

ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET MATHÉMATIQUES

Année 2011

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

Techniques d'ordonnancement d'atelier et de
fournées basées sur la programmation par
contraintes

THÈSE DE DOCTORAT

Discipline : Informatique

Spécialité : Informatique

*Présentée
et soutenue publiquement par*

Arnaud MALAPERT

Le 9 septembre 2011

à l'École Nationale Supérieure des Techniques Industrielles et des Mines de Nantes

Devant le jury ci-dessous :

Président	: Gilles Pesant, Professeur	École Polytechniques de Montréal
Rapporteur	: Christian Artigues, Chargé de recherche	LAAS-CNRS
Examineurs	: Christelle Guéret, Maître-assistant	École des Mines de Nantes
	André Langevin, Professeur	École Polytechnique de Montréal

Directeur de thèse :

Narendra Jussien, Professeur, École des Mines de Nantes

Co-encadrant :

Louis-Martin Rousseau, Professeur, École Polytechnique de Montréal

Équipe d'accueil 1 : Équipe TASC, INRIA, LINA UMR CNRS 6241

Laboratoire d'accueil 1 : Département Informatique de l'École des Mines de Nantes
la Chantrerie, 4, rue Alfred Kastler, 44 307 Nantes

Équipe d'accueil 2 : Équipe Quosseça

Laboratoire d'accueil 2 : Département de mathématiques et génie industriel
École Polytechnique de Montréal

succursale Centre-ville, H3C 3A7 Montréal, Canada, C.P. 6079

Techniques d'ordonnancement d'atelier et de fournées
basées sur la programmation par contraintes

Shop and batch scheduling with constraints

Arnaud MALAPERT



Université de Nantes

À mon père, *Éric*.

Nous sommes des nains juchés sur des épaules de géants. Nous voyons ainsi davantage et plus loin qu'eux, non parce que notre vue est plus aiguë ou notre taille plus haute, mais parce qu'ils nous portent en l'air et nous élèvent de toute leur hauteur gigantesque.

Bernard DE CHARTRES

— ... *La voie la plus féconde pour la recherche est celle qui implique la plus grande résistance à l'extérieur et la moindre à l'intérieur. L'échec doit être conçu comme une obligation de poursuivre ses efforts et de concentrer sa volonté. Si les efforts accomplis sont déjà considérables, l'échec ne doit être que plus joyeux ! C'est que notre barre a donné sur le couvercle de fer du trésor ! Et la victoire sur des difficultés accrues est d'autant plus appréciables que l'échec aura développé la puissance de l'exécutant à proportion des difficultés affrontées.*

— *Fameux ! Très fort ! approuva Nerjine du haut de son tas de rondins.*

— *Ce qui ne veut pas dire qu'il ne faut jamais renoncer à poursuivre un effort. Notre barre peut avoir heurté une pierre. Si on en est convaincu et que le milieu soit par trop hostile ou les moyens insuffisants, on est parfaitement en droit de renoncer au but. Mais il faut que cette renonciation soit très rigoureusement fondée.*

Alexandre SOLJENITSYNE, *Le Premier Cercle*

Remerciements

Tous mes remerciements vont à Monsieur Gilles Pesant, professeur à l'École Polytechnique de Montréal, et à Monsieur Christian Artigues, chargé de recherche au LAAS-CNRS, qui m'ont fait l'honneur d'être les rapporteurs de ces travaux de thèse. J'adresse autant de remerciements à mes encadrants, Narendra, Louis-Martin, Christelle et André pour leur soutien, leurs conseils, leur patience et la liberté qu'ils m'ont laissée durant ces années.

Ce travail a été l'occasion de nombreuses rencontres enrichissantes de part et d'autre de l'Atlantique. À Nantes, je tiens à remercier les membres des équipes SLP et TASC pour leur disponibilité et la gentillesse dont ils ont fait preuve au début de ma thèse en répondant à toutes mes interrogations sans jamais perdre patience. À Montréal, je remercie tous les membres de l'équipe Quosséça et du CIRRELT qui ont facilité mon acclimatation et fait découvrir une nouvelle culture. À travers le monde, mais surtout à Cork, je remercie toutes les personnes avec qui j'ai eu l'occasion de collaborer sur des aspects scientifiques ou le développement de choco.

Je n'oublie pas non plus tous les professeurs, du collège à l'université, qui ont éveillé ma curiosité et mon goût pour les sciences. Je tiens également à remercier toutes les personnes qui m'ont donné ma chance en stage, mais aussi celles qui m'ont permis de concilier des études et un emploi.

