Projet de fin d'étude
informatique

Université Lille 1

Méthode de résolution pour le problème de planification des tâches multi-objectif

INRIA - Lille Nord Europe

Auteur:
Emilie Allart

Tuteurs:
Sophie Jacquin
Laetitia Jourdan

27 janvier 2016

Remerciements

Je remercie ....

Contents

Introduction

Dans le cadre de ma dernière année de master, j'ai effectué mon projet de fin d'étude à INRIA Lille Nord Europe dans l'équipe Dolphin, afin de mettre en place une nouvelle méthode de résolution pour le problème de planification des tâches multi-objectif, encadrée par Sophie Jacquin (INRIA) et Laetitia Jourdan (INRIA/CRIStAL). Je vais donc dans un premier temps, présenter INRIA et l'équipe Dolphin, puis Paradiseo et enfin le plan de mon rapport. TODO revoir

INRIA est un établissement public de recherche à caractère scientifique et technologique. Il a été créé en 1967 et a pour mission de produire une recherche d'excellence dans les champs informatiques et mathématiques des sciences du numérique et de garantir l'impact de cette recherche. Il couvre l'ensemble du spectre des recherches au coeur de ces domaines d'activités, et intervient sur les questions en lien avec le numérique, posées par les autres sciences et par les acteurs économiques et sociétaux. INRIA rassemble 1677 chercheurs de l'institut et 1772 universitaires ou chercheurs d'autres organismes, il compte plus de 4500 articles publiés en 2013 et est à l'origine de plus de 110 start-ups. L'institut est organisé en 8 centres :Bordeaux, Grenoble, Lille, Nancy, Rennes, Rocquencourt, Saclay et Sophia-Antipolis.

INRIA Lille - Nord Europe comporte 16 équipes de recherche et possède plusieurs partenariats tels que Lille1, Lille2, Lille3, Centrale Lille, le CNRS et le CWI. La stratégie du centre est de développer autour de la métropole lilloise un pôle d'excellence de rayonnement international (en priorité vers l'Europe du nord) et à fort impact local. Pour se faire, l'institut s'appuie sur des thématiques de recherche ambitieuses dans le domaine des sciences du numérique; l'intelligence des données et les systèmes logiciels adaptatifs, plus précisément :

- Internet des données et Internet des objets
- Couplage perception/action pour l'interaction homme-machine
- Modèle patient personnalisé dynamique
- Génie logiciel pour les systèmes éternels

L'équipe Dolphin (Discrete multi-objective Optimization for Large-scale Problems with Hybrid dIstributed techNiques) entretient plusieurs relations industrielles et internationales (EDF-GDF, bioinformatique, DHL, Univ. Montréal, ...) De nombreux secteurs de l'industrie sont concernés par des problèmes d'optimisation à grande échelle et complexes mettant en jeux des coûts financiers très importants et pour lesquels les décisions doivent être prises de façon optimales. Face à des applications qui nécessitent la résolution de problèmes de taille sans cesse croissante et ce dans des délais de plus en plus court, voire en temps réel, seule la mise en oeuvre conjointe des méthodes avancées issues de l'optimisation combinatoire en Recherche Opérationnelle, de la décision en IA

et de l'utilisation du Parallélisme et de la distribution permettrait d'aboutir à des solutions satisfaisantes.

L'équipe Dolphin a pour objectif la modélisation et la résolution parallèle de problèmes d'optimisation combinatoire (multi-objectifs) de grandes tailles. Des méthodes parallèles coopératives efficaces sont développées à partir de l'analyse de la structure du problème traité. Les problèmes ciblés appartiennent aussi bien à la classe des problèmes génériques (ordonnancement flow-shop, élaboration de tournées, etc...) que des problèmes industriels issue de la logistique, du transport, de l'énergie et de la bioinformatique.

Le problème de planification des tâches (Job Shop Scheduling Problem) consiste à planifier le traitement d'un certain nombre de tâches par les machines d'un l'atelier. L'objectif le plus couramment etudié est de trouver le planning qui permette d'achever l'ensemble des tâches au plus tôt. Néanmoins, avec une politique du juste à temps, il est nécessaire de considérer simultanément, comme second critère, le respect maximal de dates d'échéance afin d'éviter les retards de livraison et les coûts de stockage. L'équipe Dolphin développe de nouvelles méthodes d'optimisation combinatoire multi-objectif pour ce problème, en particulier des métaheuristiques. Pour tester la qualité des méthodes proposées, il nous faut nous comparer aux méthodes existantes.

