



JPOC5

Journées Polyèdres et Optimisation Combinatoire

INSA Rouen
du 4 au 6 juin 2008

Comité d'organisation :

Stefan Balev (LITIS / Université du Havre)
Sylvie Borne (LIPN / Université Paris 13)
Pierre Fouilhoux (LIP6 / Université Paris 6)
Sérigne Gueye (LMAH / Université du Havre)
Mhamed Itmi (LITIS / INSA Rouen)
Arnaud Knippel (LMI/INSA Rouen)
A. Ridha Mahjoub (LAMSADE / Paris Dauphine)

PREFACE

Nous sommes heureux de vous accueillir à Rouen, ces 4,5 et 6 juin 2008, pour les 5èmes journées Polyèdres et Optimisation Combinatoire (JPOC5). Ces journées sont précédées pour une partie des participants par un cours doctoral (du 2 au 4 juin) dont le thème est : *Inégalités valides en programmation en nombres entiers*. Ce thème est donc à l'honneur également pour la conférence, mais les autres aspects de l'optimisation combinatoire et des polyèdres y ont aussi toute leur place.

L'optimisation combinatoire est une branche relativement jeune de l'informatique et de la programmation mathématique. Elle a des applications importantes dans l'industrie. Elle est utilisée dans les transports, les télécommunications, les systèmes d'information et bien d'autres domaines. C'est donc une discipline qui intéresse particulièrement la région Haute Normandie, une région qui, en raison de son histoire et de sa situation géographique, avec les deux ports maritimes du Havre et de Rouen, est très tournée vers l'industrie et considère comme une priorité la recherche dans le domaine des transports et de la logistique. Vous aurez peut-être l'occasion lors de votre séjour à Rouen de visiter les installations portuaires ou le chantier du "6ème pont", mais nous espérons que vous trouverez aussi un moment pour découvrir quelques-uns des nombreux sites touristiques, du centre historique de Rouen aux abbayes du bord de Seine.

Ce recueil contient les résumés de 22 présentations dont 6 exposés invités. Les exposés se dérouleront sous forme de sessions plénières dans les locaux de l'INSA, dans le grand amphithéâtre situé à l'entrée principale du bâtiment. Ces journées ont pu être organisées grâce au soutien du GdR Recherche Opérationnelle du CNRS, de la ROADEF, et plus localement de l'INSA, de l'école doctorale SPMII, des laboratoires de recherche LMI, LITIS et LMAH, ainsi que de la Région Haute Normandie. De nombreuses personnes ont contribué de près ou de loin à l'organisation de ces journées, que ce soient les membres du comité d'organisation, les membres du comité du groupe Polyèdres et Optimisation Combinatoire (du GdR RO) ou les personnels de l'INSA. Nous les remercions chaleureusement.

Arnaud Knippel

PROGRAMME JPOC5

Mercredi 4 juin 2008

14h00 - 14h30

Ouverture des Journées JPOC5

14h30 - 15h30

Programmation Linéaire, Lancer de Rayon et Polarité

Y. Gérard

15h30 - 16h00

Heuristiques basées sur la génération de colonnes pour résoudre un problème de tournées de véhicules combinées à la gestion de stock

S. Michel, F. Vanderbeck

16h00 - 16h30

Le polyèdre du problème de conception de réseaux robustes K-arête connexe avec 3 sauts

Q. Botton, B. Fortz, L. Gouveia

16h30 - 17h00 PAUSE CAFÉ

17h00 - 17h30

Polyèdre des arbres d'Huffman

T. H. Nguyen, J.-F. Maurras, Nguyen V. H.

17h30 - 18h00

Sur la qualité des colonnes générées dans un schéma de génération de colonnes

N. Touati, L. Létocart, A. Nagih

18h00 - 18h30

Meilleur et pire optimum pour des programmes linéaires avec second membres incertains

N. Remli

18h30 - 19h00

Sur le polytope des *st-chaînes*

V. H. Nguyen

Jeudi 5 juin 2008

9h00 - 10h00

Sur les facettes des programmes en nombres entiers mixtes avec deux variables entières

G. Cornuejols

10h00 - 10h30

Optimisation robuste et polyèdres

O. Klopfenstein

10h30 - 11h00 PAUSE CAFÉ

11h00 - 12h00

Linear and Integer Programming vs Linear Integration and Counting

J.-B. Lasserre

12h00 - 12h30

Coupes disjonctives pour la programmation mixte sous contraintes quadratiques

A. Saxena, P. Bonami, J. Lee

12h30 - 14h00 DÉJEUNER

14h00 - 15h00

L'algorithme CCB : comment améliorer l'efficacité de la méthode de Benders

M. Minoux

15h00 - 15h30

Preuve constructive d'une relation min-max multicoupe/multiflot dans les graphes série-parallèles