Pour finir, je remercie mes parents qui ont toujours cru en moi, ma famille et mes amis pour les bons moments passés ensemble, mais aussi pour leur présence dans les moments difficiles. Enfin, cette expérience n'aurait pas été la même sans la présence et l'amour de ma compagne Marie dans toutes ces pérégrinations en y apportant son enthousiasme et sa joie de vivre. Je remercie notre fils Raphaël qui m'a insufflé l'élan pour clore cette étape de ma vie et nous entraîne vers de nouvelles aventures ...

Merci à toi lecteur de t'intéresser à cette modeste contribution.

Table des matières

Remerciements	iii
Table des matières	v
1 Introduction	1
1.1 Objectifs	2
1.2 Contribution	3
1.3 Organisation du document	4
1.4 Diffusion scientifique	4
I Contexte de l'étude	
2 Programmation par contraintes	9
2.1 Modélisation d'un problème par des contraintes	10
2.1.1 Problème de satisfaction de contraintes	11
2.1.2 Problème d'optimisation sous contraintes	12
2.1.3 Exemple : le problème des n reines	12
2.2 Méthodes de résolution	13
2.2.1 Algorithmes simples de recherche	13
2.2.2 Filtrage et propagation des contraintes	13
2.2.3 Contraintes globales	15
2.2.4 Algorithmes avancés de recherche	15
2.3 Stratégies de recherche	16
2.3.1 Méthodes de séparation	16
2.3.2 Heuristiques de sélection	17
2.4 Procédures d'optimisation	19
2.4.1 Procédure bottom-up	20
2.4.2 Procédure top-down	20
2.4.3 Procédure dichotomic-bottom-up	21
2.4.4 Procédures incomplètes	21
2.5 Solveurs de contraintes	23
2.6 Conclusion	24
3 Ordonnancement sous contraintes	25
3.1 Tâches	26
3.2 Contraintes temporelles	27
3.2.1 Contraintes de précédence	27
3.2.2 Contraintes de disponibilité et d'échéance	27
3.2.3 Contraintes de disjonction	27
3.2.4 Problèmes temporels	28
3.3 Contraintes de partage de ressource	28
3.3.1 Contrainte disjonctive	29
3.3.2 Contrainte cumulative	30

3.4	Modèle disjonctif	31
3.5	Placement sous contraintes	32
3.5.1	Placement en une dimension	32
3.5.2	Placement en deux dimensions	33
3.6	Conclusion	34
4	Ordonnancement d'atelier	35
4.1	Définition des problèmes d'atelier	35
4.1.1	Définition des problèmes de base	36
4.1.2	Variantes et classification de Lawler	36
4.1.3	Applications	37
4.2	Résultats de complexité	38
4.2.1	Problèmes à deux machines	38
4.2.2	Problèmes à trois machines : la frontière	38
4.2.3	Cas général	38
4.2.4	Conclusion	39
4.3	État de l'art	39
4.3.1	Jeux d'instances	39
4.3.2	Méthodes approchées	39
4.3.3	Méthodes Exactes	41
4.4	Orientation de nos travaux	42
5	Ordonnancement d'une machine à traitement par fournées	43
5.1	Définition des problèmes de fournées	43
5.1.1	Définition du problème de base	44
5.1.2	Variantes et classification de Lawler	45
5.1.3	Applications	45
5.2	Résultats de complexité	45
5.2.1	Modèle en parallèle	46
5.2.2	Modèle en série	46
5.2.3	Conclusion	46
5.3	État de l'art	47
5.3.1	Jeu d'instances	47
5.3.2	Méthodes de résolution	47
5.4	Orientation de nos travaux	48
 II Contribution		
6	Ordonnancement d'atelier au plus tôt	51
6.1	Modèle en ordonnancement sous contraintes	52
6.1.1	Contraintes disjonctives	52
6.1.2	Contraintes temporelles	52
6.1.3	Contraintes supplémentaires	53
6.1.4	Stratégie de branchement	54
6.2	Algorithme de résolution	55
6.2.1	Principe de l'algorithme	55
6.2.2	Solution initiale	57
6.2.3	Techniques de redémarrage	57
6.3	Évaluations expérimentales	58
6.3.1	Réglage des paramètres	59
6.3.2	Analyse de sensibilité	60
6.3.3	Comparaison avec d'autres approches	64
6.4	Conclusion	67

