

Contribution à la résolution de problèmes d'optimisation combinatoire : méthodes séquentielles et parallèles

Mohamed Esseghir Lalami

▶ **To cite this version:**

Mohamed Esseghir Lalami. Contribution à la résolution de problèmes d'optimisation combinatoire : méthodes séquentielles et parallèles. Automatique. Université Paul Sabatier - Toulouse III, 2012. Français. NNT : . tel-00748546

HAL Id: tel-00748546

https://theses.hal.science/tel-00748546

Submitted on 5 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline ou spécialité : *Automatique et Informatique*

Présentée et soutenue par *Mohamed Esseghir LALAMI*
Le *Vendredi 5 octobre 2012*

Titre :
Contribution à la résolution de problèmes d'optimisation combinatoire : méthodes séquentielles et parallèles.

JURY
Mr Imed KACEM - Examinateur
Mr Boutaib DAHHOU - Examinateur
Mr. Mhand HIFI - Rapporteur
Mr Nouredine MELAB - Rapporteur
Mr Abd-El-Kader SAHRAOUI - Invité
Mr Didier EL BAZ - Directeur de thèse
Mr Moussa ELKIHEL - Directeur de thèse

Ecole doctorale : *SYSTEMES*
Unité de recherche : *LAAS-CNRS - Laboratoire d'Analyse et d'Architecture des Systèmes*
Directeur(s) de Thèse : *Didier EL BAZ et Moussa ELKIHEL*
Rapporteurs :

Remerciements

Ces quatre années de thèse furent une étape importante dans ma vie, autant sur le plan professionnel que personnel. Aussi, j'aimerais commencer ce mémoire en remerciant les personnes qui m'ont épaulé et accompagné tout au long de ce parcours.

Je remercie tout d'abord mes Directeurs de thèse, messieurs Didier EL BAZ et Moussa ELKIHEL, de m'avoir proposé ce sujet, et de l'attention qu'ils ont portée à mon travail. J'ai découvert grâce à eux le monde de la recherche, et plus précisément celui de l'optimisation, dans les meilleures conditions : conditions matérielles, mais aussi au travers de leurs conseils et de leur soutien.

J'adresse mes sincères remerciements à Messieurs Mhand HIFI et Nouredine MELAB pour l'intérêt qu'ils ont porté à cette thèse en acceptant d'en être les rapporteurs et à Messieurs Boutaib DAHHOU, Imed KACEM et Abd-El-Kader SAHRAOUI pour l'honneur qu'ils m'ont fait en acceptant de faire partie de mon jury.

Mes remerciements vont naturellement à l'ensemble des membres du groupe CDA : Messieurs Jean-Michel ENJALBERT, Vincent BOYER et The Tung NGUYEN avec lesquels j'ai partagé ces trois années.

Je tiens également à remercier l'école doctorale EDSYS ainsi que l'université Paul Sabatier de m'avoir permis un séjour scientifique de trois mois à l'université DTU du Danemark. Je remercie Monsieur David PISINGER pour la qualité de l'accueil qu'il m'a réservé dans le laboratoire de recherche opérationnelle du DTU de cette université.

Mes remerciements vont aussi à Monsieur Abd-El-Kader SAHRAOUI pour ses sages conseils et à Monsieur Francois COURTEILLE pour l'aide qu'il ma apportée.

Je remercie aussi tous mes amis doctorants et mes amis de Foot, avec lesquels j'ai vécu de très bons moments.

A mes grands parents, paix à leurs âmes, dont l'affection m'a toujours accompagné, à grand oncle Ahmed, à ma mère à mon père, à Nassima ma sœur, à Lamine mon jeune frère, à Amina ma fiancée, un grand merci pour le soutien moral que j'ai puisé en eux.

Enfin, à mon oncle Mohamed Seghir, prématurément disparu et dont je porte fièrement le nom, je dédie ce modeste labeur.

Table des matières

I. Introduction générale 1

II. Généralités sur le sac à dos 5

II.1 Introduction 5
II.2 Le problème du sac à dos (KP) 6
II.2.1 Le problème du sac à dos multiple 7
II.2.2 Le noyau du sac à dos 8
II.3 Méthodes de résolution exactes 11
II.3.1 La méthode de Branch and Bound 11
II.3.2 La Programmation Dynamique 14
II.4 Calcul de bornes et méthode de réduction de variables 19
II.4.1 Bornes supérieures pour le problème KP 19
II.4.2 Bornes inferieures pour le problème KP 21
II.4.3 Réduction de variables 21
II.5 Les solveurs existants 23
II.6 Conclusion 25