Chapter 1

Position du problème

1.1 JobShop

Enoncé du problème
cd sophie La planification de tâches (ou job shop scheduling) est un problème NP-complet. Il s'agit d'organiser N tâches au mieux en respectant des contraintes d'avance α et de retard β ainsi qu'une disponibilité des ressources r.
Pour calculer le coût d'un ordonnancement, on somme sur chaque tâche le calcul de l'avance ou du retard. Le retard étant la différence entre la complitude de la tâche i, notée C_i, c'est à dire le temps à laquelle elle est achevée, et la due date, notée d_i, le temps à laquelle elle aurait dûe être finie, le tout pondérée par le facteur β spécifique à cette tâche. Et inversement pour l'avance avec un facteur de pondération α. Les facteur α et β sont spécifique à chaque tâche car il est plus ou moins important selon la tâche de la finir dans les temps impartis. Cependant, il faut prendre garde à respecter la disponibilité r, une tâche ne peut pas être effectuée avant son temps r.

Modélisation mathématiques
En découle donc la formule ci-dessous :

$$\sum_{i=0}^{N} \max(\beta_i(C_i - d_i), \alpha_i(d_i - C_i)) \tag{1.1}$$

1.2 Optimisation multi-objectif

L'optimisation multi-objectif permet de résoudre des problèmes d'optimisation présentant plus d'un objectif. La formule ci-dessous représente un problème de minimisation multi-objectif.

$$Minimize\ F(x) = \{f_1(x), f_2(x), ...f_M(x)\}\ x \in \Omega \tag{1.2}$$

Ω est l'espace de solution, x est une solution, $f_i(x)$ est la $i^{ième}$ fonction objective et M donne le nombre d'objectif. Dans la plupart des cas, ces objectifs sont conflictuels, ce qui signifie que l'optimisation d'un objectif entraine la détérioration d'autres. Ici, comme le représente la formule ci-dessous, nos objectifs à minimiser sont le retard et l'avance, deux objectifs opposés.

$$objectif1 : earliness = \sum_{i=0}^{N} \alpha_i \max((d_i - C_i), 0) \tag{1.3}$$

$$objectif2 : tardiness = \sum_{i=0}^{N} \beta_i \max((C_i - d_i), 0) \tag{1.4}$$

Le principe d'optimalité Pareto pour l'optimisation mutli-objectif est basé sur la relation de dominance Pareto. En supposant que les fonctions objectives sont à minimiser, une solution est dite dominante par rapport à une solution y si $\forall i \in 1, ..., M, f_i(x) \leq f_i(y)$ et $\exists i \in 1, ..., M f_i(x) < f_i(y)$. Une solution x^* est Pareto optimale si elle n'est pas dominée par aucune solution de l'espace de solution. L'ensemble des solutions Pareto optimale est appelé l'ensemble Pareto optimal, et l'ensemble des vecteurs objectifs correspondant représente le front Pareto. Le but de la résolution de problèmes d'optimisation multi-objectifs (MOPs) concernant la Pareto optimalité, est de trouver l'ensemble Pareto optimal.

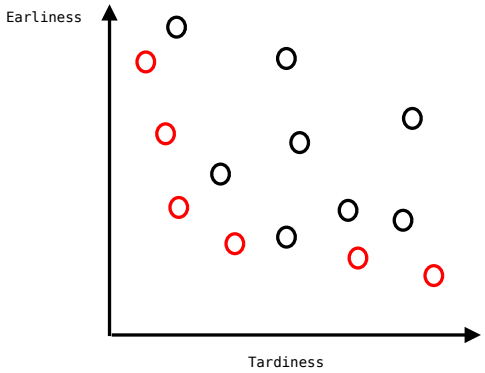

Figure 1.1: Front Pareto pour le problème de minimisation de retard et d'avance

TODO : détail de dominance Pareto + schéma cf Sophie's thesis

Chapter 2

Méthodes de résolution

2.1 Principe générale

Algorithme évolutionnaire
Les algorithmes évolutionnaires sont inspirés du concept de sélection naturelle élaboré par Charles Darwin, d'ailleurs le vocabulaire employé découle de cette théorie.