7	Heuristiques d'arbitrage dans un atelier	69
7.1	Modèles en ordonnancement sous contraintes	70
7.1.1	Contraintes temporelles	70
7.1.2	Stratégie de branchement	70
7.1.3	Notes sur les modèles	71
7.2	Algorithme de résolution	72
7.3	Évaluations expérimentales	72
7.3.1	Problèmes d'open-shop	72
7.3.2	Problèmes de job-shop	74
7.4	Conclusion	75
8	Minimisation du retard algébrique maximal sur une machine à traitement par four-	77
	nées	
8.1	Modèle en programmation par contraintes	78
8.2	Description de la contrainte globale <code>sequenceEDD</code>	79
8.2.1	Relaxation du problème	80
8.2.2	Filtrage de la variable objectif	81
8.2.3	Filtrage basé sur le coût du placement d'une tâche	82
8.2.4	Filtrage basé sur le coût du nombre de fournées non vides	83
8.3	Stratégie de branchement	84
8.4	Expérimentations	85
8.4.1	Performance des règles de filtrage	85
8.4.2	Performance des heuristiques de sélection de valeur	87
8.4.3	Comparaison avec un modèle en programmation mathématique	88
8.4.4	Comparaison avec une approche par branch-and-price	89
8.5	Conclusion	89
	Annexe 8.A : un algorithme quadratique pour le filtrage du placement d'une tâche	90
9	Implémentation dans le solveur de contraintes choco	93
9.1	Le modèle	94
9.1.1	Variables	94
9.1.2	Contraintes	95
9.2	Le solveur	96
9.2.1	Lecture du modèle	96
9.2.2	Stratégie de branchement	97
9.2.3	Redémarrages	98
9.2.4	Résolution d'un modèle	99
9.3	Contraintes de choco	99
9.3.1	<code>cumulative</code>	100
9.3.2	<code>disjunctive</code>	101
9.3.3	<code>useResources</code> (en développement)	102
9.3.4	<code>precedence</code>	103
9.3.5	<code>precedenceDisjoint</code>	103
9.3.6	<code>precedenceReified</code>	103
9.3.7	<code>precedenceImplied</code>	104
9.3.8	<code>disjoint</code> (décomposition)	104
9.3.9	<code>pack</code>	104
9.3.10	Reformulation de contraintes temporelles	106
9.4	Construction du modèle disjonctif	106
9.4.1	Traitement des contraintes temporelles	107
9.4.2	Traitement des contraintes de partage de ressource	107
9.4.3	Déduction de contraintes « coupe-cycle »	109
9.5	CSP Solver Competition 2009	110
9.5.1	Catégorie 2-ARY-INT	110
9.5.2	Catégorie Alldiff+Cumul+Elt+WSum	113

9.6 Conclusion	113
10 Conclusion	115
10.1 Problèmes d'atelier	116
10.2 Problème de fournées	116
10.3 Solveur choco	116
 Annexes	
A Outils choco : développement et expérimentation	119
A.1 Procédure générique de traitement d'une instance	119
A.2 Assistant à la création d'un programme en ligne de commande	121
A.3 Intégration dans une base de données	121
B Tutoriel choco : ordonnancement et placement	123
B.1 Gestion de projet : construction d'une maison	123
B.1.1 Approche déterministe	124
B.1.2 Approche probabiliste	126
B.2 Ordonnancement cumulatif	129
B.3 Placement à une dimension	132
B.3.1 Traitement d'une instance	132
B.3.2 Création d'une commande	134
B.3.3 Cas d'utilisation	135
C Problème de collectes et livraisons avec contraintes de chargement bidimensionnelles	139
D Problème d'allocation de ressources et d'ordonnancement pour la maintenance	149
Bibliographie	165
Résumés	178