III. Le problème du sac à dos multiple 27

III.1 Introduction 27
III.2 Le problème du sac à dos multiple MKP 28
III.3 Etat de l'art 29
III.3.1 Différentes relaxations du problème MKP 29
a. La relaxation surrogate 30
b. La relaxation continue 31
c. La relaxation Lagrangienne 31
III.3.2 Algorithmes pour le problème du MKP 33

III.4 Une Nouvelle heuristique RCH pour le problème MKP 36
III.4.1 Remplissage classique : algorithme Glouton 38
III.4.2 Remplissage efficace : utilisation du noyau 39
III.5 Résultats expérimentaux 43
III.6 Conclusions et perspectives 48

IV. Introduction à CUDA et à l'architecture GPU 49

IV.1 Introduction 50
IV.2 Algorithmes et applications 52
IV.3 Architecture GPU 55
IV.3.1 Les threads 57
IV.3.2 Les mémoires 58
IV.3.3 Host et Device 59
IV.4 Règles d'optimisations 63
IV.4.1 Instructions de base 63
IV.4.2 Instructions de contrôle 63
IV.4.3 Instruction de gestion mémoire 64
a. Mémoire globale 64
b. Mémoire locale 66
c. Mémoire constante 66
d. Registres et mémoire partagée 66
e. Nombre d²

e threads par bloc 69
f. Transferts de données CPU ↔ GPU 70
IV.5 Conclusion 70

V. Mise en œuvre CPU-GPU de la méthode de Branch and Bound 71

V.1 Introduction 71
V.2 Branch and Bound pour le sac à dos 72

V.2.1 Formulation du problème 72
V.2.2 La méthode de Branch and Bound 73
V.3 Etat de l'art 77
V.4 Calcul hybride 80
V.4.1 Initialisation et algorithme général 80
V.4.2 Algorithme parallèle 81
V.4.3 Calculs sur GPU 83
V.4.4 Calculs sur CPU 91
V.5 Résultats expérimentaux 91
V.6 Conclusions et perspectives 94

VI. Mises en œuvre CPU-GPU et Multi-GPU de la méthode du Simplexe 96

VI.1 Introduction 97
VI.2 Rappel mathématique sur la méthode du Simplexe 98
VI.3 Etat de l'art 104
VI.4 Simplexe sur un système CPU-GPU 107
VI.4.1 Initialisation et algorithme général 107
VI.4.2 Calcul de la variable entrante et la variable sortante 108
VI.4.3 Mise à jour de la base 111
VI.5 Simplexe sur un système Multi-GPU 117
VI.5.1 Initialisation 118
VI.5.2 Les threads CPU 118
VI.5.3 Calcul de la variable entrante et la variable sortante 120
VI.6 Résultats expérimentaux 121
VI.7 Conclusions et perspectives 124

VII. Conclusions et Perspectives 127

Annexe A 131

Liste des publications 133

Références bibliographiques 135

Table des figures

II.1 Arbre engendré par décomposition d'un problème 12

III.1 Déroulement de l'algorithme associant la réduction de variables à la méthode de programmation dynamique. 43

IV.1 Evolution des performances de calcul des CPUs et GPUs 50
IV.2 Architecture CPU et architecture GPU. 51
IV.3 Architecture des applications CUDA....55
IV.4 Illustration d'une grille de threads. 58
IV.5 Déroulement du programme sur CPU faisant intervenir le GPU. 60
IV.6 Multiprocesseur SIMT avec mémoire partagée embarquée....62
IV.7 Exemples d'accès à la mémoire globale 65
IV.8 Adressages à la mémoire partagée avec et sans conflit de *bancs*. 68
IV.9 Adressages à la mémoire partagée avec diffusion. 69

V.1 Arbre de décision....76
V.2 Algorithme de Branch and Bound sur CPU-GPU 82
V.3 Création de nouveaux nœuds par la grille de threads sur GPU....85
V.4 Etape d'élagage : Procédure de substitution sur GPU d'un nœud non prometteur *l* après affectation sur CPU de l'adresse *j* du nœud qui va le remplacer 90

VI.1 Représentation géométrique du polytope pour un exemple....99
VI.2 Déclaration et allocation mémoire du *tableau Simplexe*....108
VI.3 Architecture globale de l'algorithme du Simplexe sur un système CPU-GPU....109
VI.4 Operations matricielles, gestion de mémoire dans le kernel 3'....114
VI.5 Décomposition du *TableauSimplexe* et accès mémoire des threads CPU 118

VI.6 Temps d'exécution de l'algorithme du Simplexe sur un CPU et différents systèmes avec un GPU et deux GPUs (Nvidia Tesla C2050) 122