D. Cornaz

15h30 - 16h00

Approximation de problèmes de couverture de tâches en transport ferroviaire

J. Sadki, L. Alfandari, A. Nagih, A. Plateau

16h00 - 16h30 PAUSE CAFÉ

16h30 - 19h30

Réunions de travail et discussions

20h00 REPAS DE GALA

Vendredi 6 juin 2008

9h00 - 10h00

Testing Cut Generators for ILP

F. Margot

10h00 - 10h30

Polytope des cliques circulaires et calcul du nombre d'indépendance des graphes quasi-adjoints

A. Pêcher

10h30 - 11h00 PAUSE CAFÉ

11h00 - 12h00

Optimisation en présence de contraintes non holonomes

W. Respondek

12h00 - 12h30

Ordonnancement en Juste-à-Temps sur une Seule Machine

M. Rebaï, I. Kacem

12h30 - 14h00 DÉJEUNER

14h00 - 14h30

Sur le partitionnement des graphes en composantes unicycliques avec contraintes

W. Ben-Ameur, M. Hadji, A. Ouorou

14h30 - 15h00

Branchement local pour le problème de sac-à-dos multidimensionnel à choix multiples

N. Cherfi, M. Hifi

15h00 - 15h30

Une Approche Lagrangienne Appliquée aux Problèmes d'Alignement dans la Bioinformatique

S. Balev

15h30 - 16h00

Formulations linéaires en nombres entiers pour des problèmes d'isomorphisme exact et inexact

P. Le Bodic, S. Adam, P. Héroux, A. Knippel, Y. Lecourtier

16h00

Cloture des Journées

Programmation Linéaire, Lancer de Rayon et Polarité

Y. Gérard

*LAIC, Université Clermont I, Aubière, France
gerard@laic.u-clermont1.fr*

Mots-clefs : géométrie algorithmique, optimisation géométrique

Nous commencerons par présenter quelques problèmes de géométrie algorithmique ou d'optimisation géométrique dits de type LP avant de nous intéresser plus particulièrement à un problème élémentaire de cette classe, le problème de lancer de rayon sur un polytope : input : un ensemble fini S de points, une droite orientée (le rayon) output : les sommets de la face éclairée de l'enveloppe convexe de S .

Nous montrerons que non seulement ce problème est de type LP, mais qu'il s'agit exactement de programmation linéaire, c'est-à-dire que résoudre un programme linéaire consiste exactement à lancer un rayon sur un ensemble de points. La présentation de ce point de vue géométrique sur la programmation linéaire est l'objectif principal de l'exposé. Nous verrons qu'il permet d'interpréter simplement la LP-dualité et qu'on peut ainsi interpréter la relation entre contraintes et solutions comme une transformation géométrique connue sous le nom de polarité. Nous utiliserons enfin ce point de vue original pour comprendre les algorithmes classiques (Simplex, Megiddo, Ellipsoïde?, points intérieurs?) comme des algorithmes de géométrie algorithmique et en développer de nouveaux...

Heuristiques basées sur la génération de colonnes pour résoudre un problème de tournées de véhicules combinées à la gestion de stock

S. Michel^{1,3} et F. Vanderbeck^{2,3}

1. LMAH, Université du Havre,
ISEL, Quai Frissard BP 1137, 76063 Le Havre Cedex
michels@univ-lehavre.fr

2. IMB, Université Bordeaux 1,
351 cours de la Libération, 33405 Talence Cedex
fv@math.u-bordeaux1.fr

3. INRIA Bordeaux Sud-Ouest

Mots-clefs : heuristiques primales, génération de colonnes, tournées de véhicules, planification, problème industriel.

1 Heuristiques basées sur la génération de colonnes

Les heuristiques classiques de type glouton, de recherche locale ou d'arrondi de la solution continue, peuvent être adaptées dans le cadre d'une approche de génération de colonnes. Intégrées à une méthode exacte de Branch-and-Price, elles permettent d'améliorer l'incumbent et ainsi de réduire le nombre de nœuds à explorer. Ces heuristiques permettent de trouver de "bonnes" solutions pour des problèmes réels de grande taille dont la structure se prête à une décomposition de Dantzig-Wolfe.

Dans une première partie de l'exposé, nous verrons comment implémenter dans le cadre d'une approche de décomposition les quatre heuristiques classiques suivantes :

- la résolution du maître restreint en nombre entier,
 - les heuristiques de type glouton,
 - les heuristiques d'arrondi,
 - les heuristiques de recherche locale et méta-heuristiques,
- tout en pointant leur utilisation dans la littérature [1, 3, 4, 5, 7, 9, 11, 13, 14].

Dans une deuxième partie, nous exposerons les résultats obtenus sur un problème réel de tournées de véhicules combinées à la gestion de stock.