Table des figures

2.1	Exemple de CSP : n reines.	12
2.2	Exemples d'affectation : 4 reines.	12
2.3	Algorithme <i>backtrack</i> : 4 reines.	14
2.4	Exemple de CSP où l'arc-consistance est incomplète.	15
2.5	Exemple de CSP utilisant des contraintes globales: n reines.	15
2.6	Algorithmes de recherche rétrospectifs : 8 reines.	16
2.7	Influence des heuristiques de sélection sur l'algorithme <i>backtrack</i> : 4 reines.	18
2.8	Ordre d'exploration des branches par différentes recherches arborescentes.	19
2.9	Description de la procédure d'optimisation bottom-up.	20
2.10	Description de la procédure d'optimisation top-down.	21
2.11	Description de la procédure d'optimisation dichotomic-bottom-up.	22
2.12	Description de la procédure d'optimisation destructive-upper-bound.	23
3.1	Notations relatives à l'exécution d'une tâche.	26
3.2	Hierarchie des types de ressources.	29
3.3	Construction et d'arbitrage d'un graphe disjonctif.	32
3.4	Une condition nécessaire mais pas suffisante pour le problème de faisabilité.	34
4.1	Un ordonnancement optimal de délai total 1170 pour l'instance d'open-shop GP04-01.	36
5.1	Un ordonnancement optimal de retard maximal -90 pour l'instance bp10-07.	44
6.1	Exemples de stratégie de branchement.	55
6.2	Structure de l'algorithme RRCP.	56
6.3	Écart à l'optimum de la solution initiale et influence de CROSH sur les performances de RRCP.	61
6.4	Influence des redémarrages et de l'enregistrement des <i>nogoods</i>	62
6.5	Influence des redémarrages avec enregistrement des <i>nogoods</i> et équivalence des stratégies de redémarrage.	63
7.1	Comparaison des modèles <i>Light</i> et <i>Heavy</i> sur le problème d'open-shop.	73
8.1	Deux exemples de construction de la relaxation $I(\mathcal{A})$	81
8.2	Exemple de calcul de $L(\mathcal{A}_3 \cap \{M \leftarrow 4\})$ pour une affectation (vide) \mathcal{A}_3	84
8.3	Comparaison des règles de filtrage en utilisant <i>complete decreasing first fit</i>	86
8.4	Comparaison des heuristiques de sélection de valeur.	87
8.5	Comparaison à la formulation mathématique.	89
8.6	Anticipation de l'effet du placement d'une tâche sur la séquence de blocs.	90
8.7	Solution de la relaxation après le placement de la tâche 4.	92
9.1	Élimination des arbitrages créant un cycle de longueur 3 dans le graphe disjonctif.	109
9.2	Résolution de 100 problèmes d'open-shop (CSC'2009).	111
9.3	Résolution de 126 problèmes de job-shop (CSC'2009)	111
9.4	Nombre d'instances résolues dans une limite de temps (2-ARY-INT).	112
9.5	Nombre d'instances résolues dans une limite de temps (Alldiff+Cumul+Elt+WSum).	113
B.1	Visualisation des graphes disjonctifs (activée à l'étape 5) pendant les étapes 6 et 7.	127

B.2	Probabilité que le projet finisse au plus tard à une date D .	129
B.3	Une instance de notre problème d'ordonnancement cumulatif.	129
B.4	Une solution optimale de l'instance décrite en figure B.3.	132
B.5	Une solution optimale de l'instance N1C3W1_A.	136
B.6	Aperçu d'un rapport généré par oobase.	137

Liste des tableaux

5.1	Complexité des problèmes de fournées sans contrainte de précédence ni date de disponibilité.	47
6.1	Identification de bons paramètres pour les stratégies de redémarrage.	60
6.2	Résultats sur les trois instances de Brucker les plus difficiles.	63
6.3	Résultats sur les instances de Taillard.	66
6.4	Résultats sur les instances de Brucker.	67
6.5	Résultats sur les instances de Guéret-Prins.	68
7.1	Temps de résolution des 11 instances les plus difficiles du problème d'open-shop.	73
7.2	Comparaison des heuristiques de sélection de disjonction sur le problème d'open-shop.	74
7.3	Résultats sur le problème de job-shop.	75
8.1	Qualité des bornes inférieures destructives.	86
9.1	Méthodes pour la création d'une ou plusieurs tâches (<code>TaskVariable</code>).	95
9.2	Aperçu des méthodes publiques de la classe <code>TaskVar</code> .	97
9.3	Méthodes de résolution du <code>Solver</code> .	99
9.4	Principales méthodes de la classe <code>PackModel</code> .	106
9.5	Reformulation des contraintes temporelles.	106
B.1	Estimation probabiliste des durées des tâches.	128
B.2	Longueurs et variances des chemins critiques.	128

Listings

1	Calculer toutes les solutions du problème des n reines avec choco.	24
2	Méthode principale de la classe <code>AbstractInstanceModel</code>	119

Liste des Algorithmes

-	Procédure $\text{bottom-up}(lb,ub)$	20
-	Procédure $\text{top-down}(lb,ub)$	21
-	Procédure $\text{dichotomic-bottom-up}(lb,ub)$	22
-	Procédure $\text{destructive-lower-bound}(lb,ub)$	22
-	Procédure $\text{destructive-upper-bound}(lb,ub)$	23
-	Procédure jobCBF	91
-	Procédure buildDisjModT	108
-	Procédure buildDisjModR	108
-	Procédure postCoupeCycle	109