Liste des tableaux

II.1 Taille minimale du noyau $|C|$ suivant le nombre d'articles 10

II.2 Listes de la programmation dynamique 17

III.1 Temps de calcul et *gaps* pour des problèmes non corrélés 46

III.2 Temps de calcul et *gaps* pour les problèmes faiblement corrélés 46

III.3 Temps de calcul et *gaps* pour des problèmes fortement corrélés 47

V.1 Comparaison des temps moyens de calcul de l'algorithme B&B séquentiel et parallèle 92

V.2 *Speedup* de l'étape de calcul de borne 93

VI.1 Tableau du Simplexe 101

VI.2 Accélérations moyennes 123

Chapitre I

Introduction générale

Le problème du sac à dos fait partie des problèmes d'optimisation combinatoire les plus étudiés ces cinquante dernières années, en raison de ces nombreuses applications dans le monde réel. En effet, ce problème intervient souvent comme sous-problème à résoudre dans plusieurs domaines : la logistique comme le chargement d'avions ou de bateaux, l'économie comme la gestion de portefeuille ou dans l'industrie comme la découpe de matériaux.

Ce problème en variables 0-1, dont l'énoncé est assez simple, fait partie des problèmes mathématiques NP-complets. Cela explique que le nombre d'ouvrages qui lui sont consacrés est important, on peut notamment citer les ouvrages de référence [KEL 04] et [MAR 90], mais aussi les différents travaux proposant diverses méthodes pour résoudre ce problème (cf [BEL 57], [GIL 65], [KOL 67], [BAL 80], [PLA 85] , [ELK 02] , [MEL 05] , [HIF 08] et [BEL 08a]).

De nos jours le problème du sac à dos se résout de manière assez efficace. Les travaux actuels portent sur différentes variantes du problème du sac à dos qui sont beaucoup plus difficiles à résoudre. On peut en citer :

- Le problème du sac à dos multidimensionnel (MKP) : on notera les travaux de Freville et Plateau [FRE 94], Hanafi et al. [HAN 96] et Boyer et al.[BOY 10].
- Le problème du partage équitable (KSP) : on notera les travaux de Hifi et al. [HIF 05] Belgacem et Hifi [BEL 08b], Boyer et al. [BOY 11].
- Le problème du sac à dos multiple (MKP) : on notera les travaux de Hung et Fisk [HUN 78], Martello et Toth [MAR 80].

- Le problème du sac à dos disjonctif (DCKP) : on notera les travaux de Yamada et Kataoka [YAM 01], Hifi et Michrafy [HIF 07].
- Le problème de bin packing: on notera les travaux de Bekrar et al.[BEK 10] et Hifi et al. [HIF 10].

Les approches proposées dans la littérature, pour résoudre les problèmes de la famille du sac à dos sont des méthodes exactes capables de résoudre un problème à l'optimalité ou des heuristiques qui fournissent une solution approchée de bonne qualité dans des temps de résolution très raisonnables.

Les méthodes classiques telles que la programmation dynamique et le Branch and Bound peuvent être combinées de manière efficace afin de donner naissance à des méthodes coopératives ou hybrides. On note par exemple le travail de Viader [VIA 98] sur une méthode coopérative pour le problème du sac à dos. On note aussi le travail de Boyer et al. [BOY 10] sur une méthode coopérative pour le problème du sac à dos multidimensionnel. Les tests numériques présentés montrent l'efficacité des méthodes coopératives par rapport aux méthodes classiques.

Le parallélisme constitue une autre approche afin d'accélérer la résolution de problèmes d'optimisation combinatoire. L'apparition de nouvelles architectures comme les Graphics Processing Units ou GPUs semble particulièrement intéressante afin de diminuer les temps de résolution de manière économique. cf. [LUO 10], [BOY 11].
On s'est intéressé dans ce mémoire à la résolution d'une variante du problème du sac à dos à savoir le problème du sac à dos multiple (MKP). Ce problème compte plusieurs applications industrielles dont on peut citer quelques exemples : le chargement de fret sur les navires, où il s'agit de choisir certains conteneurs, dans un ensemble de n conteneurs à charger dans m navires de différentes capacités de chargement (cf [EIL 71]), le chargement de n réservoirs par m liquides qui ne peuvent pas être mélangés (cf [MAR 80]); l'affectation de tâches. Le problème MKP est NP-complet et la nécessité de trouver des algorithmes donnant une bonne solution heuristique se justifie par la complexité de ce type de problème.

La deuxième partie de notre contribution porte sur l'utilisation des GPUs pour la mise en œuvre parallèle de méthodes d'optimisation combinatoire en variables 0-1. Ces travaux font suite à une série d'études effectuées dans l'équipe CDA du LAAS-CNRS sur la mise en

œuvre parallèle de la méthode de programmation dynamique sur GPU, cf. Boyer et al. [BOY 11] et [BOY 10]. Notre travail, s'est concentré sur la mise en œuvre parallèle sur GPU de la méthode de Branch and Bound ainsi que de la méthode du Simplexe.

