Pos_{\Gamma(0)}(\pi \; est \; optimal) = 0 \\ \sup\left\{\gamma : Pos_{\Gamma(\gamma)}(\pi \; est \; optimal) = 1, \; \gamma \in [0,\,1]\right\} \quad si \quad Pos_{\Gamma(0)}(\pi \; est \; optimal) = 1 \end{cases} \tag{2.5}$$

Le problème de calcul de $Pos(\pi \; est \; optimal)$ peut être simplifié en supposant que les paramètres d'ordonnancement sont donnés par des nombres flous trapézoïdaux. Nous supposons que $\tilde{p}_i = \left(\underline{p}_i, \overline{p}_i, \alpha^p_i, \beta^p_i\right)$, $\tilde{\omega}_i = \left(\underline{\omega}_i, \overline{\omega}_i, \alpha^{\omega}_i, \beta^{\omega}_i\right)$ et $\tilde{r} = \left(\underline{r}, \overline{r}, \alpha^r, \beta^r\right)$ pour $i = \overline{1,n}$. Il en résulte de la Définition 2.9 et la Propriété 2.1 que $Pos(\pi \; est \; optimal)$ est égale à la valeur maximale de la fonction objectif du problème d'optimisation suivant :

$$\begin{cases} \max\limits_{0\leq\gamma\leq 1}(\gamma) \\ Sujet\ \text{à} \\ \underline{p}_{\pi(i)} - \alpha^{p}_{\pi(i)}(1-\gamma) \leq p_{\pi(i)} \leq \bar{p}_{\pi(i)} + \beta^{p}_{\pi(i)}(1-\gamma) \quad i=\overline{1,n} \\ \underline{\omega}_{\pi(i)} - \alpha^{\omega}_{\pi(i)}(1-\gamma) \leq \omega_{\pi(i)} \leq \bar{\omega}_{\pi(i)} + \beta^{\omega}_{\pi(i)}(1-\gamma) \quad i=\overline{1,n} \\ \underline{r} - \alpha^{r}(1-\gamma) \leq r \leq \bar{r} + \beta^{r}(1-\gamma) \\ \frac{e^{rp_{\pi(i)}}-1}{\omega_{\pi(i)}} \leq \frac{e^{rp_{\pi(i+1)}}-1}{\omega_{\pi(i+1)}} \quad i=\overline{1,n-1} \end{cases} \tag{2.6}$$

Si ce problème n'est pas faisable, alors nous aurons $Pos(\pi\ est\ optimal) = 0$. Le concept d'optimalité nécessaire d'un ordonnancement fixe est donné par la Définition ci-dessous.

Définition 2.10. *On dit qu'un ordonnancement fixe* π *est nécessairement optimal si et seulement si l'ordonnancement* π *est optimal au sens ordinaire pour toutes les configurations. La formulation mathématique de cette définition est exprimée par :* $(\pi\ est\ nécessairement\ optimal) \Leftrightarrow (\forall \tau \in \Gamma(\gamma)/\pi\ est\ optimal\ pour\ \tau)$.

L'optimalité nécessaire est notée par $Nec_{\Gamma(\gamma)}$ et nous écrivons $Nec_{\Gamma(\gamma)}(\pi\ est\ optimal) = 1$ si π est nécessairement optimal pour $\Gamma(\gamma)$. Autrement, nous aurons $Nec_{\Gamma(\gamma)}(\pi\ est\ optimal) = 0$. La propriété suivante est satisfaite par la mesure d'optimalité nécessaire.

Propriété 2.2. *Pour un ordonnancement donné* π *et un certain réel* $\gamma \in [0,1]$*, la condition suffisante pour* $Nec_{\Gamma(\gamma)}(\pi\ est\ optimal) = 1$ *est que le système non linéaire suivant soit satisfait pour* $i=\overline{1,n-1}$ *:*

$$\frac{e^{\bar{r}^{\gamma}\bar{p}^{\gamma}_{\pi(i)}}-1}{\underline{\omega}^{\gamma}_{\pi(i)}} \leq \frac{e^{\underline{r}^{\gamma}\underline{p}^{\gamma}_{\pi(i+1)}}-1}{\overline{\omega}_{\pi(i+1)}} \tag{2.7}$$

Démonstration.
Necessité des conditions ($\Rightarrow$)
Nous démontrons en premier que ces conditions ne sont pas nécessaires. Nous supposons que $Nec_{\Gamma(\gamma)}(\pi\ est\ optimal) = 1$ pour une valeur fixe de $\gamma \in [0,1]$, à partir de la Définition 2.10, π est optimal pour toutes les configurations $(p_1,...,p_n,\omega_1,...,\omega_n,r) \in \Gamma(\gamma)$. Si ces conditions ne sont pas satisfaites, il existe $k \in \{1,...,n-1\}$ tel que la valeur de $\frac{e^{\bar{r}^{\gamma}\bar{p}^{\gamma}_{\pi(i)}}-1}{\underline{\omega}^{\gamma}_{\pi(i)}}$ est strictement supérieure à $\frac{e^{\underline{r}^{\gamma}\underline{p}^{\gamma}_{\pi(i+1)}}-1}{\overline{\omega}_{\pi(i+1)}}$. Cette inégalité implique que pour des valeurs arbitraires de $p_i \in \left[\underline{p}^{\gamma}_i, \overline{p}^{\gamma}_i\right]$, $\omega_i \in \left[\underline{\omega}^{\gamma}_i, \overline{\omega}^{\gamma}_i\right]$ et $r \in \left[\underline{r}^{\gamma}, \overline{r}^{\gamma}\right]$, nous obtenons les inégalités suivantes :

$$\begin{cases} \frac{e^{\bar{r}^{\gamma}\bar{p}^{\gamma}_{\pi(k)}}-1}{\underline{\omega}^{\gamma}_{\pi(k)}} \geq \frac{e^{rp_{\pi(k)}}-1}{\omega_{\pi(k)}} \\ \frac{e^{rp_{\pi(k+1)}}-1}{\omega_{\pi(k+1)}} \geq \frac{e^{\underline{r}^{\gamma}\underline{p}^{\gamma}_{\pi(k+1)}}-1}{\bar{\omega}_{\pi(k+1)}} \end{cases}$$

Nous distinguons deux cas $\frac{e^{r\underline{p}^{\gamma}_{\pi(k)}-1}}{\omega_{\pi(k)}} > \frac{e^{rp_{\pi(k+1)}-1}}{\omega_{\pi(k+1)}}$ ou $\frac{e^{r\underline{p}^{\gamma}_{\pi(k)}-1}}{\omega_{\pi(k)}} \leq \frac{e^{rp_{\pi(k+1)}-1}}{\omega_{\pi(k+1)}}$. Dans le cas de la dernière inégalité, pour toute configuration $\tau = \left(p_1,...,p_n,\omega_1,...,\omega_n,r\right) \in \Gamma(\gamma)$, nous concluons du Théorème 2.2 que π est optimal pour toutes les configurations (les conditions nécessaires d'optimalité sont vérifiées indépendamment de la valeur de γ), ce qui signifie que $Nec_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right) = 1$ même si ces conditions ne sont pas vérifiées. D'où, les conditions ne sont pas des conditions nécessaires.

Suffisance des conditions ($\Leftarrow$)

Supposons que les conditions sont satisfaites. Donc, pour chaque configuration $\tau = \left(p_1,...,p_n,\omega_1,...,\omega_n,r\right) \in \Gamma(\gamma)$ où $p_i \in \left[\underline{p}^{\gamma}_i,\overline{p}^{\gamma}_i\right]$,$\omega_i \in \left[\underline{\omega}^{\gamma}_i,\overline{\omega}^{\gamma}_i\right]$ et $r \in \left[\underline{r}^{\gamma},\overline{r}^{\gamma}\right]$, $(i = \overline{1,n})$, nous avons également $\frac{e^{rp_{\pi(i)}-1}}{\omega_{\pi(i)}} \leq \frac{e^{\overline{r}^{\gamma}\overline{p}^{\gamma}_{\pi(k+1)}-1}}{\underline{\omega}^{\gamma}_{\pi(i)}} \leq \frac{e^{\underline{r}^{\gamma}\underline{p}^{\gamma}_{\pi(i+1)}-1}}{\overline{\omega}^{\gamma}_{\pi(i+1)}} \leq \frac{e^{rp_{\pi(i+1)}-1}}{\omega_{\pi(i+1)}}$ pour $i = \overline{1,n-1}$. Donc, du Théorème 2.2, π est optimal pour toute configuration $\tau \in \Gamma(\gamma)$. De la Définition 2.10, nous concluons que $Nec_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right) = 1$.

□

Définition 2.11. *Pour un ordonnancement fixe π, le degré d'optimalité nécessaire* Nec *est exprimé par :*

$$Nec\left(\pi\ est\ optimal\right) =$$

$$\begin{cases} 0 \quad si \quad Nec_{\Gamma(1)}\left(\pi\ est\ optimal\right) = 0 \\ 1 - \inf\left\{\gamma : Nec_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right) = 1,\ \gamma \in [0,1]\right\} \quad si \quad Nec_{\Gamma(1)}\left(\pi\ est\ optimal\right) = 1 \end{cases} \tag{2.8}$$

L'extension du Lemme 2.1 au cas flou où tous les paramètres sont représentés par des nombres flous trapézoïdaux est illustrée par le Lemme suivant.

Lemme 2.2. *Dans le cas des valeurs très petites du taux d'actualisation flou, le problème* $1\left|\tilde{\omega}_i,\tilde{p}_i,\tilde{r}\right|\sum_{i=1}^{i=n}\omega_i\left(1-e^{-rC_i}\right)$ *se réduit au problème* $1\left|\tilde{\omega}_i,\tilde{p}_i\right|\sum_{i=1}^{i=n}\omega_i C_i$.

Démonstration.

Si le taux d'actualisation flou prend des valeurs faibles ($\overline{r} \to 0$), nous pouvons l'approximer par un nombre réel ($\tilde{r} \cong (r,r,0,0)$). Dans ce cas, la fonction objectif devient elle même une fonction floue notée par $\tilde{f}$. En utilisant la notation des coupes γ, nous obtenons pour $\gamma \in [0,1]$ l'expression $\left[\tilde{f}(r)\right]^{\gamma} = \left[\underline{f}(r,\gamma),\overline{f}(r,\gamma)\right]$ avec

$$\begin{cases} \underline{f}(r,\gamma) = \sum_{i=1}^{i=n}\underline{\omega}_i(\gamma) \times \left(1-e^{-r\underline{C}_i(\gamma)}\right) \\ \overline{f}(r,\gamma) = \sum_{i=1}^{i=n}\overline{\omega}_i(\gamma) \times \left(1-e^{-r\overline{C}_i(\gamma)}\right) \end{cases}$$

Comme ces bornes sont des fonctions réelles de r, le Lemme 2.1 peut être appliqué pour arriver à :

$$\begin{cases} \underline{f}(r,\gamma) \cong r \sum_{i=1}^{i=n} \underline{\omega}_i(\gamma) \times \underline{C}_i(\gamma) \\ \\ \overline{f}(r,\gamma) \cong r \sum_{i=1}^{i=n} \overline{\omega}_i(\gamma) \times \overline{C}_i(\gamma) \end{cases}$$

ou bien sous forme plus compacte $[\tilde{f}(r)]^{\gamma} \cong r\left[\sum_{i=1}^{i=n} \underline{\omega}_i(\gamma) \times \underline{C}_i(\gamma), \sum_{i=1}^{i=n} \overline{\omega}_i(\gamma) \times \overline{C}_i(\gamma)\right]$. Sachant que les intervalles $r\left[\sum_{i=1}^{i=n} \underline{\omega}_i(\gamma) \times \underline{C}_i(\gamma), \sum_{i=1}^{i=n} \overline{\omega}_i(\gamma) \times \overline{C}_i(\gamma)\right]$ pour $\gamma \in [0,1]$ peuvent être identifiés à une constante de multiplication près au coupes γ de la fonction objectif associée au problème $1\left|\tilde{\omega}_i, \tilde{p}_i\right| \sum_{i=1}^{i=n} \omega_i C_i$, nous déduisons que le problème $1\left|\tilde{\omega}_i, \tilde{p}_i, \tilde{r}\right| \sum_{i=1}^{i=n} \omega_i\left(1-e^{-rC_i}\right)$ coïncide bien avec le problème $1\left|\tilde{\omega}_i, \tilde{p}_i\right| \sum_{i=1}^{i=n} \omega_i C_i$ pour les valeurs très faibles du taux d'actualisation $\tilde{r}$. □

Quelques résultats asymptotiques pour le problème $1\left|\tilde{\omega}_i, \tilde{p}_i, \tilde{r}\right| \sum_{i=1}^{i=n} \omega_i\left(1-e^{-rC_i}\right)$ sont obtenus si les hypothèses suivantes sont imposées sur le taux d'actualisation comme indiqué par les Corollaires suivants.

Corollaire 2.1. *Si le taux d'actualisation prend des valeurs réelles i.e.* $\tilde{r} = (r,r,0,0)$*, l'ensemble des conditions suffisantes sont aussi des conditions nécessaires pour l'optimalité nécessaire d'un ordonnancement. Dans ce cas, le degré d'optimalité nécessaire est donné par* $\mathrm{N}ec\left(\pi\ est\ optimal\right) = 1-\gamma^*$ *avec* γ^* *est la valeur minimale de la fonction objectif du problème d'optimisation suivant :*

$$\begin{cases} \min\limits_{0\le\gamma\le 1}(\gamma) \\ \\ Sujet\ \text{à} \\ \\ \dfrac{e^{r\left[\bar{p}_{\pi(i)}+\beta^{p}_{\pi(i)}(1-\gamma)\right]}}{\left[\underline{\omega}_{\pi(i)}-\alpha^{\omega}_{\pi(i)}(1-\gamma)\right]} \le \dfrac{e^{r\left[\underline{p}_{\pi(i+1)}-\alpha^{p}_{\pi(i+1)}(1-\gamma)\right]}}{\left[\bar{\omega}_{\pi(i+1)}+\beta^{\omega}_{\pi(i+1)}(1-\gamma)\right]} \quad i=\overline{1,n-1} \end{cases} \tag{2.9}$$

Démonstration.
Supposons que $\mathrm{N}ec_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right) = 1$ pour une valeur donnée de $\gamma \in [0,1]$. Donc, de la Définition 2.10, π est optimal pour toutes les configurations $\left(p_1,...,p_n,\omega_1,...,\omega_n,r\right) \in \Gamma(\gamma)$. Si les conditions ne sont pas satisfaites alors il existe $k \in \{1,...,n-1\}$ tel que $\frac{e^{\overline{r}^{\gamma}\overline{p}^{\gamma}_{\pi(k)}-1}}{\underline{\omega}^{\gamma}_{\pi(k)}} > \frac{e^{\underline{r}^{\gamma}\underline{p}^{\gamma}_{\pi(k+1)}-1}}{\overline{\omega}^{\gamma}_{\pi(k+1)}}$. Nous considérons une configuration $\tau = \left(p_1,...,p_n,\omega_1,...,\omega_N,r\right) \in \Gamma(\gamma)$ telle que $p_{\pi(k)} = \overline{p}^{\gamma}_{\pi(k)}$, $p_{\pi(k+1)} = \underline{p}^{\gamma}_{\pi(k+1)}$, $\omega_{\pi(k)} = \underline{\omega}^{\gamma}_{\pi(k)}$, $\omega_{\pi(k+1)} = \overline{\omega}^{\gamma}_{\pi(k+1)}$ et $r = \underline{r}^{\gamma} = \overline{r}^{\gamma}$. À partir de la Propriété 2.2 et Théorème 2.2, nous concluons que π n'est pas optimal pour τ (les conditions nécessaires pour l'optimalité ne sont pas satisfaites), ce qui contredit notre hypothèse initiale $\mathrm{N}ec_{\Gamma(\gamma)}\left(\pi\ est\ optimal\right) = 1$. Donc, pour une valeur réelle du taux d'actualisation $\tilde{r}$, les conditions suffisantes sont également des conditions nécessaires. Dans ce cas, par application de la Définition 2.11 et la Propriété 2.2 nous obtenons le problème de minimisation (2.9). □

Corollaire 2.2. *Pour des petites valeurs du taux d'actualisation flou* $\tilde{r}$*, les assertions suivantes sont valides :*

i) Les conditions d'optimalité possible se réduisent à :

$$\begin{cases} \underline{p}_i^{\Upsilon} \leq p_i \leq \bar{p}_i^{\Upsilon} i = \overline{1,n} \\ \underline{\omega}_i^{\Upsilon} \leq \omega_i \leq \bar{\omega}_i^{\Upsilon} i = \overline{1,n} \\ \frac{p_{\pi(i)}}{\omega_{\pi(i)}} \leq \frac{p_{\pi(i+1)}}{\omega_{\pi(i+1)}} i = \overline{1,n-1} \end{cases} \tag{2.10}$$

ii) Les conditions suffisantes d'optimalité nécessaire se réduisent à :

$$\frac{\overline{p}_{\pi(i)}^{\Upsilon}}{\underline{\omega}_{\pi(i)}^{\Upsilon}} \leq \frac{\underline{p}_{\pi(i+1)}^{\Upsilon}}{\overline{\omega}_{\pi(i+1)}^{\Upsilon}} i = \overline{1,n-1} \tag{2.11}$$

qui sont aussi des conditions nécessaires.

Démonstration.

i) Par la réduction de notre problème flou $1\left|\tilde{\omega}_i, \tilde{p}_i, \tilde{r}\right| \sum_{i=1}^{i=n} \omega_i \left(1 - e^{-r\mathrm{C}_i}\right)$ utilisant le Lemme 2.2 au problème $1\left|\tilde{\omega}_i, \tilde{p}_i\right| \sum_{i=1}^{i=n} \omega_i \mathrm{C}_i$ et en suivant la même procédure de démonstration de la Propriété 2.1, nous déduisons que la faisabilité du système d'inégalités :

$$\begin{cases} \underline{p}_i^{\Upsilon} \leq p_i \leq \bar{p}_i^{\Upsilon} i = \overline{1,n} \\ \underline{\omega}_i^{\Upsilon} \leq \omega_i \leq \bar{\omega}_i^{\Upsilon} i = \overline{1,n} \\ \frac{p_{\pi(i)}}{\omega_{\pi(i)}} \leq \frac{p_{\pi(i+1)}}{\omega_{\pi(i+1)}} i = \overline{1,n-1} \end{cases}$$

est une condition nécessaire et suffisante pour l'optimalité possible d'un ordonnancement.

ii) Nous utilisons les mêmes procédures comme pour Propriété 2.2 et Corollaire 2.1 pour le problème réduit $1\left|\tilde{\omega}_i, \tilde{p}_i\right| \sum_{i=1}^{i=n} \omega_i \mathrm{C}_i$, nous déduisons que la faisabilité du système d'inégalités :

$$\frac{\overline{p}_{\pi(i)}^{\Upsilon}}{\underline{\omega}_{\pi(i)}^{\Upsilon}} \leq \frac{\underline{p}_{\pi(i+1)}^{\Upsilon}}{\overline{\omega}_{\pi(i+1)}^{\Upsilon}} i = \overline{1,n-1}$$

est une condition nécessaire et suffisante pour l'optimalité nécessaire d'un ordonnancement.

□

En se basant sur les modèles compacts associés aux mesures d'optimalité possible et nécessaire, nous pouvons observer que ces modèles représentent des problèmes d'optimisation continue avec contraintes non linéaires. Comme dans plusieurs cas, ce type de problème est difficile à résoudre à cause de la présence des régions faisables non connexes dans l'espace de recherche ainsi que la présence de minimum locaux,

il est commode d'opter pour des techniques plus fiables telles que les approches évolutionnaires émergentes ces dernières années comme outils puissants pour la résolution et l'analyse des problèmes d'optimisation ([CASTILLO et MELIN, 2009] et [KROLL et SCHULTE, 2014]).

2.4 Conception du cadre d'expérimentations numériques

Dans cette section, nous présentons le cadre de simulation utilisé pour l'évaluation du degré d'optimalité possible d'un ordonnancement bien défini. Par contre, le degré d'optimalité nécessaire n'est pas considéré car c'est une condition forte d'optimalité qui ne peut être satisfaite que dans des cas très rares par exemple lorsque le taux d'actualisation est connu d'une façon exacte. Le formalisme Lagrangien augmenté est utilisé pour intégrer les contraintes non linéaires dans la fonction objectif et ceci dans le contexte des algorithmes d'optimisation proposés (algorithme génétique et recherche par motifs).

2.4.1 Support des contraintes non linéaires

Les problèmes d'optimisation continue non linéaires ont représentés un grand souci en pratique due à la difficulté de leur adaptation aux méthodes conventionnelles. L'approche basée sur un coût de pénalité est parmi plusieurs approches proposées pour remédier tel défi où les contraintes sont ajoutées à la fonction objectif [COELLO COELLO, 2002]. Lorsque une solution s'éloigne de la région faisable pendant le processus de recherche, elle est pénalisée par l'accroissement de la fonction objectif avec un terme de pénalité multiplié par un coefficient de pénalité [NOCEDAL et WRIGH, 1999]. La violation de contrainte est pénalisée par l'augmentation de ce coefficient d'une manière graduelle, ce qui force le processus de recherche de se rediriger vers les régions faisables du problème à résoudre. Le problème majeur dans les fonctions de pénalités et de barrières simples consiste en l'effet de bord associé au mal conditionnement. Cet obstacle peut être évité dans le cas des approches à base du Lagrangien augmenté ayant plusieurs avantages en comparaison avec d'autres techniques de programmation non linéaire [SRIVASTAVA et DEB, 2010] :

- Une solution exacte peut être obtenue ce qui est faux pour les méthodes basées sur des fonctions de pénalités simples ;
- La fonction objectif originale n'est pas distorsée, au lieu elle est décalée de telle sorte que l'optimum se déplace du minimum sans contraintes vers le minimum sous contraintes ;
- La valeur du multiplicateur de Lagrange est donnée pour chaque contrainte. Cette possibilité de calculer le multiplicateur de Lagrange offre plus de flexibilité par rapport aux autres algorithmes d'optimisation.

Cependant, parmi les critiques des méthodes à base du Lagrangien augmenté nous trouvons la nécessité d'un nombre d'itération important (méthodes gourmandes en temps de calcul) ainsi que la dépendance des phases d'optimisation actuelles de la précision des phases précédentes. Pour un problème d'optimisation continue avec des contraintes non linéaires, linéaires et de bornes, la formulation mathématique générique est donnée par :

$$\begin{cases} \min\limits_{x} \mathrm{F}(x) \\ Sujet\ \text{à} \\ \mathrm{C}_i^{ineq}(x) \leq 0,\ i = \overline{1,\mathrm{M}} \\ \mathrm{C}_i^{eq}(x) = 0,\ i = \overline{1,\mathrm{N}} \\ \mathrm{A}^{ineq} x \leq \mathrm{B}^{ineq} \\ \mathrm{A}^{eq} x = \mathrm{B}^{eq} \\ \mathrm{L}^b \leq x \leq \mathrm{U}^b \end{cases} \tag{2.12}$$

où $\mathrm{C}_i^{ineq}(x)$ représente M contraintes d'inégalité non linéaires, $\mathrm{C}_i^{eq}(x)$ représentent N contraintes d'égalité non linéaires, A^{ineq} la matrice des coefficients des contraintes d'inégalité linéaires, A^{eq} la matrice des coefficients des contraintes d'égalité linéaires, L^b et U^b sont les vecteurs des bornes inférieures et supérieures.

Un sous problème est formulé par la définition d'une nouvelle fonction objectif de la somme pondérée de la fonction objectif originale et les contraintes non linéaires utilisant les paramètres de Lagrange et de pénalité. La formulation compacte du sous problème résultant est donnée par [Conn et collab., 1997] :

$$\Psi\left(x,\lambda,s,\rho\right) = \mathrm{F}(x) - \sum_{i=1}^{\mathrm{M}} \lambda_i s_i \log\left(s_i - \mathrm{C}_i^{ineq}(x)\right) + \sum_{i=1}^{\mathrm{N}} \lambda_i \mathrm{C}_i^{eq}(x) + \frac{\rho}{2} \sum_{i=1}^{\mathrm{N}} \left(\mathrm{C}_i^{eq}(x)\right)^2 \tag{2.13}$$

Donc, nous commençons par l'affectation d'une valeur initiale au paramètre de pénalité, une séquence de sous problèmes (qui sont des approximations du problème original) est minimisée selon une méthode itérative appropriée ou une heuristique de telle sorte que les contraintes linéaires et de bornes soient satisfaites et traitées séparément. À chaque itération, si la précision nécessaire du minimum du sous problème est atteinte, les éléments estimés du vecteur de multiplicateur de Lagrange sont mis à jour. Sinon, le paramètre de pénalité est augmenté par un facteur de pénalité. Ceci donne un nouveau sous problème à minimiser suivant la même stratégie. Ces étapes sont réitérées jusqu'à l'une des conditions d'arrêt imposées sur le nombre de générations, violation de contraintes ou l'erreur sur la fonction objectif soit vérifiée.

2.4.2 Approche basée sur l'algorithme génétique

Les algorithmes génétiques introduits initialement dans les années soixante par John Holland sont des techniques de recherche stochastiques basées sur la génération d'une population de solutions à chaque itération. L'évolution de la population est guidée par plusieurs opérateurs (sélection, croisement et mutation) permettant la couverture de l'espace de solutions pendant la recherche [Glover et Kochenberger, 2003]. L'évolution de la population entre différentes itérations est gouvernée par la configuration adéquate des opérateurs génétiques responsables des bonnes performances de l'algorithme génétique. Pour atteindre ce but, des efforts supplémentaires doivent être focalisés sur ces

opérateurs lors de la conception d'un algorithme génétique destiné à un problème. En effet, dans plusieurs cas un tel algorithme dépend des caractéristiques intrinsèques du problème analysé. Par exemple, le codage par permutation est largement utilisé pour la représentation des séquences des tâches dans les problèmes d'ordonnancement NP-difficiles à une machine [LOPEZ et ROUBELLAT, 2000]. Malgré que la représentation binaire classique soit la plus répondue, notre implémentation est basée sur la représentation réelle à cause de la nature du problème étudié. De plus, des études ont confirmé que le codage réel est rapide, plus consistant et donne une meilleure précision par rapport au codage binaire [MICHALEWICZ, 1998]. Chaque individu dans la population est codé comme un vecteur de dimension $2n+2$.

Le processus de sélection des parents pour la prochaine génération est achevé utilisant une sélection stochastique uniforme pour assurer que la fréquence de sélection de chaque individu soit en ligne avec les fréquences attendues. Une valeur unique aléatoire est utilisée pour sélectionner toutes les solutions sur des intervalles équivaux d'une ligne où chaque parent correspond à une section de la ligne de longueur proportionnelle à la valeur normalisée de sa fonction objectif. Cette procédure donne aux individus faibles de la population une chance d'être choisis et permet donc de réduire le risque de la convergence prématurée de la population. En utilisant l'opérateur de croisement, il est possible de transférer les bonnes propriétés des individus à leurs descendants. Dans le croisement uniforme adopté dans ce travail, un vecteur aléatoire (ou masque) contenant des valeurs binaires est créé et par la suite la génération d'un enfant est faite par le test du masque bit par bit. Si le bit testé est zéro alors le génotype du parent est extrait tandis que le génotype du deuxième parent est considéré si le bit est à un. Un deuxième enfant peut être obtenu en inversant l'ordre des parents. Les enfants sont par la suite sujets à une mutation gaussienne où une valeur distribuée selon une loi gaussienne est ajoutée à un génotype choisi d'un enfant. Si la valeur résultante viole la contrainte de borne pour ce génotype, cette nouvelle valeur est écartée. Finalement pour la sélection élitiste ayant pour objectif de conserver l'amélioration continue de la meilleur solution d'une itération à une autre, une fraction des meilleurs individus est transferée directement à la prochaine génération sans subir les opérateurs de croisement et de mutation. Les étapes élémentaires de l'algorithme génétique sont illustrées par l'Algorithme 2.1.

Algorithm 2.1 Algorithme génétique

DEBUT
Initialiser les paramètres de l'algorithme génétique
POUR ($k = 1$ au nombre maximum d'itérations) **FAIRE**
  Évaluer la fonction objectif pour tous les individus de la population
  Choisir les parents en se basant sur une stratégie de sélection
  Appliquer l'opérateur de croisement avec le taux de croisement associé
  Appliquer l'opérateur de mutation avec le taux de mutation associé
  Préserver les individus élites et remplacer les individus restants
FIN POUR
Retourner la meilleure solution correspondante au minimum de la fonction objectif f
FIN

Dans leur forme standard, les algorithmes génétiques ont plus de chance pour éviter les optimums locaux en comparaison avec les techniques de programmation déterministes comme ces algorithmes ne nécessitent pas le calcul du gradient de la fonction objectif. Mais d'autre part, ces algorithmes supportent uniquement les contraintes de bornes et les individus qui violent ces contraintes peuvent être éliminés sans difficultés.

Ceci est inefficace dans le cas où les solutions faisables sont difficilement déterminées. En se basant sur les bonnes performances des métaheuristiques intégrées avec des fonctions de pénalité à base du lagrangien augmenté, des approches très prometteuses ont été développées pour des applications industrielles ([DEB et SRIVASTAVA, 2012], [RAHIM et collab., 2012], [ROCHA et collab., 2011] et [TALBI, 2013]).

2.4.3 Approche basée sur la recherche par motifs

La recherche par motifs est une sous classe de la famille des méthodes de recherche directionnelles proposées initialement par Lewis et Torczon [LEWIS et TORCZON, 1999]. Cette méthode ne nécessite pas le calcul du gradient ou d'une dérivée d'ordre supérieure pendant le processus de recherche de solution optimale. Son principe consiste en l'exploitation d'un ensemble de points voisins de la solution courante avec l'objectif de trouver un point ayant une valeur plus faible de la fonction objectif. La génération du voisinage est fondée sur un maillage construit sur la base d'un ensemble de vecteurs connus sous le nom de motifs définissant les directions de recherche. Le maillage est caractérisé par une taille représentant une sorte de distance entre le point courant et un point arbitraire du voisinage. Sachant que le cardinal d'une base positive de $\mathbb{R}^d$ est limité dans l'intervalle $[d+1, 2d]$ ([HARE et SONG, 2016]), deux stratégies élémentaires sont implémentées en se basant sur les valeurs extrêmes de l'intervalle pour définir un motif. La première stratégie est de considérer d vecteurs de base en plus du vecteur obtenu par l'inverse de la somme de ces d vecteurs. La deuxième stratégie est de considérer $2d$ vecteurs définis par les d vecteurs de base ainsi que les d vecteurs inverses [LEWIS et collab., 2000]. Dans notre travail, nous adoptons la deuxième stratégie ce qui est justifié par l'obtention d'un nombre plus élevé de directions de recherche.

Pour une fonction objectif donnée, la méthode de recherche par motifs commence avec une solution initiale x_0 et une taille de maillage initiale $\Delta_0 > 0$. Une séquence de solutions itérées $\{x_k\}$ est générée telle que $f(x_{k+1}) \leq f(x_k)$ et à chaque itération la méthode sauvegarde les points du maillage en évaluant leurs valeurs de la fonction objectif. Ce processus de sauvegarde est connu sous le nom de procédure de vote. Si un point a une valeur inférieure par rapport au point courant, le vote est dit réussi et ce point devient le centre courant du nouveau maillage créé selon la formule :

$$M_k = \left\{ x_k + \Delta_k \nu_z : z \in Z_+^{|\nu|} \right\} \tag{2.14}$$

où Δ_k le paramètre de maillage, $\mathbb{Z}_+$ l'ensemble des entiers non négatifs et $|\nu|$ le cardinal de la base positive. Autrement, le vote est dit échoué et le point courant est préservé avec le retrissage de l'espace de recherche [TORCZON, 1997]. Nous pouvons déduire que la structure du voisinage est modifiée d'une façon dynamique avec l'évolution de l'algorithme à cause de la taille du maillage qui dépend de l'état du vote (réussi ou échoué) ainsi que du nombre d'itérations selon l'équation de récurrence suivante :

$$\Delta_{k+1} = \tau \Delta_k \tag{2.15}$$

avec

$$\begin{cases} \tau < 1 \ si \ le \ vote \ est \ réussi \\ \\ \tau > 1 \ si \ le \ vote \ est \ échoué \end{cases}$$

En effet, les mécanismes d'exploiatation dans la méthode de recherche par motifs contiennent deux types dans le parcours de l'espace de solutions : une étape de recherche optionnelle et une étape de vote (enregistrement). Nous utilisons comme mécanisme de recherche l'ensemble de points de voisinage obtenu par la génération de $2n_v$ vecteurs aléatoires avec n_v le nombre de variables indépendantes de la fonction objectif $n_v = 2n + 2$. La différence entre les mécanismes de vote et de recherche optionnelle réside dans la manière de sélection des points tests. L'étape de recherche optionnelle est plus flexible car une heuristique ou règle de dominance peut être utilisée pour ce but. Mais il est à noter que l'étape de vote est l'élément primordial dans le paradigme de la recherche par motifs qui est plus rigide et élaborée. À ce niveau, on a entre les mains tous les ingrédients nécessaires pour la description de la méthode comme illustré par l'Algorithme 2.2.

Algorithm 2.2 Recherche par motifs

DEBUT
Initialiser les paramètres de la recherche par motifs
POUR ($k = 1$ au nombre maximum d'itérations) **FAIRE**
  Effectuer l'étape de recherche optionnelle pour obtenir une solution améliorée
  SI (l'étape de recherche est échouée) **ALORS**
    Créer l'ensemble des directions de vote
    Effectuer l'étape de vote pour obtenir une solution améliorée
  FIN SI
  SI (une solution améliorée est trouvée) **ALORS**
    Considérer la solution améliorée comme solution courante
    Étendre la taille du maillage
  SINON
    Réduire la taille du maillage
  FIN SI
FIN POUR
Retourner la meilleure solution correspondante au minimum de la fonction objectif f
FIN

Il est à mentionner que pour des problèmes continus sous contraintes, les contraintes de bornes peuvent être supportées par l'élimination des points testés non faisables où la fonction objectif est évaluée uniquement aux points faisables respectant les contraintes de bornes. D'autres parts, la délimitation de la région faisable lorsque des contraintes non linéaires compliquées sont présentes nécessite plus d'efforts et peut être dans certains cas une tâche fastidieuse. Aussi, la considération d'une fonction objectif basée sur une pénalité du lagrangien augmenté peut aider à surmonter cet obstacle et présenter une bonne alternative pour la résolution du problème d'optimisation non linéaire avec contraintes ([ARTIGUES et collab., 2008] et [COSTA et collab., 2012]).

2.4.4 Génération des instances

Dans le but d'évaluer les performances des approches proposées, il est nécessaire de générer un ensemble d'instances de test. Chaque instance est composée d'un ensemble fixe de tâches caractérisées par leurs durées opératoires, coefficients de pondération et taux d'actualisation, où les nombres flous trapézoïdaux sont utilisés pour la modélisation d'incertitude au niveau de ces paramètres. On s'inspire de la procédure de génération utilisée pour l'obtention des nombres flous triangulaires à partir des nombres réels

comme illustré dans la référence [GHRAYEB, 2003]. Un paramètre générique noté par $\left(\tilde{Z}_i = \left(\underline{z}_i, \overline{z}_i, \alpha_i^Z, \beta_i^Z\right)\right)$ exprimant l'un des paramètres flous d'ordonnancement $\tilde{p}_i$, $\tilde{\omega}_i$ et $\tilde{r}$, est généré selon les règles suivantes :

- Initialement, une valeur réelle arbitraire z_i pour le paramètre $\tilde{Z}_i$ est sélectionnée d'une distribution uniforme sur l'intervalle $\left[a_i^Z, b_i^Z\right]$ pour chaque tâche i;
- Après ce choix pour toutes les tâches, les bornes inférieures et supérieures de $\tilde{Z}_i$ sont générées aléatoirement selon l'expression :

$$\begin{cases} \underline{z}_i \leftarrow \left[\eta \times z_i, z_i\right] \\ \bar{z}_i \leftarrow \left[z_i, \left(1+\eta\right) \times z_i\right] \end{cases}$$

  avec le paramètre $\eta \in (0,1]$ une graine aléatoire;
- Par la suite, les paramètres α_i^Z et β_i^Z sont définis comme des entiers arbitraires des intervalles spécifiés par $\left[a_i^{\alpha_Z}, b_i^{\alpha_Z}\right]$ et $\left[a_i^{\beta_Z}, b_i^{\beta_Z}\right]$ respectivement;
- Une valeur élevée de la graine aléatoire η indique que nous avons en général un intervalle étendu pour les nombres flous et une valeur faible indique un intervalle étroit de valeurs affectées au paramètres flous.

2.5 Simulation et résultats

Pour valider les approches de résolution proposées, des expérimentations numériques sont réalisées utilisant un ensemble de problèmes tests. L'ensemble test contient des instances de différentes tailles allant de $n = 2$ jusqu'à $n = 6$. En total, 30 instances sont générées pour chaque nombre de tâches. Mais seulement un nombre limité sélectionné arbitrairement est utilisé dans l'optimisation à cause du temps de calcul. La génération des instances et les algorithmes de résolution sont codés utilisant l'outil MATLAB qui est devenu actuellement un environnement intégré puissant de modélisation des problèmes difficiles d'engineering [WONGA et LAI, 2011]. Les expériences numériques sont effectuées sur un processeur Intel Core i7 avec 8 GB de RAM. L'ensemble complet des instances avec toutes les données associées sont disponibles sur le lien : <http://lab.univ-batna2.dz/lap/index.php/component/content/article?id=31>.
Comme le comportement des approches est basé sur quelques paramètres aléatoires, chaque instance est résolue répétitivement n_{rep} fois avec des configurations initiales différentes. En se basant sur les résultats obtenus, il est possible d'analyser la sensibilité de la qualité des solutions face à la variation de quelques paramètres. Le Tableau 2.1 présente les valeurs choisies pour les bornes inférieures et supérieures utilisées dans la génération des nombres flous trapézoïdaux associés à chaque paramètre dans notre problème d'ordonnancement.

Le Tableau 2.2 illustre les paramètres principaux de l'algorithme génétique et la recherche par motifs. Il est à noter que tous les paramètres des approches proposées sont fixés et varient uniquement lorsque nous effectuons une analyse de sensibilité d'un paramètre spécifique. Dans ce travail, aucun effort n'est dédié à la détermination de la configuration optimale des métaheuristiques à cause des coûts de calcul élevés.

L'influence du changement de quelques paramètres sur la qualité de la solution est exploitée ci-dessous où nous adoptons une notation à trois champs pour les instances de test Sol (X) (Y) (Z). La signification de chaque champ est comme suit :

- X : la valeur de la graine aléatoire;

TABLEAU 2.1 Valeurs des bornes inférieures et supérieures des paramètres utilisées lors de la procédure de génération des instances

Paramètre	Borne inférieure	Borne supérieure
p_i	1	10
α_i^p	1	8
β_i^p	1	8
ω_i	1	5
α_i^ω	1	8
β_i^ω	1	8
r	0.1	0.04
α^r	0.001	0.005
β^r	0.001	0.005

TABLEAU 2.2 Valeurs des paramètres de configuration utilisées dans les approches d'optimisation proposées

Paramètre	Sous paramètre	Algorithme génétique	Recherche par motifs
Nombre des éxecutions répétées		30	30
Nombre des individus élites		5	/
Nombre des itérations		200	200
Taille de la population		50	/
Taux de croisement		0.8	/
Maillage	Taille initiale	/	10
	Facteur d'expansion	/	10
	Facteur de contraction	/	10
Pénalité	Valeur initiale	5	10
	Facteur d'augmentation	25	10
Tolérance	Violation des contraintes	10^{-6}	10^{-6}
	Fonction objectif	10^{-6}	10^{-6}

- Y : le nombre de tâches ;
- Z : l'indice de l'instance.

Quatre critères statistiques sont utilisés pour l'évaluation des performances des deux approches : la meilleure valeur de la fonction objectif trouvée après n_{rep} exécutions ($\mathrm{F_{B}}est$), la valeur moyenne de toutes les valeurs de la fonction objectif ($\mathrm{F_{M}}ean$), l'erreur moyenne entre les valeurs de la fonction objectif et sa meilleure valeur $\left(\mathrm{F_{A}}vg\right)$ et finalement l'écart type des valeurs de la fonction objectif ($\mathrm{F_{S}}td$). Les formulations mathématiques de ces critères sont données par :

$$\mathrm{F_{B}}est = \max_{j=1,n_{rep}} \left(f_j\right) \tag{2.16}$$

$$\mathrm{F_{M}}ean = \frac{\sum_{j=1}^{n_{rep}} f_j}{n_{rep}} \tag{2.17}$$

$$\mathrm{F_{A}}vg = \frac{\sum_{j=1}^{n_{rep}} \left|f_j - \mathrm{F_{B}}est\right|}{n_{rep}} \tag{2.18}$$

$$\mathrm{F_{S}}td = \frac{\sum_{j=1}^{n_{rep}} \left(f_j - \mathrm{F_{M}}ean\right)^2}{n_{rep}} \tag{2.19}$$

avec f_j est le degré d'optimalité possible obtenu lors de la répétition j.

2.5.1 Effet du nombre de tâches

Du Tableau 2.3 représentant les performances des approches proposées en fonction du nombre de tâches, il s'ensuit que l'augmentation du nombre de tâches affecte la qualité des solutions. Il est à noter que pour les instances avec deux, trois et quatre tâches, les approches proposées sont capables de résoudre d'une façon optimale le problème de maximisation. Par contre, pour un nombre plus élevé de tâches, la qualité des solutions obtenues est dégradée comme il est reflété par l'augmentation des valeurs de l'erreur moyenne et l'écart type. Cette observation peut être expliquée par le fait que la considération de contraintes supplémentaires rend l'espace de recherche faisable plus difficile à exploiter. En outre, la différence entre des valeurs meilleure et moyenne de la fonction objectif pour la méthode de recherche par motifs est plus faible et strictement inférieure à celle associée à l'algorithme génétique. Cette tendance montre que la méthode de recherche par motifs est plus stable envers la variation du nombre de tâches en comparaison avec l'algorithme génétique à cause de l'aspect stochastique caractérisant ce dernier.

2.5.2 Effet de la graine aléatoire η

La graine aléatoire (η) joue un rôle crucial dans la procédure de génération car elle permet de contrôler l'intervalle des valeurs associées au différents paramètres d'ordonnancement. Le Tableau 2.4 introduit les performances des approches proposées en fonction des valeurs de η, où le nombre de tâches est fixé à quatre. Il est facile de déduire que la qualité des solutions générées par l'algorithme génétique utilisant des valeurs élevées

TABLEAU 2.3 Évolution des performances des approches proposées en fonction du nombre de tâches n

Instance	Algorithme génétique				Recherche par motifs			
	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}
Sol(0.2)(2)(2)	1*	1	0	0	1*	1	0	0
Sol(0.2)(3)(7)	1*	1	0	0	1*	1	0	0
Sol(0.2)(4)(15)	1*	1	0	0	1*	1	0	0
Sol(0.2)(5)(21)	0.6	0.53	0.07	0.07	0.6	0.58	0.02	0.02
Sol(0.2)(6)(1)	0.71	0.47	0.24	0.08	0.88	0.87	0.01	0.03

* : Valeur optimale

de η est strictement inférieure à celle associée aux valeurs faibles de η. Ceci peut être justifié par l'aspect stochastique de l'algorithme en plus de l'intervalle étroit des valeurs des paramètres d'ordonnancement lorsque η prend des petites valeurs. Alors, il est possible d'obtenir de meilleures solutions en comparaison avec le cas où les valeurs de η sont élevées. La méthode de recherche par motifs conserve sa stabilité même pour des valeurs élevées du paramètre η où la meilleure valeur et la valeur moyenne de la fonction objectif sont égales.