2 Tournées de véhicules combinées à la gestion des stocks

Dans l'application industrielle qui a motivé notre étude, les tournées de véhicules sont combinées à la gestion de stock chez les sites clients. 3 décisions doivent être prises : (i) quand servir un client, (ii) quelle quantité livrer ou collecter chez le client servi, (iii) quelles routes utiliser. Ce problème est connu sous le nom d' "Inventory Routing Problem". Beaucoup de variantes ont été traitées dans la littérature [2, 6, 8, 16], mais les approches existantes tendent à faire des hypothèses restrictives (telle que la politique de partition fixe) ou d'adopter une optimisation hiérarchique où l'élaboration d'un planning est décidée avant la construction des routes. De

plus, la plupart de ces approches heuristiques ne fournissent pas de déviation à l’optimalité et sont spécifiques à la variante du problème.

Dans la variante considérée ici, une flotte homogène de véhicules est affectée à collecter un seul produit sur différents sites. Chaque site a son propre taux d’accumulation déterministe et sa capacité de stockage. A chaque visite, le stock est vidé (“order-up-to-level policy”). Dans la phase de planification tactique, nous cherchons une solution périodique qui est répétée dans le temps. L’objectif est de minimiser la taille de la flotte et les coûts de transport tout en donnant un découpage régionale de l’espace par une partition des sites entre les véhicules.

La structure du problème est exploitée pour développer une approche de décomposition de Dantzig-Wolfe [15]. Des plannings périodiques sont générés pour les véhicules en résolvant un problème de sac-à-dos à choix multiple [12]. L’élaboration des plannings de collecte pour les sites est traitée dans le programme maître. Les bornes duales sont calculées par une méthode tronquée de Branch-and-Price-and-Cut. Les bornes primales sont obtenues par des heuristiques d’arrondi et de recherche locale développées dans le contexte de la génération de colonnes. Des instances, fournies par notre partenaire industriel, sont ainsi résolues avec une déviation à l’optimalité raisonnable.

3 Branch-and-Price-and-Cut heuristique

La décomposition de Dantzig-Wolfe permet d’éliminer la symétrie liée à l’indexation des véhicules, la formulation obtenue souffre cependant d’une symétrie en temps. En-effet, décaler les départs des routes dans une solution périodique définit une solution symétrique. Pour éviter cette symétrie en temps, le problème maître *discret* est alors reformulé en terme de variables agrégées, nous obtenons un problème maître *agrégé*. Nous montrons que les problèmes maîtres agrégé et discret ont la même solution linéaire optimale. Néanmoins, la formulation agrégée est une relaxation du problème et la formulation discrète reste utile pour le calcul d’une solution entière.

En conséquence, la borne duale est obtenue en résolvant à l’optimalité la relaxation linéaire du problème maître agrégé par génération de colonnes. Pour améliorer la formulation, des coupes sont ajoutées au problème maître agrégé. Ces coupes sont dérivées de nos contraintes de couverture de la demande en utilisant une procédure d’arrondi. Notre objectif étant focalisé sur l’utilisation des véhicules, nous branchons sur la variable comptabilisant ce nombre. L’impact de ce branchement partiel sur la borne duale est important, contrairement aux coupes. Notre meilleure borne duale permet d’évaluer la qualité de nos solutions obtenues par les heuristiques primales.

La borne primale est obtenue par des heuristiques basées sur la formulation agrégée, mais à chacune de leurs étapes, nous nous assurons que la solution partielle construite est réalisable pour la formulation discrète. Plusieurs variantes des heuristiques sont comparées, les heuristiques d’arrondi et de recherche locale fournissent nos meilleures solutions.