Organisation de la thèse

Dans le **chapitre II**, nous commençons par présenter le problème du sac à dos ainsi que certaines de ces variantes comme le problème du sac à dos multiple. Nous nous intéressons en particulier à des méthodes de résolution classiques, comme la programmation dynamique et le Branch and Bound.

Nous proposons au **chapitre III** une méthode heuristique pour résoudre le problème du sac à dos multiple. Nous commençons d'abord par un état de l'art du MKP et détaillons en particulier une heuristique qui à été proposée pour le problème du sac à dos multiple, à savoir l'heuristique MTHM de Martello et Toth [MAR 81]. Nous présentons en détail notre contribution à savoir l'heuristique RCH pour Recursive Core Heuristic. Dans cette dernière méthode, nous considérons le problème du sac à dos multiple comme une succession de problème de sac à dos à résoudre. Nous définissons alors pour chaque problème KP un noyau. Nous résolvons alors pour chaque noyau excepté le dernier, un problème de *subset sum* par l'approche basée sur la programmation dynamique proposée par Elkihel [ELK 84] tandis que le dernier noyau est résolu en utilisant la programmation dynamique classique.

Le **chapitre IV** est consacré au GPU. Nous commençons d'abord par donner un état de l'art du calcul sur GPU. Puis, nous nous intéresserons à l'architecture CUDA (Compute Unified Device Architecture) proposée par NVIDIA. Les performances de cette architecture reposent sur deux éléments fondamentaux : la mémoire et la décomposition du travail en tâches.

Au **chapitre V**, nous présentons l'approche que nous avons suivie pour la mise en œuvre parallèle de l'algorithme de Branch and Bound sur GPU. Nous donnons un bref état de l'art relatant les différentes implémentations parallèles existantes pour l'algorithme de Branch and Bound, qu'elles soient sur des machines multi-cœurs, grilles de calculs ou sur architecture GPU. Nous expliquons les différents choix que nous avons faits pour aboutir à la mise en œuvre proposée. Ceux-ci concernent tout aussi bien la stratégie de séparation des nœuds, les données sauvegardés pour chaque nœuds, les mémoires utilisées mais aussi les différentes

techniques et synchronisations utilisées pour diminuer les temps de latence d'accès en mémoire. Pour finir, nous présentons nos résultats et les analysons.

Enfin, nous présentons au **chapitre VI**, l'approche que nous avons suivie pour la mise en œuvre parallèle de la méthode du Simplexe sur GPU et sur un système Multi-GPU. Nous commençons d'abord par présenter un bref état de l'art. Nous expliquons ensuite les différents choix que nous avons faits pour aboutir à la mise en œuvre proposée. Ceux-ci concernent aussi bien l'identification des tâches de cet l'algorithme qui peuvent se paralléliser de manière performante, que le choix des mémoires utilisées et le moyen utilisé pour réduire l'effet de chemins divergents induits par des instructions conditionnelles. Nous proposons, ensuite, une mise en œuvre multi-GPU de l'algorithme du Simplexe mettant à contribution plusieurs cartes GPUs disponibles dans un seul système pour résoudre un problème de programmation linéaire. Nous expliquons comment partager le tableau du Simplexe entres les différents GPUs et comment diminuer ainsi les échanges entre les GPUs et le CPU. Enfin, nous présentons et analysons les résultats obtenues pour la mise en œuvre séquentielle, sur GPU et la mise en œuvre parallèle sur un système multi-GPU.

Nous terminons ce mémoire en présentant nos conclusions générales et les perspectives de recherche.

Chapitre II

Généralités sur le sac à dos

Sommaire

II.1 Introduction 5

II.2 Le problème du sac à dos (KP) 6

II.2.1 Le problème du sac à dos multiple 7

II.2.2 Le noyau du sac à dos 8

II.3 Méthodes de résolution exactes 11

II.3.1 La méthode de Branch and Bound 11

II.3.2 La Programmation Dynamique 14

II.4 Calcul de bornes et méthode de réduction de variables 19

II.4.1 Bornes supérieures pour le problème KP 19

II.4.2 Bornes inferieures pour le problème KP 21

II.4.3 Réduction de variables 21

II.5 Les solveurs existants 23

II.6 Conclusion 25

II.1 Introduction

Dans le présent chapitre, nous présentons le contexte dans lequel vont s'inscrire nos travaux de recherche. Ces travaux s'articulent autour de la résolution de problèmes d'optimisation.

Dans la sous-section II.2, Nous définirons le problème du sac à dos ainsi que le problème du *sac à dos multiple* (MKP). Nous définirons aussi la notion de noyau d'un problème de sac à dos.