TABLEAU 2.4 Évolution des performances des approches proposées pour différentes valeurs de la graine aléatoire η

Instance	Algorithme génétique				Recherche par motifs			
	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}
Sol(0.2)(4)(6)	0.94	0.74	0.20	0.08	1*	1	0	0
Sol(0.4)(4)(13)	0.88	0.84	0.04	0.06	0.88	0.88	0	0
Sol(0.6)(4)(19)	1*	0.89	0.11	0.07	1*	1	0	0
Sol(0.8)(4)(24)	0.81	0.68	0.14	0.10	0.81	0.81	0	0
Sol(1)(4)(29)	0.91	0.73	0.18	0.11	0.97	0.97	0	0

* : Valeur optimale

2.5.3 Effet de la taille de la population de l'algorithme génétique

Pour analyser l'effet de la taille de la population sur les performances de l'algorithme génétique en comparaison avec la recherche par motifs, quelques instances sont choisies et résolues avec quatre valeurs différentes de la taille de la population. Dans ces expériences, nous prenons des populations formées de 20, 40, 60 et 80 individus. Les résultats concernant les critères statistiques obtenus après 30 répétitions sont listées dans le Tableau 2.5. Dans ce cas, nous pouvons conclure que l'efficacité et la consistance de l'algorithme pour de grandes valeurs de la graine aléatoire ne dépend pas fortement de la taille de la population car l'intervalle large des valeurs attachées au problème d'ordonnancement rend la recherche des régions faisables plus compliquée. Pour le nombre maximum de la taille de la population (80), la meilleure solution de l'algorithme génétique se rapproche de celle de la recherche par motifs mais de moindre qualité.

TABLEAU 2.5 Influence de la taille de la population p_{Size} sur les performances de l'algorithme génétique

Instance		Algorithme génétique				Recherche par motifs			
	p_{Size}	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}
Sol(0.8)(2)(7)	20	1*	1	0	0	1*	1	0	0
	40	1*	1	0	0				
	60	1*	1	0	0				
	80	1*	1	0	0				
Sol(0.4)(4)(11)	20	0.79	0.51	0.27	0.10	0.93	0.93	0	0
	40	0.85	0.65	0.20	0.11				
	60	0.91	0.75	0.16	0.06				
	80	0.92	0.81	0.11	0.06				
Sol(0.2)(4)(6)	20	0.89	0.58	0.31	0.15	1*	1	0	0
	40	0.88	0.61	0.27	0.14				
	60	0.99	0.66	0.33	0.16				
	80	1*	0.69	0.31	0.15				

* : Valeur optimale

2.5.4 Effet de la réduction du taux d'actualisation flou $\tilde{r}$

Dans le but d'exploiter l'impact du taux d'actualisation flou sur les performances des approches proposées, nous adoptons les données du premier exemple dans [CHANAS et KASPERSKI, 2004] pour notre problème avec l'introduction d'une valeur initiale arbitraire pour le taux d'actualisation flou $\tilde{r} = \left(3\times10^{-2}, 6\times10^{-2}, 2\times10^{-3}, 1\times10^{-3}\right)$. Après, nous réduisons cette valeur graduellement en la divisant par un facteur puissance de 2 et l'instance résultante est résolue comme il est montré dans le Tableau 2.6. Nous pouvons voir que les meilleures valeurs de la fonction objectif fluctuent autour de la valeur optimale de la fonction objectif associée au problème d'ordonnancement avec coûts non actualisés $F_{Opt} = 0.316$ pour l'algorithme génétique et un comportement similaire est détecté pour la recherche par motifs où la meilleure valeur de la fonction objectif converge vers la même valeur optimale. En outre, la recherche par motifs montre un comportement plus stable en fonction de la variation du taux d'actualisation flou comme indiqué par les critères de qualité. La convergence de la fonction objectif vers une valeur limite peut être considérée comme une pseudo validation numérique du Lemme 2.2.

TABLEAU 2.6 Évolution des performances des approches proposées avec la réduction du taux d'actualisation flou $\tilde{r}$

Diviseur	Algorithme génétique				Recherche par motifs				Valeur optimale
	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}	F_{Opt}
2	0.317	0.225	0.092	0.048	0.417	0.405	0.013	0.053	0.316
4	0.313	0.203	0.110	0.044	0.351	0.341	0.011	0.004	
8	0.309	0.200	0.108	0.045	0.331	0.300	0.036	0.046	
16	0.326	0.202	0.124	0.061	0.329	0.245	0.087	0.071	
32	0.315	0.200	0.115	0.069	0.319	0.239	0.079	0.054	

2.5.5 Effet de l'hybridation des approches algorithme génétique et recherche par motifs

Généralement, il est bien prédictible que l'hybridation entre différentes méthodes d'optimisation peut donner une amélioration dans les performances. Les méthodes de recherche par motifs souffrent de la limitation de la taille du voisinage généré dans les phases de recherche optionnelle et de vote. Cette taille dépend d'une façon polynômiale du nombre de paramètres du problème d'ordonnancement i.e. $O(2n+2)$. Donc, la qualité des solutions obtenues par cette famille de méthode peut être dégradée à cause de cette limitation. Pour étendre les possibilités de recherche, nous proposons d'utiliser un algorithme génétique dans la phase de recherche optionnelle sans modifier le reste des phases. Cette hybridation permet d'augmenter le nombre de solutions du voisinage exploité d'une manière aléatoire et d'offrir un compromis entre la diversification et l'intensification pendent la recherche.

Une comparaison des résultats obtenus par les approches élémentaires et l'approche hybride est illustrée dans les Tableaux 2.7 et 2.8, où le nombre de tâches est fixé à 5. Nous pouvons observer que l'approche hybride domine les deux approches élémentaires en terme de la meilleure valeur de la fonction objectif. Ceci confirme l'avantage d'inclure une autre méthode d'optimisation dans le contexte de la recherche par motifs comme phase de recherche optionnelle.

TABLEAU 2.7 Performances des approches proposées utilisées séparément

Instance	Algorithme génétique				Recherche par motifs			
	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}
Sol(0.2)(5)(1)	0.950	0.721	0.229	0.117	0.882	0.870	0.012	0.027
Sol(0.4)(5)(8)	0.924	0.644	0.281	0.104	0.965	0.964	0.001	0.001
Sol(0.6)(5)(15)	0.720	0.569	0.151	0.095	0.940	0.940	0	0
Sol(0.8)(4)(22)	0.850	0.568	0.282	0.125	0.851	0.840	0.011	0.014
Sol(1)(5)(30)	0.503	0.276	0.227	0.096	0.841	0.840	0.002	0.003

* : Valeur optimale

TABLEAU 2.8 Performances des approches proposées après hybridation

Instance	Approche hybride			
	F_{Best}	F_{Mean}	F_{Avg}	F_{Std}
Sol(0.2)(5)(1)	1*	1	0	0
Sol(0.4)(5)(8)	0.965	0.964	0.001	0.001
Sol(0.6)(5)(15)	0.940	0.940	0.001	0.004
Sol(0.8)(4)(22)	0.851	0.777	0.073	0.067
Sol(1)(5)(30)	0.841	0.813	0.028	0.054

* : Valeur optimale

2.6 Conclusion

Dans ce chapitre, nous avons presenté le problème d'ordonnancement à une machine avec des coûts actualisés et des paramètres incertains. Ce problème diffère des autres problèmes d'ordonnancement conventionnels dans le fait que la notion d'optimalité ordinaire est remplacée par les degrés possible et nécessaire pour évaluer l'optimalité d'un ordonnancement fixe. Le calcul du degré d'optimalité possible a été achevé en utilisant deux approches d'optimisation basées sur un algorithme génétique et une méthode de recherche par motifs. Dans ces approches, le formalisme lagrangien augmenté est intégré pour supporter les contraintes non linéaires dans le contexte de la fonction objectif originale. Cette intégration permet une meilleure gestion de la procédure de recherche dans les régions des solutions faisables.

L'évaluation des performances des approches proposées a été caractérisée par l'analyse de l'effet de quelques paramètres sur la qualité des solutions obtenues. En effet, nous avons trouvé que la majorité du temps de calcul est consommée dans la recherche de région faisable, ce qui reflète la difficulté de satisfaire les contraintes non linéaires. Les approches proposées peuvent être considérées comme des candidates prometeuses pour la résolution des problèmes d'ordonnancement plus compliqués lorsque les données sont sujettes à des sources d'incertitude.

Dans cette recherche, nous avons focalisé les efforts sur la quantification de la notion d'optimalité pour un ordonnancement bien défini dans un contexte flou. Il serait intéressant de pousser l'analyse vers la considération d'aspect combinatoire de l'ordonnancement et d'exploiter l'existence de problèmes ayant une complexité polynômiale. Un autre axe de recherche qui mérite d'être entamer est de voir le comportement des solutions optimales sous l'effet de contraintes agissant sur les durées opératoires des tâches sans négliger les conséquences sur les procédures de simulation associées.

2.7 Références

AHMADIZAR, F. et L. HOSSEINI. 2011, «Single-machine scheduling with a position-based learning effect and fuzzy processing times», *The International Journal of Advanced Manufacturing Technology*, vol. 56, nº 5, p. 693–698. 43

AHMADIZAR, F. et L. HOSSEINI. 2013, «Minimizing makespan in a single-machine scheduling problem with a learning effect and fuzzy processing times», *The International Journal of Advanced Manufacturing Technology*, vol. 65, nº 1, p. 581–587. 43

ARTIGUES, C., S. DEMASSEY et E. NRON. 2008, *Resource-constrained project scheduling models, algorithms, extensions and applications*, Wiley Press. 47, 59

BAKER, K. R. et D. TRIETSCH. 2009, *Principles of sequencing and scheduling*, Wiley Press. 41

BŁAŻEWICZ, J., K. H. ECKER, E. PESCH, G. SCHMIDT et J. WEGLARZ. 2007, *Handbook on scheduling from theory to applications*, Springer Press. 47

CAO, C., X. GU et Z. XIN. 2009, «Chance constrained programming models for refinery short-term crude oil scheduling problem», *Applied Mathematical Modelling*, vol. 33, nº 3, p. 1696–1707. 42

CASTILLO, O. et P. MELIN. 2009, *Soft computing models for intelligent control of non-linear dynamical systems,*, chap. 4, in W. Mitkowski et J. Kacprzyk (eds.), Modelling dynamics in processes and systems, Springer. 55

CHANAS, S. et A. KASPERSKI. 2001, «Minimizing maximum lateness in a single machine scheduling problem with fuzzy processing times and fuzzy due dates», *Engineering Applications of Artificial Intelligence*, vol. 14, nº 3, p. 377–386. 43

CHANAS, S. et A. KASPERSKI. 2004, «Possible and necessary optimality of solutions in the single machine scheduling problem with fuzzy parameters», *Fuzzy Sets and Systems*, vol. 142, nº 3, p. 359–371. 43, 64

CHRYSSOLOURIS, G. 2006, *Manufacturing systems : theory and practice*, Springer Press. 41

CHUANG, T. N. 2004, «The edd rule for fuzzy job time», *Journal of Information and Optimization Sciences*, vol. 25, nº 1, p. 1–20. 43

COELLO COELLO, C. A. 2002, «Theoretical and numerical constrainthandling techniques used with evolutionary algorithms : a survey of the state of the art», *Computer Methods in Applied Mechanics and Engineering*, vol. 191, nº 11-12, p. 1245–1287. 55

CONN, A. R., N. GOULD et P. L. TOINT. 1997, «A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds», *Mathematics of Computation*, vol. 66, nº 217, p. 261–288. 56

COSTA, L., I. A. C. P. ESPRITO SANTO et E. M. G. P. FERNANDES. 2012, «A hybrid genetic pattern search augmented lagrangian method for constrained global optimization», *Applied Mathematics and Computation*, vol. 218, nº 18, p. 9415–9426. 59

DAHAL, K. P., K. C. TAN et P. I. COWLING. 2007, *Evolutionary scheduling*, Springer Press. 41

DEB, K. et S. SRIVASTAVA. 2012, «A genetic algorithm based augmented lagrangian method for constrained optimization», *Computational Optimization and Applications*, vol. 53, nº 3, p. 869–902. 58

DONG, Y. 2003, «One machine fuzzy scheduling to minimize total weighted tardiness, earliness, and recourse cost», *International Journal of Smart Engineering System Design*, vol. 5, nº 3, p. 135–147. 41

DUBOIS, D. et H. PRADE. 1988, *Possibility theory an approach to computerized processing of uncertainty*, Plenum Press. 46

DUENAS, A. et D. PETROVIC. 2008, «Multi-objective genetic algorithm for single machine scheduling problem under fuzziness», *Fuzzy Optimization and Decision Making*, vol. 7, nº 1, p. 87–104. 42

GAWIEJNOWICZ, S. 2008, *Time-dependent scheduling*, Springer Press. 47

GEORGESCU, I. 2012, *Possibility theory and the risk*, Springer Press. 46

GHRAYEB, O. A. 2003, «A bi-criteria optimization : minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems», *Applied Soft Computing*, vol. 2/3F, p. 197–210. 60

GLOVER, F. et G. A. KOCHENBERGER. 2003, *Handbook of metaheurustics*, Kluwer Press. 56

GUPTA, S. K. et J. KYPARISIS. 1987, «Single machine scheduling research», *Omega The International Journal of Management Science*, vol. 15, nº 3, p. 207–227. 41

HAN, S., H. ISHII et S. FUJI. 1994, «One machine scheduling problem with fuzzy due dates», *European Journal of Operational Research*, vol. 79, nº 1, p. 1–12. 42

HARE, H. et H. SONG. 2016, «On the cardinality of positively linearly independent sets», *Optimization Letters*, vol. 10, nº 4, p. 649–654. 58

HARIKRISHNAN, K. K. et H. ISHII. 2005, «Single machine batch scheduling problem with resource dependent setup and processing time in the presence of fuzzy due date», *Fuzzy Optimization and Decision Making*, vol. 4, nº 2, p. 141–147. 42

JAMISON, K. D. 1998, *Modeling uncertainty using probabilistic based possibility theory with applications to optimization*, thèse de doctorat, Université de Colorado. 46

KASPERSKI, A. 2005, «A possibilistic approach to sequencing problems with fuzzy parameters», *Fuzzy Sets and Systems*, vol. 150, nº 1, p. 77–86. 44

KASPERSKI, A. et P. ZIEŁIŃSKI. 2011, «Possibilistic minmax regret sequencing problems with fuzzy parameters», *IEEE Transactions on Fuzzy Systems*, vol. 19, nº 6, p. 1072–1082. 44

KLIR, G. J. et B. YUAN. 1995, *Fuzzy sets and fuzzy logic theory and applications*, Prentice Hall Press. 44

KROLL, A. et H. SCHULTE. 2014, «Benchmark problems for nonlinear system identification and control using soft computing methods : need and overview», *Applied Soft Computing*, vol. 25, p. 496–513. 55

LEUNG, J. Y. T. 2004, *Handbook of scheduling algorithms, models,and performance analysis*, CRC Press. 41

LEWIS, R. M. et V. TORCZON. 1999, «Pattern search algorithms for bound constrained minimization», *SIAM Journal on Optimization*, vol. 9, nº 4, p. 1082–1099. 58

LEWIS, R. M., V. TORCZON et M. W. TROSSET. 2000, «Direct search methods :then and now», *Journal of Computational and Applied Mathematics*, vol. 124, nº 1-2, p. 191–207. 58

LI, J., X. YUAN, E. S. LEE et D. XU. 2011, «Setting due dates to minimize the total weighted possibilistic mean value of the weighted earliness-tardiness costs on a single machine», *Computers and Mathematics with Applications*, vol. 62, nº 11, p. 4126–4139. 44

LIAO, L. M. et C. J. LIAO. 1998, «Single machine scheduling problem with fuzzy due date and processing time», *Journal of the Chinese Institute of Engineers*, vol. 21, nº 2, p. 189–196. 42

LIU, H. C. et Y. YIH. 2013, «A fuzzy-based approach to the liquid crystal injection scheduling problem in a tft-lcd fab», *International Journal of Production Research*, vol. 51, nº 20, p. 6163–6181. 42

LODWICK, W. A. et J. KACPRZYK. 2010, *Fuzzy optimization recent advances and applications*, Springer Press. 46

LOPEZ, P. et F. ROUBELLAT. 2000, *Ordonnancement de la production*, Editions Lavoisier. 57

MICHALEWICZ, Z. 1998, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Press. 57

NGUYEN, H. T. 1978, «A note on the extension principle for fuzzy sets», *Journal of Mathematical Analysis and Applications*, vol. 64, nº 2, p. 369–380. 45

NIKULIN, Y. et A. DREXL. 2010, «Theoretical aspects of multicriteria flight gate scheduling : deterministic and fuzzy models», *Journal of Scheduling*, vol. 13, nº 3, p. 261–280. 42

NOCEDAL, J. et S. J. WRIGH. 1999, *Numerical optimization*, Springer Press. 55

OLARU, D. et B. SMITH. 2005, «Modelling behavioural rules for daily activity scheduling using fuzzy logic», *Transportation*, vol. 32, nº 4, p. 423–441. 42

ÖZELKAN, E. C. et L. DUCKSTEIN. 1999, «Optimal fuzzy counterparts of scheduling rules», *European Journal of Operational Research*, vol. 113, nº 3, p. 593–609. 43

PETROVIC, S., D. PETROVIC et E. BURKE. 2011, *Fuzzy logic-based production scheduling and rescheduling in the presence of uncertainty,*, chap. 20, in K. G. Kempf, (eds.), Planning production and inventories in the extended enterprise,Springer. 42

PINEDO, M. L. 2012, *Scheduling theory, algorithms, and systems*, Springer Press. 47

PRADE, H. 1979, «Using fuzzy set theory in a scheduling problem : a case study», *Fuzzy Sets and Systems*, vol. 2, nº 2, p. 153–165. 42

RAHIM, M. A., H. M. KHALID et A. KHOUKHI. 2012, «Nonlinear constrained optimal control problem : a pso-ga-based discrete augmented lagrangian approach», *The International Journal of Advanced Manufacturing Technology*, vol. 62, nº 1, p. 183–203. 58

ROCHA, A. M. A. C., T. F. M. C. MARTINS et E. M. G. P. FERNANDES. 2011, «An augmented lagrangian fish swarm based method for global optimization», *Journal of Computational and Applied Mathematics*, vol. 235, nº 16, p. 4611–4620. 58

ROTHKOPF, M. H. 1966, «Scheduling independent tasks on parallel processors», *Management Science*, vol. 12, nº 5, p. 437–447. 47

SAIDI MEHRABAD, M. et A. PAHLAVANI. 2009, «A fuzzy multi-objective programming for scheduling of weighted jobs on a single machine», *The International Journal of Advanced Manufacturing Technology*, vol. 45, nº 1-2, p. 122–139. 43

SCHULTMANN, F., M. FRÖHLING et O. RENTZ. 2006, «Fuzzy approach for production planning and detailed scheduling in paints manufacturing», *International Journal of Production Research*, vol. 44, nº 8, p. 1589–1612. 42

SIVANANDAM, S. N., S. SUMATHI et S. N. DEEPA. 2007, *Introduction to fuzzy logic using MATLAB*, Springer Press. 44

SRIVASTAVA, S. et K. DEB. 2010, *A genetic algorithm based augmented Lagrangian method for computationally fast constrained optimization,*, in Panigrahi, B. K. and Das, S. and Suganthan, P. N. and Dash, S. S. (eds.), Swarm, evolutionary, and memetic computing. Springer. 55

STANFIELD, P. M., R. E. KING et J. A. JOINES. 1996, «Scheduling arrivals to a production system in a fuzzy environment», *European Journal of Operational Research*, vol. 93, nº 1, p. 75–87. 42

TALBI, E. G. 2013, «Combining metaheuristics with mathematical programming, constraint programming and machine learning», *4OR A Quarterly Journal of Operations Research*, vol. 11, nº 2, p. 101–150. 58

TORCZON, V. 1997, «On the convergence of pattern search algorithms», *SIAM Journal on Optimization*, vol. 7, nº 1, p. 1–25. 58

WANG, C., D. WANG, W. H. IP et D. W. YUEN. 2002, «The single machine ready time scheduling problem with fuzzy processing times», *Fuzzy Sets and Systems*, vol. 127, nº 2, p. 117–129. 42

WONGA, B. K. et V. S. LAI. 2011, «A survey of the application of fuzzy set theory in production and operations management : 1998-2009», *International Journal of Production Economics*, vol. 129, nº 1, p. 157–168. 60

WU, C. C. et W. C. LEE. 2005, «A single-machine group schedule with fuzzy setup and processing times», *Journal of Information and Optimization Sciences*, vol. 26, nº 3, p. 683–691. 43

WU, H. C. 2010, «Solving the fuzzy earliness and tardiness in scheduling problems by using genetic algorithms», *Expert Systems with Applications*, vol. 37, nº 7, p. 4860–4866. 43

ZADEH, L. A. 1965, «Fuzzy sets», *Information and Control*, vol. 8, nº 3, p. 338–353. 41

ZADEH, L. A. 1978, «Fuzzy sets as a basis for a theory of possibility», *Fuzzy Sets and Systems*, vol. 1, p. 3–28. 46

ZIMMERMANN, H. J. 1985, *Fuzzy sets theory and applications*, Kluwer Press. 44

Chapitre 3

Problèmes d'ordonnancement à une machine avec effet d'apprentissage et paramètres incertains

« *As far as the laws of mathematics refer to reality, they are not certain ; and as far as they are certain, they do not refer to reality* »

Albert Einstein

Sommaire

3.1 Introduction et état de l'art

Il n'y a pas de doute qu'un grand nombre de travaux est publié relativement au problèmes d'ordonnancement déterministes et en particulier les problèmes d'ordonnancement à une machine. Malgré que le traitement de ces problèmes date de plusieurs décennies [PINEDO, 2012], plusieurs améliorations continuent d'avoir lieu en considérant les nouvelles contraintes technologiques de nos jours. Cette tendance peut être expliquée d'un point de vue théorique par le fait que les connaissances cumulées sur les modèles d'ordonnancement à une machine permettent une compréhension profonde pour les problèmes d'ordonnancement plus compliqués tels que les environnements job shop ou à machines parallèles. Donc, les propositions disponibles peuvent être étendues et ouvrir de futurs axes de recherche. D'un point de vue pratique, les systèmes de production dans certaines industries se comportent comme une ressource unique de service qui opère sur un produit à la fois mais d'une façon séquentielle pour le plan de production composé de plusieurs produits. En outre, dans quelques situations, où plusieurs machines existent, l'une de ces machines connue sous le nom de goulot d'étranglement domine les autres phases du processus. Toute procédure d'amélioration doit être concentrée en premier sur cette machine si les performances globales sont à augmenter. Sachant que plusieurs problèmes d'ordonnancement existant en pratique sont des problèmes NP-difficiles, dans le sens que la complexité de calcul de résolution croît exponentiellement en fonction de la taille du problème, plus d'efforts sont réservés à l'élaboration d'heuristique où d'approches évolutionnaires permettant d'avoir des solutions proches de l'optimale [GLOVER et KOCHENBERGER, 2003].

Dans plusieurs configurations d'ordonnancement réelles, la capacité de production des différentes ressources (en particulier les ressources humaines) à exécuter des tâches bien déterminées (planification de la maintenance, manipulation de logiciel) augmente d'une façon continue avec le temps. Souvent ceci peut être observé dans les compagnies avec des tâches similaires exécutées sur une machine ou machines parallèles identiques pour un nombre de clients. À un niveau de base, ceci est reflété par des durées opératoires qui se réduisent lorsque les tâches associées sont placées en aval qu'en amont dans le plan d'ordonnancement. Mathématiquement, ce phénomène connu sous le nom d'effet d'apprentissage est exprimé par une fonction décroissante pour les durées opératoires. Une représentation graphique nommée par les courbes d'apprentissage a été introduite par Wright en 1936 pour analyser les coûts directs de production dans l'industrie des avions, où les coûts de travail par unité diminuent avec l'augmentation du nombre d'avions produits [WRIGHT, 1936]. Ces courbes décrivent l'amélioration de performance comme une fonction de la reoccurrence de la même tâche. Par la suite, les courbes d'apprentissage ont été utilisées dans multitudes domaines d'applications ([YELLE, 1979] et [WOODS et collab., 1992]). La première étude intégrant l'effet d'apprentissage dans les problèmes d'ordonnancement est due à Meilzson et Tamir, où ils ont considéré un ensemble de tâches avec des unités décroissantes à exécuter sur des processeurs parallèles identiques [MEILIJSON et TAMIR, 1984]. Le travail pionnier de Biskup était le premier à présenter l'effet d'apprentissage dans le contexte d'ordonnancement à une machine en supposant que le temps de production d'un composant décroit en fonction de la position de la tâche dans la séquence. Il a également prouvé que la minimisation de la somme des durées de séjour des tâches ainsi que la minimisation de la somme des déviations des dates de fin des tâches par rapport à une date échue connue sont toutes les deux de complexité polynômiale [BISKUP, 1999]. Par contre, Mosheiov a démontré à travers des exemples numériques que plusieurs règles classiques d'ordonnancement telles que

l'algorithme de Smith ou Moore ne donne pas des solutions optimales sous l'hypothèse d'effet d'apprentissage. Donc, il est possible de développer des heuristiques pour les problèmes modifiés en se basant sur les solutions optimales du problème classique comme solution initiale [MOSHEIOV, 2001]. Bachman et Janiak ont montré que quelques cas particuliers du problème de minimisation de la somme pondérée des dates de fin des tâches avec effet d'apprentissage sont polynômiaux. Par exemple, lorsque toutes les tâches ont la même durée opératoire ou lorsque les poids sont un multiple de la durée opératoire. Ils ont prouvé aussi que le problème de minimisation du makespan avec des dates de disponibilité est au moins NP-difficile mais la procédure de démonstration reste à vérifier car quelques commentaires ont été mentionnées après par Rudek ([BACHMAN et JANIAK, 2004] et [RUDEK, 2013]). Par la suite, les efforts sont concentrés sur les cas polynômiaux des problèmes d'ordonnancement où plusieurs auteurs ont proposé des nouveaux modèles d'effet d'apprentissage et ont déterminé l'ensemble des conditions à satisfaire par les paramètres de configuration de telle sorte que l'application des règles conventionnelles reste une procédure de résolution valide. Kuo et Yang ont introduit un effet d'apprentissage qui dépend du temps comme étant une fonction de la somme des durés opératoires des tâches ordonnancées en amont. Ils ont montré que la minimisation de la somme des dates de fin sur une machine admet la séquence obtenue par la règle Délai de Fabrication le plus Court comme solution optimale [KUO et YANG, 2006]. Wu et Lee ont présenté un modèle d'apprentissage où la durée opératoire de la tâche actuelle dépend de sa position ainsi que de la somme des durées opératoires des tâches déjà ordonnancées. Le makespan et la somme des dates de fin d'exécution peuvent être résolues à l'optimalité si les durées opératoires et les poids sont liés par une contrainte d'agréabilité [WU et LEE, 2008]. Janick et Rudek ont proposé un effet d'apprentissage à multi-capacité dans lequel le processeur obtient seulement les capacités ou bien les expériences. Les auteurs ont étudié le problème de minimisation du makespan et ont élaboré un algorithme de résolution de complexité polynômiale [JANIAK et RUDEK, 2010]. Yin et ses collègues ont développé un modèle d'ordonnancement général avec effet d'apprentissage dépendant de la position et effet de détérioration dépendant du temps. Plusieurs critères d'ordonnancement à une machine ont été considérés par les auteurs et quelques règles ont été prouvées d'être optimales sous quelques conditions spécifiques. Des algorithmes d'approximations ont été également proposés pour la classe des problèmes à complexité ouverte [YIN et collab., 2012]. Dans [LAI et LEE, 2013], un nouveau modèle considérant les effets d'apprentissage et d'oublie a été décrit. Les auteurs ont prouvé que l'application de la règle Délai de Fabrication le plus Court donne des solutions optimales pour le makespan et la somme des dates de fin d'exécution. De plus, la règle Délai de Fabrication le plus Court Pondéré (en anglais WSPT) donne une solution optimale pour la somme pondérée des dates de fin et la règle Délais de Livraison le plus Proche (en anglais EDD) donne une solution optimale pour les critères maximum retard algébrique, maximum retard absolu et somme des retards absolus sous quelques contraintes d'agréabilité. Néanmoins, il est à noter qu'il existe moins de travaux dédiés à l'étude des versions NP-difficiles de ce type de problème. Eren a développé un modèle de programmation non linéaire pour le problème d'ordonnancement à une machine avec dates de disponibilité différentes et effet d'apprentissage. Il a montré en utilisant des tests numériques que le modèle proposé peut résoudre des instances de taille allant jusqu'à trente tâches d'une façon efficace [EREN, 2009]. Wu et ses collègues ont considéré un problème d'ordonnancement à une machine avec effet d'apprentissage basé sur la somme des durées opératoires et données de disponibilité pour minimiser la somme pondérée des dates de fin d'exécution. Les auteurs ont développé des approches par séparation et évaluation et algorithme

génétique pour obtenir des solutions acceptables où les résultats ont montré que l'approche par séparation et évaluation a des bonnes performances jusqu'à quinze tâches et l'erreur relative moyenne de l'algorithme génétique est tolérable [Wu et collab., 2011]. Yin et ses collègues ont abordé la minimisation de la somme des retards avec des dates de disponibilité arbitraires et effet d'apprentissage de position. Comme ce problème est NP-difficile, un modèle de programmation linéaire mixte en nombres entiers et une procédure par séparation et évaluation avec des règles de dominance et bornes inférieures ont été proposés pour obtenir une solution optimale [Yin et collab., 2014]. Pour une étude détaillée sur la complexité des problèmes d'ordonnancement avec différents types d'effet d'apprentissage, le lecteur peut consulter la référence [Biskup, 2008].

Un autre facteur qui peut altérer non pas seulement les durées opératoires des tâches mais aussi la totalité de la stratégie d'ordonnancement est l'existence de plusieurs sources d'incertitudes internes et externes du système de production. Dans la pratique, les durées opératoires sont souvent estimées relativement à deux règles : les durées opératoires associées à des tâches de même type et si ce type n'existe pas (nouveau produit) il faut opter pour l'estimation basée sur des tâches un peu similaires. Le problème majeur confronté dans ce cas réside dans le fait que les durées opératoires des tâches dépendent de plusieurs éléments contenant l'aspect subjectif qu'on ne peut pas évaluer d'une façon précise. Par exemple, les temps de configurations dépendant de la séquence dans la production de peintures ont une influence importante sur la conformité de la couleur aux besoins. Plusieurs contraintes environnementales comme la régulation de la température peuvent aussi générer des incertitudes avec impact sur le processus de production et en particulier sa durée [Schultmann et collab., 2006]. Pour cela, la théorie des ensembles flous a été adoptée par plusieurs chercheurs comme le pilier de construction d'un nouveau paradigme représenté par les modèles d'ordonnancement flous. Ce type d'ordonnancement avec durées opératoires floues a été abordé initialement par Prade qui a publié un article dans ce domaine. Il a affecté des nombres flous triangulaires aux durées des cours pour élaborer le calendrier trimestriel dans une grande école française [Prade, 1979]. Après, le concept flou a été diffusé aux différents niveaux motivé par les perturbations dans les marchés de nos jours. Liao et Liao ont considéré un problème d'ordonnancement à une machine avec durées opératoires et dates échues floues représentées par des nombres flous triangulaires et trapézoïdaux respectivement. Ils ont proposé des algorithmes de complexité polynômiale pour maximiser le degré minimum de satisfaction [Liao et Liao, 1998]. Dans [Chanas et Kasperski, 2001], il a été illustré que si la fonction coût est F-monotone, la généralisation de l'algorithme de Lawler peut être utilisée pour résoudre plusieurs problèmes d'ordonnancement avec des durées opératoires ou dates échues floues. Le problème de maximisation du degré de possibilité que le maximum flou du retard algébrique n'est pas supérieure à un certain seuil flou donné par des experts a été également introduit. Wang et ses collègues ont utilisé le principe d'extension et le concept de profil d'achèvement probable de tâche pour maximiser le temps de disponibilité commun avec durées opératoires floues et sous des conditions particulières. Ils ont établit une condition nécessaire que la solution optimale doit satisfaire lorsque les tâches ont différentes dates échues et intervalles de confiances [Wang et collab., 2002]. L'article de Dong a été consacré à une approche d'ordonnancement à deux phases pour résoudre un problème à une machine avec l'objectif de minimiser une combinaison de la somme pondérée des retards absolus, la somme pondérée des avances et la somme pondérée des coûts de recours. Il a représenté les durées opératoires incertaines par des nombres flous triangulaires et les a interprété comme étant des distributions de possibilité [Dong, 2003]. Chanas et Kasperski ont supposé que les paramètres des problèmes

d'ordonnancement flous sont mal-connus et ont introduit le concept d'optimalité possible et nécessaire exprimant la possibilité et la nécessité que l'événement "Un ordonnancement donné est optimal au sens ordinaire" [CHANAS et KASPERSKI, 2004]. Wu et Lee ont considéré un problème d'ordonnancement de groupe à une machine où les durées opératoires et les temps de configurations des groupes sont traités comme des nombres flous. Des procédures de résolution sont proposées pour trouver une séquence de groupe et une séquence de tâches qui minimisent les valeurs centroïdes de ces tâches pour que le temps total de séjour flou puisse être calculé pour la décision [WU et LEE, 2005]. Li et ses collègues ont abordé un problème d'ordonnancement pour l'affectation des dates échues à une machine avec durées opératoires incertaines et des contraintes de précédence générales entre tâches. Ils ont montré que si les contraintes de précédence générales sont incluses, le problème est NP-difficile. Autrement, si ces contraintes sont éliminées ou ayant une structure arborescente alors le problème est polynômial [LI et collab., 2010]. Dans [WU, 2010], l'auteur a exploité la minimisation de la somme pondérée de l'avance flou et le retard absolu flou à travers le concept de l'addition entre les nombres flous où les durées opératoires et les dates échues sont supposées des nombres flous. Un algorithme génétique a été proposé pour résoudre ce problème. Kasperski et Zieliński ont modélisé l'incertitude par le biais des intervalles flous où les fonctions d'appartenances sont données par des distributions de possibilité pour les valeurs des paramètres. Les auteurs ont représenté un cadre général pour les problèmes de séquencement avec critère du regret minmax en utilisant la théorie de possibilité et ont proposé quelques méthodes pour résoudre les problèmes NP-difficiles flous [KASPERSKI et ZIEŁIŃSKI, 2011]. Malgré le nombre croissant des travaux disponibles sur l'effet d'apprentissage et modélisation floue des problèmes d'ordonnancement, il est clair que les contributions relatives à la considération simultanée d'effet d'apprentissage à base de position et durées opératoires floues dans le même modèle sont à nos connaissances très rares. Ahmadizar et Hosseini ont proposé plusieurs procédures polynômiales de résolution basées sur la règle Délai de Fabrication le plus Court, programmation sous contraintes probabilistes floues et classement des nombres flous où les critères d'optimisation sont la minimisation de la somme des dates de fin et le makespan ([AHMADIZAR et HOSSEINI, 2011] et [AHMADIZAR et HOSSEINI, 2013]).

Comme l'application des approches classiques d'ordonnancement résulte en des solutions sous optimales ou non faisables lorsque l'effet d'apprentissage ou bien données incertaines sont présents, le travail présenté dans ce chapitre vise deux objectifs essentiels. En premier, nous allons développer un algorithme polynômial pour minimiser la somme pondérée des dates de fin d'exécution des tâches lorsque les deux aspects sont considérés à la fois. Mais dans ce cas, le concept d'agréabilité floue doit être satisfait pour préserver la validité des conditions d'optimalité classiques. Par la suite, lorsque ce concept n'est pas vérifié avec la présence de différentes dates de disponibilité, une analyse numérique comparative entre trois métaheuristiques à base de trajectoire (le recuit simulé, la recherche tabou et la recherche kangourou) sont implémentées pour l'obtention de solutions acceptables. Les tests numériques sont consolidés par une validation statistique pour évaluer les performances de chaque méthode.

Le reste de ce chapitre est organisé comme suit. La Section 3.2 présente les définitions de base et les concepts de la théorie des ensembles flous. La Section 3.3 est dédiée à la présentation de différentes métaheuristiques à base de trajectoire et les étapes de leur élaboration sont également détaillées. Dans la Section 3.4, nous introduisons le concept d'agréabilité et nous étendons la règle de Smith pour le cas avec effet d'apprentissage et durées opératoires floues. Les expérimentations numériques basées sur ces trois méta-

heuristiques sont décrites avec une analyse statistique dans la Section 3.5. La dernière Section est destinée à la discussion des résultats obtenus sans oublier quelques suggestions pour des futurs développements.

3.2 Présentation du cadre de modélisation floue

Dans la plus part des applications réelles, les processus de prise de décision nécessitent souvent la manipulation de données numériques dans un environnement incertain ou sous manque d'informations. Il est primordial de mentionner que les variations probabilistes ne sont pas la source unique d'imprécision qui doit être supportée. D'autres sources peuvent être considérées comme générateurs d'incertitude telles que les capteurs d'acquisions bruités ou les préférences subjectives humaines. Alors, les mesures de performances comme l'utilité espérée sont représentées en utilisant des modèles flous dans lesquels la comparaison de quantités incertaines devient un obstacle ardu à contourner. Dans cette section, nous présentons les idées de base des ensembles et nombres flous où quelques définitions sont extraites partiellement des références ([KLIR et YUAN, 1995], [ROSS, 2004], [HANSS, 2005] et [SIVANANDAM et collab., 2007]).

Un nombre flou est un sous ensemble flou de la ligne réelle $\mathbb{R}$ avec des fonctions d'appartenance continues et convexes. L'espace des nombres flous est noté par $\mathbb{F}$. En général, un nombre flou est défini uniquement sur un sous ensemble $X \subseteq \mathbb{R}$ connu sous le nom de l'univers de discours.

Définition 3.1. *Pour un nombre flou donné* $\tilde{A} = (a,b,c,d) \in \mathbb{F}$*, sa fonction d'appartenance possède les propriétés suivantes :*

$$\mu_{\tilde{A}}(x) = \begin{cases} \mu^{L}_{\tilde{A}}(x) & si \quad a \leq x \leq b \\ 1 & si \quad b \leq x \leq c \\ \mu^{R}_{\tilde{A}}(x) & si \quad c \leq x \leq d \\ 0 & autrement \end{cases} \tag{3.1}$$

avec $\mu^{L}_{\tilde{A}}$ *et* $\mu^{R}_{\tilde{A}}$ *sont des fonctions monotones, continues et strictement croissante (décroissante) définies sur* $\mathbb{R}$ *dans l'intervalle fermé* $[0,1]$.

À cause de la tendance monotone des deux fonctions, leurs fonctions inverses existent et sont aussi monotones.

Définition 3.2. *Pour tout nombre réel* $\gamma \in [0,1]$*, la coupe* γ *de* $\tilde{A}$ *notée par* $A(\gamma)$ *est donnée par :*

$$A(\gamma) = \left\{ x \middle| \mu_{\tilde{A}}(x) \geq \gamma \right\} \forall x \in X \tag{3.2}$$

La coupe γ résultante est un intervalle compact exprimé sous forme paramétrique par $[\underline{A}(\gamma), \bar{A}(\gamma)] \subseteq \mathbb{R}$ avec $\underline{A}(\gamma) = \inf_{x \in X} \left\{ x \middle| \mu_{\tilde{A}}(x) \geq \gamma \right\}$ et $\bar{A}(\gamma) = \sup_{x \in X} \left\{ x \middle| \mu_{\tilde{A}}(x) \geq \gamma \right\}$. Ici, nous distinguons deux cas limites lorsque $\gamma = 0$ et $\gamma = 1$. Le premier cas limite est le support de $\tilde{A}$ défini par $\text{supp}(\tilde{A}) = \overline{\left\{ x \middle| \mu_{\tilde{A}}(x) \geq 0 \right\}}$ avec $\overline{\{\}}$ signifie la fermeture de l'ensemble

$\{x \,|\, \mu_{\tilde{A}}(x) \geq 0\}$. Le deuxième cas limite est le noyau de $\tilde{A}$ défini par $\ker(\tilde{A}) = \{x \,|\, \mu_{\tilde{A}}(x) = 0\}$. Le Théorème de représentation de Zadeh permet de représenter $\tilde{A}$ par [ZADEH, 1971] :

$$\tilde{A} = \bigcup_{\gamma \in [0,\,1]} (\gamma A(\gamma)) \tag{3.3}$$

avec

$$\gamma A(\gamma) = \begin{cases} \gamma & si \quad x \in A(\gamma) \\ 0 & si \quad x \notin A(\gamma) \end{cases}$$

Nous concluons de ce Théorème que si toutes les coupes γ d'un nombre flou peuvent être déterminées, le nombre flou est complètement défini. Donc, il y a une équivalence entre un nombre flou et ses coupes γ pour $\gamma \in [0,1]$.

Dans [MATARAZZO et MUNDA, 2001], il a été mentionné que la représentation L-R des nombres flous est la forme générale la plus adoptée à cause de sa précision et faible complexité de calcul lorsqu'elle est appliquée à des instances des problèmes réelles.

Définition 3.3. *Un nombre flou noté par* $\tilde{A} = (\underline{a}, \overline{a}, \alpha_{\tilde{A}}, \beta_{\tilde{A}})_{L-R}$ *est de type* L-R *si sa fonction d'appartenance a la forme suivante :*

$$\mu_{\tilde{A}}(x) = \begin{cases} L\left(\frac{\underline{a}-x}{\alpha_{\tilde{A}}}\right) & pour \quad -\infty < x < \underline{a} \\ 1 \quad pour \quad \underline{a} < x < \bar{a} \\ R\left(\frac{x-\bar{a}}{\beta_{\tilde{A}}}\right) & pour \quad \bar{a} < x < +\infty \end{cases} \tag{3.4}$$

avec les paramètres $\alpha_{\tilde{A}}$ *et* $\beta_{\tilde{A}}$ *expriment les déviations gauche et droite, respectivement. Les fonctions* L *et* R *sont des fonctions décroissantes avec* $L(0) = R(0) = 1$.

Les nombres flous trapézoïdaux sont caractérisés par des fonctions linéaires L (R) et $\underline{a} < \overline{a}$, tandis que les nombres flous triangulaires permettent la linéarité des fonctions L et R avec $\underline{a} = \overline{a}$. Les coupes γ du nombre flou trapézoïdal $\tilde{A}$ sont données par l'ensemble des intervalles $A_\gamma = [\underline{a} - \alpha_{\tilde{A}} + \alpha_{\tilde{A}}\gamma, \overline{a} + \beta_{\tilde{A}} - \beta_{\tilde{A}}\gamma]$. En utilisant le principe d'extension de Zadeh, il est possible de généraliser le concept d'application définie sur des ensembles réels à une application définie sur des ensembles flous. Par conséquent les opérations arithmétiques élémentaires peuvent être également déduites à partir des opérations simples.