- La fonction objectif F est appelé fonction de fitness
- Les points de l'espace de recherche Ω (pour notre cas se sont des ordonnancement) sont appelés des individus (ou chromosomes)
- L'ensemble des individus est une population
- Un tour de boucle principale de l'algorithme correspond à une génération

L'espace de recherche est déterminé par le problème (la fonction objectif).

Initialisation
L'initialisation consiste à échantilloner le plus uniformément possible l'espace de recherche Ω. Pour le cas de ce projet, on génère aléatoirement des ordonnancements de job.
Avec l'aide d'opérateurs génétiques, de nouveaux individus sont créés à partir des individus sélectionnés au préalable. On distingue les opérateurs de croisement (ou crossover) et les opérateurs de mutation. Ces transformations sont stochastiques, leur application nécessite des tirages aléatoires. Ainsi, tout opérateur doit respecter des conditions d'ergodicité (tous points de l'espace doit être ateignable).

Mutation
La mutation permet de parcourir l'ensemble de l'espace, de se diversifier. Elle est appliquée avec une probabilité P_m sur chaque individu et permet de modifier chacun aléatoirement. Les détails des opérateurs choisis pour ce projet sont détaillés dans le chapitre suivant.

Crossover
Ce croisement reflète la "transmission d'information génétique" des parents. Si deux chromosomes parents sont performants, on peut considérer le fait qu'un de leur enfant pourrait hériter des parties qui les rendent performants et être ainsi meilleurs encore. Le croisement est appliqué avec une certaine probabilité P_c, ainsi il est possible de conserver certain chromosome optimaux. On est dans le cadre d'une intensification (on garde les traits importants d'un individu). Il existe également une grande quantité d'opérateurs de crossover, ceux utilisés pour ce projet seront expliqués un peu plus loin.

Selection / Remplacement
L'intégration de l'idée darwinienne s'intègre sous la forme de deux étape dans l'algorithme : la sélection et le remplacement. Il existe deux type de procédures, la sélection déterministe lors de laquelle on sélectionne les meilleurs individus selon la fitness. Les individus moins performants sont totalement éliminés, seul les meilleurs restent, on parle d'élitisme. Et la sélection stochastique, le but est là aussi de sélectionner les meilleurs individus, mais de manière stochastique. Le meilleur n'est pas choisi à tous les coups.

- Le tirage de roulette consiste à donner à chaque individu une probabilité d'être sélectionné proportionnelle à sa performance.
- La selection par le rang consiste à faire une sélection en utilisant une roulette dont les portions sont proportionnelles au rang des individus.
- La selection par tournoi consiste à tirer T individus uniformément dans la population et à sélectionner le meilleur.

Les algorithmes évolutionnaires reposent donc sur 3 principes :

- Assignation d'une fitness : Le rôle de ce processus est de privilégier les solutions proches du front Pareto. Il doit permettre d'assurer la convergence de l'ensemble solution fourni par l'algorithme vers le front Pareto.
- Assignation d'une valeur de diversification : Cette stratégie a pour but d'éviter la convergence de la population vers un petit sous-ensemble de solutions du front Pareto et donc d'assurer la bonne diversité de l'ensemble solution.
- Élitisme : La notion d'élitisme a pour but de préserver et d'utiliser les solutions élites qui sont les solutions non dominées trouvées par l'algorithme.

TODO : Développer ces 3 points rapidement.

Plusieurs algorithmes évolutionnaires ont été proposés au cours du temps, on peut distinguer ceux utilisant directement le principe de dominance pour la procédure d'assignation de fitness et ceux utilisant des indicateurs de qualité.