Définition 3.4. *Soient* $M, N \in \mathbb{F}(\mathbb{R})$ *deux nombres flous et* $\circ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *une opération binaire. Si nous notons par* $\tilde{\circ}$ *l'opération généralisée sur l'espace des nombres flous, alors par application du principe d'extension,* $M \tilde{\circ} N$ *est donnée par un ensemble flou de* $\mathbb{R}$ *avec la fonction d'appartenance :*

$$(M \tilde{\circ} N)(z) = \left\{ \begin{array}{l} \sup\limits_{z = x \circ y} \min\{M(x), N(y)\} \quad si \quad (\circ)^{-1}(z) \neq \phi \\ \\ 0 \quad si \quad (\circ)^{-1}(z) = \phi \end{array} \right\} \forall z \in \mathbb{R} \tag{3.5}$$

L'application de cette définition résulte en les expressions floues des opérations binaires classiques pour tout triplet de nombres flous $\tilde{A} = (\underline{a}, \overline{a}, \alpha_{\tilde{A}}, \beta_{\tilde{A}})_{L-R}$, $\tilde{B} = \left(\underline{b}, \overline{b}, \alpha_{\tilde{B}}, \beta_{\tilde{B}}\right)_{L-R}$ et

$\tilde{C} = \left(\underline{c}, \overline{c}, \alpha_{\tilde{C}}, \beta_{\tilde{C}}\right)_{R-L}$ comme suit :

- $\tilde{A} + \tilde{B} = \left(\underline{a} + \underline{b}, \overline{a} + \overline{b}, \alpha_{\tilde{A}} + \alpha_{\tilde{B}}, \beta_{\tilde{A}} + \beta_{\tilde{B}}\right)_{L-R}$
- $\tilde{A} - \tilde{C} = \left(\underline{a} - \overline{c}, \overline{a} - \underline{c}, \alpha_{\tilde{A}} + \beta_{\tilde{C}}, \beta_{\tilde{A}} + \alpha_{\tilde{C}}\right)_{L-R}$
- $\lambda\tilde{A} = \begin{cases} \left(\lambda\underline{a}, \lambda\overline{a}, \lambda\alpha_A, \lambda\beta_A\right)_{L-R} \ \ pour \ \ \lambda \geq 0 \\ \left(\lambda\overline{a}, \lambda\underline{a}, |\lambda|\alpha_A, |\lambda|\beta_A\right)_{L-R} \ \ pour \ \ \lambda < 0 \end{cases}$

Au contraire des nombres réels pour lesquelles il existe un ordre total, il est très difficile de comparer les nombres flous représentés par des distributions de possibilité car il est possible d'avoir un chevauchement. Mais un outil efficace consiste à utiliser les fonctions de classement [JAIN, 1976], qui associe à chaque nombre flou un point sur la ligne réelle $\mathbb{R}$ et la comparaison est rendue naturelle après cette transformation. Dépendamment de la position relative des nombres flous, deux relations de dominance sont établies. Une dominance forte caractérisant les nombres flous séparés et la dominance faible caractérisant les nombres flous avec chevauchement en utilisant une fonction de classement. Leurs définitions formelles sont représentées ci-dessous [ÖZELKAN et DUCKSTEIN, 1999]. Soit $E : \mathbb{F} \rightarrow \mathbb{R}$ une fonction de classement, $\tilde{A}$ et $\tilde{B}$ sont deux nombres flous.

Définition 3.5. *On dit que* $\tilde{A}$ *domine faiblement* $\tilde{B}$ *si* $E\left(\tilde{A}\right) \geq E\left(\tilde{B}\right)$. *La dominance faible est notée par* $\tilde{A} \geq_w \tilde{B}$ *et dans ce cas nous avons* $\max_w \left\{\tilde{A}, \tilde{B}\right\} = \tilde{A}$.

Définition 3.6. *On dit que* $\tilde{A}$ *domine fortement* $\tilde{B}$ *si* $\underline{A}(\gamma) \geq \underline{B}(\gamma)$ *et* $\overline{A}(\gamma) \geq \overline{B}(\gamma)$ *pour toutes les valeurs de* $\gamma \in [0,1]$. *La dominance forte est notée par* $\tilde{A} \geq_s \tilde{B}$ *et dans ce cas nous avons* $\max_s \left\{\tilde{A}, \tilde{B}\right\} = \tilde{A}$.

Une condition plus simple pour la dominance forte est écrite $\underline{a} - \alpha_{\tilde{A}} \geq \overline{b} + \beta_{\tilde{B}}$, ou d'une façon equivalente $A(0) \cap B(0) = \phi$.

Pour remédier les inconvénients des fonctions de classement existantes qui sont dans certains cas en contradiction avec l'intuition humaine ou ne sont pas discriminantes, Abasabandy et Hajjari ont proposé une nouvelle approche de classement des nombres flous trapézoïdaux basée sur les déviations gauche et droite à une certaine coupe γ des nombres flous trapézoïdaux [ABBASBANDY et HAJJARI, 2009]. Le principe de cette méthode est décrit ci-dessous.

Définition 3.7. *La magnitude d'un nombre flou arbitraire* $\tilde{A} = \left(\underline{a}, \overline{a}, \alpha_A, \beta_A\right)_{L-R}$ *avec forme paramétrique* $\left[\underline{A}(\gamma), \overline{A}(\gamma)\right]$ *est définie par :*

$$Mag\left(\tilde{A}\right) = \frac{1}{2}\int_{r=0}^{r=1}\left(\underline{A}(\gamma) + \overline{A}(\gamma) + \underline{a} + \overline{a}\right) f(r)\, dr \tag{3.6}$$

avec la fonction f *est une fonction croissante sur* $[0,1]$ *avec* $f(0) = 0$, $f(1) = 1$ *et* $\int_{r=0}^{r=1} f(r)\, dr = \frac{1}{2}$.

Il est à mentionner que si f est prise égale à 1 et seulement les bornes des coupes γ sont

considérées alors l'expression de la magnitude se réduit à la formule proposée par Yager [YAGER, 1981]. Si les fonctions d'appartenance sont supposées linéaires, il est possible de déduire une expression analytique donnée par $Mag\left(\tilde{A}\right) = \frac{1}{2}\left(\underline{a} + \overline{a}\right) - \frac{1}{12}\left(\alpha_{\tilde{A}} - \beta_{\tilde{A}}\right)$.

Dans ce qui suit, nous démontrons que la propriété de linéarité est valable pour la fonction de classement magnitude dans le cas des nombres flous trapézoïdaux et coefficients réels positifs.

Propriété 3.1. *La fonction de classement exprimée par la magnitude est linéaire pour tous les nombres flous* $\tilde{A}, \tilde{B} \in \mathbb{F}$ *et les réels* $c_1, \ c_2 \in \mathbb{R}$ *dans le sens de :*

$$Mag\left\{\left(c_1 \otimes \tilde{A}\right) \oplus \left(c_2 \otimes \tilde{B}\right)\right\} = c_1 Mag\left(\tilde{A}\right) + c_2 Mag\left(\tilde{B}\right) \tag{3.7}$$

Démonstration.
En se basant sur les expressions floues pour la somme et la multiplication par un scalaire, nous pouvons écrire :

$$Mag\left\{\left(c_1 \otimes \tilde{A}\right) \oplus \left(c_2 \otimes \tilde{B}\right)\right\} =$$

$$Mag\left\{\left(c_1\underline{a} + c_2\underline{b}, c_1\overline{a} + c_2\overline{b}, c_1\alpha_{\tilde{A}} + c_2\alpha_{\tilde{B}}, c_1\beta_{\tilde{A}} + c_2\beta_{\tilde{B}}\right)_{L-R}\right\}$$

$$= \frac{1}{2}\left(c_1\underline{a} + c_2\underline{b} + c_1\overline{a} + c_2\overline{b}\right) - \frac{1}{12}\left(c_1\alpha_{\tilde{A}} + c_2\alpha_{\tilde{B}} - c_1\beta_{\tilde{A}} - c_2\beta_{\tilde{B}}\right)$$

$$= c_1\left\{\frac{1}{2}\left(\underline{a} + \overline{a}\right) - \frac{1}{12}\left(\alpha_{\tilde{A}} - \beta_{\tilde{A}}\right)\right\} + c_2\left\{\frac{1}{2}\left(\underline{b} + \overline{b}\right) - \frac{1}{12}\left(\alpha_{\tilde{B}} - \beta_{\tilde{B}}\right)\right\}$$

Comme les expressions entre crochet sont les fonctions de classement (les magnitudes) des nombres flous $\tilde{A}$ et $\tilde{B}$, nous obtenons l'équation $Mag\left\{\left(c_1\tilde{A}\right) + \left(c_2\tilde{B}\right)\right\} = c_1 Mag\left(\tilde{A}\right) + c_2 Mag\left(\tilde{B}\right)$. □

3.3 Métaheuristiques à base de trajectoire

Dans cette section, nous présentons trois types de métaheuristiques à base de trajectoire : recherche tabou, recuit simulé et recherche kangourou. L'avantage de ces méthodes réside dans le temps de calcul acceptable car elles sont construites à la base d'une solution en plus de la simplicité d'application des opérateurs d'évolution. Pour chaque méthode, nous discutons le codage de l'espace de recherche et la structure du voisinage qui joue un rôle critique dans la convergence rapide de ces approches. Les paramètres de configuration sont aussi illustrés avec les règles adéquates pour la détermination de leurs valeurs.

3.3.1 Recherche tabou

Le paradigme de la recherche tabou peut être considéré parmi les métaheuristiques les plus populaires appliquées pour la résolution d'un grand nombre de problèmes d'ordonnancement [LEE et collab., 1997]. Cette approche est une méthode de recherche basée sur une solution et a été développée par Glover. Elle constitue une extension de la fameuse heuristique de recherche locale où une mémoire à courte terme nommée liste tabou est ajoutée pour éviter d'être piégé dans un minimum local ([GLOVER, 1989] et [GLOVER, 1990]). La procédure de recherche commence avec une solution initiale faisable enregistrée comme solution courante et meilleure. Ensuite, utilisant une structure

de voisinage pour obtenir les solutions candidates, elle génère un sous ensemble de solutions et évalue leurs fonctions objectifs correspondantes. Le meilleur candidat non tabou ou tabou mais satisfaisant le critère d'aspiration est sélectionné comme une nouvelle solution courante et le mouvement associé est ajouté à la liste tabou gérée selon la stratégie Premier Entré Premier Sorti (en anglais FIFO) pour que le même mouvement soit interdit pendant un nombre fini des prochaines itérations. Sinon, choisir le premier mouvement non tabou et le considérer comme la nouvelle solution courante. Si cette solution est meilleure que la meilleure solution courante, elle est enregistrée comme nouvelle meilleure solution. Cette procédure est répétée pour un nombre fixe d'itérations. À la fin, la meilleure solution obtenue est la solution du problème d'optimisation [AL-TURKI et collab., 2001]. L'Algorithme 3.1 est une description de différentes phases de la recherche tabou.

Algorithm 3.1 Recherche tabou

DEBUT
Initialiser la liste tabou TL $\leftarrow \phi$
Sélectionner la solution initiale x_0 arbitrairement et la considérer comme solution courante
Initialiser la meilleure solution $x^* \leftarrow x_0$
POUR ($k = 1$ au nombre maximum d'itérations) **FAIRE**
  Définir le type de mouvement δ et déterminer un sous voisinage de cardinal s de la solution courante $N_\delta^s(x_k)$
  POUR ($i = 1$ à s) **FAIRE**
    SI ($f(x_i) < f(x_k)$) **ALORS**
      Ajouter les attributs du mouvement inverse à la tête de la liste tabou selon la strategie FIFO
      Mettre $x_{k+1} \leftarrow x_i$
      Sortir de la boucle
    FIN SI
    SI ($f(x_i) < f(x^*)$ et les attributs de mouvement ne sont pas tabous) **ALORS**
      Mettre $x_{k+1} \leftarrow x_i$
      Sortir de la boucle
    FIN SI
  FIN POUR
  SI ($f(x_{k+1}) < f(x^*)$) **ALORS**
    Mettre $x^* \leftarrow x_{k+1}$
  FIN SI
FIN POUR
Retourner la solution x^* comme la meilleure solution correspondante au minimum de la fonction objectif f
FIN

Comme plusieurs paramètres affectent le processus de recherche à travers la qualité de solution et le temps d'exécution, nous abordons dans ce qui suit les détails de l'implémentation de l'approche adoptée dans ce chapitre.

- Solution initiale (x_0) : malgré que pour le cas des données numériques réelles, plusieurs méthodes telles que les règles de répartition de priorité sont utilisées pour obtenir une solution initiale, moins d'efforts sont alloués aux données floues. Ceci est due au manque de connaissance sur les propriétés intrinsèques des problèmes

d'ordonnancement flous. Mais généralement, une telle situation ne pose pas de problème sérieux car les méthodes de choix d'une solution initiale n'influent pas généralement sur la qualité de la solution mais sur le temps de calcul. Donc, dans notre travail la recherche est initialisée avec une solution aléatoire ;

- Génération de voisinage (N_x) : un nouvel ensemble de solutions est obtenu de la solution courante par application de petites perturbations et ce passage est connu sous le nom de mouvement. Le type de mouvement doit être défini avec précision car il existe une forte corrélation entre la structure du voisinage et l'efficacité de la recherche. Le mouvement de permutation général (GPIM) est adopté où deux tâches choisies arbitrairement sont permutées jusqu'à l'obtention d'un voisinage ayant une taille prédéfinie ;
- Liste tabou (TL) : la liste tabou est définie comme étant une chaîne circulaire avec une taille généralement fixe. Mais une étude empirique suggère l'utilisation d'une liste tabou avec une taille dynamique pour augmenter la possibilité d'avoir de bonnes solutions. Donc, nous suivons la même approche et nous supposons que la taille de la liste varie aléatoirement dans l'intervalle $[TL_{min}, TL_{max}]$ à chaque période finie d'itérations (TL_{Up}) [SALHI, 2002]. Après chaque mouvement où les tâches i et j occupant les positions pos_i et pos_j respectivement sont permutées, les attributs de mouvement sont enregistrés dans la liste tabou dans l'ordre inverse $\big((j, pos_i), (i, pos_j)\big)$. Chaque solution voisine créée par une permutation déjà existante dans la liste est interdite jusqu'à la satisfaction du critère d'aspiration ou la suppression de l'état tabou ;
- Critère d'aspiration : les critères d'aspirations sont des règles qui modifient l'état tabou d'un mouvement. Le critère basé sur la valeur de la fonction objectif est largement utilisé où un mouvement est permis s'il mène à une solution meilleure que la meilleure solution trouvée auparavant ;
- Nombre de solutions de test (N_{tr}) : nous avons fixé la taille du voisinage à $(n-1)$ solutions pour permettre l'exploitation complète des possibilités en plus de l'exécution rapide. En effet, des valeurs élevées pour le nombre de solutions de test rendent la recherche basée sur l'intensification, tandis que les valeurs faibles génèrent moins de solutions voisines pour l'optimisation et la recherche est accentuée sur la diversification ;
- Critères de terminaison : l'algorithme s'arrête lorsque un nombre maximum d'itérations I_{max} est atteint ou la valeur de la fonction objectif est stable pour un nombre donné d'itérations I_{fstab}.

3.3.2 Recuit simulé

D'une façon analogue au phénomène physique de recuit dans lequel un cristal est chauffé et ensuite refroidi très lentement jusqu'à l'atteinte de l'état d'énergie minimale par le réseau. Le recuit simulé a été développé comme une recherche stochastique itérative pour mimer ce type de comportement thermodynamique dans l'optimisation. Les concepts de base ont été introduits pour la première fois par Metropolis, tandis que l'application à des problèmes apparaissant dans la conception des ordinateurs est due à Kirkpatrick et ses collègues [KIRKPATRICK et collab., 1983]. L'algorithme commence par une solution initiale générée aléatoirement ou selon des règles spécifiques. À chaque itération de l'algorithme, la valeur de la fonction objectif est évaluée pour la solution courante (x_k)

et la solution sélectionnée du voisinage (x'). Si la valeur de la fonction objectif est améliorée alors la solution associée est toujours acceptée. Mais même pour des solutions non améliorées, il y a une probabilité d'acceptation dépendente d'un facteur de température. Ce mécanisme est adopté pour éviter d'être piégé dans les optimums locaux pendant le processus de recherche. La probabilité d'acceptation est donnée par :

$$\mathrm{P}\left(Accepter\ x'\ comme\ prochaine\ solution\right) = \begin{cases} e^{-\left(\frac{f(x')-f(x_k)}{\mathrm{T}_k}\right)} & si \quad f(x')-f(x_k)>0 \\ 1 \quad si \quad f(x')-f(x_k)\leq 0 \end{cases} \tag{3.8}$$

avec le paramètre T_k définit la température à l'itération k et $\forall k \quad \mathrm{T}_k \geq 0$ et $\lim_{k\to+\infty} \mathrm{T}_k = 0$. Il est à mentionner que la convergence du recuit simulé est assurée par la tendance décroissante du paramètre de température. Par la réduction des valeurs vers zéro, la chaîne de Markov non homogène équivalente converge vers une forme pour laquelle la probabilité est concentrée sur l'ensemble des solutions optimales locales ou globales [VAN LAARHOVEN et AARTS, 1987]. Les différentes étapes du recuit simulé sont indiquées sur l'Algorithme 3.2.

Algorithm 3.2 Recuit simulé

DEBUT
Construire la solution initiale x_0 et la considérer comme la solution courante
Initialiser la température à une valeur élevée $\mathrm{T} = \mathrm{T}_{ini}$
Initialiser la meilleure solution $x^* \leftarrow x_0$
Définir le nombre et les longueurs des époch
POUR ($k = 1$ au nombre maximum d'itérations) **FAIRE**
  Générer une solution voisine $(x' \in \mathrm{N}(x_k))$ utilisant un type de mouvement défini
  SI $((f(x')-f(x_k)\leq 0))$ **ALORS**
    Accepter l'état nouvel $(x_k \leftarrow x')$
  SINON
    Générer un nombre aléatoir $q \in [0,1]$
    SI $\left(q < \mathrm{e}^{-\frac{(f(x')-f(x_k))}{\mathrm{T}_k}}\right)$ **ALORS**
      Accepter l'état nouvel $(x_k \leftarrow x')$
    FIN SI
  FIN SI
  Mettre à jour la température à chaque époch utilisant la loi de refroidissement $(\mathrm{T}_{k+1} = g(\mathrm{T}_k))$
FIN POUR
Retourner la solution x^* comme la meilleure solution correspondante au minimum de la fonction objectif f
FIN

Les détails les plus importants concernant l'implémentation de la technique du recuit simulé sont donnés ci-dessous.

- Solution initiale (x_0) : la solution initiale est générée aléatoirement ;
- Génération de voisinage (N_x) : le mouvement d'insertion (IM) est utilisé pour la génération du voisinage où une tâche dans la position j est sélectionnée au hasard

ce qui permet par la suite d'avoir $(n-1)$ nouvelles solutions par l'insertion de cette tâche à chaque fois dans une position des positions restantes $\{1,..,j-1,j+1,..n\}$;

- Température initiale (T_0) : si la température initiale dans le processus du recuit est suffisamment élevée, le système peut atteindre tous les états possibles. En se basant sur cette réalité physique, l'algorithme peut trouver une solution qui ne dépend pas fortement de la configuration initiale. Dans ce travail et comme une première suggestion de la température initiale, nous générons le voisinage de la solution initiale, ensuite nous calculons la différence entre les valeurs meilleure et mauvaise de la fonction objectif. La température initiale est approximée par $-\frac{(f(x_{worst})-f(x_{best}))}{\log(p_0)}$ avec p_0 la probabilité d'acceptation d'une solution médiocre ;
- Nombre d'itérations à chaque température (I_T) : il n'est pas très difficile d'assimiler le passage d'une configuration à une autre voisine avec certaine probabilité dans le processus du recuit à une chaîne de Markov. Alors, le nombre d'itérations à chaque température doit être choisi avec précision. Il est important d'allouer un temps long pour les températures faibles et un temps bref pour les températures élevés. Ceci est fait dans le but de consolider la décroissance de taux de probabilité des solutions avec la réduction de la température. Comme la borne supérieure peut être proportionnelle à la taille du voisinage, le nombre d'itérations à chaque température est de l'ordre de n ($O(n)$).
- Taux de refroidissement (r_{cool}) : le taux de refroidissement designe le taux de réduction de la température qui est souvent modélisé comme une loi de dégradation linéaire $T_k = r_{cool} \times T_{k-1}$ avec la pente choisie dans l'intervalle $[0.8, 0.99]$ pour avoir une décroissance lente de la température. Mais il existe d'autres types de refroidissement analysés dans la littérature [NOURANI et ANDRESEN, 1989] ;
- Conditions d'arrêt : en plus du nombre maximum d'itérations I_{max}, une autre condition est utilisée lorsque la meilleure configuration reste figée pour un nombre spécifique de phases de réduction de la température. Cette condition est proportionnelle au logarithme du cardinal de l'espace de recherche (la taille de l'espace de toutes les permutations). Donc, une condition d'arrêt supplémentaire est la stabilité de la fonction objectif pour un nombre d'itérations de l'ordre de $log(n!)$.

3.3.3 Recherche kangourou

La méthode kangourou est une technique d'optimisation basée sur la descente stochastique. Elle consiste en l'application d'une descente aléatoire aux différentes solutions prises arbitrairement dans l'espace de recherche. Cette méthode a été proposée par Fleury en 1993 par analogie avec le recuit simulé mais utilisant une stratégie de recherche complètement différente [FLEURY, 1993]. Dans cette approche, nous avons une solution initiale faisable obtenue aléatoirement ou utilisant une heuristique. À chaque itération, la structure de voisinage est exploitée dans le but d'obtenir une solution meilleure que la solution courante. Si la meilleure solution n'est pas améliorée pendant un certain nombre d'itérations, alors un nouveau voisinage généré après un saut de la solution courante est exploité. L'exécution est arrêtée si le nombre maximum d'itérations est atteint ou un autre critère d'arrêt est vérifié. À ce moment, un optimum local qui peut être proche du global est obtenu [DUTA, 2006]. La structure générale de cette méthode est détaillée dans l'Algorithme 3.3.

Les aspects élémentaires de la recherche kangourou sont clarifiés ci-dessous.

Algorithm 3.3 Recherche kangourou

DEBUT
Construire une solution initiale x_0 et la considérer comme la solution courante
Initialiser le compteur de stabilité C $\leftarrow$ 1
Initialiser le nombre maximum d'itérations sans améliorations A
POUR (k = 1 au nombre maximum d'itérations) **FAIRE**
  SI ((C<A)) **ALORS**
    Appliquer la procédure de descente à la solution courante x_k
    SI (la valeur de la fonction objectif de la solution obtenue n'est pas changée) **ALORS**
      Mettre à jour la solution courante x_k
      Incrémenter le compteur de stabilité C $\leftarrow$ C + 1
    SINON
      Mettre à jour la solution courante x_k
      Réinitialiser le compteur de stabilité C $\leftarrow$ 0
    FIN SI
  SINON
    Appliquer l'opérateur de saut à la solution courante x_k
    SI (la valeur de la fonction objectif de la solution obtenue n'est pas changée) **ALORS**
      Mettre à jour la solution courante x_k
      Incrémenter le compteur de stabilité C $\leftarrow$ C + 1
    SINON
      Mettre à jour la solution courante x_k
      Réinitialiser le compteur de stabilité C $\leftarrow$ 0
    FIN SI
  FIN SI
  SI ($f(x_{k+1}) < f(x^*)$) **ALORS**
    Mettre $x^* \leftarrow x_{k+1}$
  FIN SI
FIN POUR
Retourner la solution x^* comme la meilleure solution correspondante au minimum de la fonction objectif f
FIN

- Solution initiale (x_0) : comme pour les approches précédentes, la solution initiale est générée aléatoirement ;
- Génération de voisinage (N_x) : pour la génération des solutions du voisinage, nous utilisons une stratégie hybride basée sur l'exploitation alternée de deux types de mouvement, le mouvement d'insertion et le mouvement de permutation adjacente (IMAPIM). Cette stratégie a pour objectif d'assurer le compromis entre l'exploration et l'exploitation. D'autre part, elle réduit le risque d'être piégé dans un optimum local ;
- Opérateur de saut (J_{op}) : l'opérateur de saut est utilisé comme un mécanisme de diversification lorsque une valeur de la fonction objectif est dominante dans le processus de recherche. Dans ce travail, l'opérateur 2-opt est appliqué à la solution courante pour transférer la recherche à d'autres régions inexploitées. Pour l'opérateur 2-opt, deux tâches sont tirées au hasard, les six combinaisons sont formées et évaluées. La recherche locale est initialisée autour de la meilleure combinaison. Cet opérateur a été également proposé dans d'autres approches d'optimisation [Koç, 2010] ;
- L'algorithme s'arrête lorsque le nombre d'itérations atteint la valeur maximale I_{max} ou lorsque la valeur de la fonction objectif reste constante pendant un nombre donnée d'itérations I_{fstab}.

3.4 Algorithme polynômial pour le problème d'ordonnancement à une machine avec effet d'apprentissage et durées opératoires floues

La formulation du problème d'ordonnancement avec effet d'apprentissage et durées opératoires floues étudié est comme suit. Il existe une ressource avec capacité d'apprentissage (être humain ou machine intelligente) pour effectuer un ensemble de n tâches. Chaque tâche est caractérisée par un poids noté ω_i et une durée opératoire réelle p_i ou floue $\tilde{p}_i$ selon l'environnement. Lorsque une tâche est ordonnancée à la $r^{ème}$ position dans la séquence, sa durée opératoire actuelle devient $p_{i,r} = p_i r^a$. Le fonctionnement de la ressource est sujet aux règles suivantes :

- Il n'y a pas de contraintes de précédence entre les tâches ;
- Les périodes de réparation et de maintenance sont négligeables ;
- Une tâche est servie à la fois ;
- La préemption d'une tâche en exécution n'est pas permise ;
- Toutes les tâches sont supposées prêtes au même temps.

La notation de Graham à trois champs $\alpha|\beta|\gamma$ est utilisée pour décrire les différents problèmes d'ordonnancement analysés dans ce chapitre [Graham et collab., 1979].

3.4.1 Optimalité du problème d'ordonnancement à une machine avec effet d'apprentissage

Dans le cas du problème d'ordonnancement avec des données réelles et sans effet d'apprentissage, l'algorithme de Smith résout à l'optimalité ce problème à une machine avec le critère de la somme pondérée des dates de fin d'exécution en considérant

les tâches dans l'ordre décroissant du rapport $\frac{p_i}{\omega_i}$ connu aussi sous le nom de la règle WSPT [SMITH, 1956]. Si l'effet d'apprentissage est introduit, la règle reste valide mais sous quelques conditions comme il est indiqué par le Théorème suivant [ZHAO et collab., 2004].

Théorème 3.1. *Pour le problème* $1\left|p_{i,r}=p_i r^a\right|\sum_{i=1}^{i=n}\omega_i C_i$, *si les tâches ont des poids agréables i.e.* $p_j \le p_k$ *implique* $\omega_j \ge \omega_k$ *pour toutes les tâches* j *et* k, *alors un ordonancement* Δ *est optimal si et seulement si les conditions suivantes sont satisfaites :*

$$\frac{p_{\Delta(i)}}{\omega_{\Delta(i)}} \le \frac{p_{\Delta(i+1)}}{\omega_{\Delta(i+1)}}, i=1,...,n-1 \tag{3.9}$$

Démonstration.
Necessité des conditions ($\Rightarrow$)
Nous considérons un ordonnancement optimal π pour lequel la règle WSPT n'est pas applicable. Alors, il existe au moins $i \in \{1,...,n-1\}$ tel que $\frac{p_{\pi(i)}}{\omega_{\pi(i)}} > \frac{p_{\pi(i+1)}}{\omega_{\pi(i+1)}}$, ce qui implique $p_{\pi(i)} > p_{\pi(i+1)}$ à cause de l'hypothèse d'agréabilité. En effectuant une permutation adjacente des tâches $\pi(i)$ et $\pi(i+1)$ dans π, nous obtenons un nouvel ordonnancement π'. Les dates de fin d'exécution ne sont pas affectées par la permutation des tâches comme indiqué sur la Figure 3.1.

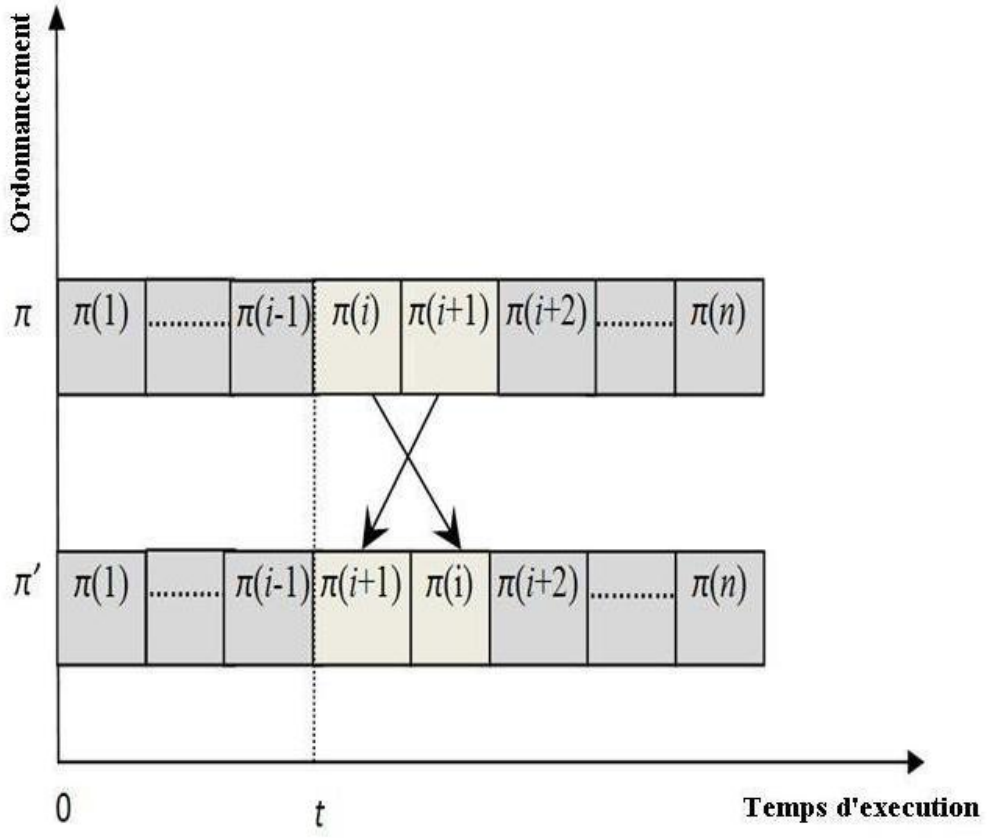

FIGURE 3.1 Application de l'argument de permutation pour les tâches aux positions i et $i+1$.

Dans le but d'évaluer la dominance de l'ordonnancement π' par rapport à l'ordonnance-

ment π, nous procédons comme suit.
Sous l'ordonnancement π nous avons :

$$\begin{cases} C_i(\pi) = t + p_{\pi(i)} i^a \\ \\ C_{i+1}(\pi) = t + p_{\pi(i)} i^a + p_{\pi(i+1)} (i+1)^a \end{cases}$$

Sous l'ordonancement π' nous avons :

$$\begin{cases} C_i(\pi') = t + p_{\pi(i+1)} i^a \\ \\ C_{i+1}(\pi') = t + p_{\pi(i+1)} i^a + p_{\pi(i)} (i+1)^a \end{cases}$$

Nous calculons la différence des deux dates de fin d'exécution de tâche dans la position $(i+1)$ pour les deux ordonnancements :

$$C_{i+1}(\pi') - C_{i+1}(\pi) = \left(p_{\pi(i+1)} - p_{\pi(i)}\right)\left(i^a - (i+1)^a\right)$$

Comme $p_{\pi(i+1)} - p_{\pi(i)} < 0$ et $i^a - (i+1)^a > 0$ alors $C_{i+1}(\pi') < C_{i+1}(\pi)$, ce qui garantie que toutes les tâches ordonnancées après la position $(i+1)$ dans π' ont des dates de fin non supérieures à leurs dates de fin dans π. Maintenant, nous calculons la somme pondérée des deux tâches aux positions (i) et $(i+1)$ pour les deux ordonnancements :

$$\left(\omega_{\pi(i+1)} C_i(\pi') + \omega_{\pi(i)} C_{i+1}(\pi')\right) - \left(\omega_{\pi(i)} C_i(\pi) + \omega_{\pi(i+1)} C_{i+1}(\pi)\right) =$$

$$\omega_{\pi(i+1)}\left(t + p_{\pi(i+1)} i^a\right) + \omega_{\pi(i)}\left(t + p_{\pi(i+1)} i^a + p_{\pi(i)} (i+1)^a\right) -$$

$$\omega_{\pi(i)}\left(t + p_{\pi(i)} i^a\right) - \omega_{\pi(i+1)}\left(t + p_{\pi(i)} i^a + p_{\pi(i+1)} (i+1)^a\right)$$

Après simplification, nous obtenons :

$$\frac{\left(\omega_{\pi(i+1)} C_i(\pi') + \omega_{\pi(i)} C_{i+1}(\pi')\right) - \left(\omega_{\pi(i)} C_i(\pi) + \omega_{\pi(i+1)} C_{i+1}(\pi)\right)}{\left(\omega_{\pi(i+1)} + \omega_{\pi(i)}\right) p_{\pi(i)} i^a} =$$

$$\frac{p_{\pi(i+1)}}{p_{\pi(i)}}\left[1 - \left(\frac{\omega_{\pi(i+1)}}{\omega_{\pi(i+1)} + \omega_{\pi(i)}}\right)\left(\frac{i+1}{i}\right)^a\right] - \left[1 - \left(\frac{\omega_{\pi(i)}}{\omega_{\pi(i+1)} + \omega_{\pi(i)}}\right)\left(\frac{i+1}{i}\right)^a\right]$$

Comme $p_{\pi(i)} \geq p_{\pi(i+1)}$ et $\omega_{\pi(i+1)} \geq \omega_{\pi(i)}$, nous deduisons que la quantité suivante est négative $\left(\omega_{\pi(i+1)} C_i(\pi') + \omega_{\pi(i)} C_{i+1}(\pi')\right) - \left(\omega_{\pi(i)} C_i(\pi) + \omega_{\pi(i+1)} C_{i+1}(\pi)\right)$, ce qui garantie que la contribution à la somme pondérée des dates de fin des tâches aux positions (i) et $(i+1)$ dans l'ordonnancement π' est inférieure à leur contribution dans l'ordonnancement π.
Il résulte des deux inégalités que la somme pondérée des dates de fin sous π' est strictement inférieure à celle sous π, ce qui contredit l'optimalité de π. Donc, les conditions de la règle WSPT sont des conditions nécessaires pour l'optimalité.
Suffisance des conditions (⇐)
Supposons qu'il existe un ordonnancement arbitraire violant l'ensemble des conditions WSPT. Alors, il existe aux moins deux tâches consécutives dans l'ordonnancement qui violent aussi ces conditions. Si nous effectuons une permutation locale de ces tâches, une amélioration de la somme pondérée des dates de fin de l'ordonnancement original aura lieu. En répétant cette procédure pour toutes les tâches non ordonnancées selon la règle WSPT jusqu'à la stabilité de l'amélioration, nous obtenons un ordonnancement optimal. La procédure de classement nécessite un temps de $O(n \times log(n))$. D'où l'ensemble des conditions considérées sont des conditions suffisantes pour l'optimalité d'un

ordonnancement dans le problème $1\left|p_{i,r}=p_i r^a\right|\sum_{i=1}^{i=n}\omega_i C_i$.

□

3.4.2 Optimalité du problème d'ordonnancement à une machine avec effet d'apprentissage et durées opértoires floues

Supposons que chaque tâche est caracterisée par une durée operatoire floue et coefficient de ponderation réel. Dans le but de préserver la validité de la règle de Smith sous paramètres incertains, nous devons en premier généraliser le concept d'agreabilité des poids au cas flou.

Définition 3.8. *On dit que des tâches ont des poids agréables flous si* $\tilde{p}_j \geq_w \tilde{p}_i$ *implique que* $\omega_i \geq \omega_j$ *pour toutes les tâches* i *et* j*. On écrit mathématiquement :*

$$\forall (i,j) \in \mathrm{J}\times\mathrm{J}\left(\tilde{p}_j \geq_w \tilde{p}_i\right) \Rightarrow \left(\omega_i \geq \omega_j\right) \tag{3.10}$$

avec J *est l'ensemble de toutes les tâches.*

Si la fonction de classement est identifiée à la fonction magnitude, nous avons $\left(\tilde{p}_j \geq_w \tilde{p}_i\right) \Leftrightarrow \left(Mag\left(\tilde{p}_j\right) \geq Mag\left(\tilde{p}_i\right)\right)$.
Nous proposons le théorème suivant présentant une approche polynômiale pour le problème $1\left|\tilde{p}_i, p_{i,r}=p_i r^a\right| Mag\left(\sum_{i=1}^{i=n}\omega_i \otimes \tilde{C}_i\right)$. Ce théorème peut être considérée comme une extension du Théorème 3.1 lorsque le modèle d'ordonnancement est sujet à des données incertaines.

Théorème 3.2. *Si les tâches ont des poids agréables flous i.e.* $\tilde{p}_k \geq_w \tilde{p}_j$ *implique que* $\omega_j \geq \omega_k$ *pour toutes tâches arbitraires* j *et* k*, alors, l'ordonnancement* Φ *est optimal si et seulement si les conditions suivantes sont satisfaites :*

$$\frac{\tilde{p}_{\Phi(i)}}{\omega_{\Phi(i)}} w \leq \frac{\tilde{p}_{\Phi(i+1)}}{\omega_{\Phi(i+1)}}, i = 1,...,n-1 \tag{3.11}$$

Démonstration.
Necessité des conditions ($\Rightarrow$)
Nous démontrons la nécessité de ces conditions par contradiction. Nous supposons qu'un ordonnancement donné π est optimal sans satisfaire ces conditions. Donc, il existe au moins deux tâches adjacentes avec $\frac{\tilde{p}_{\pi(i)}}{\omega_{\pi(i)}} >_w \frac{\tilde{p}_{\pi(i+1)}}{\omega_{\pi(i+1)}}$, ce qui implique $p_{\pi(i)} >_w p_{\pi(i+1)}$ due à l'agréabilité floue des tâches. Si nous effectuons une permutation locale des tâches $\pi(i)$ et $\pi(i+1)$ dans π, nous obtenons un nouvel ordonnancement :

$\pi' = (\pi(1),...,\pi(i-1),\pi(i+1),\pi(i),\pi(i+2),...,\pi(n))$.

Les dates de fin floues des tâches ordonnancées en amont de la tâche i ne sont pas affectées par l'opération de permutation. Nous adoptons la même méthodologie que dans le cas réel pour la vérification de la dominance de l'ordonnancement π' par rapport à l'ordonnancement π.
Sous l'ordonnancement π nous avons :

$$\begin{cases} \tilde{C}_i(\pi) = \tilde{t} \oplus \left(i^a \otimes \tilde{p}_{\pi(i)}\right) \\ \tilde{C}_{i+1}(\pi) = \tilde{t} \oplus \left(i^a \otimes \tilde{p}_{\pi(i)}\right) \oplus \left((i+1)^a \otimes \tilde{p}_{\pi(i+1)}\right) \end{cases}$$

et pour l'ordonnancement π' :

$$\begin{cases} \tilde{C}_i(\pi') = \tilde{t} \oplus \left(i^a \otimes \tilde{p}_{\pi(i+1)}\right) \\ \tilde{C}_{i+1}(\pi') = \tilde{t} \oplus \left(i^a \otimes \tilde{p}_{\pi(i+1)}\right) \oplus \left((i+1)^a \otimes \tilde{p}_{\pi(i)}\right) \end{cases}$$

Dans le but d'assurer que toutes les tâches ordonnancées en aval de la position $(i+1)$ dans π' ont une magnitude des dates de fin pas supérieure à leur magnitude de dates de fin dans π, nous calculons la différence :

$$Mag\left(\tilde{C}_{i+1}(\pi')\right) - Mag\left(\tilde{C}_{i+1}(\pi)\right)$$

$$= Mag\left\{\tilde{t} \oplus \left(i^a \otimes \tilde{p}_{\pi(i+1)}\right) \oplus \left((i+1)^a \otimes \tilde{p}_{\pi(i)}\right)\right\}$$

$$-Mag\left\{\tilde{t} \oplus \left(i^a \otimes \tilde{p}_{\pi(i)}\right) \oplus \left((i+1)^a \otimes \tilde{p}_{\pi(i+1)}\right)\right\}$$

L'application de la propriété de linéarité de la fonction magnitude donne :

$$Mag\left(\tilde{C}_{i+1}(\pi')\right) - Mag\left(\tilde{C}_{i+1}(\pi)\right) =$$

$$i^a Mag\left(\tilde{p}_{\pi(i+1)}\right) + (i+1)^a Mag\left(\tilde{p}_{\pi(i)}\right) - i^a Mag\left(\tilde{p}_{\pi(i)}\right) + (i+1)^a Mag\left(\tilde{p}_{\pi(i+1)}\right)$$

$$= \left(Mag\left(\tilde{p}_{\pi(i+1)}\right) - Mag\left(\tilde{p}_{\pi(i)}\right)\right)\left(\left(i^a - (i+1)^a\right)\right)$$

Comme $\tilde{p}_{\pi(i)} >_w \tilde{p}_{\pi(i+1)}$ et $i^a - (i+1)^a > 0$ alors nous déduisons l'inégalité ci-dessous : $Mag\left(\tilde{C}_{i+1}(\pi')\right) < Mag\left(\tilde{C}_{i+1}(\pi)\right)$.

Maintenant, pour garantir que la contribution à la magnitude de la somme pondérée des dates de fin des tâches aux positions (i) et $(i+1)$ dans l'ordonnancement π' est inférieure à leur contribution dans l'ordonnancement π, Nous calculons la différence entre le premier terme donné par $Mag\left\{\left(\omega_{\pi(i+1)} \otimes \tilde{C}_i(\pi')\right) \oplus \left(\omega_{\pi(i)} \otimes \tilde{C}_{i+1}(\pi')\right)\right\}$ et le deuxième terme donné par $Mag\left\{\left(\omega_{\pi(i)} \otimes \tilde{C}_i(\pi)\right) \oplus \left(\omega_{\pi(i+1)} \otimes \tilde{C}_{i+1}(\pi)\right)\right\}$.
En se basant sur la propriété de linearité et après quelques simplifications nous obtenons :

$$\frac{Mag\left(\omega_{\pi(i+1)} \otimes \tilde{C}_i(\pi') \oplus \omega_{\pi(i)} \otimes \tilde{C}_{i+1}(\pi')\right) - Mag\left(\omega_{\pi(i)} \otimes \tilde{C}_i(\pi) \oplus \omega_{\pi(i+1)} \otimes \tilde{C}_{i+1}(\pi)\right)}{i^a\left(\omega_{\pi(i+1)} + \omega_{\pi(i)}\right) Mag\left(\tilde{p}_{\pi(i)}\right)} =$$

$$\frac{Mag(p_{\pi(i+1)})}{Mag(p_{\pi(i)})}\left[1 - \left(\frac{\omega_{\pi(i+1)}}{\omega_{\pi(i+1)} + \omega_{\pi(i)}}\right)\left(\frac{i+1}{i}\right)^a\right] - \left[1 - \left(\frac{\omega_{\pi(i)}}{\omega_{\pi(i+1)} + \omega_{\pi(i)}}\right)\left(\frac{i+1}{i}\right)^a\right]$$

Comme $\tilde{p}_{\pi(i)} \geq_w \tilde{p}_{\pi(i+1)}$ et $\omega_{\pi(i+1)} \geq \omega_{\pi(i)}$, nous déduisons que cette différence est strictement inférieure à zéro. Il en résulte de ces deux inégalités que la magnitude de la somme pondérée des dates de fin sous π' est strictement inférieure à celle sous π, ce qui contredit l'optimalité de π. Donc, les conditions considérées sont des conditions nécessaires pour l'optimalité.