2.2 NSGA II

Les algorithmes utilisant le principe de dominance sont ceux utilisant une procédure d'assignation de fitness basée sur la relation de dominance. Cette fitness permet de classer les individus en fonction de leur proximité avec le front Pareto mais elle ne garantit pas la construction d'une approximation du front ayant une bonne diversification. C'est pourquoi les individus se voient également assigner une mesure de diversification. Ainsi les individus sont classés en considérant dans un premier temps leur fitness puis leur mesure de diversification. Voici, par exemple, la description du NSGA II (Non-dominated Sorting Genetic Algorithm II)

1. Assignation de fitness : Profondeur de dominance.
2. Mesure de diversité : Plus proche voisin en utilisant la crowding distance.
3. Élitisme : Lors de l'étape de remplacement les individus parents et enfants sont classés selon leurs fitness, puis à fitness égale selon la valeur de leur apport en diversification, les meilleurs individus de ce classement sont ceux gardés pour constituer la génération suivante.

.

2.3 IBEA

Plus récemment des méthodologies basées sur l'utilisation d'un ou plusieurs indicateurs de qualité permettant d'évaluer l'apport d'une solution d'un point de vue convergence et/ou diversification ont été proposées. Ces méthodes n'utilisent pas forcément les deux processus d'assignation car l'indicateur de performance peut être choisi par rapport au critère que l'on souhaite évaluer (convergence et/ou diversification). Parmis ces algorithmes nous comptons l'IBEA (Indicator-Based Evolutionary Algorithm) définit comme suit :

1. Assignation de fitness : Soit un indicateur binaire I, la fonction fitness se calcule comme suit : où est un réel à fixer. Cette fonction mesure la perte en qualité si la solution x est retirée de la poulation P.
2. Mesure de diversité : Aucune.
3. Elitisme : Remplacement élitiste. Les individus ayant les meilleurs fitness parmis les parents et les enfants sont sélectionnés pour appartenir à la génération suivante.

Chapter 3

Application au problème

3.1 Etat de l'art

3.1.1 Choix des opérateurs de mutation

Après avoir étudier la littérature, nous avons découvert l'article de S. Kedad-Sidhoum et F. Sourd qui représente les données de façon originale. Il repose sur le principe d'idle time, c'est-à-dire qu'il est possible d'avoir une pause entre deux tâches pour répondre au mieux aux objectifs. Ainsi, une solution n'est pas représentée comme une suite de tâches mais comme une succession de blocs. Un bloc correspond à un ensembles de tâches successives ne nécessitant pas de pause. Chaque bloc, un temps de démarrage s est donné avec une sous-séquence de tâches ordonnées et adjacentes après s. L'atout de cet algorithme est de permettre une recherche rapide du voisinage. Pour cela, il permet premièrement, de faire un ensemble de changement au sein d'un bloc. Et deuxièmement, il utilise un opérateur temporel, qui permet de bouger en avant ou en arrière un bloc.
Pour résumer la modélisation mathématique. Pour chaque tâche J_i, nous utiliserons la function de coût

$$f_i(t) = \max(\beta_i(t - d_i), \alpha_i(d_i - t))$$

, qui indique le coût de J_i selon son temps de complétion t. Par exemple, le coût de l'ordonancement π, noté $c(\pi)$, est égal à

$$sum_{i=0}^{n} f_i(C_i)$$

. Les tâches d'un ordonnancement π sont partitionnées dans des blocs qui sont des séquences maximales de tâches ordonnancées sans pause. Posons $b(\pi)$ le nombre de blocs. De ce fait, les blocs d'un ordonnancement donné sont notés B_1, B_2, ..., B_b. $|B|$ correspond au nombre de jobs dans le bloc B et ${J_i}^B$,...,${J_{|B|}}^B$ identifie le job de B indéxé dans l'ordre de π. De même, nous utilisons les notations ${f_i}^B$, ${p_i}^B$ et ${C_i}^B$.

Swap

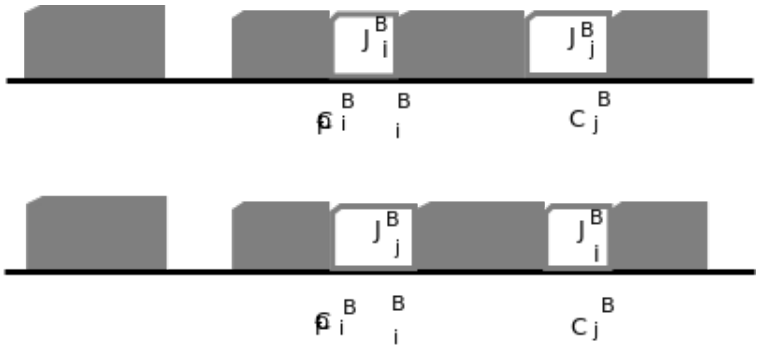

Figure 3.1: Swap de ${J_i}^B$ et ${J_j}^B$

Avec un ordonnancement π, le voisinage de swap consiste à échanger la position de deux jobs ${J_i}^B$ et ${J_j}^B$ d'un même bloc. Sans entrer dans les détails, qui nécessiterait une longue démonstration, le voisinage complet de π est exploré en temps $O(n^2)$.