Suffisance des conditions ($\Leftarrow$)

Pour un ordonnancement arbitraire non optimal π, il existe au moins deux tâches adjacentes dans l'ordonnancement qui viole ces conditions. Nous pouvons améliorer la valeur de magnitude de la somme pondérée des dates de fin en permutant ces tâches. Si cette procédure est itérée pour toutes les tâches non respectant la règle WSPT

jusqu'à l'obtention d'une stabilité, un ordonnancement optimal est obtenu dans un temps polynômial $O(n \times log(n))$. Donc, nous concluons que l'ensemble des conditions considérées sont aussi des conditions suffisantes d'optimalité d'un ordonnancement dans le problème $1\left|\tilde{p}_i, p_{i,r} = p_i r^a\right| \mathrm{M}ag\left(\sum_{i=1}^{i=n} \omega_i \otimes \tilde{\mathrm{C}}_i\right)$.

□

L'exemple suivant illustre l'utilité du concept d'agréabilité floue dans l'application du Théorème 3.2.

Exemple 3.1. *Nous considérons une instance du problème* $1\left|\tilde{p}_i, p_{i,r} = p_i r^a\right| \mathrm{M}ag\left(\sum_{i=1}^{i=n} \omega_i \otimes \tilde{\mathrm{C}}_i\right)$ *avec cinq tâches. L'ensemble des valeurs des paramètres du modèle est présenté dans le Tableau 3.1. Il est possible de vérifier sur ce tableau que les différentes tâches satisfaient le concept d'agréabilité floue.*

TABLEAU 3.1 Valeurs des parametèrs d'ordonnancement associées à l'ensemble des tâches J

$\mathrm{J}_{i=1,5}$	Durée opératoire floue $\tilde{p}_i$	Magnitude $\mathrm{M}ag\left(\tilde{p}_i\right)$	Poids ω_i
1	(31,35,3,2)	1.49	22
2	(9,18,1,0)	0.45	30
3	(28,54,1,2)	4.11	10
4	(31,42,0,3)	2.63	14
5	(24,38,1,2)	1.35	23

La Figure 3.2 illustre les durées opératoires floues des tâches où le chevauchement des nombres flous rend l'élaboration d'un classement au sens conventionnel impossible. Le calcul de la fonction magnitude pour les différents ordonnancements utilisant une recherche exhaustive donne la solution optimale $\pi^* = (2,5,1,4,3)$*, qui correspond bien au résultat de l'application du Théorème 3.2.*

Dans la Figure 3.3, l'évolution de la fonction magnitude de la somme pondérée des dates de fin pour la solution optimale est représentée. Comme attendu, la fonction objectif a une allure décroissante en fonction de la valeur absolue du taux d'apprentissage, ce qui confirme la réduction des coûts à cause de l'amélioration de la capacité de la ressource à réaliser les tâches allouées.

Mais l'hypothèse d'agréabilité floue est une contrainte très forte et n'est pas satisfaite dans la majorité des situations pratiques. Pour cette raison, il est plus approprié de concentrer les efforts sur le développement d'outils génériques ayant la capacité de résoudre une multitude de problèmes sophistiqués sans avoir une connaissance approfondie sur les propriétés intrinsèques de ces problèmes.

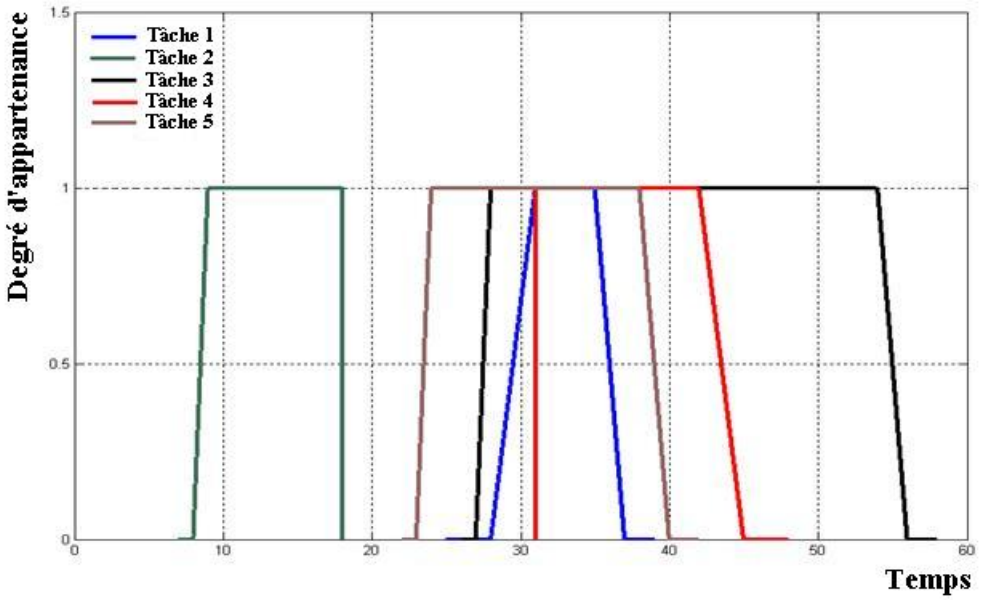

FIGURE 3.2 Représentation graphique des durées opératoires floues.

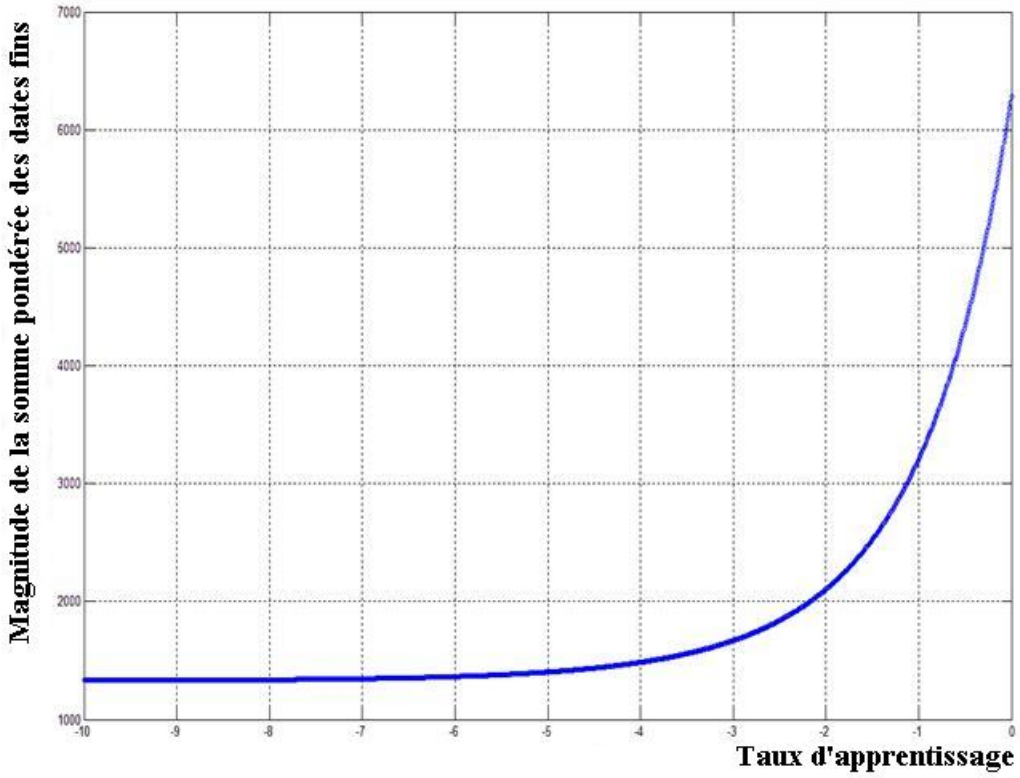

FIGURE 3.3 Évolution de la magnitude de la somme pondérée des dates de fin en fonction du taux d'apprentissage.

3.5 Approches d'optimisation pour le problème d'ordonnancement à une machine avec dates de disponibilité floues, durées opératoires floues et effet d'apprentissage à base de position

Dans cette section, nous présentons en premier la formulation floue du problème d'ordonnancement proposé. Ensuite, nous abordons la procédure de génération des instances utilisées dans les expérimentations numériques et finalement nous évaluons les performances des trois métaheuristiques adoptées à la base d'une analyse statistique.

3.5.1 Formulation du problème d'ordonnancement

Dans le but d'approcher les spécificités des problèmes d'ordonnancement réels, nous conservons la description générale du problème précédent avec la considération de quelques contraintes supplémentaires. La restriction des poids avec contrainte d'agréabilité floue est relaxée et des dates de disponibilité floues différentes $\tilde{r}_i = \left(\underline{r}_i, \overline{r}_i, \alpha_{\tilde{r}_i}, \beta_{\tilde{r}_i}\right)_{\mathrm{L-R}}$ pour $i = \overline{1,n}$ sont affectées aux tâches. Dans ce cas, la date fin floue de chaque tâche i peut être calculée utilisant l'opérateur de somme floue, multiplication avec un scalaire et la dominance faible pour la détermination du maximum de deux nombres flous. Comme fonction de classement, nous adoptons la magnitude d'un nombre flou proposée par Abasabandy et Hajjeri [ABBASBANDY et HAJJARI, 2009], qui est également utilisée dans la comparaison entre différentes ordonnancements. La date fin floue d'une tâche i dans un ordonnancement π est calculée utilisant la formule de récurrence suivante :

$$\begin{cases} \tilde{C}_1(\pi) = \tilde{r}_1(\pi) \oplus \tilde{p}_1(\pi) \\ \tilde{C}_i(\pi) = \max_w \left\{\tilde{r}_i(\pi), \tilde{C}_{i-1}(\pi)\right\} \oplus \tilde{p}_i(\pi) \quad pour \quad i = \overline{2,\ n} \end{cases} \tag{3.12}$$

La manipulation mathématique de cette formule donne :

$$\begin{cases} \tilde{C}_1(\pi) = \left(\underline{r}_1 + \underline{p}_1, \overline{r}_1 + \overline{p}_1, \alpha_{\tilde{r}_1} + \alpha_{\tilde{p}_1}, \beta_{\tilde{r}_1} + \beta_{\tilde{p}_1}\right) \\ \tilde{C}_i(\pi) = \begin{cases} \left(\underline{r}_i + \underline{p}_i i^a, \overline{r}_i + \overline{p}_i i^a, \alpha_{\tilde{r}_i} + \alpha_{\tilde{p}_i} i^a, \beta_{\tilde{r}_i} + \beta_{\tilde{p}_i} i^a\right) \\ si \ \ Mag(\tilde{r}_i(\pi)) > Mag\left(\tilde{C}_{i-1}(\pi)\right) \\ \left(\underline{C}_{i-1} + \underline{p}_i i^a, \overline{C}_{i-1} + \overline{p}_i i^a, \alpha_{\tilde{C}_{i-1}} + \alpha_{\tilde{p}_i} i^a, \beta_{\tilde{C}_{i-1}} + \beta_{\tilde{p}_i} i^a\right) \\ si \ \ Mag(\tilde{r}_i(\pi)) \leq Mag\left(\tilde{C}_{i-1}(\pi)\right) \end{cases} \quad i = \{2, \ldots, n\} \end{cases} \tag{3.13}$$

La fonction objectif à optimiser est la magnitude de la somme pondérée des dates de fin floues des tâches. En adoptant la notation à trois champs, nous notons ce problème par $1\left|\tilde{r}_i, \tilde{p}_i, p_{i,r} = p_i r^a\right| Mag\left(\sum_{i=1}^{i=n} \omega_i \otimes \tilde{C}_i\right)$. Si l'effet d'apprentissage n'est pas considéré et tous les paramètres ont des valeurs réelles, ce problème se réduit au problème $1|r_i|\sum_{i=1}^{i=n} \omega_i C_i$ qui est prouvé NP-difficile au sens fort [LENSTRA et collab., 1977]. Donc, notre version

floue a au moins la même complexité de calcul. La résolution de ce problème est basée sur les trois métaheuristiques à base de trajectoire (la recherche tabou, le recuit simulé et la recherche kangourou).

3.5.2 Procédure de génération des jeux tests

La procédure de génération des instances est une étape primordiale avant l'évaluation des performances des méthodes d'optimisation. Le nombre de tâches n est fixé pour chaque instance où l'incertitude au niveau des durées opératoires et les dates de disponibilité sont modélisées par des nombres flous trapézoïdaux tandis que les poids sont modélisés par des valeurs entières positives. La procédure de génération proposée est basée essentiellement sur les travaux [GHRAYEB, 2003], [SAIDI MEHRABAD et PAHLAVANI, 2009] et [YIN et collab., 2014]. Alors, les paramètres d'ordonnancement sont générés selon les règles suivantes :

i) Une valeur aléatoire p_i est prise d'une distribution uniforme sur l'intervalle $\left[a_{\tilde{p}_i}, b_{\tilde{p}_i}\right]$ pour chaque tâche $i = \overline{1,n}$.

ii) Les bornes inférieures et supérieures des durées opératoires floues $\tilde{p}_i$ sont générées aléatoirement selon les relations :

$$\begin{cases} \underline{p}_i \leftarrow \left[\left(1-\rho_1\right)p_i, p_i\right] \\ \overline{p}_i \leftarrow \left[p_i, \left(1+\rho_1\right)p_i\right] \end{cases}$$

avec ρ_1 une première graine aléatoire appartenant à l'intervalle $[0,1]$.

iii) Les déviations gauche et droite ($\alpha_{\tilde{p}_i}$ et $\beta_{\tilde{p}_i}$) des durées opératoires floues sont définies arbitrairement comme des entiers de l'intervalle $\left[0, 0.1 \times \left(\frac{\underline{p}_i + \overline{p}_i}{2}\right)\right]$.

iv) Les bornes supérieures et inférieures des dates de disponibilité floues $\tilde{r}_i$ sont générées selon les relations :

$$\begin{cases} \underline{r}_i = \left[0,\ \rho_2 \sum_{i=1}^{i=n} \underline{p}_i\right] \\ \bar{r}_i = \left[\rho_2 \sum_{i=1}^{i=n} \underline{p}_i,\ \rho_2 \sum_{i=1}^{i=n} \bar{p}_i\right] \end{cases}$$

avec ρ_2 une deuxième graine aléatoire supérieure à zéro.

v) Les déviations gauche et droite ($\alpha_{\tilde{r}_i}$ et $\beta_{\tilde{r}_i}$) des dates de disponibilité floues sont définies d'une façon analogue comme des entiers arbitraires de l'intervalle $\left[0, 0.1 \times \left(\frac{\underline{r}_i + \overline{r}_i}{2}\right)\right]$.

vi) Les poids ω_i sont générés comme des entiers positifs d'une distribution discrète uniforme $\left[a_{\omega_i}, b_{\omega_i}\right]$

vii) Des valeurs élevées de ρ_1 et ρ_2 indiquent des intervalles étendus des durées opératoires $\tilde{p}_i$ et dates de disponibilité floues $\tilde{r}_i$ respectivement. Alors que des valeurs faibles indiquent des intervalles étroits pour les valeurs affectées à ces paramètres.

3.5.3 Évaluation des résultats

Les expérimentations numériques sont réalisées à l'aide d'un ensemble d'instances pour évaluer l'efficacité des méthodes proposées, où chaque ensemble test contient des

instances de taille fixe. Le nombre de tâches varie de 7 à 40 tâches. Comme l'exactitude des résultats dépend fortement de la taille de l'échantillon, un nombre défini d'instances n_{inst} est généré pour chaque nombre de tâches. La procédure de génération ainsi que les trois métaheuristiques sont implémentées utilisant l'outil MATLAB qui est un environnement émergeant pour les problèmes de modélisation et d'optimisation [WONGA et LAI, 2011]. Les expériences sont exécutées sur un processeur $i7$ avec 8 GB de RAM. Les instances complètes du problème traité avec toutes ses données sont disponibles sur le lien : <http://lab.univ-batna2.dz/lap/index.php/component/content/article?id=31>.
Le Tableau 3.2 donne les bornes inférieures et supérieures des paramètres nécessaires pour la procédure de génération.

TABLEAU 3.2 Valeurs des différents paramètres utilisées lors de la procédure de génération des instances

Paramètre	Borne inférieure	Borne supérieure
p_i	1	40
ω_i	1	15
n_{inst}	40	40

Comme les performances des méthodes d'optimisation dépendent du choix adéquat de leurs paramètres de configuration, nous effectuons quelques tests de sensibilité pour les valeurs des paramètres. Ces tests sont achevés uniquement pour les instances de petite taille à cause des coûts de calcul. Heureusement, nous n'avons pas détecté une grande différence dans la qualité des résultats. Dans le Tableau 3.3, nous listons les paramètres principaux ainsi que leurs valeurs associées pour les trois métaheuristiques. Il est à noter que ces paramètres ne sont pas changés à l'exception du cas où nous réalisons une analyse de sensibilité en fonction d'un paramètre spécifique.

TABLEAU 3.3 Valeurs de configuration associées aux métaheuristiques à base de trajectoire

Paramètre	Recherche tabou	Recuit simulé	Recherche kangourou
x_0	Aléatoire	Aléatoire	Aléatoire
N_x	GPIM	IM	IMAPIM
TL	$\lfloor\sqrt{n}, 2n\rfloor$	/	/
TL_{Up}	n	/	/
N_{tr}	$n-1$	$n-1$	$n-1$
T_0	/	$-\frac{(f(x_{worst})-f(x_{best}))}{\log(p_0)}$	/
p_0	/	0.8	/
I_T	/	n	/
r_{cool}	/	0.9	/
J_{op}	/	/	2-opt
I_{fstab}	$7.5 \times n$	$3 \times log(n!)$	$7.5 \times n$
I_{max}	$15 \times n$	$15 \times n$	$15 \times n$

/ : Non applicable

D'une façon similaire à l'étude de Allahverdi et Al-Anzi [ALLAHVERDI et AL-ANZI, 2006], trois critère statistiques sont adoptés pour comparer les performances des mé-

thodes d'optimisation utilisées dans ce chapitre : la moyenne des erreurs relatives, l'écart type des erreurs relatives et le pourcentage des meilleures solutions trouvées pour les 40 instances. Ces critères sont notés par Avg, Std et POB respectivement. L'erreur relative est définie pour chaque métaheuristique dans le cas de petites instances ($n \leq 10$) par :

$$Erreur = 100 \times \left(\frac{f\left(Métaheuristique\right) - f\left(Optimale\right)}{f\left(Optimale\right)} \right) \tag{3.14}$$

et pour les moyennes et les grandes instances par :

$$Erreur = 100 \times \left(\frac{f\left(Métaheuristique\right) - f\left(Meilleure\right)}{f\left(Meilleure\right)} \right) \tag{3.15}$$

avec f représente la fonction objectif $Mag\left(\sum_{i=1}^{i=n} \omega_i \otimes \tilde{C}_i\right)$.

Dans le Tableau 3.4, les trois métaheuristiques à base de trajectoire sont exploitées avec variation du nombre de tâches. Nous pouvons déduire que ces méthodes s'adaptent bien à l'ensemble des instances car la moyenne des erreurs relatives obtenue pour 40 instances est inférieure à 1%, ce qui reflète de bonnes solutions proches de l'optimale. En outre, la recherche kangourou donne les meilleures performances en fonction du pourcentage des meilleures solutions obtenues, tandis que le recuit simulé est classé deuxième avant la recherche tabou. La performance supérieure de la recherche kangourou est justifiée par l'efficacité de sa méthode de recherche locale à deux phases combinée avec un opérateur 2-opt de saut, permettant un équilibre entre l'intensification et la diversification dans l'espace de recherche.

TABLEAU 3.4 Variation des performances des métaheuristiques proposées en fonction du nombre de tâches n pour $L_{eff} = 0$, $\rho_1 = 0.1$ et $\rho_2 = 0.1$

Paramètre	Recherche tabou			Recuit simulé			Recherche kangourou		
n	Avg	Std	POB	Avg	Std	POB	Avg	Std	POB
07	0.000	0.000	100	0.032	0.138	92.5	0.000	0.000	100
08	0.042	0.202	82.5	0.057	0.213	90.0	0.000	0.000	100
09	0.114	0.462	87.5	0.058	0.254	90.0	0.056	0.350	95.0
10	0.058	0.143	55.0	0.094	0.233	52.5	0.037	0.233	67.5
15	0.105	0.113	20.0	0.158	0.531	35.0	0.002	0.013	97.5
20	0.108	0.097	07.5	0.097	0.229	27.5	0.125	0.262	75.0
25	0.268	0.592	17.5	0.187	0.491	17.5	0.369	1.218	67.5
30	0.378	0.673	02.5	0.219	0.396	17.5	0.070	0.219	80.0
35	0.196	0.449	12.5	0.215	0.555	25.0	0.238	0.454	62.5
40	0.332	0.595	20.0	0.359	0.950	25.0	0.492	1.032	55.0

La Figure 3.4 résume le temps CPU moyen (en seconde) de l'exécution des trois métaheuristiques. Comme indiqué sur cette figure, les temps CPU moyens des recherches tabou et kangourou sont prèsque superposés et donnent approximativement des résultats identiques tandis que le recuit simulé montre une petite différence relativement aux deux approches précédentes avec des grandes valeurs pour un nombre de tâches supérieur à 25. Pour les valeurs inférieures à 15 tâches, il n'y a pas de différance remarquable entre les temps CPU des trois métaheuristiques.

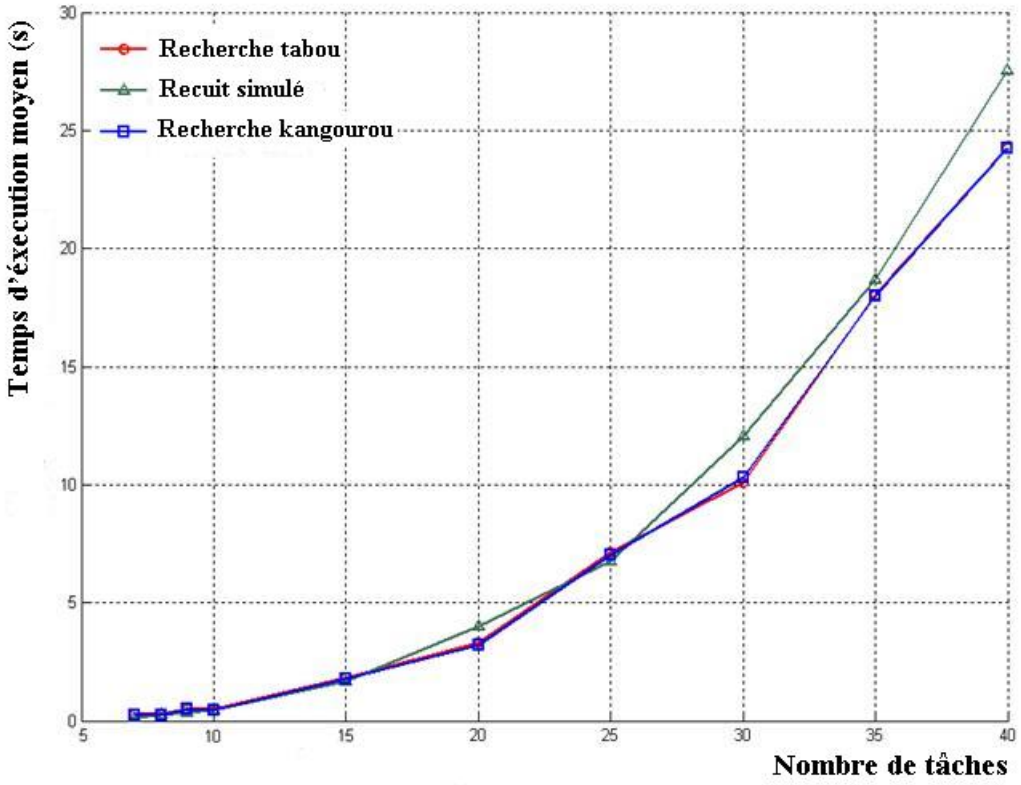

FIGURE 3.4 Variation du temps d'exécution moyen en fonction du nombre de tâches (a = 0, $\rho_1 = 0.1$ et $\rho_2 = 0.1$).

Le Tableau 3.5 reporte les variations des performances des métaheuristiques en fonction du coefficient d'apprentissage. Il est à noter dans ce cas que la recherche kangourou dépasse les deux autres techniques en fonction de pourcentage des meilleures solutions obtenues. Pour des petites valeurs du coefficient d'apprentissage, la majorité des meilleures solutions est associée à cette méthode avec des valeurs très faibles de la moyenne des erreurs relatives et l'écart type des erreurs relatives. En effet, le problème d'ordonnancement considéré a l'allure de devenir plus facile à résoudre lorsque le coefficient d'apprentissage est augmenté en valeur absolue résultant en une réduction significative des durées opératoires floues des tâches. Pour cette raison, les performances des trois métaheuristiques s'améliorent avec la valeur absolue du coefficient d'apprentissage. Les performances de recherche tabou et le recuit simulé restent inférieures à la recherche kangourou à cause de l'efficacité de la recherche locale hybride utilisée par cette dernière métaheuristique.

Les Figures 3.5, 3.6 et 3.7 clarifient la variation de la moyenne des erreurs relatives en fonction des graines aléatoires (ρ_1 et ρ_2)pour les trois méthodes proposées avec $n = 20$ et $L_{eff} = 0$. Toutes les deux graines sont variées de 0.1 à 1 avec un pas de 0.1. Une caractéristique intéressante commune à toutes les courbes est reliée à la complexité de résoudre les instances générées utilisant différentes configurations des graines peut être observée. Les courbes de contour projetées sur le plan formé de ρ_1 et ρ_2 permettent de détecter les valeurs élevées (en rouge) et les valeurs faibles (en bleu) de la moyenne des erreurs relatives. Les configurations associées avec des valeurs élevées indiquent des instances difficiles tandis que celles associées avec des valeurs faibles indique les instances faciles à résoudre. À partir de ces courbes, les instances difficiles sont situées en géné-

TABLEAU 3.5 Variation des performances des métaheuristiques proposées en fonction du coefficient d'apprentissage L_{eff} pour $\rho_1 = 0.1$ et $\rho_2 = 0.1$

Paramètre	Recherche tabou			Recuit simulé			Recherche kangourou		
L_{eff}	Avg	Std	POB	Avg	Std	POB	Avg	Std	POB
0.0	0.105	0.126	17.5	0.086	0.155	20.0	0.062	0.189	80.0
-0.2	0.102	0.103	10.0	0.154	0.245	17.5	0.139	0.344	82.5
-0.4	0.152	0.328	10.0	0.159	0.242	12.5	0.078	0.240	85.0
-0.6	0.079	0.096	22.5	0.151	0.218	22.5	0.242	0.655	70.0
-0.8	0.104	0.132	07.5	0.129	0.166	17.5	0.102	0.332	85.0
-1.0	0.215	0.902	15.0	0.139	0.283	12.5	0.093	0.294	77.5
-1.2	0.040	0.077	22.5	0.257	0.511	15.0	0.177	0.471	67.5
-1.4	0.134	0.346	35.0	0.242	0.353	10.0	0.069	0.142	62.5
-1.6	0.399	0.119	35.0	0.276	0.389	02.5	0.063	0.146	65.0
-1.8	0.254	0.036	30.0	0.189	0.192	05.0	0.115	0.570	70.0
-2.0	0.042	0.069	22.5	0.139	0.138	05.0	0.015	0.052	75.0
-2.2	0.041	0.061	12.5	0.145	0.224	05.0	0.012	0.042	87.5
-2.4	0.032	0.045	12.5	0.098	0.144	02.5	0.023	0.085	87.5
-2.6	0.046	0.069	05.0	0.115	0.156	07.5	0.002	0.007	95.0
-2.8	0.020	0.029	17.5	0.070	0.165	17.5	0.003	0.019	95.0
-3.0	0.034	0.049	07.5	0.066	0.110	02.5	0.000	0.001	95.0
-3.2	0.031	0.042	07.5	0.060	0.156	10.0	0.000	0.001	95.0
-3.4	0.029	0.042	02.5	0.040	0.062	07.5	0.000	0.000	100
-3.6	0.345	0.070	00.0	0.040	0.068	10.0	0.000	0.000	100
-3.8	0.286	0.053	02.5	0.079	0.203	10.0	0.000	0.000	100
-4.0	0.025	0.052	05.0	0.146	0.405	02.5	0.000	0.000	100

rale au voisinage des valeurs centrales de ρ_1 et ρ_2. Tandis que les instances faciles sont localisées aux petites valeurs de ρ_2. Cette dernière remarque est expliquée par le fait que les valeurs faibles de la deuxième graine aléatoire mène à des dates de disponibilité très serrées proches du zero comme indiqué dans la procédure de génération des instances. Ce problème est connu dans le cas avec paramètres réels comme étant un problème de complexité polynômiale.

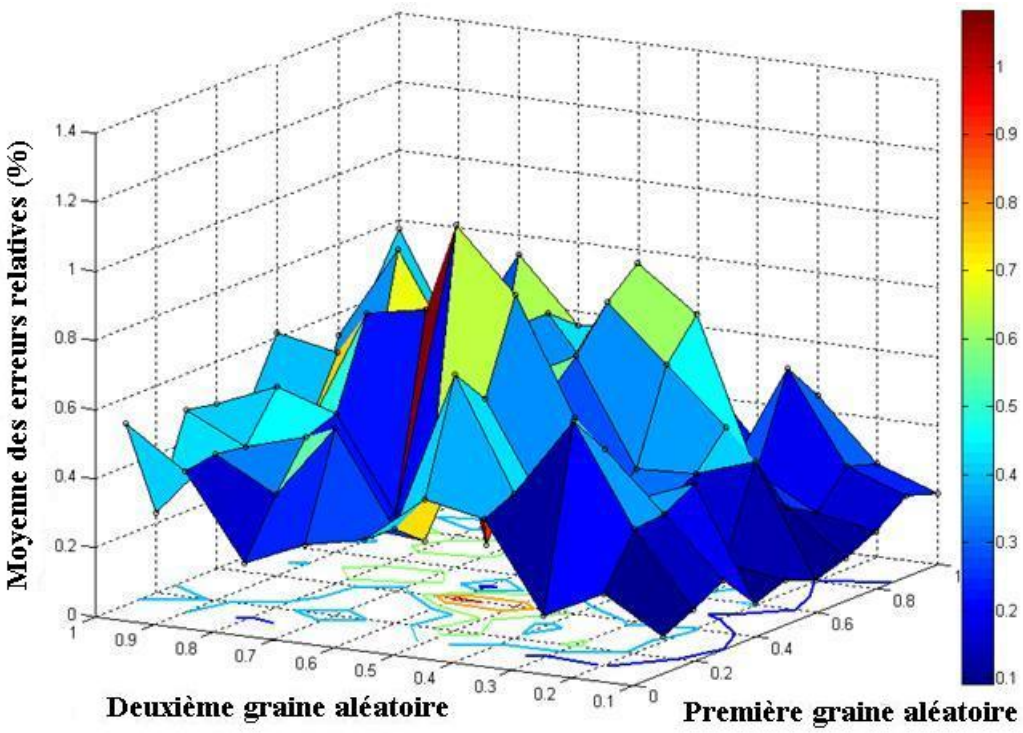

FIGURE 3.5 Variation de la moyenne des erreurs relatives en fonction des valeurs des graines aléatoires dans le cas de la recherche tabou.

Le même comportement concernant les instances difficiles et faciles peut être observé en fonction de l'écart type des erreurs relatives pour les métaheuristiques proposées comme indiquée par les Figures 3.8, 3.9 et 3.10. Mais avec l'existence des régions des deux types d'instances le long du plan de projection.

Pour une analyse plus fiable des performances de la recherche tabou et le recuit simulé, nous conduisons un test d'hypothèse pour comparer ces deux méthodes. La recherche kangourou n'est pas prise en compte car elle donne dans la majorité des cas les meilleurs résultats. Mais avant d'entamer ce test, nous avons besoin de vérifier l'hypothèse de normalité qui est primordiale dans plusieurs méthodes de test paramétrique telles que t-test et F-test [LEHMANN et ROMANO, 2005]. Nous effectuons le test de Lilliefors sur l'échantillon formé par les erreurs relatives au Tableau 3.4 pour la recherche tabou et le recuit simulé, ce qui donne deux échantillons chacune contient 400 observations. Les valeurs p des deux méthodes sont inférieures au niveau de signification (1%) et l'hypothèse nulle de normalité est rejetée à 1% de niveau de signification. Alors, nous concluons que les deux échantillons ne peuvent pas être considérés comme issus de la même population avec distribution normale. Il est possible de valider ce résultat par un diagramme de probabilité normale utilisant les données des deux échantillons comme représenté dans la Figure 3.11, où une différence claire peut être observée entre l'ensemble

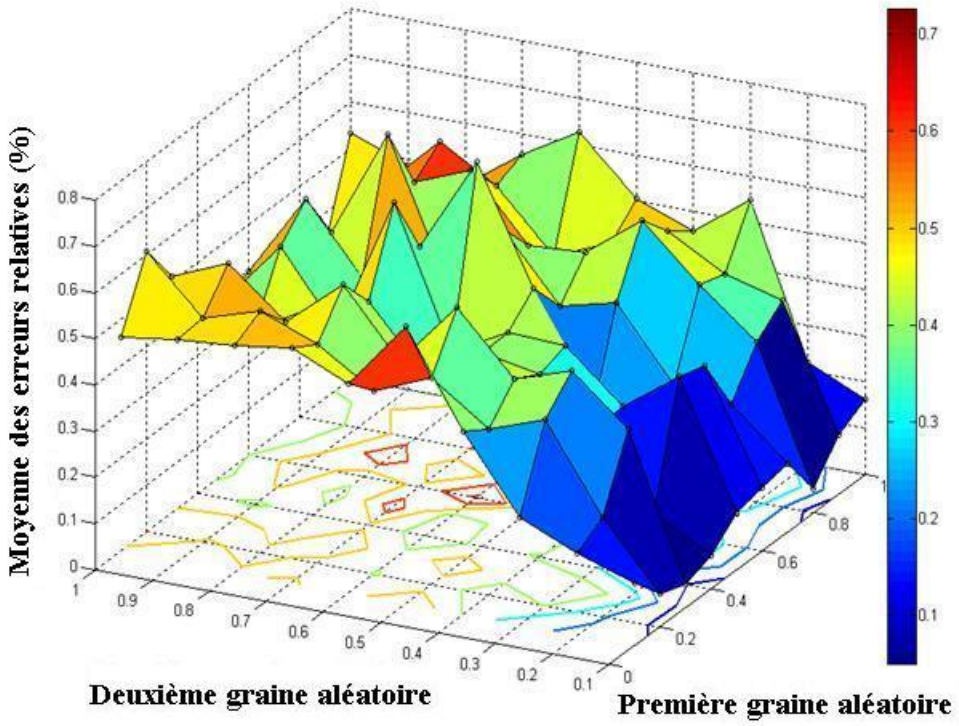

FIGURE 3.6 Variation de la moyenne des erreurs relatives en fonction des valeurs des graines aléatoires dans le cas du recuit simulé.

des données et l'interpolation linéaire des ordres statistiques des échantillons des deux métaheuristiques.

Comme l'hypothèse de normalité n'est pas satisfaite, nous proposons d'utiliser un test d'hypothèse non paramétrique. La médiane est utilisée au lieu de la moyenne comme paramètre statistique de comparaison et les deux hypothèses suivantes sont considérées :

H_0 : La médiane des erreurs relatives de la recherche tabou = La médiane des erreurs relatives du recuit simulé (Hypothèse nulle)
H_1 : La médiane des erreurs relatives de la recherche tabou $\neq$ La médiane des erreurs relatives du recuit simulé (Hypothèse alternative)

Donc, nous utilisons le test de la somme des rangs de Wilcoxon pour évaluer la signification statistique de l'hypothèse nulle H_0 que les échantillons sont extraits des populations avec des médianes égales contre l'hypothèse alternative H_1 que les médianes sont différentes [GIBBONS et CHAKRABORTI, 2003]. La valeur calculée de p est trouvée supérieure au niveau de signification ($0.02 > 0.01$), ce qui reflète que le test a échoué de rejeter l'hypothèse nulle à un niveau de signification de 1%. Nous concluons de ce test d'hypothèse que la recherche tabou et le recuit simulé sont statistiquement très proches en termes de qualité de solutions.

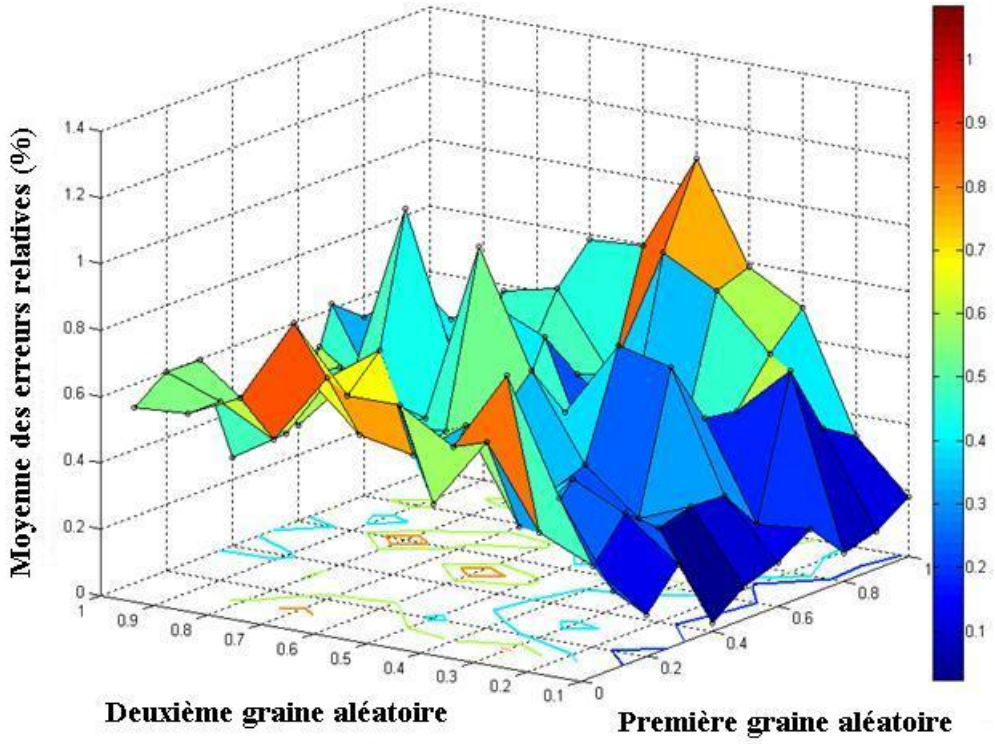

FIGURE 3.7 Variation de la moyenne des erreurs relatives en fonction des valeurs des graines aléatoires dans le cas de la recherche kangourou.

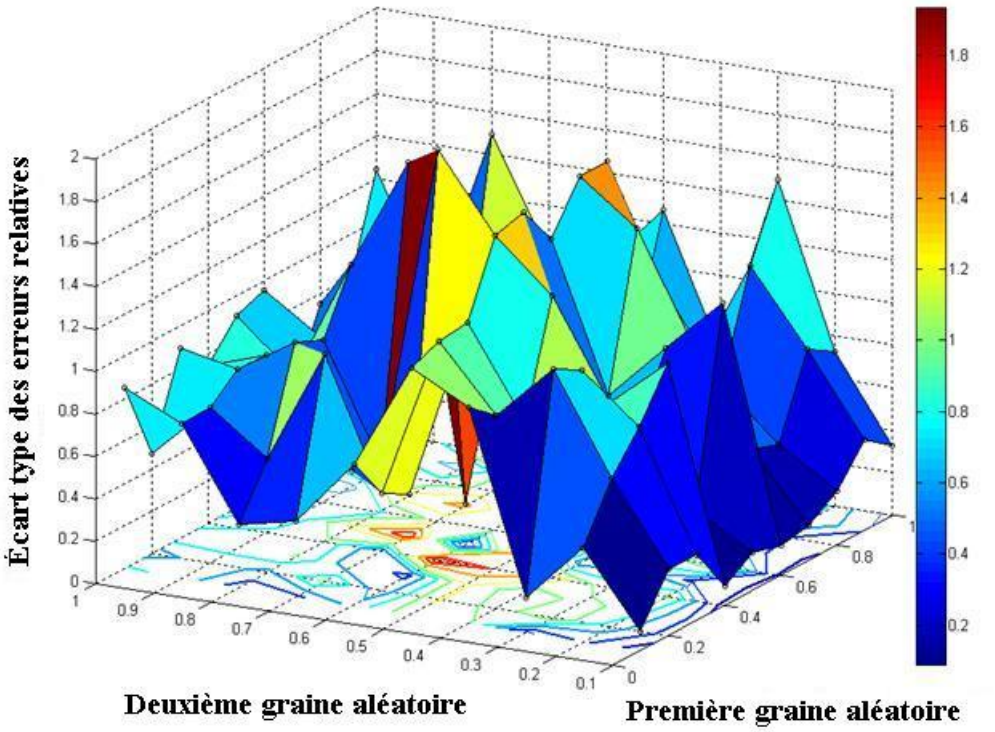

FIGURE 3.8 Variation de l'écart type des erreurs relatives en fonction des valeurs des graines aléatoires dans le cas de la recherche tabou.

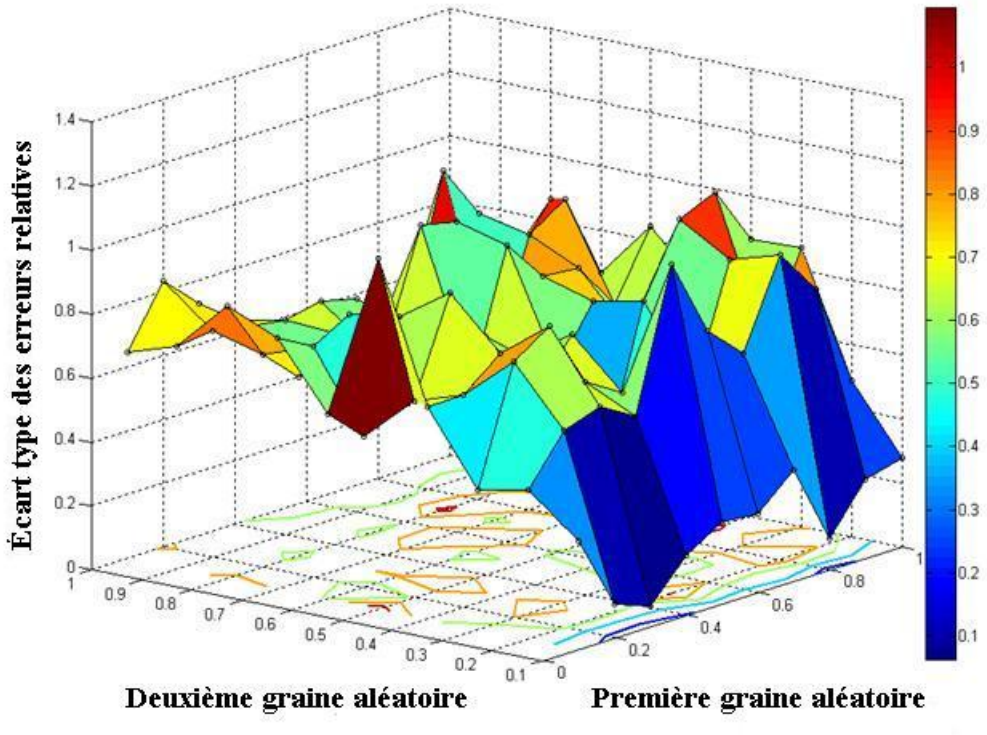

FIGURE 3.9 Variation de l'écart type des erreurs relatives en fonction des valeurs des graines aléatoires dans le cas du recuit simulé.

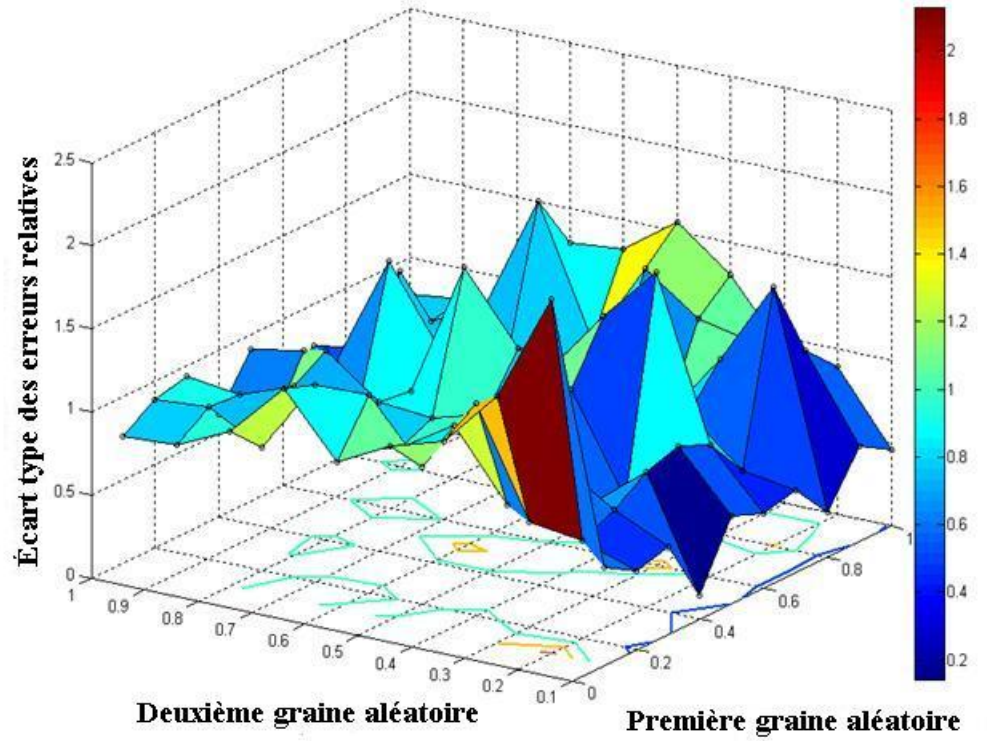

FIGURE 3.10 Variation de l'écart type des erreurs relatives en fonction des valeurs des graines aléatoires dans le cas de la recherche kangourou.

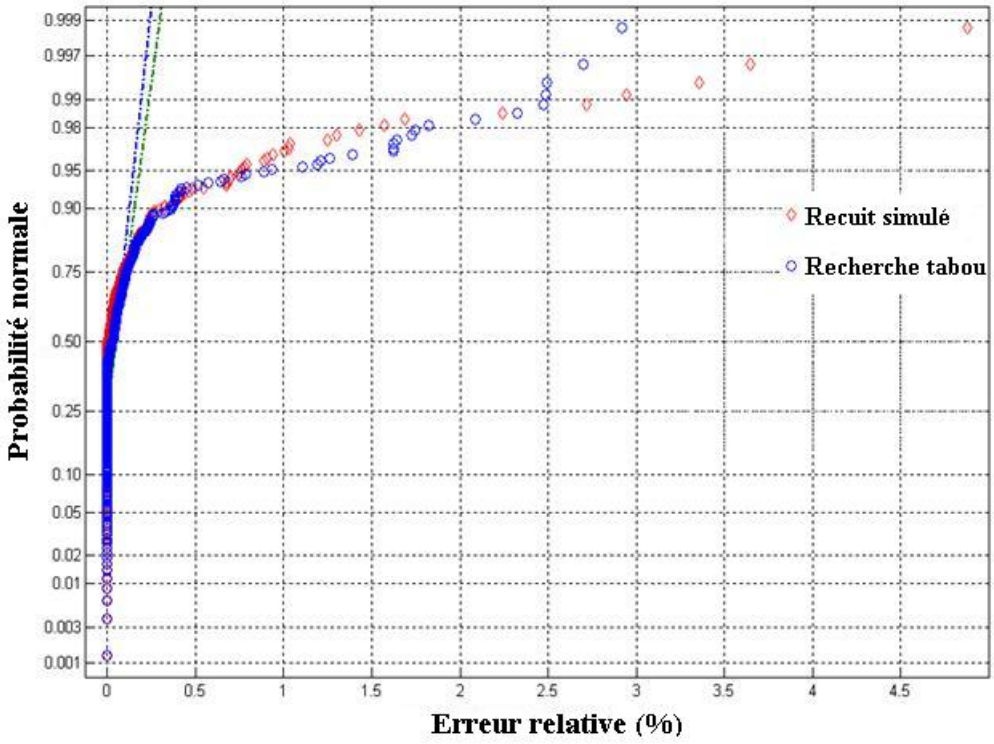

FIGURE 3.11 Représentation du diagramme de probabilité normale pour l'échantillon des données dans le cas de la recherche tabou et le recuit simulé.

3.6 Conclusion

Dans ce chapitre, nous avons considéré un problème d'ordonnancement à une machine avec effet d'apprentissage à base de position et paramètres incertains. Nous avons concentré sur deux aspects principaux notamment la proposition d'un algorithme polynômial pour résoudre la version simplifiée du problème lorsque le concept d'agréabilité floue est satisfait pour les poids. Le deuxième aspect est relatif au développement des métaheuristiques à base de trajectoire (la recherche tabou, le recuit simulé et la recherche kangourou) pour la résolution d'une version NP-difficile du problème lorsque des dates de disponibilité différentes sont présentes et l'hypothèse d'agréabilité est omise. Des expérimentations numériques intensives ont été conduites en fonction de plusieurs paramètres en plus de l'analyse statistique basée sur les tests d'hypothèse. Les résultats ont montré que les métaheuristiques proposées ont des performances acceptables en termes de la bonne qualité des solutions et le coût de calcul raisonnable. Mais quelques difficultés ont été détectées pendant le choix des valeurs des paramètres de configuration.

Les approches de modélisation et de résolution concrétisées dans ce chapitre ont constitué le point de départ à des contributions relatives à deux axes de recherche. Le premier axe concerne le développement de modèles pour des fonctions objectifs dans le cas où les nombres flous décrivant les paramètres incertains sont d'allure plus générale notamment les intervalles flous à base de puissance. Le deuxième axe consiste à évaluer les performances des approches d'optimisation multi-objectif en présence de différents critères de domination. Ces deux aspects seront traités dans le dernier chapitre de cette thèse.

3.7 Références

ABBASBANDY, S. et T. HAJJARI. 2009, «A new approach for ranking of trapezoidal fuzzy numbers», *Computers and Mathematics with Applications*, vol. 57, nº 3, p. 413–419. 79, 93

AHMADIZAR, F. et L. HOSSEINI. 2011, «Single-machine scheduling with a position-based learning effect and fuzzy processing times», *The International Journal of Advanced Manufacturing Technology*, vol. 56, nº 5, p. 693–698. 76

AHMADIZAR, F. et L. HOSSEINI. 2013, «Minimizing makespan in a single-machine scheduling problem with a learning effect and fuzzy processing times», *The International Journal of Advanced Manufacturing Technology*, vol. 65, nº 1, p. 581–587. 76

AL-TURKI, U., C. FEDJKI et A. ANDIJANI. 2001, «Tabu search for a class of single-machine scheduling problems», *Computers and Operations Research*, vol. 28, nº 12, p. 1223–1230. 81

ALLAHVERDI, A. et F. AL-ANZI. 2006, «A pso and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application», *Computers and Operations Research*, vol. 33, nº 4, p. 1056–1080. 95

BACHMAN, A. et A. JANIAK. 2004, «Scheduling jobs with position-dependent processing times», *The Journal of the Operational Research Society*, vol. 55, nº 3, p. 257–264. 74

BISKUP, D. 1999, «Single-machine scheduling with learning considerations», *European Journal of Operational Research*, vol. 115, nº 1, p. 173–178. 73

BISKUP, D. 2008, «A state-of-the-art review on scheduling with learning effects», *European Journal of Operational Research*, vol. 188, nº 2, p. 315–329. 75

CHANAS, S. et A. KASPERSKI. 2001, «Minimizing maximum lateness in a single machine scheduling problem with fuzzy processing times and fuzzy due dates», *Engineering Applications of Artificial Intelligence*, vol. 14, nº 3, p. 377–386. 75

CHANAS, S. et A. KASPERSKI. 2004, «Possible and necessary optimality of solutions in the single machine scheduling problem with fuzzy parameters», *Fuzzy Sets and Systems*, vol. 142, nº 3, p. 359–371. 76

DONG, Y. 2003, «One machine fuzzy scheduling to minimize total weighted tardiness, earliness, and recourse cost», *International Journal of Smart Engineering System Design*, vol. 5, nº 3, p. 135–147. 75

DUTA, L. 2006, *Contribution à l'étude de la conduite des systemes de desassemblage*, thèse de doctorat, Université de Franche-Comte. 84

EREN, T. 2009, «Minimizing the total weighted completion time on a single machine scheduling with release dates and a learning effect», *Applied Mathematics and Computation*, vol. 208, nº 2, p. 355–358. 74

FLEURY, G. 1993, *Méthodes stochastiques et deterministes pour les problemes NP-difficiles*, thèse de doctorat, Université de Clermont Ferrand 2. 84

GHRAYEB, O. A. 2003, «A bi-criteria optimization : minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems», *Applied Soft Computing*, vol. 2/3F, p. 197–210. 94

GIBBONS, J. D. et S. CHAKRABORTI. 2003, *Non parametric statistical inference*, Marcel Dekker Press. 100

GLOVER, F. 1989, «Tabu search-part i», *ORSA Journal on Computing*, vol. 1, nº 3, p. 190–206. 80

GLOVER, F. 1990, «Tabu search-part ii», *ORSA Journal on Computing*, vol. 2, nº 1, p. 4–32. 80

GLOVER, F. et G. A. KOCHENBERGER. 2003, *Handbook of metaheurstics*, Kluwer Press. 73

GRAHAM, R. L., E. L. LAWLER, J. K. LENSTRA et A. H. G. RINNOOY KAN. 1979, «Optimization and approximation in deterministic sequencing and scheduling : a survey», *Annals of Discrete Mathematics*, vol. 5, p. 287–326. 86

HANSS, M. 2005, *Applied fuzzy arithmetic an introduction with engineering applications*, Springer Press. 77

JAIN, R. 1976, «Decision-making in the presence of fuzzy variables», *IEEE Transactions on Systems Man and Cybernetics*, vol. 6, nº 10, p. 698–703. 79

JANIAK, A. et R. RUDEK. 2010, «A note on a makespan minimization problem with a multi-ability learning effect», *Omega The International Journal of Management Science*, vol. 38, nº 3-4, p. 213–217. 74

KASPERSKI, A. et P. ZIEŁIŃSKI. 2011, «Possibilistic minmax regret sequencing problems with fuzzy parameters», *IEEE Transactions on Fuzzy Systems*, vol. 19, nº 6, p. 1072–1082. 76

KIRKPATRICK, S., C. D. GELATT et M. P. VECCHI. 1983, «Optimization by simulated annealing», *Science*, vol. 220, nº 4598, p. 671–680. 82

KLIR, G. J. et B. YUAN. 1995, *Fuzzy sets and fuzzy logic theory and applications*, Prentice Hall Press. 77

KOÇ, E. 2010, *The bees algorithm theory, improvements and applications*, thèse de doctorat, Université de Wales. 86

KUO, W. H. et D. L. YANG. 2006, «Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect», *European Journal of Operational Research*, vol. 174, nº 7, p. 1184–1190. 74

LAI, P. J. et W. C. LEE. 2013, «Single-machine scheduling with learning and forgetting effects», *Applied Mathematical Modelling*, vol. 37, nº 6, p. 4509–4516. 74

LEE, C. Y., L. LEI et M. PINEDO. 1997, «Current trends in deterministic scheduling», *Annals of Operations Research*, vol. 70, nº 0, p. 1–41. 80

LEHMANN, E. L. et J. P. ROMANO. 2005, *Testing statistical hypotheses*, Springer Press. 99

LENSTRA, J. K., A. H. G. RINNOOY KAN et P. BRUCKER. 1977, «Complexity of machine scheduling problems», *Annals of Discrete Mathematics*, vol. 1, p. 343–362. 93

LI, J., K. SUN, D. XU et H. LI. 2010, «Single machine due date assignment scheduling problem with customer service level in fuzzy environment», *Applied Soft Computing*, vol. 10, nº 3, p. 894–858. 76

LIAO, L. M. et C. J. LIAO. 1998, «Single machine scheduling problem with fuzzy due date and processing time», *Journal of the Chinese Institute of Engineers*, vol. 21, nº 2, p. 189–196. 75

MATARAZZO, B. et G. MUNDA. 2001, «New approaches for the comparison of l-r fuzzy numbers : a theoretical and operational analysis», *Fuzzy Sets and Systems*, vol. 118, nº 3, p. 407–418. 78

MEILIJSON, I. et A. TAMIR. 1984, «Minimizing flow time on parallel identical processors with variable unit processing time», *Operations Research*, vol. 32, nº 2, p. 440–448. 73

MOSHEIOV, G. 2001, «Scheduling problems with a learning effect», *European Journal of Operational Research*, vol. 132, nº 3, p. 687–693. 74

NOURANI, Y. et B. ANDRESEN. 1989, «A comparison of simulated annealing cooling strategies», *Journal of Physics A : Mathematical and General*, vol. 31, nº 41, p. 8373–8385. 84

ÖZELKAN, E. C. et L. DUCKSTEIN. 1999, «Optimal fuzzy counterparts of scheduling rules», *European Journal of Operational Research*, vol. 113, nº 3, p. 593–609. 79

PINEDO, M. L. 2012, *Scheduling theory, algorithms, and systems*, Springer Press. 73

PRADE, H. 1979, «Using fuzzy set theory in a scheduling problem : a case study», *Fuzzy Sets and Systems*, vol. 2, nº 2, p. 153–165. 75

ROSS, T. J. 2004, *Fuzzy logic with engineering applications*, Wiley Press. 77

RUDEK, R. 2013, «A note on proving the strong np-hardness of a scheduling problem with position dependent job processing times», *Optimization Letters*, vol. 7, nº 3, p. 613–616. 74

SAIDI MEHRABAD, M. et A. PAHLAVANI. 2009, «A fuzzy multi-objective programming for scheduling of weighted jobs on a single machine», *The International Journal of Advanced Manufacturing Technology*, vol. 45, nº 1-2, p. 122–139. 94

SALHI, S. 2002, «Defining tabu list size and aspiration criterion within tabu search methods», *Computers and Operations Research*, vol. 29, nº 1, p. 67–86. 82

SCHULTMANN, F., M. FRÖHLING et O. RENTZ. 2006, «Fuzzy approach for production planning and detailed scheduling in paints manufacturing», *International Journal of Production Research*, vol. 44, nº 8, p. 1589–1612. 75

SIVANANDAM, S. N., S. SUMATHI et S. N. DEEPA. 2007, *Introduction to fuzzy logic using MATLAB*, Springer Press. 77

SMITH, W. E. 1956, «Various optimizers for single stage production», *Naval Research Logistics*, vol. 3, nº 1-2, p. 59–66. 87

VAN LAARHOVEN, P. J. M. et E. H. L. AARTS. 1987, *Simulated annealing : theory and applications*, Springer Press. 83

WANG, C., D. WANG, W. H. IP et D. W. YUEN. 2002, «The single machine ready time scheduling problem with fuzzy processing times», *Fuzzy Sets and Systems*, vol. 127, nº 2, p. 117–129. 75

WONGA, B. K. et V. S. LAI. 2011, «A survey of the application of fuzzy set theory in production and operations management : 1998-2009», *International Journal of Production Economics*, vol. 129, nº 1, p. 157–168. 95

WOODS, J. R., R. M. SAYWELL et A. W. NYHUIS. 1992, «The learning curve and the cost of heart transplantation», *Health Services Research*, vol. 27, nº 2, p. 219–238. 73

WRIGHT, T. P. 1936, «Factors affecting the cost of airplanes», *Journal of the Aeronautical Sciences*, vol. 3, nº 4, p. 122–128. 73

WU, C. C., P. H. HSU, J. C. CHEN et N.-S. WANG. 2011, «Genetic algorithm for minimizing the total weighted completion time scheduling problem with learning and release times», *Computers and Operations Research*, vol. 38, nº 7, p. 1025–1034. 75

WU, C. C. et W. C. LEE. 2005, «A single-machine group schedule with fuzzy setup and processing times», *Journal of Information and Optimization Sciences*, vol. 26, nº 3, p. 683–691. 76

WU, C. C. et W. C. LEE. 2008, «Single-machine scheduling problems with a learning effect», *Applied Mathematical Modelling*, vol. 32, nº 7, p. 1191–1197. 74

WU, H. C. 2010, «Solving the fuzzy earliness and tardiness in scheduling problems by using genetic algorithms», *Expert Systems with Applications*, vol. 37, nº 7, p. 4860–4866. 76

YAGER, R. R. 1981, «A procedure for ordering fuzzy subsets of the unit interval», *Information Sciences*, vol. 24, nº 2, p. 143–161. 80

YELLE, L. E. 1979, «The learning curves : historical review and comprehensive survey», *Decision Sciences*, vol. 10, nº 2, p. 302–328. 73

YIN, Y., M. LIU et J. HAO. 2012, «Single-machine scheduling with job-position-dependent learning and time-dependent deterioration», *IEEE Transactions on Systems, Man, and Cybernetics - Part A : Systems and Humans*, vol. 42, nº 1, p. 192–200. 74

YIN, Y., W. H. WU et W. H. WU. 2014, «A branch-and-bound algorithm for a single machine sequencing to minimize the total tardiness with arbitrary release dates and position-dependent learning effects», *Information Sciences*, vol. 256, p. 91–108. 75, 94

ZADEH, L. A. 1971, «Similarity relations and fuzzy orderings», *Information Sciences*, vol. 3, nº 2, p. 177–200. 78

ZHAO, C. L., Q. L. ZHANG et H. Y. TANG. 2004, «Machine scheduling problems with learning effects», *Dynamics of Continuous, Discrete and Impulsive Systems, Series A : Mathematical Analysis*, vol. 11, nº 5-6, p. 741–750. 87

Chapitre 4

Problème d'ordonnancement bi-objectif à une machine avec effet d'apprentissage et paramètres incertains

« Il n'est pas certain que tout soit incertain »

Blaise Pascal

Sommaire

4.1 Introduction et état de l'art

Dans le but d'avoir un taux de production élevé, les stratégies d'ordonnancement ont été le sujet d'une analyse condensée comme elles constituent la liaison entre les stocks de la matière première, les outils de fabrication et la clientèle ciblée [PINEDO, 2009]. En effet, les problèmes d'ordonnancement ont déclenché pendant ces dernières années un nombre colossal de travaux de recherche à cause de leur adaptabilité au traitement mathématique. Les efforts initiaux dédiés à la résolution des problèmes d'ordonnancement faciles ont élargi la vision sur des problèmes plus sophistiqués avec des critères et contraintes plus sévères [GUPTA et KYPARISIS, 1987]. Le problème d'ordonnancement à une machine a joué un rôle clé sur le niveau théorique que pratique pour les raisons suivantes : abondance de cas réels avec une seule ressource, possibilité d'agrégation des machines multiples en une seule machine et facilitation de la compréhension des problèmes plus compliqués à machines multiples [AL-TURKI et collab., 2001].

Traditionnellement, les problèmes d'ordonnancement à une machine ont été introduits en pratique avec l'hypothèse de satisfaire un objectif unique. Mais sous la pression des contraintes des marchés de nos jours, il est primordial de prendre des décisions à multiples facettes, ce qui a stimulé les gestionnaires à utiliser plus qu'une seule mesure de performance dans l'évaluation des ordonnancements dans un contexte plus vaste connu sous le nom de problèmes d'ordonnancement multi-objectifs. Dans ce cas, il est très rare qu'une configuration optimise à la fois tous les objectifs et pour cela, un compromis entre les critères de performance devient inévitable. Comme résultat, plusieurs solutions optimales sont offertes aux gestionnaires pour un choix flexible dépendant de la situation économique rencontrée [HOOGEVEEN, 2005]. Dans le modèle standard d'ordonnancement à une machine, la classification la plus répondue est celle basée sur la considération des dates échues dans la formulation de la fonction objectif. Dans la première classe, les objectifs sont fonction des durées opératoires des tâches. La minimisation de ces critères conduit à une meilleure utilisation des ressources de système. Par exemple, la minimisation de la somme pondérée des dates de fin est un problème classique résolu à l'optimalité en classant les tâches selon la règle WSPT. Le poids d'une tâche peut exprimer un coût de maintien par unité de temps dans le stock. La deuxième classe des objectifs dépendant des dates échues des tâches est perçue comme une mesure des coûts associés aux services fournis aux clients. Par conséquent, en considérant la minimisation conjointe des critères appartenant à des classes différentes, il sera plus raisonnable d'établir des outils d'ordonnancement fiables.

Avec l'accroissement des marchés actuels aux niveaux national et international, la majorité des systèmes de production sont sujets à des hautes pressions pour réduire le délai de mise en oeuvre et pour respecter les délais spécifiés par les clients [LI et collab., 2011]. Les transactions de ventes sont influées par la capacité du producteur à livrer la marchandise au client à une date bien définie. Cette vision étroitement liée au paradigme Juste À Temps (en anglais JIT) motive les gestionnaires à se concentrer sur des nouvelles approches d'ordonnancement plus efficaces en termes de la manipulation des coûts relatifs à l'indemnisation des clients suite à la violation des dates échues. Une étude sur les pratiques de fabrication a démontré que la satisfaction des dates échues en gardant un niveau faible de stock est l'objectif le plus important ciblé par les entreprises [SINGH et AHUJA, 2012]. Parmi les critères basés sur la date échue nous trouvons la somme pondérée du nombre de tâches en retard qui est considérée une mesure flexible comme elle est utilisée pour différencier les clients. La satisfaction des tâches en retard engendre des coûts supplémentaires au niveau de l'entreprise à cause du payement des frais élevés pour les

employées hors les horaires de travail normaux. De plus, la productivité peut être réduite en raison de la surexploitation des ouvriers pendant leurs jours de repos. Il a été démontré par Karp que la minimisation de cette mesure est NP-difficile au sens ordinaire [KARP, 1972]. Donc, ce problème est pseudo polynômial ce qui permet sa résolution par une méthode de programmation dynamique en un temps $O\left(n\sum p_j\right)$. Par contre, la version non pondérée du problème peut être résolue à l'optimalité en $O\left(n \times \log(n)\right)$ utilisant le fameux algorithme de Moore-Hodgson [LAWLER et MOORE, 1969]. Malgré la simplicité apparente de ce problème d'ordonnancement à une machine, il a été traité sous plusieurs situations. Par exemple, Potts et Wassenhove ont relaxé la contrainte d'intégrité à une formulation avec variables binaires et ont résolu le problème de programmation linéaire résultant par un algorithme de complexité $O\left(n \times \log(n)\right)$. La borne inférieure calculée est adoptée dans une procédure de réduction pour élaborer une approche par séparation et évaluation efficace même lorsque appliquée aux grandes instances [POTTS et VAN WASSENHOVE, 1988]. Dauzère-Pérès a étudié le cas général où les dates de disponibilité et différentes dates échues sont incluses. Une borne inférieure efficace basée sur la relaxation de la formulation du programme linéaire mixte en nombres entiers du problème a été proposée. L'auteur a proposé également une nouvelle heuristique et a comparé ses performances avec la procédure de la borne inférieure où l'heuristique a indiqué des meilleures performances en termes de qualité et temps de calcul [DAUZÈRE-PÉRÈS, 1995]. Dans [BRUCKER et KOVALYOV, 1996], la somme pondérée du nombre de retards a été étudié par Brucker et Kovalyov sous l'hypothèse que la totalité des tâches est partitionnée en lots. Les auteurs ont présenté un algorithme de programmation dynamique résolvant le problème avec poids égaux en temps $O\left(n^3\right)$. L'application de cet algorithme au problème avec poids échelonnés donne un schéma d'approximation polynômial pour le problème original. Dauzère-Pérès et Sevaux ont introduit la notion de la séquence maître pour dériver une formulation originale de la programmation linéaire mixte en nombres entiers. Ils ont donné un algorithme de relaxation lagrangienne permettant d'obtenir les bornes supérieures et inférieures où les expériences numériques ont montré que l'approche proposée est capable de résoudre le problème avec une taille de 100 tâches [DAUZÈRE-PÉRÈS et SEVAUX, 2003]. En outre, M'Hallah et Bulfine ont proposé une heuristique ainsi qu'un algorithme exact basé sur la procédure de séparation et évaluation avec les bornes obtenues par une représentation à base du problème de sac à dos. Les expérimentations réalisées ont indiqué que toutes les deux approches résultent en des performances très compétitives avec des instances allant jusqu'à 2500 tâches dans un temps réduit [M'HALLAH et BULFIN, 2003]. Sevaux et Dauzère-Pérès ont conçu un algorithme génétique où trois stratégies différentes pour l'évaluation de la fonction objectif des individus, quatre types d'opérateurs de croisement et trois types d'opérateurs de mutation ont été employés pour obtenir la meilleure configuration de l'algorithme. L'algorithme génétique obtenu est amélioré par application d'une recherche locale et initialisation de la population avec des solutions de bonne qualité [SEVAUX et DAUZÈRE-PÉRÈS, 2003]. Cheng et ses collègues ont étudié un modèle de faisabilité d'ordonnancement multi-agent où la fonction objectif de chaque agent est de minimiser la somme pondérée du nombre des tâches en retards. Ils ont montré que lorsque le nombre d'agents est fixé, le problème peut être résolu dans un temps pseudo polynômial pour des poids entiers et dans un temps polynômial pour des poids unitaires [CHENG et collab., 2006]. Le travail de M'Hallah et Bulfine a montré que l'heuristique proposée est bonne et très rapide pour les problèmes non pondérés. La méthode exacte proposée résout rapidement tous les problèmes non pondérés non résolus au préalable. Elle permet également de résoudre les grands problèmes exploités à nos jours. Leurs résultats ont prouvé que l'algorithme est

robuste en termes du type de problème ainsi que les niveaux de ses paramètres [M'HALLAH et BULFIN, 2007]. Shabtay et Steiner ont étudié des modèles d'ordonnancement bi-critères avec des durées opératoires contrôlables convexes, coûts de consommation de ressources et des dates échues affectées par les méthodes CON et SLK. Pour chaque méthode d'affectation des dates échues, les auteurs ont fourni une analyse bi-critère. Le premier critère est de minimiser la somme pondérée du nombre de retards plus les coûts d'affectation des dates échues et le deuxième critère est de minimiser la somme pondérée des consommations des ressources [SHABTAY et STEINER, 2011]. Dans le contexte de la gestion des chaînes logistiques, Rasti-Barzoki et Hejazi ont étudié un modèle d'ordonnancement qui considère à la fois l'affectation des dates échues, ordonnancement de la production et livraison du produit. Le problème de minimiser la somme des coûts de livraison en lots dans une chaîne d'approvisionnement est modélisé comme un programme en nombre entier, une heuristique et une approche par séparation et évaluation ont été présentées pour la résolution où les performances sont sensibles à quelques paramètres du problème, sa structure et sa taille [RASTI-BARZOKI et HEJAZI, 2013]. De plus, Dauzère-Pérès et Mönch ont considéré une machine unique opérant en lots avec des familles de tâches incompatibles où deux formulations différentes à base de la programmation linéaire mixte en nombres entiers ont été développées. La première formulation a comme inconvénient la nécessité d'introduire un coefficient M ayant une grande valeur (méthode du grand M). Les expérimentations ont montré que les deux formulations sont capables de trouver des solutions proches de l'optimale en un temps raisonnable pour les instances de petites et moyennes tailles mais le gap peut être aussi grand dans le cas des instances de grande taille. La deuxième formulation est capable de trouver des meilleures bornes inférieures en comparaison avec la première approche tandis que cette dernière donne des meilleures bornes supérieures [DAUZÈRE-PÉRÈS et MÖNCH, 2013]. Malgré qu'un grand nombre de travaux cumulé concernant les problèmes d'ordonnancement avec différentes contraintes est disponible, le nombre d'articles réservé à l'état de l'art de minimisation de la somme pondérée du nombre de retards dans le cas des paramètres réels est limité. Les lecteurs intéressés peuvent consulter ([MUNDT et WICH, 2007] et [ADAMU et ADEWUMI, 2014]). Une analyse vigilante de la structure interne des systèmes de production permet de distinguer deux ressources de variabilité dans les données d'ordonnancement : variation déterministe et variation stochastique. La variation déterministe est produite en générale à cause d'un effet ayant un comportement monotone telle que le phénomène de vieillissement ou d'apprentissage. L'effet d'apprentissage a été présenté pour la première fois par Wright qui a démontré que le coût unitaire dans l'industrie d'avions décroît lorsque les firmes produisent plus d'unités ou acquirent d'avantage d'expérience [WRIGHT, 1936]. Il est à mentionner qu'une littérature riche sur les problèmes d'ordonnancement polynômiaux à une machine avec effet d'apprentissage est disponible ([YIN et collab., 2012], [WU et LEE, 2008], [JANIAK et RUDEK, 2010], [LAI et LEE, 2011] et [WANG et collab., 2012]). En général, les papiers traitant les approches polynômiales de ces problèmes partagent un aspect commun dans le sens de proposer une nouvelle formule d'apprentissage et par la suite déterminer les conditions sous lesquelles les règles d'optimalité conventionnelles restent valables. Le travail récent de Rudek a été consacré au problème d'ordonnancement à une machine avec effet d'apprentissage basé sur la somme des durées opératoires. L'auteur a prouvé que ce problème est fortement NP-difficile en utilisant le problème de Partition et il a implémenté un algorithme par séparation et évaluation pour traiter de manière optimale des instances modérées ayant jusqu'à 25 tâches [RUDEK, 2017]. Quelques études relatives à la classification et la résolution des problèmes d'ordonnancement avec durées opératoires variables sont présentées

dans ([ALIDAEE et WOMER, 1999] et [BISKUP, 2008]).

Pour les variations stochastiques, quelques travaux ont été dédiés aux difficultés liées à l'application des problèmes d'optimisation. Le débat est focalisé autour de quatre catégories d'incertitudes :

- Les valeurs de la fonction objectif sont affectées par le bruit ;
- Les variables de conception /environnement peuvent être changées après le processus d'optimisation ;
- L'approximation de la fonction objectif peut amener à des erreurs d'approximation significatives ;
- L'optimum a un comportement dynamique ce qui nécessite un suivi continu en fonction du temps.

Sous ces contraintes, des mesures supplémentaires doivent être allouées aux algorithmes évolutionnaires dans le but de conserver un niveau satisfaisant de performance [JIN et BRANKE, 2005]. D'autres alternatives sont utilisées pour capturer la variabilité des données au lieu de l'utilisation des modèles déterministes. Les distributions de probabilité ont été largement appliquées pour la prise en compte des incertitudes internes relatives aux systèmes de production. Cependant, dans le cas d'absence d'enregistrement de données dans le passée (historique de données) ou manque de données disponibles, les gestionnaires échouent dans la dérivation d'une distribution de probabilité exacte [CAI et collab., 2014]. Alors, la considération d'aspect flou dans les modèles d'ordonnancement devient inévitable pour la décision. Il est remarquable que malgré plusieurs cas pratiques ont été manipulés considérant l'effet d'apprentissage et formulations floues séparément, moins d'efforts ont été alloués au traitement des problèmes de minimisation de la somme pondérée du nombre de retards lorsque ces deux sources d'imprécision sont présentes. Qian et Steiner ont introduit les effets d'apprentissage/vieillissement et durées opératoires dépendantes du temps dans les problèmes d'ordonnancement à une machine avec la considération d'affectation des dates échues. Ils ont montré par la réduction du problème à un problème d'affectation la possibilité de résolution optimale en $O(n^3)$ [QIAN et STEINER, 2013]. Dans [KASPERSKI, 2005], le degré de retard d'une tâche est remplacé par la possibilité de l'événement que la date fin d'exécution de cette tâche dépasse la date échue données toutes les deux par des nombres flous. Par la réduction polynômiale des problèmes d'ordonnancement flous résultant, l'auteur a montré que la somme pondérée des possibilités de retard est NP-difficile. Il est à noter qu'aucune expérimentation numérique n'a été conduite dans ce travail et tous les développements théoriques ont été élaborés dans le contexte d'un objectif unique. Kasperski et Zieliński ont étendu l'approche de regret minmax au cas flou. Ils ont supposé que chaque tâche a une durée opératoire unitaire et date échue réelle, tandis que les poids des tâches sont des paramètres incertains. Il a été démontré que ce problème flou peut être résolu en $O(n \min\{d, n-d\} \log \varepsilon^{-1})$ par une recherche binaire dans le cas des dates échues égales. Autrement, un modèle de programmation en nombre entier est fourni. Le nombre de tâches juste à temps dans une séquence optimale peut être déterminé en $O(n^2)$ par un algorithme glouton [KASPERSKI et ZIEŁIŃSKI, 2011]. Bentrcia et ses collègues ont traité le problème d'ordonnancement à une machine avec la somme pondérée actualisée des dates de fin d'exécution où ils ont supposée que tous les paramètres sont décrits par des nombres flous trapézoïdaux. Des algorithmes génétique et recherche par motifs ont été proposés pour l'évaluation de degré d'optimalité d'un ordonnancement fixe. Le cas limite lorsque le facteur d'actualisation est réduit à des valeurs très faibles a été validé numériquement [BENTRCIA et collab., 2015]. Il est essentiel de souligner que la majorité des travaux cité ci-dessus sont basés

sur une vision unique d'incertitude dans le sens que l'incertitude des paramètres et l'effet d'apprentissage sont manipulés séparément. Malgré que nous sommes au courant de l'existence de quelques études intégrant ces deux aspects dans le même contexte de modélisation, ces contributions sont établies dans un cadre mono objectif, ce qui limite leur application en pratique. Par exemple, Ahmadizar et Hosseini ont proposé un problème d'ordonnancement à une machine avec la considération des durées opératoires floues et effet d'apprentissage à base de position. Les valeurs affectées aux durées opératoires sont supposées d'allure triangulaire. Comme l'objectif est de minimiser la somme des dates de fin, la procédure de solution est basée sur l'extension de la règle SPT pour supporter les durées opératoires incertaines [AHMADIZAR et HOSSEINI, 2011]. Les mêmes auteurs ont étudié le problème d'ordonnancement à une machine avec effet d'apprentissage à base de position et durées opératoires floues où l'objectif est de minimiser le makespan. Deux algorithmes polynômiaux ont été développés pour ce problème en se basant sur la programmation par contraintes stochastiques floues et une méthode de classement de nombres flous. Les simulations numériques résultent en des performances équivalentes [AHMADIZAR et HOSSEINI, 2013]. Dans [BENTRCIA et MOUSS, 2015], les auteurs ont exploité l'effet d'apprentissage et les paramètres flous avec l'objectif de minimiser la somme pondérée des dates de fin d'exécution des tâches. Dans ce contexte, ils ont généralisé la règle de Smith au cas flou en redéfinissant le concept d'agréabilité ordinaire. En outre, trois métaheuristiques à base de trajectoire ont été implémentées et appliquées pour résoudre le cas avec dates de disponibilité floues différentes. Les tests statistiques non paramétriques ont été adoptés dans la comparaison des performances des métaheuristiques proposées. Aydilek et al ont considéré un système de fabrication à machine unique dans le but de minimiser le nombre de tâches en retard pour lesquelles les durées opératoires sont données dans certains intervalles pour exprimer l'incertitude. Sur la base d'une relation de dominance globale, plusieurs alternatives de résolution ont été évaluées et les expérimentations numériques ont révélé que l'algorithme Updated Sequence (US) est le plus approprié par rapport aux autres versions [AYDILEK et collab., 2017].

Dans ce chapitre, nous conduisons une analyse comparative basée sur les performances de deux métaheuristiques : l'algorithme génétique élitiste de tri non dominé et la recherche coucou combinée avec une procédure de recherche locale. Le problème d'ordonnancement est formulé comme un problème d'optimisation bi-objectif où les critères de performances sont notés par la magnitude de la somme pondérée des dates de fin flous et la somme pondérée des possibilités des retards des tâches. Même avec des données réelles, ce problème a une importance cruciale car le premier objectif exprime la configuration efficace des ressources internes du système de production et le deuxième objectif reflète l'état de satisfaction du client envers la qualité de service délivré. Dans ce contexte, nous considérons l'effet d'apprentissage ainsi que l'aspect flou au niveau des durées opératoires et dates échues dans le même modèle. À notre connaissance, ce problème est introduit pour la première fois. Dans le but de générer les instances, nous proposons une extension de la méthode conventionnelle pour le support des nombres flous. L'amélioration de la recherche coucou par des procédures de recherche locale et vol de Lévy a été utilisée pour garantir un compromis entre l'intensification et la diversification de la recherche. Le reste de ce chapitre est organisé comme suit. Dans la Section 4.2, quelques définitions formelles associées aux concepts de base des nombres flous et théorie de possibilité sont illustrées. La Section 4.3 est dédiée à la formulation du problème ainsi que sa complexité intrinsèque. Nous décrivons dans la Section 4.4 les deux approches à base de population utilisées dans ce chapitre. Nous présentons dans la Section 4.5 les expérimentations numériques pour évaluer les performances des stratégies proposées et nous com-

parons leur signification statistique. La Section 4.6 résume les principales contributions et énumère quelques recommandations potentielles pour des futures améliorations.

4.2 Présentation du cadre de modélisation floue

En se basant sur le fait que plusieurs situations réelles de la vie ne peuvent pas être décrites par des relations mathématiques précises à cause de l'inclusion d'un type d'informations qui ne sont pas nécessairement déterministes. La théorie de la logique floue a été proposée pour remédier à la limitation de la vision classique [ZADEH, 1965]. Il est essentiel de savoir que les variations stochastiques ne sont pas la seule source d'incertitude à prendre en compte. D'autres facteurs peuvent être considérés des générateurs d'incertitudes comme les procédures d'acquisition ou la perception humaine des données manipulées. Dans ce qui suit, quelques définitions de base et des propriétés caractérisant les nombres flous ainsi que les mesures de possibilité sont présentées et expliquées.

4.2.1 Nombres flous

Le concept de nombre flou a émergé comme un type particulier des ensembles flous, où il est considéré comme une généralisation des nombres réels. Cette représentation est utile pour assurer la modélisation adéquate des paramètres pendant l'analyse des problèmes de gestion et d'engineering [KAHRAMAN et YAVUZ, 2010]. Les nombres flous sont utilisés pour résoudre les problèmes pratiques de types différents à cause de la présence de subjectivité et imprécision étroitement liées au processus de décision humaine. Une définition formelle du nombre flou est donnée ci-dessous [DIAMOND et KLOEDEN, 1994].

Définition 4.1. *Un nombre flou est un ensemble flou* $\tilde{u} : \mathbb{R} \to [0, 1]$ *ayant les propriétés suivantes :*

i) $\tilde{u}$ *est normal, i.e.* $\exists x_0 \in \mathbb{R}$ *avec* $\tilde{u}(x_0) = 1$ *;*

ii) $\tilde{u}$ *est un ensemble flou convexe (i.e.* $\tilde{u}\left(\lambda x + (1-\lambda) y\right) \geq \min\left\{\tilde{u}(x), \tilde{u}(y)\right\} \ \forall x, y \in \mathbb{R}, \forall \lambda \in [0, 1]$*) ;*

iii) $\tilde{u}$ *est semi continu supérieur sur* $\mathbb{R}$ *;*

iv) $\overline{\{x_0 \in \mathbb{R} : \tilde{u}(x) > 0\}}$ *est un ensemble compact, avec* $\overline{\{\}}$ *est la fermeture de l'ensemble.*

À cause de la complexité de calcul élevée lorsque les problèmes manipulés sont de taille réelle en particulier avec des paramètres flous, des représentations alternatives augmentant l'efficacité mais sans affecter la capacité de généralisation en dehors des limites tolérables sont à considérer. Souvent, il est plus commode d'utiliser les nombres flous du type L-R pour la description des données floues [CHANAS, 2001].

Définition 4.2. *Un nombre flou* $\tilde{A}$ *est dit de type L-R si sa fonction d'appartenance a la forme :*

$$\mu_{\tilde{A}}(x) = \begin{cases} L\left(\frac{\underline{a}-x}{\alpha_{\tilde{A}}}\right) & pour \quad x \leq \underline{a} \\ 1 \quad pour \quad \underline{a} \leq x \leq \bar{a} \\ R\left(\frac{x-\bar{a}}{\beta_{\tilde{A}}}\right) & pour \quad x \geq \bar{a} \end{cases} \tag{4.1}$$

avec L *et* R *des fonctions continues décroissantes et satisfaisant à la condition* $L(0) = R(0) =$

1. *Les paramètres* $\alpha_{\tilde{A}}$ *et* $\beta_{\tilde{A}}$*, connus sous le nom de déviations gauche et droite de la fonction d'appartenance, sont des nombres réels non négatifs.*

Une forme spéciale de l'allure des fonctions L et R est donnée par la loi de puissance :

$$\begin{cases} L(z) = \max\{0, 1 - z^{p_1}\} \\ \\ R(z) = \max\{0, 1 - z^{p_2}\} \end{cases} \tag{4.2}$$

avec $p_1, p_2 \geq 1$.
Si $p_1 = p_2 = 1$, les fonctions deviennent linéaires et donnent les nombres flous trapézoïdaux connus également sous le nom des intervalles flous. Cette définition est très générale et permet la modélisation de plusieurs types d'information :

$$\tilde{A} = \begin{cases} (a, a, 0, 0)_{L-R} \ \forall L, \ \forall R \ pour\ un\ nombre\ réel\ a \\ \\ \left(\underline{a}, \bar{a}, 0, 0\right)_{L-R} \ \forall L, \ \forall R \ pour\ un\ intervalle\ réel\ \left[\underline{a}, \bar{a}\right] \end{cases}$$

Le principe d'extension introduit par Zadeh joue un rôle prépondérant dans la généralisation des applications définies sur $\mathbb{R}$ vers des applications définies sur des ensembles flous $\mathbb{F}$. Les opérations arithmétiques élémentaires dans le cas flou peuvent être déduites des opérations classiques opérant sur des nombres réels [TZAFESTAS et VENETSANOPOULOS, 1994].