Extract-and-reinsert
Pour un ordonnancement π, le voisinage est obtenu en enlevant un job J de sa place dans π et en l'insérant devant ou derrière dans le même bloc. Les tâches du bloc sont décalées pour pouvoir insérer ce job à la position souhaitée. Pour un job ${J_i}^B$ d'un bloc B de π et une position $i \leq j \leq |B|$ telle que $i \neq j$, notons $[v_{ij}^B$ l'ordonnancement dérivé de π en décalant ${J_i}^B$ à une position derrière j dans B. Si on compare le voisinage de ${v_{ij}}^B$ et ${v_{i,j-1}}^B$ pour $j < |B|$ et $i < j$, on peut

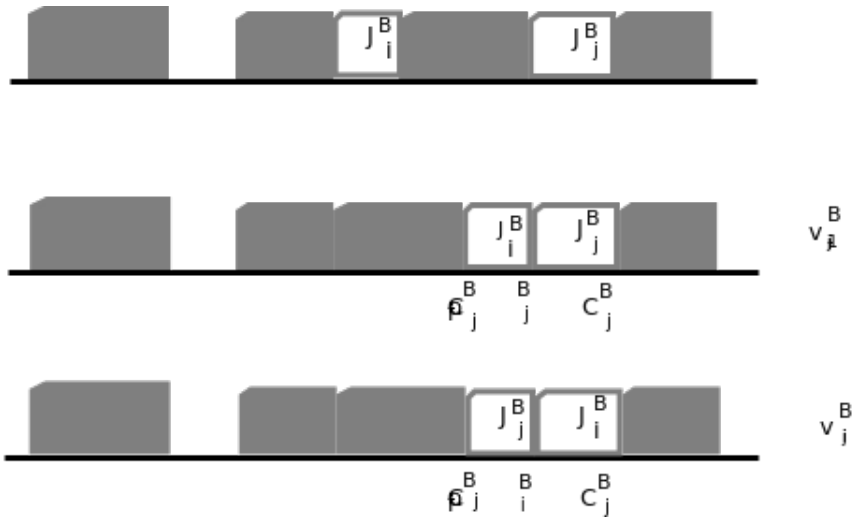

Figure 3.2: Extrait et reinsert ${J_i}^B$ à la position j-1 et j

voir que la seule différence est la position de ${J_i}^B$ et ${J_j}^B$. Donc, on a :

$$c(v_{ij}^B) - c(v_{i,j-1}^B) = {f_i}^B({C_j}^B) + {f_j}^B({C_j}^B - {p_i}^B) - {f_i}^B({C_j}^B - {p_j}^B) - {f_j}^B({C_j}^B)$$

Ce qui signifie que $c(v_{ij}^B)$ peut être dérivé de $c(v_{i,j-1}^B)$ en $O(1)$. On a $O(|B|)$ insertions possible pour tous les jobs de B, donc le voisinage associé à B peut être exploré en temps $O(|B|^2)$. C'est pourquoi, le voisinage de tout bloc est exploré en temps $O(n^2)$.

subblock shift
Swap et extract-and-reinsert, permettent de modifier l'ordonnancement à l'intérieur d'un bloc. Subblock shift lui est un opérateur temporel, il permet de décaler les blocs en avant ou en arrière en réduisant les temps de pause, tout en conservant une solution réalisable. Son mécanisme est de sélectionner un job aléatoirement ${J_i}^B$. Si ce job se trouve en début de bloc, on a la possibilité de réduire son temps de pause I_B et ainsi fusionner avec le bloc de devant si le décalage $t = I_B$. Ou d'augmenter I_B en réduisant I_{B+1} pour ne pas modifier l'ordonnancement du bloc suivant et de fusionner avec le bloc suivant si $t = I_{B+1}$.