Définition 4.3. *Soit* X *le produit cartésien des espaces* $X_1, X_2, ..., X_r$ *et* $A_1, A_2, ..., A_r$ *sont r ensemble flous de* $X_1, X_2, ..., X_r$ *respectivement. Soit* f *une application de* X *sur l'espace* Y*, i.e.* $y = f(x_1, ..., x_r)$ *et* $f^{-1}(y) = \{x \in X / f(x) = y\}$*. Alors, le principe d'extension permet d'induire des r ensembles flous* A_i*, un ensemble flou* B *de* Y *par* f *comme suit :*

$$B(y) = \left\{ \begin{array}{ll} \sup\limits_{y=f(x_1,x_2,..,x_r)} \min\{A_1(x_1), ..., A_r(x_r)\} & si \quad f^{-1}(y) \neq \phi \\ \\ 0 \quad si \quad f^{-1}(y) = \phi \end{array} \right\} \forall y \in Y \tag{4.3}$$

Il est possible de généraliser les opérations unaires et binaires sur les nombres flous en concentrant sur les cas particuliers $r = 1$ et $r = 2$. Soient $M, N \in \mathbb{F}(\mathbb{R})$ deux élements de l'espace des nombres flous $\mathbb{F}$ et $\circ : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ une opération binaire. Si nous notons l'opération généralisée par $\tilde{\circ}$, la relation $M\tilde{\circ}N$ represente un ensemble flou de $\mathbb{R}$ avec la fonction d'appartenance :

$$(M\tilde{\circ}N)(z) = \left\{ \begin{array}{ll} \sup\limits_{z=x\circ y} \min\{M(x), N(y)\} & si \quad (\circ)^{-1}(z) \neq \phi \\ \\ 0 \quad si \quad (\circ)^{-1}(z) = \phi \end{array} \right\} \forall z \in \mathbb{R} \tag{4.4}$$

En appliquant cette formule, nous obtenons les expressions floues équivalentes à la somme et la multiplication par un scalaire positif λ pour tous nombres flous de type L-R $\tilde{A}$ et $\tilde{B}$:

$$\tilde{A} \oplus \tilde{B} = \left(\underline{a}, \bar{a}, \alpha_A, \beta_A\right)_{L-R} \oplus \left(\underline{b}, \bar{b}, \alpha_B, \beta_B\right)_{L-R} = \left(\underline{a} + \underline{b}, \bar{a} + \bar{b}, \alpha_A + \alpha_B, \beta_A + \beta_B\right)_{L-R}$$

$$\lambda \otimes \tilde{A} = \lambda \otimes \left(\underline{a}, \bar{a}, \alpha_A, \beta_A\right)_{L-R} = \left(\lambda \times \underline{a}, \lambda \times \bar{a}, \lambda \times \alpha_A, \lambda \times \beta_A\right)_{L-R}$$

Au contraire à la structure de l'ensemble des réels $\mathbb{R}$ pour lequel un ordre total existe, il est très difficile de comparer des nombres flous représentés par des distributions de possibilité car ils peuvent se chevaucher. Cependant, une alternative efficace consiste en l'utilisation des fonctions de classement pour établir un ordre comme proposée dans [JAIN, 1976]. Cette approche attache à chaque nombre flou un point sur la ligne réelle $\mathbb{R}$, ce qui rend la comparaison possible. Deux relations de dominance sont élaborées dépendamment de la localisation relative des nombres flous. La dominance forte caractérise des nombres flous complètement distincts tandis que la dominance faible caractérise des nombres flous chevauchés en utilisant les fonctions de classement. Une définition formelle de la dominance faible est présenté ci-dessous [ÖZELKAN et DUCKSTEIN, 1999].

Définition 4.4. *Soient* $E : \mathbb{F} \to \mathbb{R}$ *une fonction de classement,* $\tilde{A}$ *et* $\tilde{B}$ *deux nombres flous. On dit que* $\tilde{A}$ *domine faiblement* $\tilde{B}$ *si* $E\left(\tilde{A}\right) \geq E\left(\tilde{B}\right)$*, la dominance faible est notée par* $\tilde{A} \geq_w \tilde{B}$*. Dans ce cas, on a* $\max_w \left\{\tilde{A}, \tilde{B}\right\} = \tilde{A}$.

En réponse aux limites des fonctions de classement existantes qui sont dans certains cas en contradiction avec l'intuition humaine ou bien ne sont pas discriminantes, Abasabandy et Hajjari ont proposé une nouvelle approche pour le classement des nombres flous trapézoïdaux en se basant sur les déviations gauche et droite à une certaine coupe γ du nombre flou trapézoïdal [ABBASBANDY et HAJJARI, 2009]. Le calcul de cette méthode est détaillé dans la définition suivante.

Définition 4.5. *La magnitude d'un nombre flou arbitraire* $\tilde{A} = \left(\underline{a}, \bar{a}, \alpha_{\tilde{A}}, \beta_{\tilde{A}}\right)_{L-R}$ *avec forme paramétrique* $\left[\underline{A}(\gamma), \bar{A}(\gamma)\right]$ *est donnée par* $Mag\left(\tilde{A}\right) = \frac{1}{2}\int_{\gamma=0}^{\gamma=1}\left(\underline{A}(\gamma) + \bar{A}(\gamma) + \underline{a} + \bar{a}\right) f(\gamma)\, d\gamma$ *où la fonction* f *est une fonction décroissante sur* $[0,1]$ *avec* $f(0) = 0, f(1) = 1$ *et* $\int_{\gamma=0}^{\gamma=1} f(\gamma)\, d\gamma = \frac{1}{2}$.

Il est à noter que si f est égale à 1 (au sens des fonctions) et seulement les bornes des coupes γ sont considérées, alors l'expression de la magnitude se réduit à la formule proposée initialement par Yager [YAGER, 1981].

Comme la capacité des nombres flous plats (intervalles flous) ayant des fonctions d'appartenance non linéaires à représenter les variables linguistiques est plus grande [BOJADZIEV et BOJADZIEV, 1996], le Corollaire suivant donne l'expression explicite de la magnitude d'un nombre flou plat avec $p_1 = p_2 = 2$ pour les fonctions L et R.

Corollaire 4.1. *Soit* $\tilde{A} = \left(\underline{a}, \bar{a}, \alpha_{\tilde{A}}, \beta_{\tilde{A}}\right)_{L-R}$ *un nombre flou plat (intervall flou) avec forme paramétrique* $\left[\underline{A}(\gamma), \bar{A}(\gamma)\right] = \left[\underline{a} - \alpha_{\tilde{A}}\sqrt{1-\gamma}, \bar{a} + \beta_{\tilde{A}}\sqrt{1-\gamma}\right]$ *i.e. allure parabolique. Donc, la magnitude de* $\tilde{A}$ *est définie par* $Mag\left(\tilde{A}\right) = \frac{1}{2}\left(\underline{a} + \bar{a}\right) - \frac{2}{15}\left(\alpha_{\tilde{A}} - \beta_{\tilde{A}}\right)$.

Démonstration.
En substituant les bornes de la forme paramétrique dans l'expression générale de la Définition 4.5, nous obtenons :

$$Mag\left(\tilde{A}\right) = \frac{1}{2}\int_{\gamma=0}^{\gamma=1}\left(\underline{a} - \alpha_{\tilde{A}}\sqrt{1-\gamma} + \bar{a} + \beta_{\tilde{A}}\sqrt{1-\gamma} + \underline{a} + \bar{a}\right) f(\gamma)\, d\gamma$$

Si nous considérons la forme simple de f ($f(\gamma) = \gamma$) nous pouvons écrire :

$$Mag\left(\tilde{A}\right) = \frac{1}{2}\int_{\gamma=0}^{\gamma=1}\left(2\underline{a}+2\bar{a}-\left(\alpha_{\tilde{A}}-\beta_{\tilde{A}}\right)\sqrt{1-\gamma}\right)\gamma d\gamma$$

$$Mag\left(\tilde{A}\right) = \frac{1}{2}\int_{\gamma=0}^{\gamma=1}\left(2\underline{a}+2\bar{a}\right)\gamma d\gamma - \frac{1}{2}\left(\alpha_{\tilde{A}}-\beta_{\tilde{A}}\right)\int_{\gamma=0}^{\gamma=1}\left(\gamma\sqrt{1-\gamma}\right)d\gamma$$

$$Mag\left(\tilde{A}\right) = \frac{1}{2}\int_{\gamma=0}^{\gamma=1}\left(2\underline{a}+2\bar{a}\right)\gamma d\gamma - \frac{1}{2}\left(\alpha_{\tilde{A}}-\beta_{\tilde{A}}\right)\left(-\frac{2(3\gamma+2)(1-\gamma)^{\frac{3}{2}}}{15}\right)\Bigg|_{\gamma=0}^{\gamma=1}$$

Après quelques manipulations mathématiques, nous obtenons :

$$Mag\left(\tilde{A}\right) = \frac{1}{2}\left(\underline{a}+\bar{a}\right) - \frac{2}{15}\left(\alpha_{\tilde{A}}-\beta_{\tilde{A}}\right)$$

□

4.2.2 Mesures de possibilité

La construction de la théorie de possibilité a été initiée par les travaux du professeur Lotfi Zadeh ([ZADEH, 1968] et [ZADEH, 1978]), où il a démontré que la théorie des ensembles flous peut jouer un rôle similaire à la théorie de mesure dans l'élaboration des fondations des probabilités. Avec le développement accéléré de la théorie des ensembles flous et en profitant de la structure topologique riche des nombres flous, les mesures de possibilité sont devenues un concurrent compétitif aux notions classiques de probabilité. Les mesures floues sont nécessaires pour l'introduction de plusieurs concepts d'analyse mathématique tels que l'ordre ou la convergence des entités floues. Donc, une conceptualisation bien définie doit être donnée sur la base des ensembles flous [KLIR et YUAN, 1995].

Définition 4.6. *Soient* X *un ensemble universel et* ξ *une famille des sous ensembles de X non vide, la mesure floue sur* $\langle X,\xi\rangle$ *est une fonction* $g:\xi\rightarrow[0,1]$ *qui satisfait les conditions suivantes :*

i) $g\left(\phi\right)=0\wedge g\left(X\right)=1$;

ii) $\forall A,B\in\xi,(A\subseteq B)\Rightarrow\left(g\left(A\right)\leq g\left(B\right)\right)$;

iii) $\{\forall A_1,A_2,..\in\xi/A_1\subset A_2\subset ...\},\left(\bigcup_{i=1}^{\infty}A_i\in\xi\right)\Rightarrow\left(\lim_{i\rightarrow\infty}g\left(A_i\right)=g\left(\bigcup_{i=1}^{\infty}A_i\right)\right)$;

iv) $\{\forall A_1,A_2,..\in\xi/A_1\supset A_2\supset ...\},\left(\bigcap_{i=1}^{\infty}A_i\in\xi\right)\Rightarrow\left(\lim_{i\rightarrow\infty}g\left(A_i\right)=g\left(\bigcap_{i=1}^{\infty}A_i\right)\right)$.

Définition 4.7. *Soit* Pos *une mesure floue sur* $\langle X,\xi\rangle$. *Alors,* Pos *est dite une mesure de possibilité si et seulement si* $Pos\left(\bigcup_{k\in K}A_k\right)=\sup_{k\in K}Pos\left(A_k\right)$ *pour toute famille* $\{A_k/k\in K\}$ *dans* ξ *telle que* $\bigcup_{k\in K}A_k\in\xi$, *où* K *est un ensemble d'indices arbitraires.*

Théorème 4.1. *Toute mesure de possibilité* Pos *sur l'ensemble puissance fini* $P\left(X\right)$ *est uniquement detrminée par une fonction de distribution de possibilité* $r:X\rightarrow[0,1]$ *via la formule* $Pos\left(A\right)=\max_{x\in A}r\left(x\right)$ *pour tout* $A\in P\left(X\right)$.

La mesure de possibilité est directement liée aux ensembles flous à travers les fonctions de distribution de possibilité associées. L'événement "la variable Ξ prend la valeur ξ dans un ensemble universel Σ" est donnée par l'équation $\Xi=\xi$. Une contrainte flexible H est imposée sur Σ limitant ainsi les valeurs affectées à Ξ. L'assertion $H(\xi)$ peut être interprétée comme le degré de compatibilité de ξ avec la contrainte décrite par H. Mais

en incluant la proposition "Ξ est H", il sera plus approprié d'interpréter H(ξ) comme le degré de possibilité que $\Xi = \xi$. Cette interprétation de la possibilité d'un événement est en générale moins expressive en comparaison avec les probabilité mais utilisant moins d'informations [DUBOIS et collab., 2004].

Définition 4.8. *Soient* H *un ensemble flou sur* Σ *et la proposition "*Ξ *est* H*", la possibilité* $r_H(\xi)$ *de* $\Xi = \xi$ *est numériquement égale au degré H(ξ) pour lequel* ξ *appartient à* H *pour tout* $\xi \in \Sigma$. *La mesure de possibilité associée* Pos_H *est définie pour tout* $A \in P(\Sigma)$ *par l'équation :*

$$Pos_H(A) = \sup_{\xi \in A} r_H(\xi) \tag{4.5}$$

Comme une discrimination quantitative claire entre les nombres flous ne peut pas être achevée dans certains cas (tels que l'intersection des parties croissantes ou décroissantes des fonctions d'appartenance) utilisant uniquement les opérateurs généralisés $m\tilde{a}x$ et $m\tilde{i}n$, des approches plus efficaces peuvent être développées en se basant sur les mesures de possibilité comme indiqué par les définitions suivantes [DUBOIS et PRADE, 1988].

Définition 4.9. *Soient* P *et* Q *deux quantités floues. Pour déterminer si* P *est supérieur à* Q, *la comparaison des deux ensembles* P *et* $]Q, +\infty]$ *par la possibilité d'un évènement flou au sens de* μ_P, *de l'évènement flou* $]Q, +\infty]$ *est faite. Dans ce contexte, on obtient l'indice fondamental de comparaison suivant :*

$$\Pi_P(]Q, +\infty]) = \sup_u \min\left(\mu_P(u), \inf_{v \geq u}\left[1 - \mu_Q(v)\right]\right) = \sup_u \inf_{v \geq u} \min\left(\mu_P(u), \left[1 - \mu_Q(v)\right]\right) \tag{4.6}$$

Si X et Y sont les variables associées avec les domaines controlés par μ_P et μ_Q respectivement, alors cet indice peut être interprété comme la possibilité que les valeurs maximales que X peut prendre sont supérieures aux valeurs maximales que Y peut prendre $Pos(\bar{X} > \bar{Y})$. Le calcul explicite de la valeur de cet indice se réduit à trouver les cordonnées des points d'intersection des parties extrèmes droites des fonctions d'appartenances μ_P et μ_Q. Pour des nombres flous du type L-R donnés par $P = \left(\underline{p}, \bar{p}, \alpha_P, \beta_P\right)_{L-R}$ et $Q = \left(\underline{q}, \bar{q}, \alpha_Q, \beta_Q\right)_{L-R}$, la résolution de l'équation $f(z) = 1 - \left(\frac{z-\bar{p}}{\beta_P}\right) - \left(\frac{z-\bar{q}}{\beta_Q}\right) = 0$ donne $z^* = \frac{\beta_Q\beta_P + \beta_Q\bar{p} + \beta_P\bar{q}}{\beta_P + \beta_Q}$.

Corollaire 4.2. *Pour des nombres flous trapézoïdaux (*$L(z) = R(z) = \max(0, 1 - z)$*), en substituant la valeur de* z^* *dans* μ_Q, *on obtient l'expression suivante pour le degré de possibilité en fonction des valeurs des déviations et modales des nombres flous :*

$$Pos(\bar{X} > \bar{Y}) = \max\left(0, \min\left(1, \frac{\bar{p} - \bar{q} + \beta_P}{\beta_P + \beta_Q}\right)\right) \tag{4.7}$$

Sans grande difficulté, il est possible d'étendre la formule précédente de possibilité (obtenue pour le cas trapézoïdal) au cas où l'allure des fonctions d'appartenance est donnée par un polynôme de deuxième degré (allure parabolique).

Corollaire 4.3. *Pour des nombres flous plats (intervalles flous) d'allure parabolique (*$L(z) = R(z) = \max\left(0, 1 - z^2\right)$*), l'expression du degré de possibilité en fonction des paramètres de*

déviations gauche et droite ainsi que les paramètres modaux associés aux nombres flous est comme suit :

$$Pos\left(\bar{X} > \bar{Y}\right) = \begin{cases} 0 & si \quad \left(\left(\bar{p} + \beta_P\right) \leq \bar{q}\right) \\ 1 & si \quad \left(\bar{p} \geq \left(\bar{q} + \beta_Q\right)\right) \\ 1 - \left(\frac{z_1 - \bar{p}}{\beta_P}\right)^2 & si \quad \left(z_1 \in \left(\left[\bar{p}, \bar{p} + \beta_P\right] \cap \left[\bar{q}, \bar{q} + \beta_Q\right]\right)\right) \\ 1 - \left(\frac{z_2 - \bar{p}}{\beta_P}\right)^2 & si \quad \left(z_2 \in \left(\left[\bar{p}, \bar{p} + \beta_P\right] \cap \left[\bar{q}, \bar{q} + \beta_Q\right]\right)\right) \end{cases} \tag{4.8}$$

avec z_1 et z_2 représentent les solutions de l'équation du seconde ordre donnée par $g(z) = 1 - \left(\frac{z-\bar{p}}{\beta_P}\right)^2 - \left(\frac{z-\bar{q}}{\beta_Q}\right)^2 = 0$.

Démonstration.

Comme les deux parties gauche et droite des fonctions d'appartenance ont une allure parabolique, les cordonnées des points d'intersection des fonctions d'appartenance ne sont que les racines de l'équation du deuxième ordre :

$g(z) = 1 - \left(\frac{z-\bar{p}}{\beta_P}\right)^2 - \left(\frac{z-\bar{q}}{\beta_Q}\right)^2 = 0$. La résolution de cette équation utilisant la méthode classique du discriminant permet d'avoir les expressions explicites des racines z_1 et z_2 comme indiqué ci-dessous :

$$\begin{cases} z_1 = \left(\frac{\beta_Q^2\bar{p} + \beta_P^2\bar{q} - \beta_P\beta_Q\sqrt{\left\{\left(\beta_Q^2 + \beta_P^2\right) - \left(\bar{p} - \bar{q}\right)^2\right\}}}{\beta_P^2 + \beta_Q^2}\right) \\ z_2 = \left(\frac{\beta_Q^2\bar{p} + \beta_P^2\bar{q} + \beta_P\beta_Q\sqrt{\left\{\left(\beta_Q^2 + \beta_P^2\right) - \left(\bar{p} - \bar{q}\right)^2\right\}}}{\beta_P^2 + \beta_Q^2}\right) \end{cases}$$

La procédure de développement est maintenant simplifiée en considérant les différentes situations de la position des parties gauche et droite des courbes des fonctions d'appartenance en plus des intervalles de validité des solutions obtenues. Avec quelques manipulations mathématiques nous pouvons déduire que :

$$Pos\left(\bar{X} > \bar{Y}\right) = \begin{cases} 0 & si \quad \left(\left(\bar{p} + \beta_P\right) \leq \bar{q}\right) \\ 1 & si \quad \left(\bar{p} \geq \left(\bar{q} + \beta_Q\right)\right) \\ 1 - \left(\frac{z_1 - \bar{p}}{\beta_P}\right)^2 & si \quad \left(z_1 \in \left(\left[\bar{p}, \bar{p} + \beta_P\right] \cap \left[\bar{q}, \bar{q} + \beta_Q\right]\right)\right) \\ 1 - \left(\frac{z_2 - \bar{p}}{\beta_P}\right)^2 & si \quad \left(z_2 \in \left(\left[\bar{p}, \bar{p} + \beta_P\right] \cap \left[\bar{q}, \bar{q} + \beta_Q\right]\right)\right) \end{cases}$$

□

4.3 Déscription du problème d'ordonnancement

Le problème étudié dans ce chapitre peut être considéré comme une généralisation du problème d'ordonnancement mono objectif de minimisation de la somme pondérée du nombre des tâches en retard. Un objectif supplémentaire est pris en compte et

l'ensemble des tâches est à exécuter sur une ressource unique. Des contraintes additionnelles sur l'évaluation des deux fonctions objectifs sont incluses pendant la procédure d'optimisation. La formulation du problème est élaborée en se basant sur les hypothèses suivantes :

- La préemption des tâches n'est pas permise ;
- Seulement une tâche est traitée à la fois par la machine ;
- Les dates de disponibilité des tâches sont fixées au début de l'horizon du temps $t = 0$;
- Les durées de configuration dépendantes des séquences et les relations de précédences entre les tâches ne sont pas considérées ;
- Les durées opératoires ainsi que les dates échues de chaque tâche sont connues d'une façon approximative ;
- Les deux types des coefficients de pénalité et le coefficient d'apprentissage sont considérés comme des paramètres déterministes (valeurs entières pour les facteurs de pénalité et valeurs réelles pour le coefficient d'apprentissage.)

Les notations suivantes sont utilisées pour décrire notre problème d'ordonnancement à une machine avec paramètres incertains :

- n : Nombre de tâches ;
- $\tilde{p}_i$: Durée opératoire de tâche i $(i = 1, 2, ..., n)$;
- $\tilde{\mathrm{C}}_i$: Date fin d'exécution de tâche i $(i = 1, 2, ..., n)$;
- $\tilde{d}_i$: Date échue de tâche i $(i = 1, 2, ..., n)$;
- J : Ensemble de tâches avec $\mathrm{C}ard(\mathrm{J}) = n$;
- ω_i : Coefficient de pénalité de tâche i associé à la somme pondérée des possibilités de retard ;
- $\omega_i^{'}$: Coefficient de pénalité de tâche i associé à la somme pondérée des dates de fin d'exécution ;
- L_{eff} : Facteur d'apprentissage.

Comme la durée opératoire et la date échue d'une tâche i sont considérées comme des paramètres flous, nous proposons de modéliser l'incertitude au niveau de ces paramètres utilisant des fonctions d'appartenance (fonctions L et R) d'allure parabolique. En effet, les nombres flous plats ont été utilisés dans plusieurs problèmes pratiques pour la modélisation convenable des paramètres incertains (voir par exemple [Gupta et collab., 2013] et [Ebrahimnejad, 2016]). Par conséquent, la fonction d'appartenance générique des durées opératoires floues est donnée par :

$$\mu_{\tilde{p}_i}(x) = \begin{cases} 1 - \left(\frac{\underline{p}_i - x}{\alpha_{\tilde{p}_i}}\right)^2 & pour \quad x \leq \underline{p}_i \\ 1 \quad pour \quad \underline{p}_i \leq x \leq \tilde{p}_i \\ 1 - \left(\frac{x - \tilde{p}_i}{\beta_{\tilde{p}_i}}\right)^2 & pour \quad x \geq \tilde{p}_i \end{cases} \tag{4.9}$$

et la fonction d'appartenance des dates échues floues est donnée par :

$$\mu_{\tilde{d}_i}(x) = \begin{cases} 1 - \left(\frac{\underline{d}_i - x}{\alpha_{\tilde{d}_i}}\right)^2 & pour \quad x \leq \underline{d}_i \\ 1 \quad pour \quad \underline{d}_i \leq x \leq \bar{d}_i \\ 1 - \left(\frac{x - \bar{d}_i}{\beta_{\tilde{d}_i}}\right)^2 & pour \quad x \geq \bar{d}_i \end{cases} \tag{4.10}$$

D'une façon équivalente, une notation plus compacte peut être obtenue comme un quadruplet (représentation L-R) pour les deux paramètres $\tilde{p}_i = \left(\underline{p}_i, \bar{p}_i, \alpha_{\tilde{p}_i}, \beta_{\tilde{p}_i}\right)_{\text{L-R}}$ et $\tilde{d}_i = \left(\underline{d}_i, \bar{d}_i, \alpha_{\tilde{d}_i}, \beta_{\tilde{d}_i}\right)_{\text{L-R}}$. En pratique, la détermination exacte des valeurs de déviations gauche/droite et la valeur modale nécessite l'intervention des experts [BUCKLEY, 2005]. Dans le cadre de notre problème d'ordonnancement bi-objectif, nous considérons que la durée opératoire floue actuelle d'une tâche est affectée par la position de la tâche dans la séquence selon le modèle d'apprentissage de Biskup donné dans le cas déterministe par [BISKUP, 1999] :

$$p_i^{\text{A}} = i^{\text{L}_{eff}} \times p_i \tag{4.11}$$

Utilisant le principe d'extension pour les durées opératoires floues nous pouvons écrire :

$$\tilde{p}_i^{\text{A}} = i^{\text{L}_{eff}} \otimes \tilde{p}_i = \left(i^{\text{L}_{eff}} \times \bar{p}_i, i^{\text{L}_{eff}} \times \underline{p}_i, i^{\text{L}_{eff}} \times \alpha_{\tilde{p}_i}, i^{\text{L}_{eff}} \times \beta_{\tilde{p}_i}\right)_{\text{L-R}} \tag{4.12}$$

Avant de calculer les deux fonctions objectifs, nous avons besoin d'obtenir les expressions des dates de fin floues pour une tâche fixe i dans l'ordonnancement. La date fin floue d'exécution est calculée par la formule $\tilde{\text{C}}_i = \oplus \sum_{j=1}^{i} \tilde{p}_j^{\text{A}} = \oplus \sum_{j=1}^{i} j^{\text{L}_{eff}} \otimes \tilde{p}_j$. Dans la notation L-R, nous avons $\tilde{\text{C}}_i = \left(\sum_{j=1}^{i} j^{\text{L}_{eff}} \times \underline{p}_j, \sum_{j=1}^{i} j^{\text{L}_{eff}} \times \bar{p}_j, \sum_{j=1}^{i} j^{\text{L}_{eff}} \times \alpha_{\tilde{p}_j}, \sum_{j=1}^{i} j^{\text{L}_{eff}} \times \beta_{\tilde{p}_j}\right)_{\text{L-R}}$. En se basant sur la notation L-R de la date fin floue $\tilde{\text{C}}_i$, il est possible de déduire la possibilité de retard exprimant le degré de la réalisation de l'évènement "la tâche i est en retard" comme suit :

$$Pos\left(\tilde{C}_i > \tilde{d}_i\right) = \begin{cases} 0 \quad si \quad \left(\left(\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j + \sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\tilde{p}_j}\right) \leq \bar{d}_i\right) \\ 1 \quad si \quad \left(\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j \geq \left(\bar{d}_i + \beta_{\tilde{d}_i}\right)\right) \\ 1 - \left(\frac{x_1 - \sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j}{\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}}\right)^2 \quad si \quad \left(x_1 \in \left[\underline{h}_1\left(L_{eff}, \tilde{p}, \tilde{d}\right), \bar{h}_1\left(L_{eff}, \tilde{p}, \tilde{d}\right)\right]\right) \\ 1 - \left(\frac{x_2 - \sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j}{\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}}\right)^2 \quad si \quad \left(x_2 \in \left[\underline{h}_1\left(L_{eff}, \tilde{p}, \tilde{d}\right), \bar{h}_1\left(L_{eff}, \tilde{p}, \tilde{d}\right)\right]\right) \end{cases} \tag{4.13}$$

avec

$$\left[\underline{h}_1\left(L_{eff}, \tilde{p}, \tilde{d}\right), \bar{h}_1\left(L_{eff}, \tilde{p}, \tilde{d}\right)\right] = \left(\left[\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j, \sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j + \sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\tilde{p}_j}\right] \cap \left[\bar{d}_i, \bar{d}_i + \beta_{\tilde{d}_i}\right]\right)$$

$$x_1 = \frac{\beta^2_{\tilde{d}_i}\left(\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j\right) + \left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)^2 \bar{d}_i - \left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)\beta_{\tilde{d}_i}\left\{\left(\beta^2_{\tilde{d}_i} + \left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)^2\right) - \left(\left(\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j\right) - \bar{d}_i\right)^2\right\}^{\frac{1}{2}}}{\left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)^2 + \beta^2_{\tilde{d}_i}}$$

$$x_2 = \frac{\beta^2_{\tilde{d}_i}\left(\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j\right) + \left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)^2 \bar{d}_i + \left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)\beta_{\tilde{d}_i}\left\{\left(\beta^2_{\tilde{d}_i} + \left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)^2\right) - \left(\left(\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j\right) - \bar{d}_i\right)^2\right\}^{\frac{1}{2}}}{\left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\bar{p}_j}\right)^2 + \beta^2_{\tilde{d}_i}}$$

Donc, la première fonction objectif est exprimée par la somme pondérée des possibilités de retard des tâches dans l'ordonnancement ($\sum_{i=1}^{i=n} \omega_i \times Pos\left(\tilde{C}_i > \tilde{d}_i\right)$).
Concernant la deuxième fonction objectif, la somme pondérée des dates de fin floues d'exécution des tâches est exprimée par :

$$\sum_{i=1}^{n} \omega_i^{'} \otimes \tilde{C}_i = \left(\begin{array}{c} \sum_{i=1}^{n} \omega_i^{'} \times \left(\sum_{j=1}^{i} j^{L_{eff}} \times \underline{p}_j\right), \sum_{i=1}^{n} \omega_i^{'} \times \left(\sum_{j=1}^{i} j^{L_{eff}} \times \bar{p}_j\right), \\ \\ \sum_{i=1}^{n} \omega_i^{'} \times \left(\sum_{j=1}^{i} j^{L_{eff}} \times \alpha_{\tilde{p}_j}\right), \sum_{i=1}^{n} \omega_i^{'} \times \left(\sum_{j=1}^{i} j^{L_{eff}} \times \beta_{\tilde{p}_j}\right) \end{array} \right)_{L-R}$$

La formule finale de la deuxième fonction objectif exprimée par la magnitude de la somme pondérée des dates de fin floues ($Mag\left(\sum_{i=1}^{i=n} \omega_i^{'} \otimes \tilde{C}_i\right)$) est donnée par :

$$\mathrm{M}ag\left(\sum_{i=1}^{n}\omega_i^{'}\otimes\tilde{\mathrm{C}}_i\right)=\frac{1}{2}\left(\sum_{i=1}^{n}\omega_i^{'}\times\left(\sum_{j=1}^{i}j^{\mathrm{L}_{eff}}\times\underline{p}_j\right)+\sum_{i=1}^{n}\omega_i^{'}\times\left(\sum_{j=1}^{i}j^{\mathrm{L}_{eff}}\times\bar{p}_j\right)\right)$$

$$-\frac{2}{15}\left(\sum_{i=1}^{n}\omega_i^{'}\times\left(\sum_{j=1}^{i}j^{\mathrm{L}_{eff}}\times\alpha_{\tilde{p}_j}\right)-\sum_{i=1}^{n}\omega_i^{'}\times\left(\sum_{j=1}^{i}j^{\mathrm{L}_{eff}}\times\beta_{\tilde{p}_j}\right)\right)$$

Il est à mentionner que les coefficients de pénalité sont choisis de telle sorte que $\omega_i > \omega_i^{'}$, ce qui permet de refléter l'hypothèse économique que le coût de retard de livraison d'une unité d'un produit au client est souvent supérieure au coût par unité de temps par unité de produit engendré lors de la prolongation du séjour de cette unité au sein du système de production. Dans ce qui suit, nous présentons deux théorèmes annonçant la complexité des problèmes d'ordonnancements mono objectifs donnés par $1\left|\mathrm{L}_{eff},\tilde{p}_i,\tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i\times\mathrm{P}os\left(\tilde{\mathrm{C}}_i>\tilde{d}_i\right)$ et $1\left|\mathrm{L}_{eff},\tilde{p}_i\right|\mathrm{M}ag\left(\sum_{i=1}^{i=n}\omega_i^{'}\otimes\tilde{\mathrm{C}}_i\right)$ respectivement.

Théorème 4.2. *Le problème* $1\left|\mathrm{L}_{eff},\tilde{p}_i,\tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i\times\mathrm{P}os\left(\tilde{\mathrm{C}}_i>\tilde{d}_i\right)$ *est NP-difficile au sens fort.*

Démonstration.
En premier, nous avons besoin de reformuler notre problème d'optimisation sous forme décisionnelle.
Instance : Un ensemble fini de tâches J = {1, ..., n}. Pour chaque tâche $j \in$ J, une durée opératoire floue, une date échue floue, un coefficient de pénalité réel et un facteur d'apprentissage sont associés. Une valeur fixe B (borne supérieure) est donnée pour exprimer la taille de la contrainte.
Question : Existe t'il un ordonnancement pour J tel que $\sum_{i=1}^{i=n}\omega_i\times\mathrm{P}os\left(\tilde{\mathrm{C}}_i>\tilde{d}_i\right)\leq \mathrm{B}$?
Il est facile d'observer que la version décisionnelle du problème $1\left|\mathrm{L}_{eff},\tilde{p}_i,\tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i\times\mathrm{P}os\left(\tilde{\mathrm{C}}_i>\tilde{d}_i\right)$ appartient à la classe NP car un algorithme (machine de Turing) non déterministe nécessite seulement de prévoir une permutation des tâches et de tester dans un temps polynômial si la somme pondérée des possibilités de retard ne dépasse pas la taille de la contrainte B.
Maintenant, nous démontrons que le problème d'ordonnancement $1\left|\mathrm{L}_{eff}\right|\sum_{i=1}^{i=n}\mathrm{U}_i$ peut être réduit polynômialement à notre problème $1\left|\mathrm{L}_{eff},\tilde{p}_i,\tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i\times\mathrm{P}os\left(\tilde{\mathrm{C}}_i>\tilde{d}_i\right)$. Nous rappelons qu'une instance de $1\left|\mathrm{L}_{eff}\right|\sum_{i=1}^{i=n}\mathrm{U}_i$ consiste d'un ensemble de tâches J = {1, ..., n}, durée opératoire positives p_i, dates échues positives d_i et facteur d'apprentissage L_{eff} commun à toutes les tâches $i \in$ J. Le retard d'une tâche i est défini par :

$$\mathrm{U}_i=\begin{cases}1 & si \quad \mathrm{C}_i>d_i\\ 0 & si \quad \mathrm{C}_i\leq d_i\end{cases}$$

Alors, la fonction objectif dans ce problème est de déterminer la séquence pour laquelle le nombre de tâches en retard est minimal. Soit $\Pi = \left(n, p_{i=\overline{1,n}}, d_{i=\overline{1,n}}, \mathrm{L}_{eff}\right)$ une instance du problème $1\left|\mathrm{L}_{eff}\right|\sum_{i=1}^{i=n}\mathrm{U}_i$. Nous construisons l'instance correspondante

$\Pi' = \left(n, \tilde{p}_{i=\overline{1,n}}, \tilde{d}_{i=\overline{1,n}}, \omega_{i=\overline{1,n}}, L_{eff}\right)$ du problème $1\left|L_{eff}, \tilde{p}_i, \tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i \times Pos\left(\tilde{C}_i > \tilde{d}_i\right)$ comme suit :

$$\begin{cases} \tilde{p}_i = (p_i, p_i, 0, 0) \\ \tilde{d}_i = (d_i, d_i, 0, 0) \\ \omega_i = 1 \end{cases}$$

Comme dans ce cas, les durés opératoires et les dates échues sont des paramètres réels, l'assertion suivante est vraie pour chaque tâche $\left(Pos\left(\tilde{C}_i > \tilde{d}_i\right) = U_i\right) \Rightarrow \left(\sum_{i=1}^{i=n}\omega_i \times Pos\left(\tilde{C}_i > \tilde{d}_i\right) = \sum_{i=1}^{i=n} U_i\right)$. De cette implication, nous pouvons déduire que la solution optimale de $1\left|L_{eff}\right|\sum_{i=1}^{i=n} U_i$ pour l'instance Π est la même solution optimale de $1\left|L_{eff}, \tilde{p}_i, \tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i \times Pos\left(\tilde{C}_i > \tilde{d}_i\right)$ pour Π'. De plus, il est claire que l'instance Π' peut être déduite de Π dans un temps d'ordre polynômial $O(n)$. Ceci signifie qu'en implémentant un algorithme polynômial au problème $1\left|L_{eff}, \tilde{p}_i, \tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i \times Pos\left(\tilde{C}_i > \tilde{d}_i\right)$, nous serons capable de résoudre le problème $1\left|L_{eff}\right|\sum_{i=1}^{i=n} U_i$ dans un temps polynômial également. Le problème $1\left|L_{eff}\right|\sum_{i=1}^{i=n} U_i$ est NP-difficile au sens fort comme prouvé dans [Lin, 2007], d'où notre problème $1\left|L_{eff}, \tilde{p}_i, \tilde{d}_i\right|\sum_{i=1}^{i=n}\omega_i \times Pos\left(\tilde{C}_i > \tilde{d}_i\right)$ est fortement NP-difficile. □

Théorème 4.3. *Si les tâches ont des poids agréables flous i.e.* $\forall (i,j) \in J \times J$: $\left(Mag\left(\tilde{p}_j\right) \geq Mag\left(\tilde{p}_i\right)\right)$ *implique* $\left(\omega_i' \geq \omega_j'\right)$, *alors pour la minimisation du problème* $1\left|L_{eff}, \tilde{p}_i\right|Mag\left(\sum_{i=1}^{i=n}\omega_i' \otimes \tilde{C}_i\right)$, *il suffit de classer les tâches dans l'ordre croissant du rapport* $\frac{Mag(\tilde{p}_k)}{\omega_k'}$ *pour* $k = \overline{1,n}$

Démonstration.
La démonstration est basée sur la linéarité de la fonction magnitude (qui est vérifiée car les coefficients de pénalité prennent des valeurs positives) et la procédure de permutation des tâches comme détaillé pour un cas similaire dans [Bentrcia et Mouss, 2015].

□

Malgré que le théorème précédent déclare que le problème peut être résolu dans un temps d'ordre polynômial, mais dans le cas général lorsque l'hypothèse d'agréabilité floue n'est pas satisfaite, la complexité du problème $1\left|L_{eff}, \tilde{p}_i\right|Mag\left(\sum_{i=1}^{i=n}\omega_i' \otimes \tilde{C}_i\right)$ reste ouverte comme c'est le cas pour le même problème avec donnés déterministes [Moshieov, 2001]. À ce niveau de développement, il est possible de formuler notre problème d'ordonnancement bi-objectif à une machine avec données incertaines et effet d'apprentissage basé sur la position, les deux fonctions objectifs sont à minimiser. Donc, la notation à trois champs de notre problème est donnée par

$1\left|L_{eff},\tilde{p}_i,\tilde{d}_i\right|\left(\sum_{i=1}^{i=n}\omega_i\times Pos\left(\tilde{C}_i>\tilde{d}_i\right),Mag\left(\sum_{i=1}^{i=n}\omega_i^{'}\otimes\tilde{C}_i\right)\right)$. Si l'approche de la somme pondérée est utilisée pour élaborer une seule fonction objectif, l'analyse de la complexité sera une mission facile à accomplir et la complexité de calcul dépendra principalement de la complexité du problème le plus sophistiqué. Mais, quand les deux objectifs sont manipulés individuellement dans le contexte d'une stratégie de classement non dominée, le nombre des ordonnancements non dominés du front optimal peut croître d'une façon exponentielle en fonction de la taille du problème. Le théorème suivant est très utile car il relie la complexité des problèmes d'optimisations mono-objectifs aux problèmes multi-objectifs [CHEN et BULFIN, 1993].

Théorème 4.4. *Soit* $1||\gamma_1$ *et* $1||\gamma_2$ *deux problèmes d'ordonnancement à une machine avec les critères à minimiser* γ_1 *et* γ_2 *respectivement. Si le problème* $1||\gamma_1$ *(ou* $1||\gamma_2$*) est NP-difficile, alors les problèmes d'ordonnancement à une machine basés sur les approches de : critère primaire /secondaire* $1||(\gamma_1/\gamma_2)$*, non domination* $1||(\gamma_1,\gamma_2)$ *et somme pondérée* $1||(\lambda_1\gamma_1+\lambda_2\gamma_2)$ *sont tous des problèmes NP-difficiles.*

Sans restriction, ce théorème reste valide même avec l'introduction des contraintes du deuxième champ dans la notation de Graham.

Corollaire 4.4. *Le problème d'ordonnancement bi-objectif à une machine noté par* $1\left|L_{eff},\tilde{p}_i,\tilde{d}_i\right|\left(\sum_{i=1}^{i=n}\omega_i\times Pos\left(\tilde{C}_i>\tilde{d}_i\right),Mag\left(\sum_{i=1}^{i=n}\omega_i^{'}\otimes\tilde{C}_i\right)\right)$ *est NP-difficile au sens fort.*

Démonstration.
Ce Corollaire découle directement de l'application des Théorèmes 4.2 et 4.4. □

4.4 Présentation des métaheuristiques multi-objectifs à base de population

Les approches métaheuristiques peuvent être vues comme des outils de niveau abstrait ou génériques permettant l'obtention des solutions acceptables pour une large gamme de problèmes NP-difficiles. Les propriétés suivantes sont communes à la majorité des métaheuristiques : leurs principes sont inspirés des phénomènes naturels ; leurs évolutions sont sujettes à des altérations stochastiques ; elles n'exploitent pas les informations relatives au gradient ou dérivées supérieures de la fonction objectif et finalement elles contiennent des paramètres de configuration qui nécessitent un ajustement selon le problème étudié dans le but d'avoir des bonnes performances [BOUSSAÏD et collab., 2013].

Les métaheuristiques à base de population maintiennent plusieurs solutions candidates et essayent d'améliorer leurs qualités par l'utilisation de la théorie évolutionnaire de Darwin ou en exploitant d'autres comportements sociaux de coopération. Dans cette section, nous étudions l'applicabilité de deux métaheuristiques pour la résolution du problème d'ordonnancement bi-objectif à une machine $1\left|L_{eff},\tilde{p}_i,\tilde{d}_i\right|\left(\sum_{i=1}^{i=n}\omega_i\times Pos\left(\tilde{C}_i>\tilde{d}_i\right),Mag\left(\sum_{i=1}^{i=n}\omega_i^{'}\otimes\tilde{C}_i\right)\right)$. En particulier, nous focalisons les efforts sur la détermination de tous les ordonnancements non dominés par rapport aux deux objectifs simultanément : la somme pondérée des possibilités de retard des tâches et la somme pondérée des dates de fin floues d'exécution. La première métaheuristique est le fameux algorithme génétique élitiste de tri non dominé tandis que la deuxième

est la recherche coucou combinée avec une procédure de recherche locale afin d'intensifier la recherche autour des solutions rencontrées. Les deux approches cherchent à générer toutes les solutions optimales dans le but de permettre aux décideurs d'établir un compromis entre les objectifs considérés. Il est à mentionner que ces métaheuristiques sont à base de population commençant avec des configurations initiales aléatoires ou obtenues selon une heuristique et améliorent avec les itérations jusqu'à la vérification d'un critère d'arrêt.

4.4.1 Algorithme génétique élitiste de tri non dominé

Originalement, le paradigme de la programmation génétique a été proposé par Holland comme un outil d'optimisation mono-objectif, où un ensemble de solutions évolue par la modification et/ou maintien des propriétés des individus utilisant les opérateurs génétiques tels que la mutation, le croisement et la sélection. Pendant les générations consécutives, les solutions de bonne qualité ont plus de chance d'être sélectionnées pour contribuer à la progression de la population tandis que les solutions de qualité médiocre sont éliminées. La qualité d'une solution spécifique peut être évaluée avec la fonction finesse déduite à partir de la fonction objectif utilisant quelques règles [COLEY, 1999]. Mais plusieurs applications de la vie quotidienne incluent des objectifs multiples ce qui nécessite des approches d'optimisation plus élaborées pour l'évaluation de l'optimalité des solutions. Dans le travail pionnier de Deb et ses collègues, l'extension de l'algorithme génétique basé sur la non dominance a été proposée [DEB et collab., 2002]. La version modifiée baptisée algorithme génétique élitiste de tri non dominé version 2 (en anglais NSGA II) est caractérisée par une procédure de tri non dominé plus rapide, prise en considération de l'élitisme et l'élimination du paramètre de partage.

Dans le processus de cet algorithme génétique, une population d'individus (solutions) est initialement distribuée aléatoirement le long de l'espace de recherche ou créée selon une heuristique du problème traité. Après, la population initiale est triée selon le degré de dominance en des fronts différents où le premier front est l'ensemble non dominé dans la population courante. Chaque élément dans un front est dominé par les éléments des fronts prédécesseurs et domine les éléments des fronts successeurs jusqu'à l'atteinte du dernier front qui est dominé par tous les autres fronts. Afin d'établir un ordre entre les éléments du même front, un critère supplémentaire doit être introduit. Alors, deux attributs principaux sont affectés à chaque individu dans un front : le rang (ou bien le niveau de non dominance) et la distance d'encombrement. Cette dernière peut être considérée comme une mesure de proximité d'un individu de ces voisins, ce qui signifie que les grandes valeurs de la distance d'encombrement résultent en une meilleure diversité dans la population à travers l'espace de recherche. Ces deux attributs permettent l'élaboration d'un ordre partiel connu sous le nom de l'opérateur de comparaison d'encombrement et il est défini pour chaque pair de solution $(s,\ t)$ par :

$$\left(\left(s_{rang} < t_{rang}\right) \vee \left(\left(s_{rang} = t_{rang}\right) \wedge \left(s_{distance} > t_{distance}\right)\right)\right) \Rightarrow \left(s <_{n_{pop}} t\right)$$

À ce niveau du développement, nous avons entre les mains tous les ingrédients pour décrire l'exécution pas à pas de l'algorithme génétique multi-objectif adopté. En premier, la population initiale est combinée avec la population des enfants de taille n_{pop} créée par application des opérateurs de sélection par tournoi binaire, recombinaison et mutation. Dans ce cas, l'élitisme est assuré car les populations des parents et des enfants sont supportées dans la même génération. Maintenant, la population des parents associée à

la prochaine génération est remplie des fronts non dominants dans l'ordre de leurs rang jusqu'à dépasser la taille de la population n_{pop}. Les éléments du front responsable de cet excès sont sélectionnés dans l'ordre descendant de la distance d'encombrement jusqu'à atteindre la taille exacte de la population. Donc, la prochaine population des parents est employée pour la sélection, croisement et mutation pendant la création d'une population des enfants de même taille. Ces étapes sont réitérées tant que le critère d'arrêt n'est pas satisfait comme illustré sur le pseudo code suivant (Algorithme 4.1) [DEB, 2001].

Algorithm 4.1 Algorithme génétique élitiste de tri non dominé

DEBUT
Créer une population initiale des parents P_0 aléatoirement ou selon une heuristique.
Classer la population des parents en des niveaux non dominés
Associer à chaque solution une finesse égale à son niveau de non dominance
Créer une population des enfants Q_0 de taille n_{pop} utilisant les opérateurs de sélection par tournoi binaire, croisement et mutation
TANT QUE (Condition d'arrêt non satisfaite) **FAIRE**
  Combiner les populations des parents et des enfants pour obtenir une population $R_i = P_i \cup Q_i$ de taille $2n_{pop}$
  Effectuer un classement de non domination à R_i avec identification des différents fronts F_j
  Initialiser la nouvelle population des parents $P_{i+1} \leftarrow \phi$
  Initialiser le compteur $j \leftarrow 1$
  TANT QUE $(Card(P_{i+1}) + Card(F_j) < n_{pop})$ **FAIRE**
    Affecter la distance d'encombrement aux solution dans le front F_j
    Mettre à jour la population des parents $P_{i+1} \leftarrow P_{i+1} \cup F_j$
    Incrémenter le compteur j
  FIN TANT QUE
  Effectuer la procédure de tri à base de distance d'encombrement $(<_{n_{pop}}, F_j)$
  Inserer les solutions les plus dispersées dans la population P_{i+1} utilisant les valeurs de la distance d'encombrement du front classé F_j
  Créer la population des enfants Q_{i+1} de P_{i+1} utilisant les opérateurs de sélection par tournoi, croisement et mutation
FIN TANT QUE
Retourner le meilleur front correspondant à l'ensemble des fonctions objectifs $\{f_i\}$
FIN

Dans la majorité des problèmes d'optimisation continue, la configuration d'une instance est codée comme étant une suite finie de chaîne de caractères, ce qui résulte en un temps gaspillé pour le codage et le décodage. Donc, pour notre problème nous adoptons le codage de permutation qui permet la manipulation directe des paramètres de configuration en plus de la représentation de l'ensemble des ordonnancements comme délivré par le diagramme de Gantt. Nous devons aussi mentionner que quelques étapes de l'algorithme génétique multi-objectif original sont modifiées pour bien s'adapter au contexte de notre travail. Par exemple, notre stratégie de sélection est légèrement différente de celle proposée dans la version originale. La sélection est basée sur un processus de sélection par tournoi où n_{tour} individus compétiteurs sont sélectionnés arbitrairement. Malgré que la taille du tournoi dans l'algorithme est fixée à deux, nous la considérons comme variable dans le but de tester les performances de l'algorithme en fonction du nombre de tâches n. Les n_{tour} individus sélectionnés sont triés selon l'opérateur de comparaison à base d'encombrement et les deux premiers individus sont sujets à un opérateur de croisement avec

une probabilité fixe pour l'obtention de deux descendants ajoutés à la population des enfants. La méthode de croisement utilisée dans ce travail est le croisement à deux points, où deux positions sont générées aléatoirement et les génotypes localisés entre ces deux points sont copiés d'un parent et la séquence est complétée par les génotypes de l'autre parent. En inversant l'ordre de sélection des parents, il est possible d'avoir un deuxième enfant dans le processus de croisement. Cette procédure est répétée jusqu'au remplissage des positions dans la population des enfants. En ce qui concerne l'opérateur de mutation, nous adoptons la mutation basée sur la permutation des tâches où deux positions dans un ordonnancement sont choisies arbitrairement et les tâches associées sont interchangées si la probabilité générée est égale ou dépasse le taux de mutation. La complexité de l'approche NSGA II proposée en fonction du nombre de tâches n sous l'hypothèse que la taille de la population et le nombre maximum d'itération dépend linéairement de n peut être calculée comme suit en commençant par la phase d'initialisation :

- Création aléatoire de la population initiale des parents P_0 : $O\left(n^3\right)$;
- Tri de la population initiale des parents en des niveaux de non dominance : $O\left(n^3\right)$ pour l'évaluation des fonctions objectifs et $O\left(n^2\right)$ pour la procédure du tri rapide ;
- Affectation d'un niveau de non dominance à chaque solution : $O(n)$;
- Création de la population des enfants Q_0 : $O\left(n^2\sqrt{n}\right)$.

Pour la simplification, nous concentrons dans la boucle POUR principale sur la complexité des instructions pour une itération comme illustré ci-dessous.

- Combinaison des populations des parents et des enfants : $O(n)$;
- Le classement de non dominance de la population combinée : $O\left(n^3\right)$ pour l'évaluation des fonctions objectifs et $O\left(n^2\right)$ pour la procédure rapide de classement ;
- Exécution de la boucle intérieure TANT QUE : $O\left(n^3\right)$;
- Exécution de la procédure de tri basé sur l'opérateur de comparaison d'encombrement : $O\left(n \times \log n\right)$;
- Inclusion des solutions les plus dispersées : $O(n)$;
- Création de la nouvelle population des enfants : $O\left(n^3\right)$.

Donc, la complexité totale après la considération du nombre d'itérations (qui est de l'ordre $O(n)$) devient $O\left(n^4\right)$, ce qui exprime que la complexité totale de l'approche est fortement gouvernée par les opérations de création des populations et de tri en des niveaux de non dominance. Par conséquent, toute amélioration relativement au temps de calcul doit procéder par la réduction de la complexité de ces sections.

4.4.2 Algorithme recherche coucou

En comparaison avec plusieurs métaheuristiques existantes, la recherche coucou est relativement une méthode d'optimisation récente proposée par Yang et Deb ([YANG et DEB, 2009] et [YANG et DEB, 2010]). Les bases de cette méthode sont inspirées du comportement parasitaire de la femelle coucou qui imite pendant la ponte les propriétés des oeufs appartenant aux nids d'autres espèces des oiseaux pour obtenir une probabilité de reproduction plus élevée. Juste après leurs éclosion, les oisillons essayent par l'instinct

naturelle d'expulser les oeufs du oiseau hôte du nid en plus de simuler les sons des oissillons hôtes afin d'augmenter leur part de la nourriture. Lorsque l'implémentation algorithmique du comportement du coucou est appliquée à des problèmes d'optimisation continue, l'utilisation de vol de Lévy au lieu d'un mouvement brownien pur mène à une meilleure performance. Via l'utilisation de cette règle de transition, la diversification dans les populations générées par la recherche coucou devient plus importante car la distribution des longeurs des pas est à queue lourde, et toute taille du pas est permise. La distance totale de l'origine converge à une distribution stable après un grand nombre de pas. La transition arbitraire utilisant le vol de Lévy entre deux états est décrite par l'expression suivante [PAVLYUKEVICH, 2007] :

$$x_{j+1}^{i} = x_{j}^{i} + \kappa \oplus \text{Lévy}(s, \sigma) \tag{4.14}$$

où κ represente un facteur d'échelle de la taille du pas choisi dépendamment du problème considéré et σ est un paramètre de controle choisi dans l'intervalle $]1, 3[$. Sous plusieurs situations, l'utilisation d'une valeur de $O\left(\frac{L}{100}\right)$ peut être un bon choix où L est l'ordre charactéristique du problème. L'implémentation du vol de Lévy nécessite un algorithme précis mais facile à utiliser pour approximer la distribution Lévy. La comparaison donnée dans [SCALAS et LECCARDI, 2005] a montré que l'algorithme de Mantegna est le plus efficace. Cet algorithme s'exécute en trois phases pour générer la longueur du pas définie par [MANTEGNA, 1994] :

$$s = \frac{u}{|v|^{\frac{1}{\beta}}} \tag{4.15}$$

Les deux paramètres u et v sont supposés suivant une distribution de la loi normale :

$$\begin{cases} u = N\left(0, \sigma_u^2\right) \\ \\ v = N\left(0, \sigma_v^2\right) \end{cases}$$

Les variances sont calculées selon les expressions suivantes :

$$\begin{cases} \sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{\frac{1}{\beta}} \\ \\ \sigma_v = 1 \end{cases}$$

avec Γ est la fonction Gamma.
Les étapes principales de l'algorithme standard de la recherche coucou sont résumées sur l'Algorithme 4.2 [MOHAMAD et collab., 2014].

Mais cette représentation standard émergée initialement pour le cas continu ne peut pas être appliquée directement dans le contexte de notre problème d'ordonnancement à cause de la nature discrète de l'espace de solution en plus de l'existence des objectifs multiples. Donc, plus d'efforts doivent être dédiés à l'adaptation adéquate des composantes principales de la recherche coucou. Pour achever cette adaptation, le concept du tri non dominé employé dans l'algorithme génétique multi-objectif (NSGA II) est importé pour construire une version multi-objectif de la recherche coucou. En outre, une nouvelle reformulation du vol de Lévy pour satisfaire les propriétés discrètes des ordonnancements est recommandable. En se basant sur les travaux de ([HANOUN et collab., 2014] et [BALASUBBAREDDY et collab., 2015]), nous introduisons un mécanisme

Algorithm 4.2 Version originale de la recherche coucou

DEBUT
Générer une population initiale de n_{nid} nids hôtes.
TANT QUE (Condition d'arrêt non satisfaite) **FAIRE**
  Sélectionner arbitrairement une solution coucou
  Générer une solution en appliquant le vol de Lévy
  Évaluer la qualité de la solution obtenue f_i
  Choisir un nid (j) arbitrairement de n_{nid}
  SI ($f_i < f_j$) **ALORS**
    Remplacer j par la nouvelle solution
  FIN SI
  Abondonner une fraction $p_a \times n_{nid}$ des mauvais nids et générer des nouvelles solutions
  Préserver les nids avec meilleures qualités de solutions
  Effectuer un classement pour obtenir la meilleure solution
FIN TANT QUE
Retourner la meilleure solution correspondante au minimum de la fonction objectif f
END

de séquence voisine au lieu du vol de Lévy continu. La stratégie de génération des solutions voisines est mise à jour dynamiquement avec l'évolution des itérations. Pour des petites valeurs de nombre d'itérations, des solutions aléatoires sont crées pour constituer un type de diversification dans l'espace de recherche. Avec l'augmentation du nombre d'itérations, les solutions voisines sont générées à partir de la meilleure solution trouvée dans l'historique. Donc, il est possible de préserver un compromis entre la diversification et l'intensification de la recherche, où le seuil de probabilité de génération (P_{th}) dépendant du compteur d'itérations (i_c) est donné par :

$$P_{th} = \frac{i_c}{Maxgeneration} \tag{4.16}$$

L'ensemble des opérations utilisées pour générer une nouvelle solution à partir de la meilleure solution dans la stratégie d'intensification est composée de trois modifications locales. L'opérateur d'insertion supprime une tâche de sa position originale et l'insert dans une nouvelle position. L'opérateur de permutation a pour objectif d'obtenir une nouvelle solution par l'échange des contenues de deux positions différentes. L'opérateur de décalage décale circulairement l'ordre des tâches par un pas fixe à gauche ou à droite selon la valeur aléatoire binaire générée. Plusieurs études ont prouvé que l'hybridation d'une métaheuristique avec une procédure de recherche locale résulte en des avantages tels que la convergence rapide ainsi qu'une meilleure qualité des solutions proches de l'optimale [BLUM et collab., 2011]. Dans ce contexte, la recherche de voisinage variable est adoptée dans l'intégration avec la recherche coucou. La recherche du voisinage variable est une méthode de recherche locale stochastique basée sur l'alternance stochastique des voisinages pendant la phase de recherche. Dans chaque étape, la recherche utilise certains opérateurs élémentaires pour la génération et l'exploitation du voisinage d'une solution avec l'objectif de tomber sur une solution améliorée. Alors, le processus commence avec la sélection d'un opérateur qui est abandonné et remplacé par un autre lorsque la recherche échoue dans l'amélioration de la solution courante. La recherche s'arrête avec l'identification d'une solution qui est localement optimale par rapport à tous les voisi-

nages générés [HANSEN et MLADENOVIĆ, 2001].

Dans ce chapitre, notre recherche à voisinage variable est basée sur les mêmes opérateurs utilisées dans le mécanisme de séquence voisine (insertion, permutation et décalage). Ces opérateurs sont appliqués consécutivement à tous les individus de la population courante avec l'espérance d'améliorer la qualité des mauvaises solutions. Lorsque une solution de meilleure qualité est obtenue, la recherche à voisinage variable relance la même procédure pour trouver une solution améliorée jusqu'à la satisfaction du critère d'arrêt. Malgré que la taille totale de tous les voisinages associés aux opérateurs élémentaires est $O\left(n^2\right)$ (la taille de voisinage par insertion est $n(n-1)$, la taille de voisinage de permutation est $\frac{n(n-1)}{2}$ et la taille du voisinage de décalage est 1). Nous limitons la condition d'arrêt à n qui ne permet pas seulement de couvrir une sous partie du voisinage d'une solution mais aussi de réduire la complexité de calcul en particulier que la totalité de la population est testée. Les détails de la recherche à voisinage variable sont présentées dans l'Algorithme 4.3 (Cet algorithme est une adaptation de la recherche à voisinage variable présentée dans [HANOUN et collab., 2014]).

Algorithm 4.3 Différentes phases de la formulation standard de la recherche à voisinage variable adaptée à notre problème

DEBUT
Initialiser le compteur d'itérations ($k \leftarrow 1$)
Fixer le critère d'arrêt ($Stop \leftarrow n$)
TANT QUE $k \leq Stop$ **FAIRE**
  Générer une nouvelle solution de la solution courante utilisant l'opérateur d'insertion
  SI (la nouvelle solution domine la solution courante) **ALORS**
    Considérer la nouvelle solution comme solution courante
    Initialiser le compteur d'itérations à zero ($k \leftarrow 0$)
  SINON
    Générer une nouvelle solution de la solution courante utilisant l'opérateur de permutation
    SI (la nouvelle solution domine la solution courante) **ALORS**
      Considérer la nouvelle solution comme solution courante
      Initialiser le compteur d'itérations à zero ($k \leftarrow 0$)
    SINON
      Générer une nouvelle solution de la solution courante utilisant l'opérateur de décalage
      SI (la nouvelle solution domine la solution courante) **ALORS**
        Considérer la nouvelle solution comme solution courante
        Initialiser le compteur d'itérations à zero ($k \leftarrow 0$)
      FIN SI
    FIN SI
  FIN SI
  Incrémenter le compteur d'itérations ($k \leftarrow k+1$)
FIN TANT QUE
Retourner la meilleure solution trouvée
END

La formulation hybride de l'algorithme recherche coucou avec la nouvelle définition du vol de Lévy et la recherche à voisinage variable est complète à ce niveau où le pseudo code de cette version multi-objectif est illustré dans l'Algorithme 4.4.

Algorithm 4.4 Version multi-objectif hybride de la recherche coucou

DEBUT
Créer une population initiale P_0 aléatoirement ou selon une heuristique.
Trier la population des parents en des niveaux de non domination
TANT QUE (Condition d'arrêt non satisfaite) **FAIRE**
  Initialiser la population des enfants $Q_k \leftarrow P_k$
  POUR ($i = 1$ à n_{nid}) **FAIRE**
    Prendre un individu aléatoire x_1^{rand}
    Générer une nouvelle solution cuck en appliquant le vol de Lévy à la solution x_1^{rand}
    Prendre un autre individu aléatoire x_2^{rand}
    SI (x_2^{rand} est dominée par cuck) **ALORS**
      Affecter cuck à x_2^{rand} ($x_2^{rand} \leftarrow cuck$)
    FIN SI
  FIN POUR
  Prendre un individu aléatoire x_3^{rand}
  Initialiser aléatoirement une fraction de la population des enfants de taille $p_a \times n_{nid}$ utilisant x_3^{rand} comme graine initiale
  Appliquer la recherche à voisinage variable à chaque solution de la population des enfants Q_k
  Combiner les populations des parents et des enfants pour obtenir une population $R_k = P_k \cup Q_k$ de taille $2 \times n_{nid}$
  Trier la population resultante en differents niveaux de non domination
  Créer la population des parents prochaine P_{k+1} en sélectionnant les meilleurs n_{nid} solutions non dominées de R_k
FIN TANT QUE
Retourner le meilleur front correspondant à l'ensemble des fonctions objectifs $\{f_j\}$
END

Pour l'évaluation de la complexité totale de la recherche coucou hybride, nous établissons les résultats élémentaires suivantes :

- Création aléatoire de la population initiale des parents $P_0 : O\left(n^3\right)$
- Classement de la population initiale des parents en des niveaux de non dominance : $O\left(n^3\right)$ pour l'évaluation des fonctions objectifs et $O\left(n^2\right)$ pour la procédure rapide de classement

Pour des raisons de simplification, nous concentrons, dans la principale boucle TANT QUE, sur la complexité des instructions pour une itération comme suit :

- Initialisation de la population des enfants $Q_k : O\left(n^2\right)$;
- Exécution de la boucle POUR : $O(n)$ pour le tirage d'un individu arbitraire x_1^{rand}, $O\left(n^2\right)$ pour la génération d'une nouvelle solution cuck par application du vol de Lévy, $O(n)$ pour le tirage d'un deuxième individu arbitraire x_2^{rand} et $O\left(n^2\right)$ pour le test de non dominance ;
- Tirage du troisième individu arbitraire : $O(n)$;
- Initialization aléatoire d'un pourcentage de taille $p_a \times n$ de la population des enfants utilisant le troisième individu arbitraire x_3^{rand} comme graine : $O\left(n^2\right)$;
- Application de la recherche à voisinage variable : $O\left(n \times g(n)\right)$;
- Combinaison des populations des parents et des enfants : $O(n)$;
- Classement non dominé de la population combinée : $O\left(n^3\right)$ pour l'évaluation de la fonction objectif et $O\left(n^2\right)$ pour la procédure de classement rapide ;
- Création de la prochaine population des parents : $O(n)$.

Si on note la complexité de la recherche à voisinage variable par $g(n)$, la complexité totale de l'approche hybride proposée avec la considération du nombre de génération est alors donnée par $O\left(n^4 + n^2 \times g(n)\right)$. On peut noter que le nombre d'opérations necessaire pour la recherche à voisinage variable est dans le pire des cas de l'ordre de $n!$. De ce résultat, nous pouvons déduire que la complexité de l'algorithme est basée sur les sections de recherche à voisinage variable et le tri en niveaux de non dominance. Donc, toute amélioration doit être focalisée sur la réduction de la complexité de ces deux méthodes.

4.4.3 Stratégies de tri non dominé

Dans cette sous section, nous décrivons formellement les propriétés de base des critères de dominance utilisés dans le contexte des approches algorithme génétique élitiste de tri non dominé et recherche coucou. La règle de dominance de Pareto est un critère classique utilisé abondamment dans la littérature de l'optimisation multi-objectif. Cependant, cette règle de dominance peut être dans certains cas non suffisante, ce qui motive la proposition d'autres critères de dominance. Dugardin et al ont montré qu'un algorithme génétique multiobjectif basé sur la dominance de Lorenz a des meilleures performances en comparaison avec un algorithme génétique conventionnelle basé sur la dominance de Pareto [DUGARDIN et collab., 2010]. Dans [MOGHADDAM et collab., 2015], six métaheuristiques ont été proposées pour la résolution d'un problème d'ordonnancement bi-objectif à une machine avec critère de rejet. L'utilisation de différents schémas de dominance a permis de réduire considérablement le nombre de solutions obtenues. Dans ce qui suit, nous présentons quelques définitions concernant les critères de Pareto et de Lorenz dans le cadre du problème d'optimisation multi-objectif

défini par $\pi^* = \min_{\pi\in\Omega}\left(f_1(\pi), f_2(\pi)\right)$ avec $f_1(\pi) = \sum_{i=1}^{i=n} \omega_i'(\pi) \times Pos\left(\tilde{C}_i(\pi) > \tilde{d}_i(\pi)\right)$, $f_2(\pi) = Mag\left(\sum_{i=1}^{i=n} \omega_i(\pi) \otimes \tilde{C}_i(\pi)\right)$ et Ω est l'ensemble des ordonnancements faisables π.

Définition 4.10. *On dit que l'ordonnancement π_1 domine l'ordonnancement π_2 selon le critère de dominance de Pareto si les inégalités suivantes sont satisfaites :*

$$\begin{cases} \forall k \in \{1,2\}\ f_i(\pi_1) \leq f_i(\pi_2) \\ \\ \exists k \in \{1,2\}\ f_i(\pi_1) < f_i(\pi_2) \end{cases} \tag{4.17}$$

Définition 4.11. *On dit que l'ordonnancement π_1 domine l'ordonnancement π_2 selon le critère de dominance de Lorenz si le vecteur Lorenz de l'ordonnancement π_1 domine le vecteur Lorenz de l'ordonnancement π_2 au sens du critère de Pareto, où le vecteur Lorenz de l'ordonnancement π est exprimé par* $L(\pi) = \left(\max_{i\in\{1,2\}}\left(f_i(\pi)\right), \sum_{i=1}^{i=2} f_i(\pi)\right)$.

Donc, il est recommandé d'étudier l'influence du couplage entre différents critères de dominance sur les performances des approches multi-objectifs. En effet, cette combinaison peut être un outil efficace pour guider la recherche vers des régions prometteuses ou bien d'étendre les parties extrêmes des fronts non dominés, ce qui aide à croître la diversité des solutions.

4.5 Expérimentation numérique et résultats

Dans le but d'évaluer l'efficacité des approches métaheuristiques proposées, des expérimentations numériques sont conduites. Toutes les méthodes considérées sont programmées sous l'environnement MATLAB à cause des nombreuses facilitées offertes en plus de son utilisation comme un outil de modélisation des applications industriels pendant ces dernières années [Wonga et Lai, 2011]. Les tests sont exécutés sur un processeur i7 avec 8 GB de RAM sous système d'exploitation Windows 7. L'ensemble test contient des problèmes avec nombre de tâches $n \in \{5,6,7,8,9,10,15,20,25,30,35,40\}$ avec un totale de 20 instances générées pour chaque taille. Alors, un nombre total de $12 \times 20 = 240$ instances est obtenu. Une tâche est caractérisée par sa durée opératoire, coefficients de pénalité associés à l'état de retard et à la date échue, où les nombres flous du type L-R sont utilisés pour modéliser l'incertitude au niveau de ces paramètres. Le facteur d'apprentissage est supposé à valeurs réelles dans l'intervalle $[-0.1, 0]$. Dans ce travail, les instances de test sont générées numériquement en se basant sur les règles élucidées dans les articles [Kim, 1993] et [Saidi Mehrabad et Pahlavani, 2009]. La structure générale de la procédure adoptée dans la génération des instances de taille fixe est gouvernée par les règles suivantes :

i) Pour définir les durées opératoires floues $\tilde{p}_j$ de la tâche j, $\underline{p}_j$ et $\bar{p}_j$ sont attribués des valeurs entières des distributions uniformes discrètes $\left[a_{\tilde{p}_j}, b_{\tilde{p}_j}\right]$ et $\left[\underline{p}_j, b_{\tilde{p}_j}\right]$ respectivement. Donc, $\alpha_{\tilde{p}_j}$ et $\beta_{\tilde{p}_j}$ sont définies comme des entiers aléatoires dans l'intervalle $\left[1, 0.1 \times \left(\frac{\underline{p}_j + \bar{p}_j}{2}\right)\right]$;

ii) Les coefficients de pénalité ω_j, associés à la somme pondérée des possibilités de retard sont générés comme des entiers positifs de la distribution uniforme discrète $\left[a_{\omega_j}, b_{\omega_j}\right]$;

iii) Les coefficients de pénalité $\omega_j^{'}$, associés avec la somme pondérée des dates de fin d'exécution, sont obtenus en multipliant par 10 les nombres aléatoires générés à partir de la distribution uniforme discrète $\left[a_{\omega_j^{'}}, b_{\omega_j^{'}}\right]$;

iv) En analogie avec les modèles d'ordonnancement déterministes, la gamme des dates échues (notée par RDD) et le facteur d'étroitesse des dates échues (noté par TF) sont introduits pour contrôler la difficulté à générer les problèmes où une valeur élevée de RDD indique une large plage de dates échues, tandis qu'une faible valeur indique une plage étroite de dates échues. Des valeurs de TF proche de 1 indique que les dates échues sont serrées et des valeurs proche de 0 indique que les dates échues sont lâches. Les valeurs des deux paramètres sont sélectionnées de l'intervalle $[0,1]$;

v) Pour la date échue floue $\tilde{d}_j$ de la tâche j, $\underline{d}_j$ et $\bar{d}_j$ sont affectés des valeurs entières de la distribution uniforme $\left[\left(\sum_{j=1}^{n} \underline{p}_j\right) \times \left(1 - \text{TF} - \frac{\text{RDD}}{2}\right), \left(\sum_{j=1}^{n} \bar{p}_j\right) \times \left(1 - \text{TF} + \frac{\text{RDD}}{2}\right)\right]$ et $\left[\underline{d}_j, \left(\sum_{j=1}^{n} \bar{p}_j\right) \times \left(1 - \text{TF} + \frac{\text{RDD}}{2}\right)\right]$ respectivement. Par la suite, $\alpha_{\tilde{d}_j}$ et $\beta_{\tilde{d}_j}$ sont tirés aléatoirement de l'intervalle $\left[1, 0.2 \times \left(\frac{\underline{d}_j + \bar{d}_j}{2}\right)\right]$.

Une description complète de toutes les instances ainsi que les données de configurations utilisées dans ce travail peut être téléchargée sur le lien :<http://lab.univ-batna2.dz/lap/index.php/component/content/article?id=31>.
Les valeurs affectées aux bornes inférieures et supérieures nécessaires comme des entrées pendant la procédure de génération des instances sont illustrées sur le Tableau 4.1.

TABLEAU 4.1 Valeurs des différents paramètres nécessaires à la génération des instances

Paramètre	Notation	Borne inférieure	Borne supérieure
Durée opératoire floue	$\tilde{p}_j$	1	40
Pénalité sur la possibilité de retard	ω_j	1	15
Pénalité sur la date fin d'exécution	$\omega_j^{'}$	1	15

Comme le front non dominé optimal est dans la plus part des cas inconnu à l'exception des problèmes de petites tailles pour lesquels le calcul par recherche exhaustive est raisonnable, nous évaluons les performances des approches proposées à travers trois métriques capables d'estimer les différents aspects des fronts approximés. Il est à noter que ces métriques ne nécessitent pas en générale le calcul du front optimal. La première métrique est le nombre d'ordonnancements (i.e. Cardinal) constituant le front produit par un algorithme alg_i et il est défini par :

$$n^f_{alg_i} = Card(front_{alg_i}) \tag{4.18}$$

La deuxième métrique proposée pour la première fois par Zitzler et ses collègues [ZITZLER et collab., 2003] est la proportion des ordonnancements trouvés par une méthode d'optimisation alg_i qui sont dominés par au moins un ordonnancement obtenu par une autre méthode alg_j. Cette métrique est indépendante de l'échelle dans le sens qu'elle ne nécessite pas la normalisation des valeurs de la fonction objectif même si l'ordre de

grandeur des fonctions objectifs est relativement différent. La proportion des ordonnancements dominés est exprimée par :

$$\mathrm{R}_{alg_i - alg_j} = \frac{Card\left(\left\{\pi_2 \in front_{alg_j} \middle| \exists \pi_1 \in front_{alg_i} : \pi_1 \geq \pi_2\right\}\right)}{Card\left(front_{alg_j}\right)} \tag{4.19}$$

La troisième métrique est la distance modifiée introduite initialement par Riise et étendue par Dugardin et ses collègues dans le but de supporter les valeurs qui partagent les fronts comparés ([Riise, 2002] et [Dugardin et collab., 2010]). L'idée derière cette métrique réside dans la mesure de la distance algébrique totale entre les solutions du front alg_i à leurs projections orthogonales sur le front de référence alg_j. La normalisation est achevée pour éviter la sensibilité à la variation du nombre de solutions. Donc, différentes solutions sont connectées et la ligne est extrapolée pour former des demi droites continues pour permettre la projection des points localisés en dehors de l'étendu du front de référence [Lacomme et collab., 2006]. Cette métrique peut être formulée dans une notation normalisée comme suit :

$$\mu_{alg_i - alg_j} = \frac{1}{Card\left(front_{alg_j}\right)} \frac{\sum_{k=1}^{Card\left(front_{alg_j}\right)} d_k}{\sqrt{\left(f_{1\max} - f_{1\min}\right)^2 + \left(f_{2\max} - f_{2\min}\right)^2}} \tag{4.20}$$

avec d_k est la distance entre un point du front généré par alg_i et sa projection orthogonale sur le front généré par alg_j. Le reste des paramètres dans l'expression sont données par :

$$\begin{cases} f_{1\min} = \min\left\{f_1(\pi) \middle| \pi \in front_{alg_i} \cup front_{alg_j}\right\} \\ f_{1\max} = \max\left\{f_1(\pi) \middle| \pi \in front_{alg_i} \cup front_{alg_j}\right\} \\ f_{2\min} = \min\left\{f_2(\pi) \middle| \pi \in front_{alg_i} \cup front_{alg_j}\right\} \\ f_{2\max} = \max\left\{f_2(\pi) \middle| \pi \in front_{alg_i} \cup front_{alg_j}\right\} \end{cases}$$

Mais il existe quelques situations où la projection n'est pas possible à cause de la position relative de la solution considérée. Par exemple, comme indiqué sur la Figure 4.1, la solution avec label B ne peut pas être projetée sur le front de référence. En effet, soit $(\Delta_{1,2})$ la droite reliant les points 1 et 2, $(\Delta_{2,3})$ la droite reliant les points 2 et 3 respectivement. Les équations de ces droites sont données par :

$$\begin{cases} \left(\Delta_{1,2}\right) : y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1 \\ \left(\Delta_{2,3}\right) : y = \frac{y_3 - y_2}{x_3 - x_2}(x - x_2) + y_2 \end{cases}$$

Les droites orthogonales correspondantes sont notées par $\left(\Delta_{1,2}^{\perp}\right)$ et $\left(\Delta_{2,3}^{\perp}\right)$ où leurs équations sont exprimées par :

$$\begin{cases} \left(\Delta^{\perp}_{1,2}\right) : y = -\frac{x_2 - x_1}{y_2 - y_1}(x - x_1) + y_1 \\ \left(\Delta^{\perp}_{2,3}\right) : y = -\frac{x_3 - x_2}{y_3 - y_2}(x - x_2) + y_2 \end{cases}$$

En se basant sur ces équations, on peut démontrer que les points appartenant à l'ensemble $\left\{(x,y) \in \mathbb{R}^2 : \left(-\frac{x_2 - x_1}{y_2 - y_1}(x - x_1) + y_1 < 0\right) \wedge \left(-\frac{x_3 - x_2}{y_3 - y_2}(x - x_2) + y_2 < 0\right)\right\}$ ne peuvent pas être projetés sur le front de référence (front non dominé 1). Une telle situation peut dégrader considérablement l'efficacité de la métrique de la distance modifiée dans l'évaluation des performances des approches d'optimisation multi-objectif. Selon nos connaissances, aucune solution n'est fournie dans la littérature pour cette limitation et pour cela nous proposons pour la solution avec label B de calculer la distance à la plus proche solution du front de référence au lieu de la distance de projection. En effectuant ce changement, nous espérons que le nombre de positionnements aberrants entre les deux fronts peut être réduit considérablement.

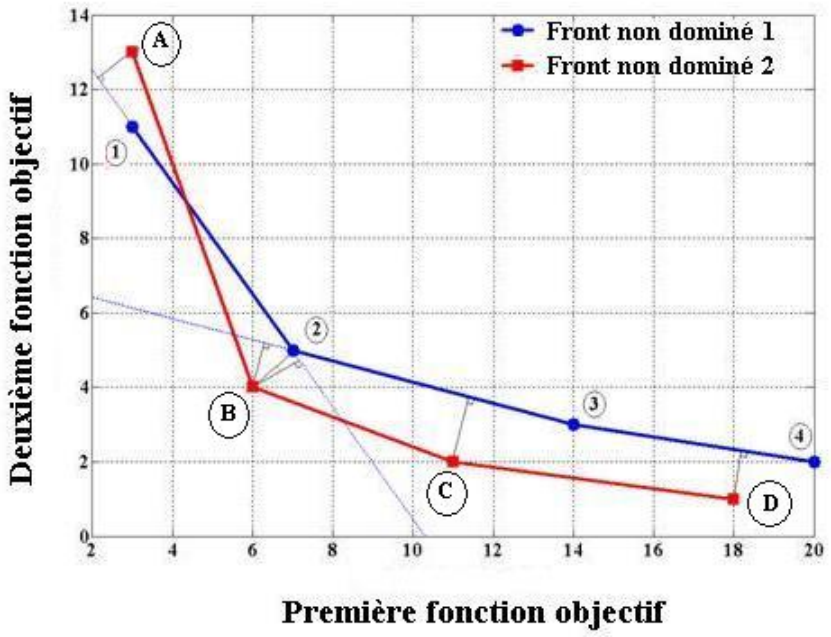

FIGURE 4.1 Situations possibles de projection de deux fronts non dominés obtenus par des métaheuristiques différentes.

Il est à mentionner que les deux directions doivent être considérées dans le calcul de la proportion des ordonnancements dominés et la distance modifiée. Ceci peut être expliqué par le fait que les deux métriques ne sont pas symétriques ni complémentaires, ce qui nécessite l'évaluation des performances des algorithmes avec la permutation de l'algorithme de référence chaque fois. À cause du nombre d'instances associées à chaque configuration (20 instance pour chaque nombre fixe de tâches), les métriques citées sont calculées sur la base de la moyenne des résultats issus de l'ensemble des instances pour une configuration bien déterminée. Alors, les quantités moyennes suivantes $\bar{n}^{f}_{alg_i}$, $\bar{\mathrm{R}}_{alg_i - alg_j}$ et $\bar{\mu}_{alg_i - alg_j}$ sont utilisées pour l'évaluation avec $\left(alg_i, alg_j\right) \in$

$\{Optimal$, NSGA II, $recherche\ coucou\}$ et $alg_i \neq alg_j$. Cette notation permet de différencier entre ces approches : optimale (recherche exhaustive), recherche coucou et algorithme génétique élitiste de tri non dominé sans oublier la non symétrie des métriques considérées. Mais à cause du temps de calcul important consommé par une énumération exhaustive, l'approche optimale n'est appliquée que pour les petites instances. À cause de la dépendance des performances des méthodes d'optimisation du choix adéquat des valeurs des paramètres de configurations, nous avons effectué quelques tests pour déterminer une configuration appropriée en termes des performances. Cette procédure de test est achevée uniquement pour les petites instances dû au coût de calcul lourd. Dans le Tableau 4.2, nous résumons les principaux paramètres ainsi que les valeurs associées aux métaheuristiques proposées. Ces valeurs sont retenues fixes à l'exception lorsque une analyse de sensibilité est conduite. En ce qui concerne le nombre de générations, nous fixons la valeur pour notre algorithme génétique multi-objectif à $10 \times n$ et le temps enregistré (en secondes) pour ce nombre d'itérations est alloué à l'exécution de la recherche coucou pour que la comparaison soit légale entre ces deux approches.

TABLEAU 4.2 Valeurs des paramètres de configuration associées aux métaheuristiques proposées.

Paramètre	Algorithme génétique élitiste de tri non dominé	Recherche coucou
Type de croisement	Croisement à deux points	/
Probabilité de croisement	0.8	/
Type de mutation	Permutation arbitraire	/
Probabilité de mutation	0.1	/
Taille de la population	$2 \times n$	$2 \times n$
Nombre de générations	$10 \times n$	$\equiv$ secondes
Stratégie de sélection	Sélection par tournoi	/
Taille du tournoi	$Round\left(\sqrt{n}\right)$	/
Probabilité d'acceptation	/	0.5

/ : Non applicable

Dans les Tableaux 4.3, 4.4 et 4.5, nous résumons les résultats de simulation obtenus par l'algorithme génétique multi-objectif et la recherche coucou dans l'espace de domination de Pareto. Il est à mentionner que les valeurs moyennes des cardinaux des meilleurs fronts non dominés pour les vingt instances sont calculées par une méthode d'énumération complète uniquement pour les configurations de petites tailles (5, 6 et 7 tâches) à cause des coûts de calcul. Nous observons que l'approche hybride (recherche coucou/recherche à voisinage variable) est plus efficace dans l'obtention des solutions optimales en comparaison avec l'algorithme génétique, qui donne des résultats légèrement inférieurs à la valeur moyenne des cardinaux optimaux. Pour un nombre de tâches supérieure à 10, la valeur moyenne des cardinaux des fronts non dominés générés par la recherche coucou est considérablement réduite en comparaison avec l'algorithme génétique. La moyenne des proportions des ordonnancements trouvés par la recherche coucou et dominés par ceux de l'algorithme génétique croît de zéro pour les petites instances pour atteindre une valeur de l'ordre de 30% pour un nombre de tâches égale à 40. En ce qui concerne la moyenne des distances modifiées normalisées, elle est toujours positive lorsque le front optimal est considéré comme référence car les solutions optimales ne sont jamais dominées. Nous observons également que cette mesure n'est pas signi-

ficative lorsque elle est calculée pour la recherche coucou comparativement aux solutions optimales, ce qui confirme que les fronts obtenus sont très proches l'un de l'autre. Pour un nombre de tâches égale ou supérieur à dix, la moyenne des distances modifiées normalisées $\bar{\mu}_{cuck-nsga}$ exprimant la position relative du meilleur front obtenu par la recherche coucou par rapport au meilleur front obtenu par l'algorithme génétique est négative. Cette remarque indique que les fronts de la recherche coucou sont généralement situés au dessous des fronts de l'algorithme génétique multi-objectif, ce qui confirme la performance supérieure (d'une manière globale) de l'approche coucou hybride. Pour un nombre de tâches inférieur à 10, nous remarquons que les deux moyennes des distances modifiées normalisées ($\bar{\mu}_{cuck-nsga}$ et $\bar{\mu}_{nsga-cuck}$) sont positives car dans certains cas les fronts coucou de certaines instances sont meilleures que les fronts de NSGAII et c'est l'inverse pour les instances restantes. Sous cette situation, la moyenne des distances modifiées normalisées ne reflète pas d'une façon précise la position relative des vingt instances associées à une configuration, ce qui implique la consultation de la métrique de la moyenne des proportions des ordonnancements dominés et la valeur moyenne des cardinaux des meilleurs fronts pour avoir une interprétation plus réaliste.

TABLEAU 4.3 Variation des performances (la moyenne des cardinaux des fronts générés) des approches proposées en fonction du nombre de tâches avec règle de dominance de Pareto ($L_{eff} = 0$, TF = 0.5 et RDD = 0.5).

n	$\bar{n}^f_{opt}$	$\bar{n}^f_{cuck}$	$\bar{n}^f_{nsga}$
5	3.85	3.75	3.6
6	4.7	4.55	4.25
7	5.25	4.85	4.85
8	/	7.15	6.7
9	/	7.25	6.75
10	/	9.7	9.3
15	/	15	17.85
20	/	19.5	27.4
25	/	24.15	36.4
30	/	22.7	46.15
35	/	25.7	54.7
40	/	27.45	63.85

/ : Non applicable

Le Tableau 4.6 présente les variations des performances des deux algorithmes en fonction de la gamme et du facteur d'étroitesse gouvernant la génération des dates échues floues pour un nombre fixe de tâches (20 tâches). Ce tableau montre que la recherche coucou donne un nombre réduit des solutions non dominées dans la majorité des cas à l'exception de la configuration TF = 1 et RDD = 0.2. En outre, la recherche coucou est considérablement plus performante en terme de la moyenne des proportions des ordonnancements dominés qui varie entre 0% à 16% tandis qu'il varie de 20% à 52% pour la moyenne des proportions des solutions de l'algorithme génétique dominées au moins par une solution de la recherche coucou. Pour la métrique de la moyenne des distances modifiées normalisées, dans la majorité des cas de $\bar{\mu}_{cuck-nsga}$, nous avons des valeurs négatives indiquant que la plupart des fronts de la recherche coucou sont localisés au dessous ou très proches des fronts obtenus par l'algorithme génétique. Alors, nous pouvons conclure que l'approche hybride de la recherche coucou est (globalement) plus robuste

TABLEAU 4.4 Variation des performances (la moyenne des proportions des ordonnancements dominés) des approches proposées en fonction du nombre de tâches avec règle de dominance de Pareto ($L_{eff} = 0$, TF = 0.5 et RDD = 0.5).

n	$\bar{R}_{cuck-opt}$	$\bar{R}_{nsga-opt}$	$\bar{R}_{cuck-nsga}$	$\bar{R}_{nsga-cuck}$
5	0	417×10^{-4}	0	417×10^{-4}
6	0	125×10^{-4}	0	125×10^{-4}
7	0	825×10^{-4}	0	825×10^{-4}
8	/	/	0	147×10^{-3}
9	/	/	38×10^{-4}	229×10^{-3}
10	/	/	45×10^{-4}	25×10^{-2}
15	/	/	487×10^{-4}	33×10^{-2}
20	/	/	959×10^{-4}	424×10^{-3}
25	/	/	171×10^{-3}	427×10^{-3}
30	/	/	227×10^{-3}	345×10^{-3}
35	/	/	296×10^{-3}	321×10^{-3}
40	/	/	296×10^{-3}	302×10^{-3}

/ : Non applicable

TABLEAU 4.5 Variation des performances (la moyenne des distances modifiées normalisées) des approches proposées en fonction du nombre de tâches avec règle de dominance de Pareto ($L_{eff} = 0$, TF = 0.5 et RDD = 0.5).

n	$\bar{\mu}_{cuck-opt}$	$\bar{\mu}_{nsga-opt}$	$\bar{\mu}_{cuck-nsga}$	$\bar{\mu}_{nsga-cuck}$
5	0	33×10^{-4}	65×10^{-4}	27×10^{-4}
6	0	2×10^{-6}	192×10^{-4}	2×10^{-6}
7	0	7×10^{-4}	83×10^{-4}	20×10^{-4}
8	/	/	59×10^{-4}	146×10^{-4}
9	/	/	115×10^{-4}	20×10^{-4}
10	/	/	-16×10^{-4}	36×10^{-4}
15	/	/	-6×10^{-4}	24×10^{-4}
20	/	/	-5×10^{-4}	15×10^{-4}
25	/	/	-3×10^{-4}	10×10^{-4}
30	/	/	-7×10^{-4}	10×10^{-4}
35	/	/	-3×10^{-4}	3×10^{-4}
40	/	/	-8×10^{-4}	2×10^{-4}

/ : Non applicable

envers les variations de la gamme et le facteur d'étroitesse des dates échues des tâches.