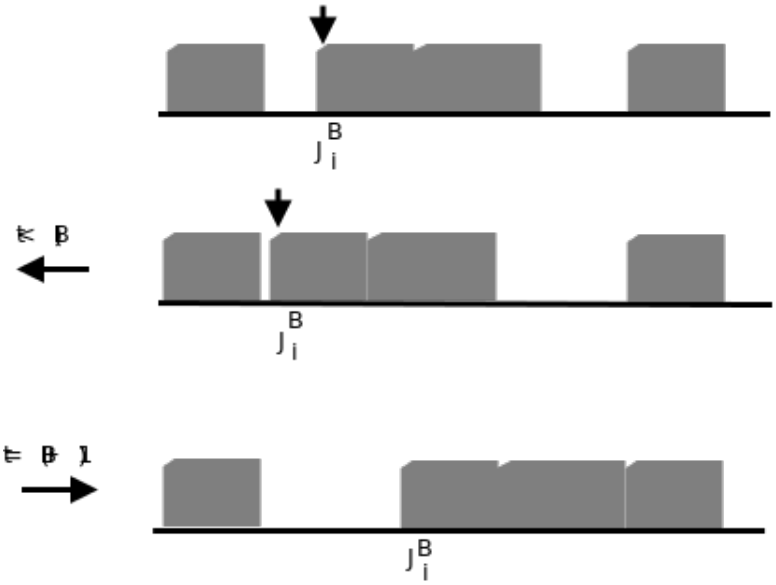

Figure 3.3: Décalage du bloc B

Dans les autres cas, le bloc est divisé en deux sous-bloc en intégrant un temps de pause t au sous-bloc de droite avac $t \leq I_{B+1}$, ce nouveau bloc peut éventuellement fusionner avec le bloc suivant.

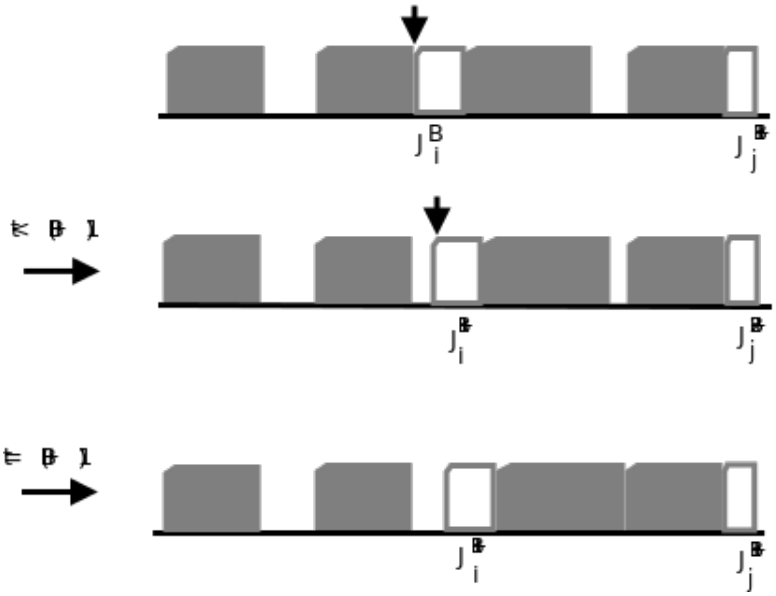

Figure 3.4: Décalage d'un sous-bloc de B

3.1.2 Choix des opérateurs de crossover

Après analyse de la littérature, nous avons relevé 3 opérateurs intéressant. Le 2-point qui est un classique mais reste le meilleur. L'utilisation deMasque et LOX. Au niveau performance, on sait que $2 - point > LOX > Masque > PMX > CX$. TODO à développer