TABLEAU 4.6 Variation des performances des approches proposées en fonction de la gamme et du facteur d'étroitesse avec règle de dominance de Pareto ($L_{eff} = 0$ et $n = 20$).

TF	RDD	$\bar{n}^f_{cuck}$	$\bar{n}^f_{nsga}$	$\bar{R}_{cuck-nsga}$	$\bar{R}_{nsga-cuck}$	$\bar{\mu}_{cuck-nsga}$	$\bar{\mu}_{nsga-cuck}$
0.2	0.2	18.30	21.35	0.11	0.30	-15×10^{-4}	30×10^{-4}
	0.6	8.80	13.70	0.14	0.24	-4×10^{-4}	20×10^{-4}
	1	5.20	7.25	0.04	0.28	-376×10^{-4}	408×10^{-4}
0.6	0.2	20.45	24.35	0.09	0.42	-11×10^{-4}	13×10^{-4}
	0.6	20.90	25.55	0.11	0.47	-10×10^{-4}	25×10^{-4}
	1	17.95	25.15	0.16	0.31	56×10^{-4}	12×10^{-4}
1	0.2	4.55	3.75	0	0.20	68×10^{-4}	92×10^{-4}
	0.6	14.5	15.15	0.02	0.43	57×10^{-4}	26×10^{-4}
	1	17.75	22.6	0.07	0.52	-10×10^{-4}	19×10^{-4}

Pour étudier l'influence du facteur d'apprentissage L_{eff} sur les métriques de performance des deux approches d'optimisation, nous considérons la configuration avec un nombre de tâches égale à dix où TF et RDD sont les deux égales à 0.5 (voir Tableau 4.7). Les expériences numériques montrent clairement que la valeur moyenne des nombres de solutions non dominées est approximativement les mêmes pour les deux approches, ce qui peut être justifié par le petit nombre de tâches considérées dans ce cas. Par contre, la moyenne des proportions attachée aux solutions coucou dominées est très petite en comparaison avec celle attachée aux solutions de l'algorithme génétique où celle de la recherche coucou varie de 0% à 6% tandis que celle de l'algorithme génétique change de 16% à 24% comme indiqué sur le tableau. De plus, nous observons que la recherche coucou est légèrement meilleure en fonction de la moyenne des distances modifiées normalisées pour les grandes valeurs du facteur d'apprentissage. Pour les cas des petites valeurs, la moyenne des distances modifiées normalisées devient positive même avec la modification du front de référence. Nous interprétons cette tendance par le fait que les fronts générés par l'algorithme génétique et la recherche coucou ne se chevauchent pas régulièrement, ce qui résulte en des valeurs de déviations alternées.

Dans la Figure 4.2, les ordonnancements non dominés obtenus par les approches algorithme génétique et recherche coucou pour une instance de 20 tâches sont représentés dans l'espace des fonctions objectifs. Nous observons que dans la majorité des cas, les ordonnancements générés par la recherche coucou dominent ceux générés par l'algorithme génétique. De plus, la combinaison des différentes règles de dominance permet aux algorithmes d'optimisation proposés de guider la recherche vers des zones plus intéressantes afin d'élarger la couverture des ordonnancements à travers le front de non domination. Par conséquent, une telle combinaison permet d'aboutir à une estimation meilleure des ordonnancements du front de non dominance.

Les métriques de performance des approches proposées avec considération simultanée des critères de dominances Pareto et Lorenz sont illustrées sur les Tableaux 4.8, 4.9 et 4.10 pour les différentes tailles du problème. La valeur moyenne des cardinaux des meilleurs fronts obtenus par l'algorithme génétique et recherche coucou devient plus grande dû à la combinaison des ordonnancements selon les deux critères de dominance. Par conséquent, la taille de ces fronts approche la taille des fronts optimaux pour les instances de petites tailles. Notons que la moyenne des proportions des ordonnancements

TABLEAU 4.7 Variation des performances des approches proposées en fonction du coefficient d'apprentissage avec règle de dominance de Pareto ($n = 10$, TF = 0.5 et RDD = 0.5).

L_{eff}	$\bar{n}^f_{cuck}$	$\bar{n}^f_{nsga}$	$\bar{R}_{cuck-nsga}$	$\bar{R}_{nsga-cuck}$	$\bar{\mu}_{cuck-nsga}$	$\bar{\mu}_{nsga-cuck}$
0	9.7	9.3	45×10^{-4}	2495×10^{-4}	-16×10^{-4}	36×10^{-4}
-0.01	12.8	13.5	697×10^{-4}	1177×10^{-4}	76×10^{-4}	11×10^{-4}
-0.02	12.95	13.15	483×10^{-4}	2040×10^{-4}	-3×10^{-6}	23×10^{-4}
-0.03	13.15	14.25	328×10^{-4}	1767×10^{-4}	13×10^{-4}	57×10^{-4}
-0.04	13.1	13	567×10^{-4}	1914×10^{-4}	8×10^{-4}	14×10^{-4}
-0.05	13	13.65	305×10^{-4}	1565×10^{-4}	109×10^{-4}	17×10^{-4}
-0.06	13.10	12.8	543×10^{-4}	1762×10^{-4}	44×10^{-4}	2×10^{-4}
-0.07	12.8	13.15	334×10^{-4}	1945×10^{-4}	11×10^{-4}	21×10^{-4}
-0.08	12.55	13.05	654×10^{-4}	1657×10^{-4}	193×10^{-4}	14×10^{-4}
-0.09	11.75	11.85	253×10^{-4}	2075×10^{-4}	276×10^{-4}	16×10^{-4}
-0.1	12.35	12.45	295×10^{-4}	2167×10^{-4}	186×10^{-4}	43×10^{-4}

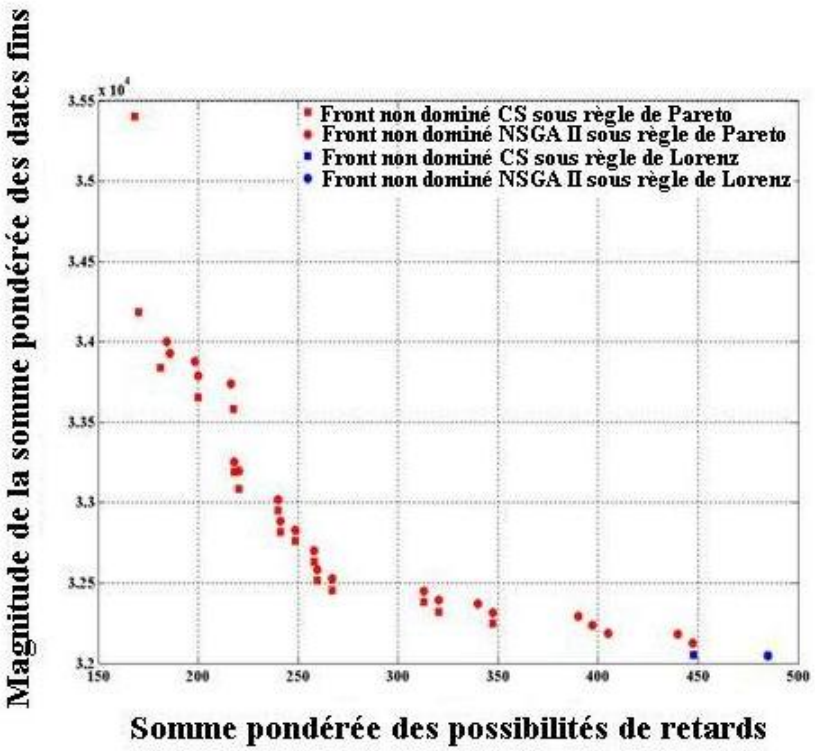

FIGURE 4.2 Représentation des meilleurs fronts obtenus par les deux approches avec considération simultanée des règles de dominance Pareto et Lorenz (L'instance numero 17 de la configuration n=20).

dominés dans le cas de la recherche coucou est considérablement réduite de même pour le cas de l'algorithme génétique multi-objectif pour lequel cette mesure est légèrement améliorée en comparaison avec l'application unique du critère de Pareto comme donné sur les Tableaux 4.3, 4.4 et 4.5. Cette remarque confirme l'avantage de l'utilisation conjointe de différents types de règles de dominance avec les mécanismes d'intensifi-

cation de recherche dans les méthodes d'optimisation multi-objectif. Pour la métrique de la distance modifiée normalisée, les valeurs moyennes calculées pour les douze configurations décèlent la difficulté de déduire une conclusion car on a une alternance des signes positifs et négatifs ce qui rend les résultats globaux de cette mesure très difficiles à interpréter ou non conclusifs.

TABLEAU 4.8 Variation des performances (la moyenne des cardinaux des fronts générés) des approches proposées en fonction du nombre de tâches avec considération simultanée des règles de dominance de Pareto et Lorenz ($L_{eff} = 0$, TF = 0.5 et RDD = 0.5).

n	$\bar{n}^f_{opt}$	$\bar{n}^f_{cuck}$	$\bar{n}^f_{nsga}$
5	3.85	3.8	3.6
6	4.7	4.55	4.25
7	5.25	5	4.85
8	/	7.3	6.75
9	/	7.25	6.8
10	/	10.10	9.5
15	/	16.7	18.15
20	/	22.7	29.35
25	/	26.8	38.4
30	/	26.35	49.7
35	/	28.7	57.35
40	/	30.85	66.95

/ : Non applicable

En se concentrant uniquement sur les valeurs moyennes des métriques de performance pendant la comparaison, la différence entre le comportement des métaheuristiques proposées ne peut pas être détectée effectivement. Dans le but de remédier tel obstacle, le test d'hypothèse peut être utilisé pour inférer quelques conclusions concernant une population à travers l'examination d'un échantillon défini. Une stratégie largement utilisée pour la comparaison des performances des approches d'optimisation consiste en le comptage de nombre des cas pour lesquels une méthode de référence est la gagnante [DERRAC et collab., 2011]. Dans ce contexte, nous considérons l'échantillon formé par l'intégration de toutes les instances utilisées dans les expérimentations du Tableau 4.3, ce qui donne un totale de 240 instances. Si la performance de la recherche coucou proposée est égale à celle de l'algorithme génétique, nous attendons d'avoir approximativement la moitié des instances gagnée par la première approche et la moitié gagnée par la deuxième. Autrement, un décalage sera détecté dans le nombre de victoires au profit de l'algorithme ayant une meilleure performance. Donc, nous pouvons distinguer trois cas comme suit. La recherche coucou est la gagnante dans le cas de $\left((\mu_{cuck-nsga} < 0) \wedge (\mu_{nsga-cuck} > 0)\right)$, l'algorithme génétique élitiste de tri non dominé est le gagnant dans le cas de $\left((\mu_{cuck-nsga} > 0) \wedge (\mu_{nsga-cuck} < 0)\right)$ et l'équivalence des deux algorithmes dans le cas de $\left((\mu_{cuck-nsga} \leq 0) \wedge (\mu_{nsga-cuck} \leq 0)\right) \vee \left((\mu_{cuck-nsga} \geq 0) \wedge (\mu_{nsga-cuck} \geq 0)\right)$. Nous notons la victoire de la recherche coucou par +1, la victoire de l'algorithme génétique élitiste de tri non dominé par -1 et la situation neutre par 0. Il est à mentionner que le cas neutre i.e $\left((\mu_{cuck-nsga} \leq 0) \wedge (\mu_{nsga-cuck} \leq 0)\right) \vee \left((\mu_{cuck-nsga} \geq 0) \wedge (\mu_{nsga-cuck} \geq 0)\right)$ n'est pas compté mais partitionné d'une façon équitable entre les deux algorithmes considérés [FONG et collab., 2003]. Le test des signes est adopté pour évaluer l'hypothèse

TABLEAU 4.9 Variation des performances (la moyenne des proportions des ordonnancements dominés) des approches proposées en fonction du nombre de tâches avec considération simultanée des règles de dominance de Pareto et Lorenz (L_{eff} = 0, TF = 0.5 et RDD = 0.5).

n	$\bar{R}_{cuck-opt}$	$\bar{R}_{nsga-opt}$	$\bar{R}_{cuck-nsga}$	$\bar{R}_{nsga-cuck}$
5	0	417×10^{-4}	0	417×10^{-4}
6	0	125×10^{-4}	0	125×10^{-4}
7	0	825×10^{-4}	0	825×10^{-4}
8	/	/	0	104×10^{-3}
9	/	/	38×10^{-4}	182×10^{-3}
10	/	/	45×10^{-4}	195×10^{-3}
15	/	/	417×10^{-4}	278×10^{-3}
20	/	/	828×10^{-4}	394×10^{-3}
25	/	/	156×10^{-3}	406×10^{-3}
30	/	/	195×10^{-3}	346×10^{-3}
35	/	/	267×10^{-3}	326×10^{-3}
40	/	/	263×10^{-3}	298×10^{-3}

/ : Non applicable

TABLEAU 4.10 Variation des performances (la moyenne des distances modifiées normalisées) des approches proposées en fonction du nombre de tâches avec considération simultanée des règles de dominance de Pareto et Lorenz (L_{eff} = 0, TF = 0.5 et RDD = 0.5).

n	$\bar{\mu}_{cuck-opt}$	$\bar{\mu}_{nsga-opt}$	$\bar{\mu}_{cuck-nsga}$	$\bar{\mu}_{nsga-cuck}$
5	0	61×10^{-4}	65×10^{-4}	33×10^{-4}
6	0	2×10^{-6}	192×10^{-4}	2×10^{-6}
7	0	6×10^{-4}	85×10^{-4}	8×10^{-4}
8	/	/	61×10^{-4}	141×10^{-4}
9	/	/	128×10^{-4}	12×10^{-4}
10	/	/	-5×10^{-4}	20×10^{-4}
15	/	/	-4×10^{-4}	12×10^{-4}
20	/	/	-3×10^{-4}	12×10^{-4}
25	/	/	-2×10^{-4}	7×10^{-4}
30	/	/	-3×10^{-4}	7×10^{-4}
35	/	/	-3×10^{-4}	3×10^{-4}
40	/	/	-1×10^{-4}	2×10^{-4}

/ : Non applicable

nulle que la médiane des valeurs gagnantes est égale à zéro contre l'hypothèse alternative que la médiane des valeurs gagnantes est supérieure à zéro. Les hypothèses nulle et alternative sont définies comme suit :

$$\begin{cases} H_0 : La\ médiane\ de\ nombre\ des\ victoires\ de\ la\ recherche\ coucou\ = 0 \\ \\ H_1 : La\ médiane\ de\ nombre\ des\ victoires\ de\ la\ recherche\ coucou\ > 0 \end{cases}$$

À cause de la taille de notre échantillon (240), il est possible d'utiliser une approximation normale pour la distribution de probabilité binomiale dans la détermination de la région du rejet du test des signes [WACKERLY et collab., 2008]. Comme les valeurs faibles de p constituent une évidence forte contre l'hypothèse nulle, le test rejette l'hypothèse nulle en faveur de l'hypothèse alternative à un niveau de signification de 1% car la valeur-p obtenue est inférieure au niveau de signification ($1.3327 \times 10^{-9} < 1\%$). Donc, il est très probable que la mesure de performance de l'approche recherche coucou en fonction de la distance modifiée normalisée soit supérieure à la mesure de performance de l'algorithme génétique multi-objectif.

4.6 Conclusion

Dans ce chapitre, nous avons abordé deux objectifs notamment la modélisation et la résolution d'un nouveau problème d'ordonnancement bi-objectif à une machine avec effet d'apprentissage à base de position. Dues aux contraintes des environnements de nos jours, l'imprécision des données a été considérée pendant les procédures de modélisation et d'optimisation à travers l'introduction de la théorie de possibilité. Alors, nous avons développé des formulations généralisées pour la somme pondérée des dates de fin d'exécution ainsi que la somme pondérée des possibilités des retards des tâches, adoptées comme des fonctions objectifs. Des cas particuliers du problème résultant ont été analysés et il a été prouvé que la version générale est NP-difficile au sens fort. Dans le but de résoudre ce problème, nous avons proposé deux métaheuristiques comme outils de résolution. La première approche est la fameuse approche d'algorithme génétique élitiste de tri non dominé avec quelques modifications pour l'adapter aux spécificités du problème étudié. L'algorithme génétique a été adopté comme méthode benchmark pour évaluer les performances de la recherche coucou combinée avec une recherche à voisinage variable. La recherche coucou hybride proposée a réussi à trouver un nombre réduit des solutions non dominées meilleures que celles obtenues par l'algorithme génétique multi-objectif. Dans le but d'améliorer la qualité des solutions obtenues, les solutions délivrées par des schémas de dominance différentes (Pareto et Lorenz) ont été combinées dans le même ensemble et les mesures de performance ont été recalculées. Cette intégration a confirmé qu'une efficacité améliorée peut être atteinte en comparaison avec un processus d'optimisation basé uniquement sur une règle de dominance.

Finalement, nous pensons que la pertinence des méthodologies menées dans cette recherche peut solliciter des futures contributions. Ceci permettra par la suite de mettre clairement en relief l'efficacité de la théorie floue pour la description des quantités incertaines ou vagues dans le cadre de stratégies d'ordonnancement.

4.7 Références

ABBASBANDY, S. et T. HAJJARI. 2009, «A new approach for ranking of trapezoidal fuzzy numbers», *Computers and Mathematics with Applications*, vol. 57, nº 3, p. 413–419. 118

ADAMU, M. O. et A. O. ADEWUMI. 2014, «A survey of single machine scheduling to minimize weighted number of tardy jobs», *Journal of Industrial and Management Optimization*, vol. 10, nº 1, p. 219–241. 113

AHMADIZAR, F. et L. HOSSEINI. 2011, «Single-machine scheduling with a position-based learning effect and fuzzy processing times», *International Journal of Advanced Manufacturing Technology*, vol. 56, nº 5, p. 693–698. 115

AHMADIZAR, F. et L. HOSSEINI. 2013, «Minimizing makespan in a single-machine scheduling problem with a learning effect and fuzzy processing times», *International Journal of Advanced Manufacturing Technology*, vol. 65, nº 1, p. 581–587. 115

AL-TURKI, U., C. FEDJKI et A. ANDIJANI. 2001, «Tabu search for a class of single-machine scheduling problems», *Computers and Operations Research*, vol. 28, nº 12, p. 1223–1230. 111

ALIDAEE, B. et N. K. WOMER. 1999, «Scheduling with time dependent processing times : Review and extensions», *The Journal of the Operational Research Society*, vol. 50, nº 7, p. 711–720. 114

AYDILEK, A., H. AYDILEK et A. ALLAHVERDI. 2017, «Algorithms for minimizing the number of tardy jobs for reducing production cost with uncertain processing times», *Applied Mathematical Modelling*, vol. 45, p. 982–996. 115

BALASUBBAREDDY, M., S. SIVANAGARAJU et C. V. SURESH. 2015, «Multi-objective optimization in the presence of practical constraints using non-dominated sorting hybrid cuckoo search algorithm», *Engineering Science and Technology, an International Journal*, vol. 18, nº 4, p. 603–615. 131

BENTRCIA, T. et L.-H. MOUSS. 2015, *Fuzzy Modeling of the Single Machine Scheduling Problems including the Learning Effect,*, chap. 14, in Talbi, E-G. and Yalaoui, F. and Amodeo, L. (eds.), Metaheuristics for Production Systems, Series : Operations Research/Computer Science Interfaces Series, Springer. 115, 126

BENTRCIA, T., L.-H. MOUSS, N.-K. MOUSS, F. YALAOUI et L. BENYOUCEF. 2015, «Evaluation of optimality in the fuzzy single machine scheduling problem including discounted costs», *International Journal of Advanced Manufacturing Technology*, vol. 80, nº 5, p. 1369–1385. 114

BISKUP, D. 1999, «Single-machine scheduling with learning considerations», *European Journal of Operational Research*, vol. 115, nº 1, p. 173–178. 123

BISKUP, D. 2008, «A state-of-the-art review on scheduling with learning effects», *European Journal of Operational Research*, vol. 188, nº 2, p. 315–329. 114

BLUM, C., J. PUCHINGER, G. R. RAIDL et A. ROLI. 2011, «Hybrid metaheuristics in combinatorial optimization : A survey», *Applied Soft Computing*, vol. 11, nº 6, p. 4135–4151. 132

BOJADZIEV, G. et M. BOJADZIEV. 1996, *Fuzzy Sets, Fuzzy Logic, Applications*, World Scientific. 118

BOUSSAÏD, I., J. LEPAGNOT et P. SIARRY. 2013, «A survey on optimization metaheuristics», *Information Sciences*, vol. 237, p. 82–117. 127

BRUCKER, P. et M. Y. KOVALYOV. 1996, «Single machine batch scheduling to minimize the weighted number of late jobs», *Mathematical Methods of Operations Research*, vol. 43, nº 1, p. 1–8. 112

BUCKLEY, J. J. 2005, *Fuzzy Probabilities, New Approach and Applications*, Springer Press. 123

CAI, X. Q., X. WU et X. ZHOU. 2014, *Optimal Stochastic Scheduling*, Springer Press. 114

CHANAS, S. 2001, «On the interval approximation of a fuzzy number», *Fuzzy Sets and Systems*, vol. 122, nº 2, p. 353–356. 116

CHEN, C.-L. et R. L. BULFIN. 1993, «Complexity of single machine, multi-criteria scheduling problems», *European Journal of Operational Research*, vol. 70, nº 1, p. 115–125. 127

CHENG, T. C. E., C. T. NG et J. J. YUAN. 2006, «Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs», *Theoretical Computer Science*, vol. 362, nº 1-3, p. 273–281. 112

COLEY, D. A. 1999, *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific. 128

DAUZÈRE-PÉRÈS, S. 1995, «Minimizing late jobs in the general one machine scheduling problem», *European Journal of Operational Research*, vol. 81, nº 1, p. 134–142. 112

DAUZÈRE-PÉRÈS, S. et L. MÖNCH. 2013, «Scheduling jobs on a single batch processing machine with incompatible job families and weighted number of tardy jobs objective», *Computers and Operations Research*, vol. 40, nº 5, p. 1224–1233. 113

DAUZÈRE-PÉRÈS, S. et M. SEVAUX. 2003, «Using lagrangean relaxation to minimize the weighted number of late jobs on a single machine», *Naval Research Logistics*, vol. 50, nº 3, p. 273–288. 112

DEB, K. 2001, *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley. 129

DEB, K., A. PRATAP, S. AGARWAL et T. MEYARIVAN. 2002, «A fast and elitist multiobjective genetic algorithm : Nsga-ii», *IEEE Transactions on Evolutionary Computation*, vol. 6, nº 2, p. 182–197. 128

DERRAC, J., S. GARCÍA, D. MOLINA et F. HERRERA. 2011, «A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms», *Swarm and Evolutionary Computation*, vol. 1, nº 1, p. 3–18. 145

DIAMOND, P. et P. KLOEDEN. 1994, *Metric Spaces of Fuzzy Sets : Theory and Applications*, World Scienific Press. 116

DUBOIS, D., L. FOULLOY, G. MAURIS et H. PRADE. 2004, «Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities», *Reliable Computing*, vol. 10, nº 4, p. 273–297. 120

DUBOIS, D. et H. PRADE. 1988, *Possibility theory an approach to computerized processing of uncertainty*, Plenum Press. 120

DUGARDIN, F., F. YALAOUI et L. AMODEO. 2010, «New multi-objective method to solve reentrant hybrid flow shop scheduling problem», *European Journal of Operational Research*, vol. 203, nº 1, p. 22–31. 135, 138

EBRAHIMNEJAD, A. 2016, «A new method for solving fuzzy transportation problem with lr flat fuzzy numbers», *Information Sciences*, vol. 357, p. 108–124. 122

FONG, D. Y. T., C. W. KWAN, K. F. LAM et K. S. L. LAM. 2003, «Use of the sign test for the median in the presence of ties», *The American Statistician*, vol. 57, nº 4, p. 237–240. 145

GUPTA, A., A. KUMAR et M. SHARMA. 2013, «Applications of fuzzy linear programming with generalized lr flat fuzzy parameters», *Fuzzy Information and Engineering*, vol. 5, nº 4, p. 475–492. 122

GUPTA, S. K. et J. KYPARISIS. 1987, «Single machine scheduling research», *Omega International Journal of Management Sciences*, vol. 15, nº 3, p. 207–227. 111

HANOUN, S., D. CREIGHTON et S. NAHAVANDI. 2014, «A hybrid cuckoo search and variable neighborhood descent for single and multiobjective scheduling problems», *International Journal of Advanced Manufacturing Technology*, vol. 75, nº 9, p. 1501–1516. 131, 133

HANSEN, P. et N. MLADENOVIĆ. 2001, «Variable neighborhood search : Principles and applications», *European Journal of Operational Research*, vol. 130, nº 3, p. 449–467. 133

HOOGEVEEN, H. 2005, «Multicriteria scheduling», *European Journal of Operational Research*, vol. 167, nº 3, p. 592–623. 111

JAIN, R. 1976, «Decision-making in the presence of fuzzy variables», *IEEE Transactions on Systems Man and Cybernetics*, vol. 6, nº 10, p. 698–703. 118

JANIAK, A. et R. RUDEK. 2010, «A note on a makespan minimization problem with a multi-ability learning effect», *Omega The International Journal of Management Science*, vol. 38, nº 3-4, p. 213–217. 113

JIN, Y. et J. BRANKE. 2005, «Evolutionary optimization in uncertain environments - a survey», *IEEE Transactions on Evolutionary Computation*, vol. 9, nº 3, p. 303–317. 114

KAHRAMAN, C. et M. YAVUZ. 2010, *Production Engineering and Management under Fuzziness*, Springer Press. 116

KARP, R. M. 1972, *Reducibility among combinatorial problems,*, in R. E. Miller and J. W. Thatcher and J. D. Bohlinger (eds.), Complexity of Computer Computations, Plenum Press. 112

KASPERSKI, A. 2005, «A possibilstic approach to sequencing problems with fuzzy parameters», *Fuzzy Sets and Systems*, vol. 150, nº 1, p. 77–86. 114

KASPERSKI, A. et P. ZIEŁIŃSKI. 2011, «Possibilistic minmax regret sequencing problems with fuzzy parameters», *IEEE Transactions on Fuzzy Systems*, vol. 19, nº 6, p. 1072–1082. 114

KIM, Y. D. 1993, «A new branch and bound algorithm for minimizing mean tardiness in two-machine flowshops», *Computers and Operations Research*, vol. 20, nº 4, p. 391–401. 136

KLIR, G. et B. YUAN. 1995, *Fuzzy Sets and Fuzzy Logic : Theory and Applications*, Prentice Hall Press. 119

LACOMME, P., C. PRINS et M. SEVAUX. 2006, «A genetic algorithm for a bi-objective capacitated arc routing problem», *Computers and Operations Research*, vol. 33, nº 12, p. 3473–3493. 138

LAI, P.-J. et W.-C. LEE. 2011, «Single-machine scheduling with general sum-of-processing-time-based and position-based learning effects», *Omega The International Journal of Management Science*, vol. 39, nº 5, p. 467–471. 113

LAWLER, E. L. et J. M. MOORE. 1969, «A functional equation and its application to resource allocation and sequencing problems», *Management Science*, vol. 16, nº 1, p. 77–84. 112

LI, J., X. YUAN, E. S. LEE et D. XU. 2011, «Setting due dates to minimize the total weighted possibilistic mean value of the weighted earliness-tardiness costs on a single machine», *Computers and Mathematics with Applications*, vol. 62, nº 1, p. 4126–4139. 111

LIN, B. M. T. 2007, «Complexity results for single-machine scheduling with positional learning effects», *Journal of the Operational Research Society*, vol. 58, nº 8, p. 1099–1102. 126

MANTEGNA, R. 1994, «Fast, accurate algorithm for numerical simulation of lévy stable stochastic processes», *Physical Review E*, vol. 49, nº 5, p. 4677–4683. 131

M'HALLAH, R. et R. L. BULFIN. 2003, «Minimizing the weighted number of tardy jobs on a single machine», *European Journal of Operational Research*, vol. 145, nº 1, p. 45–56. 112

M'HALLAH, R. et R. L. BULFIN. 2007, «Minimizing the weighted number of tardy jobs on a single machine with release dates», *European Journal of Operational Research*, vol. 176, nº 2, p. 727–744. 113

MOGHADDAM, A., F. YALAOUI et L. AMODEO. 2015, «Efficient meta-heuristics based on various dominance criteria for asingle-machine bi-criteria scheduling problem with rejection», *Journal of Manufacturing Systems*, vol. 34, p. 12–22. 135

MOHAMAD, A. B., A. M. ZAIN et N. E. N. BAZIN. 2014, «Cuckoo search algorithm for optimization problems - a literature review and its applications», *Applied Artificial Intelligence : An International Journal*, vol. 28, nº 5, p. 419–448. 131

MOSHIEOV, G. 2001, «Scheduling problems with a learning effect», *European Journal of Operational Research*, vol. 132, nº 3, p. 687–693. 126

MUNDT, A. et T. WICH. 2007, *Single Machine Scheduling with Tardiness Involved Objectives : A Survey*, mémoire de maîtrise, Université de Linköpings, Sweden. 113

ÖZELKAN, E. C. et L. DUCKSTEIN. 1999, «Optimal fuzzy counterparts of scheduling rules», *European Journal of Operational Research*, vol. 113, nº 3, p. 593–609. 118

PAVLYUKEVICH, I. 2007, «Lévy flights, non-local search and simulated annealing», *Journal of Computational Physics*, vol. 226, nº 1, p. 1830–1844. 131

PINEDO, M. L. 2009, *Planning and Scheduling in Manufacturing and Services*, Springer Press. 111

POTTS, C. N. et L. N. VAN WASSENHOVE. 1988, «Algorithms for scheduling a single machine to minimize the weighted number of late jobs», *Management Science*, vol. 34, nº 7, p. 843–858. 112

QIAN, J. et G. STEINER. 2013, «Fast algorithms for scheduling with learning effects and time-dependent processing times on a single machine», *European Journal of Operational Research*, vol. 225, nº 3, p. 547–551. 114

RASTI-BARZOKI, M. et S. R. HEJAZI. 2013, «Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains», *European Journal of Operational Research*, vol. 228, nº 2, p. 345–357. 113

RIISE, A. 2002, «Comparing genetic algorithms and tabu search for multi-objective optimization», dans *International Federation of Operational Research Studies Conference IFORS*, Edinburhg, UK. 138

RUDEK, R. 2017, «The single machine total weighted completion time scheduling problem with the sum-of-processing time based models : Strongly np-hard», *Applied Mathematical Modelling*, vol. 50, p. 314–332. 113

SAIDI MEHRABAD, M. et A. PAHLAVANI. 2009, «A fuzzy multi-objective programming for scheduling of weighted jobs on a single machine», *The International Journal of Advanced Manufacturing Technology*, vol. 45, nº 1-2, p. 122–139. 136

SCALAS, E. et M. LECCARDI. 2005, «Comparison of three algorithms for lèvy noise generation», dans *Fifth EUROMECH Nonlinear Dynamics Conference*, Eindhoven, The Netherlands. 131

SEVAUX, M. et S. DAUZÈRE-PÉRÈS. 2003, «Genetic algorithms to minimize the weighted number of late jobs on a single machine», *European Journal of Operational Research*, vol. 151, nº 2, p. 296–306. 112

SHABTAY, D. et G. STEINER. 2011, «A bicriteria approach to minimize the total weighted number of tardy jobs with convex controllable processing times and assignable due dates», *Journal of Scheduling*, vol. 14, nº 5, p. 455–469. 113

SINGH, G. et I. S. AHUJA. 2012, «Just-in-time manufacturing : literature review and directions», *International Journal of Business Continuity and Risk Management*, vol. 3, nº 1, p. 57–98. 111

TZAFESTAS, S. et A. VENETSANOPOULOS. 1994, *Fuzzy Reasoning in Information, Decision and Control Systems*, Kluwer Press. 117

WACKERLY, D. D., W. MENDENHALL et R. L. SCHEAFFER. 2008, *Mathematical Statistics with Applications*, Brooks/Cole Cengage Learning Press. 147

WANG, J.-B., Y.-B. WU et P. JI. 2012, «Erratum : A revision of some single-machine and m-machine flowshop scheduling problems with learning considerations», *Information Sciences*, vol. 190, p. 227–232. 113

WONGA, B. K. et V. S. LAI. 2011, «A survey of the application of fuzzy set theory in production and operations management : 1998-2009», *International Journal of Production Economics*, vol. 129, nº 1, p. 157–168. 136

WRIGHT, T. P. 1936, «Factors affecting the cost of airplanes», *Journal of the Aeronautical Sciences*, vol. 3, nº 4, p. 122–128. 113

WU, C.-C. et W.-C. LEE. 2008, «Single-machine scheduling problems with a learning effect», *Applied Mathematical Modelling*, vol. 32, nº 7, p. 1191–1197. 113

YAGER, R. R. 1981, «A procedure for ordering fuzzy subsets of the unit interval», *Information Sciences*, vol. 24, nº 2, p. 143–161. 118

YANG, X. S. et S. DEB. 2009, «Cuckoo search via lévy flights», dans *World congress on nature and biologically inspired computing*, Coimbatore, India. 130

YANG, X. S. et S. DEB. 2010, «Engineering optimisation by cuckoo search», *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, nº 4, p. 330–343. 130

YIN, Y., M. LIU, J. HAO et M. ZHOU. 2012, «Single-machine scheduling with job-position-dependent learning and time-dependent deterioration», *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans*, vol. 42, nº 1, p. 192–200. 113

ZADEH, L. A. 1965, «Fuzzy sets», *Information and Control*, vol. 8, nº 3, p. 338–353. 116

ZADEH, L. A. 1968, «Probability measures of fuzzy events», *Journal of Mathematical Analysis and Applications*, vol. 23, nº 2, p. 421–427. 119

ZADEH, L. A. 1978, «Fuzzy sets as a basis for a theory of possibility», *Fuzzy Sets and Systems*, vol. 1, p. 3–28. 119

ZITZLER, E., L. THIELE, M. LAUMANNS, C. M. FONSECA et V. G. DA FONSECA. 2003, «Performance assessment of multiobjective optimizers : an analysis and review», *IEEE Transactions on Evolutionary Computation*, vol. 7, nº 2, p. 117–132. 137

Conclusion générale

« *Sometimes life hits you in the head with a brick. Don't lose faith* »

Steve Jobs

Synthèse

Au cours de cette thèse, la recherche bibliographique réalisée relativement aux problèmes d'ordonnancement à une machine a permis de déceler le gap significatif existant entre les développements théoriques et les recommandations pratiques notamment en termes du traitement de certaines contraintes. En effet, les approches d'ordonnancement étant souvent élaborées sur une base déterministe pour laquelle l'imprécision des données due à des facteurs internes/externes est ignorée ou pas prise convenablement en compte. Par conséquent, une solution optimale au sens classique peut s'avérer invalide ou obsolète suite à la présence des sources d'incertitudes dans l'environnement. Le contexte de la thèse s'est donc naturellement focalisé sur le support des paramètres incertains au sein des approches d'ordonnancement pendant les phases de modélisation ainsi que de résolution.

Les travaux effectués dans cette thèse ont permis, du point de vue mathématique, non seulement de revisiter quelques opérations arithmétiques définies sur l'espace des nombres flous mais aussi d'apporter des extensions intéressantes au niveau des propriétés structurelles de ces nombres. Nos contributions liées à l'ordonnancement sont localisées au croisement de deux champs thématiques. Le premier fait le point sur la modélisation des problèmes d'ordonnancement à une machine dans l'incertain en intégrant d'autres contraintes alors que le deuxième champ concerne le developpement d'outils d'optimisation et l'évaluation de leurs performances. Dans le premier champ, nous avons représenté les paramètres d'ordonnancement (durées opératoires, dates d'échéance,...etc.) par des nombres flous ayant une allure bien spécifique et nous avons considéré la notion d'optimalité selon plusieurs interprétations : notion d'optimalité graduelle pour un ordonnancement fixe, notion d'optimalité ordinaire dans le cadre d'un problème d'optimisation combinatoire mono-objectif et notion d'optimalité ordinaire dans le cadre d'un problème d'optimisation combinatoire multi-objectif. Les résultats issus de l'analyse de complexité basée sur ces concepts d'optimalité nous ont permis de démontrer que malgré l'obtention de quelques problèmes d'ordonnancement ayant une complexité polynômiale, la résolution des modèles résultants sans hypothèses simplificatrices sur les paramètres d'ordonnancement est loin d'être un but trivial. Dans le second champ de recherche englobant les outils d'optimisation et leurs caractéristiques, nous avons développé suivant cet axe plusieurs approches de résolution en particulier des métaheuristiques à base de trajectoire et de population, qui ont fait preuve dans le domaine d'ordonnancement. Nous avons consolidé l'efficacité de ces algorithmes en proposant des schémas d'intégration divers où la motivation vient du constat que l'hybridation entre différentes approches d'optimisation peut conduire à une amélioration de la qualité des solutions obtenues. En particulier, nous avons étudié l'utilité d'adopter un algorithme génétique comme phase optionnelle dans la recherche par motifs. De plus, nous avons proposé des versions multi-objectives des approches algorithme génétique et recherche coucou destinées aux problèmes d'optimisation discrets avec analyse de l'influence de deux critères de dominance. Finalement, nous avons approuvé ces approches de résolution à travers des tests statistiques mettant en évidence une validation plus fine des résultats en comparaison avec les déductions grossières basées sur les valeurs moyennes des mesures de performance.

L'exploitation des modèles et des approches proposés nous a permis d'aboutir aux remarques générales suivantes :

- La possibilité de préserver la validité de la règle de Smith pondérée sous paramètres incertains si la linéarité de la fonction de classement des nombres flous est assurée ;

- La possibilité de préserver la validité de la règle de Smith pondérée sous paramètres incertains et effet d'apprentissage si la linéarité de la fonction de classement et l'agréabilité floue des poids sont vérifiées ;
- L'aspect NP-difficile au sens fort d'un problème d'ordonnancement à une machine avec paramètres incertains découle directement de cet aspect pour le problème d'ordonnancement associé avec paramètres réels dû à la possibilité d'élaboration d'une transformation polynômiale entre les instances des classes des deux problèmes ;
- La nécessité d'entreprendre des améliorations au niveau des mesures de performance utilisées dans le cadre des approches d'optimisation comme ces mesures souffrent d'inconsistances sous certaines situations ou offrent des estimations grossières.

Il est à souligner que les idées implémentées dans cette thèse s'appliquent sans grande difficulté à d'autres variantes des problèmes d'ordonnancement à une machine telles que le cas avec des temps de préparation incertains. La résolution de tel problème passe par les mêmes étapes abordées dans notre étude en adoptant les extensions floues des opérations arithmétiques.

Perspectives

Après l'analyse des problèmes d'ordonnancement à une machine avec paramètres incertains et l'adoption de la théorie de possibilité pour la modélisation d'aspects incertains, nous estimons que les résultats obtenus peuvent donner naissance à de futures perspectives concernant les problèmes d'ordonnancement flous. Comme la présence des périodes d'indisponibilité de la machine due par exemple à une défaillance n'est pas évoquée dans cette thèse, il serait très intéressant de considérer ce type de contraintes plus en détail via l'utilisation de la théorie des probabilités floues. Une autre ouverture de ce travail peut être l'exploitation des méthodes exactes pour les problèmes d'ordonnancement flous, ce qui nécessite la reformulation des notions associées afin de justifier l'efficacité des approches métaheuristiques proposées. Enfin, un dernier axe de recherche concerne la généralisation des idées implémentées pour les systèmes de production de type flow-shop, job-shop et pourquoi pas aller même vers le cas open-shop.

Résumé

Les systèmes de production actuels sont soumis à une forte pression de l'environnement. Cette situation peut conduire à l'altération des stratégies conventionnelles d'ordonnancement. En effet, plusieurs paramètres et contraintes relatifs aux tâches nécessitent une reformulation à la base d'outils adéquats. C'est dans ce contexte que s'inscrit ce travail de thèse dans lequel nous proposons plusieurs approches intégrées pour la modélisation du problème d'ordonnancement à une machine avec prise en compte simultanément d'une variété de sources d'incertitude. En outre, nous analysons la complexité des formulations résultantes et nous implémentons des règles d'optimalité globale ainsi que des métaheuristiques pour élaborer des solutions exactes ou approchées. L'évaluation des métriques de performance atteste l'efficacité des approches de résolution proposées.

Mots clefs : Ordonnancement à une machine, théorie floue, règle d'optimalité et méthodes d'optimisation

Abstract

Current production systems are subject to a strong pressure from their environments. Such situation can lead to the alteration of conventional scheduling strategies. Indeed, several parameters and constraints relative to scheduled tasks require a reformulation on the basis of adequate tools. In this framework, we propose several integrated approaches for modeling the problem of single machine scheduling with simultaneous consideration of a variety of sources of uncertainty. In addition, we analyze the complexity of the resulting formulations and implement global optimality rules in addition to metaheuristics in order to obtain exact or approximate solutions. The performance metrics evaluation validates the effeciency of the proposed resolution approaches.

Keywords : Single machine scheduling, fuzzy theory, optimality rules and optimization methods

ملخص

تخضع نظم الإنتاج الحالية لضغط هائل من البيئة المحيطة. يمكن أن تؤدي هذه الوضعية إلى تغيير في مناهج الجدولة التقليدية للإنتاج. العديد من المتغيرات والقيود على ارض الواقع مرتبطة بعمليات الانتاج ما يتطلب إعادة الصياغة بناء على الأدوات الملائمة. يندرج العمل المقدم في هذه الأطروحة في هذا السياق أين اقترحنا العديد من المناهج المتكاملة لنمذجة مسائل جدولة الانتاج بآلة واحدة مع الأخذ بعين الاعتبار لعدة مصادر من الارتياب إضافة إلى تحليل درجة تعقيد النماذج الناتجة. كما طورنا قواعد للأمثلية العامة وخوارزميات تجويد مستوحاة من الطبيعة للحصول على حلول دقيقة أو تقريبية. وفي الختام قمنا بتقييم مقاييس الأداء لتأكيد مدى فعالية مناهج الحل المقترحة.

الكلمات المفتاحية: جدولة آلة واحدة، نظرية الغموض، قواعد الأمثلية و طرق التجويد