2-point
Hybrid genetic algo with dominance prop

LOX
An adaptative GA for solving single ... linear order crossover

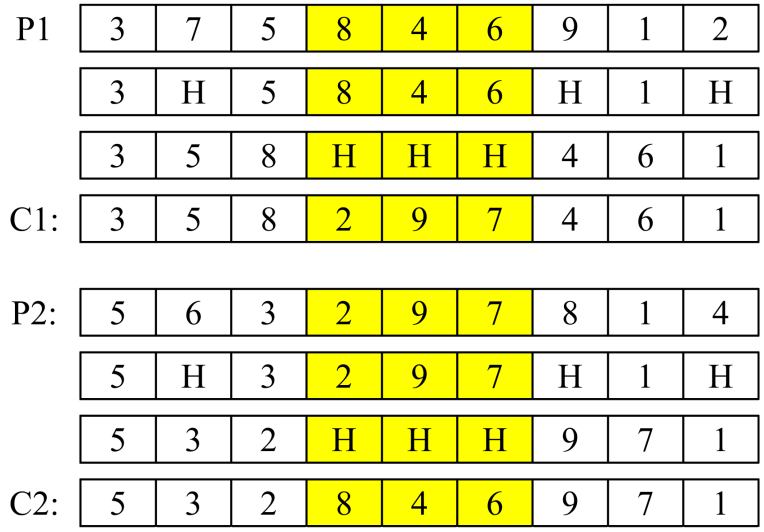

Figure 3.5: Déroulement du crossover lox

onversion. Crossover operator progressively constructs near-optimum solutions from good partial solutions and recombines the chromosome's characteristics. This operator in the scheduling domain must be such that infeasible solutions are avoided. Several crossover operators including partially mapped crossover (PMX), linear order crossover (LOX) and cycle crossover (CX) were experimented with. In initial tests, we found that the crossover LOX works the best for this problem. The main advantage of LOX is that it can preserve as much as possible the relative positions between the genes and the absolute positions relative to the extremities of the chromosome [39]. Therefore, two cutting points of the parents are chosen randomly, and the entries in the cross-section of the first parent are removed from the second parent leaving some "holes". Then, the holes are slid from the extremities toward the center until they reach the

crossover section. Finally, the crosssection is substituted with that of the corresponding parent to obtain the children [39]. An example of the performance of the LOX operator is shown in Fig. 4.

Masque
JIT single mach scheduling

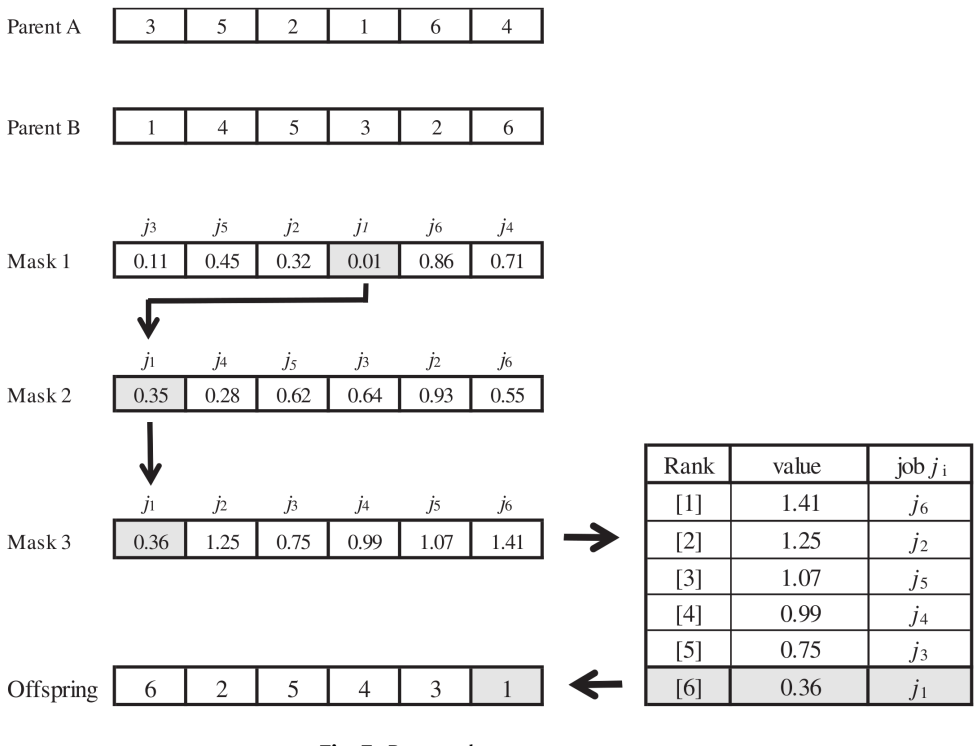

Figure 3.6: Déroulement du crossover masque

Mis en avant par l'article *A genetic algorithm for JIT single machine scheduling with preemption and machine idle time* , voici l'opérateur de crossover masque. Dans celui-ci, deux nouveaux chromosomes *mask1* et *mask2* avec pour chaque gène une valeur une distribution aléatoire comprise entre $[0, 1]$ et un nombre de gènes égal à N. Comme nous pouvons le voir sur la figure ci-dessus, le *mask1* est associé au parent 1 et le deuxième au parentB. Finalement, le *mask3* correspond à la somme des valeurs des gènes qui correspondent au même nombre dans le *mask1* et le *mask2*. Dans notre exemple, le gène associé au nombre 1 dans le *mask1* vaut 0.01 et celui associé à ce même nombre dans les *mask2* vaut 0.35. Donc le premier gène du *mask3*, quiest associé au nombre 1, est égal à la somme de ces deux valeurs donc 0.36. Ensuite, le *mask3* est trié selon la valeur de ses gènes, donnant en sortie le nombre qui lui est associé. Le nombre avec la plus grande valeur associée dans le *mask3* a la priorité pour se placer dans l'enfant. Ici, le nombre 1 est en $6^{ème}$ position.

3.2 Implémentation

3.2.1 Paradiseo

Toutes les méthodes présentées ont été implémentées à l'aide du logiciel Paradiseo. Paradiseo est un logiciel open-source qui fournit des framework permettant d'implémenter facilement des métaheuristiques. Expliquer le but + Schéma paradiseo

3.2.2 Modélisation

Schéma de la structure des données

Chapter 4

Protocole

4.1 Jeu de données

jeu de données de Sourd

4.2 Comparaison

Hypervolume et epsilon de Kruskall Wallis influence des différents opérateurs mut(0.1 .. 1.0) cross(0.1 .. 1.0) Sur différents jeu de données

Chapter 5

Résultats et Discussion

graphe et analyse

	2-point		Mask		LOX		BestC		Mutation		N	
	eps	hyp	eps	hyp	eps	hyp	eps	hyp	eps	hyp	eps	hyp
20	>0.5	=	>0.3	=	.	.	.	.	.	.	.	.
100	.	.	.	.	.	.	.	.	.	.	.	.
200	.	.	.	.	.	.	.	.	.	.	.	.

Table 5.1: Tableau comparatif pour NSGA

	2-point		Mask		LOX		BestC		Mutation		N	
	eps	hyp	eps	hyp	eps	hyp	eps	hyp	eps	hyp	eps	hyp
20	.	.	.	.	.	.	.	.	.	.	.	.
100	.	.	.	.	.	.	.	.	.	.	.	.
200	.	.	.	.	.	.	.	.	.	.	.	.

Table 5.2: Tableau Comparatif pour IBEA

Conclusion

Les compétences acquises
Grâce au travail effectué à INRIA, j'ai pu acquérir plusieurs compétences:

Les apports personnelles

Les apports à l'entreprise
Ajout de fonctionnalité à Paradiseo Comparatif d'algo

Glossaire

INRIA Institut National de Recherche en Informatique et en Automatique

CRIStal Centre de Recherche en Informatique, Signal et Automatique de Lille

CNRS Centre National de la Recherche Scientifique

CWI Centrum Wiskunde Informatica, organisme de recherche d'Amsterdam

Références

- Biblio TODO
- http://www.inria.fr/
- http://dolphin.lille.inria.fr/
- http://vision.ucsd.edu/ sagarwal/nsga2.pdf
- http://www.tik.ee.ethz.ch/sop/publicationListFiles/zk2004a.pdf
- http://paradiseo.gforge.inria.fr/index.php
- https://hal.inria.fr/inria-00376770/document
- A. Liefooghe, L. Jourdan, and E-G. Talbi. A software framework based on a conceptual unified model for evolutionary multiobjective optimization : Paradiseo-moeo. European Journal of Operational Research, 209(2) :104–112, 2011.
- S.Jacquin. Hybridation des métaheuristiques et de la programmation dynamique pour les problèmes d'optimisation mono et multi-objectif : Application à la production d'énergie.