Références

- [1] C. Archetti, MWP. Savelsbergh, MG. Speranza, An Optimization-Based Heuristic for the Split Delivery Routing Problem, *Working Paper, University of Brescia*, 2006.
- [2] J. Bramel, D. Simchi-Levi, A location based heuristic for general routing problems, *Operations Research*, 43 :649–660, 2005.
- [3] A. Ceselli, G. Righini, M. Salani, A column generation algorithm for a vehicle routing problem with economies of scale and additional constraints, *Tristan*, 2007.
- [4] A. Chabrier, E. Danna, C. Le Pape, Coopération entre génération de colonnes et recherche locale appliquées au problème de routage de véhicules, *JNPC*, 2002.
- [5] A. Chabrier, Heuristic Branch-and-Price-and-Cut to solve a network design problem, *Proceedings CPAIOR*, 2003.
- [6] AM. Campbell, LW Clarke, MWP. Savelsbergh, Inventory routing in practice, *The Vehicle Routing Problem*, eds P. Toth, D. Vigo, SIAM Monographs on Discrete Mathematics and Applications, 2002.
- [7] FL. Cimelière, Optimisation du traitement de l'ordre de fabrication dans l'industrie textile, *Thèse, Université de Bordeaux1*, 2004.
- [8] V. Gaur, ML. Fisher, A periodic inventory routing problem at a supermarket chain, *Operations Research*, 52 :813–822, 2004.
- [9] O. Gulunk, T. Kimbel, L. Ladangi, B. Schieber, G. Sorkin, Vehicle routing and staffing for sedan service, *Transportation Science*, 2005.
- [10] S. Michel Optimisation des tournées de véhicules combinée à la gestion des stocks *Thèse, Université de Bordeaux1*, 2006.
- [11] N. Perrot, Integer Programming Column Generation Strategies for the Cutting Stock Problem and its Variants, *Thèse, Université de Bordeaux1*, 2005.
- [12] D. Pisinger, A minimal Algorithm for the Multiple-Choice Knapsack Problem, *European Journal of Operational Research*, 83 :394–410, 1995.
- [13] E. Taillard, A heuristic column generation method for the heterogeneous VRP, *RAIRO Operations Research*, 33 :1–14, 1999.
- [14] F. Vanderbeck, Exact Algorithm for minimising the number of setups in the one-dimensional cutting stock problem, *Operations Research*, 48 :915–926, 2000.
- [15] F. Vanderbeck, M.W.P. Savelsbergh, A Generic View of Dantzig-Wolfe Decomposition in Mixed Integer Programming, *Operations Research Letters*, 34 :296-306, 2006.
- [16] I. Webb, R. Larson, Period and Phase of customer replenishment : A new approach to the Strategic Inventory/Routing Problem *European Journal of Operational Research*, 85 :132–148, 1995.

Le polyèdre du problème de conception de réseaux robustes K-arête connexe avec 3 sauts.

Q.Botton¹, B.Fortz² et L.Gouveia³

1. *Université catholique de Louvain, Institut d'Administration et de Gestion,
1, Place des Doyens,
1348, Louvain-la-Neuve, Belgique
quentin.botton@uclouvain.be*

2. *Université libre de Bruxelles, Département d'informatique,
Campus de la Plaine, Boulevard du Triomphe,
1050 Bruxelles, Belgique
bernard.fortz@ulb.ac.be*

3. *Université de Lisbonne, Département de statistiques et de recherche opérationnelle,
C6, Campo Grande,
1749-016 Lisbonne, Portugal
legouveia@fc.ul.pt*

Mots-clefs : Conception de Réseaux Robustes, Qualité de Service, Contraintes de Sauts, Chemins Disjoints.

1 Introduction

Ce papier considère l'étude du problème de conception de réseaux robustes K-arête connexe avec un maximum de 3 sauts. Celui-ci consiste à trouver le sous-graphe de coût minimum de telle manière qu'il existe, pour chaque demande ou commodité, au moins K-chemins arête-disjoint d'une part et assurant que le nombre d'arêtes utilisées dans chaque chemin n'est pas supérieur à un paramètre L fixé, d'autre part. Dans le cas d'espèce, nous nous focalisons dans un premier temps sur la version simple commodité et nous fixons le paramètre $L \leq 3$. Ce problème a été souvent étudié par le passé dans l'espace des variables de design, variables binaires associées à chaque arête potentiellement sélectionnable. Dans cet article, nous proposons une formulation étendue utilisant uniquement les variables de flot pour modéliser les chemins. Nous démontrons que la relaxation linéaire de notre modèle donne pour $L \leq 3$ et pour $K \geq 1$ d'une part une description complète du polyèdre associé pour la version arc-disjoint et d'autre part pour la version arête-disjoint nous montrons que la solution optimale de la relaxation linéaire est toujours entière si et seulement si les coûts d'installation des arêtes sont non-négatifs. Enfin, nous étudions la qualité de la borne inférieure proposée par la relaxation linéaire de notre modèle lorsque L est supérieur à 3.

2 Revue de la littérature

La robustesse dans le domaine de la conception de réseaux est une classe de problèmes sur laquelle la recherche de ces dernières années s'est fortement concentrée. Sous l'hypothèse de panne(s), certains chercheurs préconisent des techniques de reroutage dit local, qui consistent à trouver un ou plusieurs nouveau(x) plus court(s) chemin(s) entre les extrémités de l'arête (pour des pannes d'arêtes) endommagée. Alors que d'autres, et nous nous rattachons à ces personnes dans le cadre de cette étude, s'attachent à trouver de nouveaux chemins entre l'origine et la

destination de toutes les demandes affectées par la panne. Dans le cadre de cet article, nous traitons la robustesse en trouvant pour chaque commodité K -chemins arête(arc)-disjoint, (si une arête (un arc) est utilisée par un des K -chemins, alors cette arête ne sera plus utilisable par la suite). Nous garantissons dès lors un certain degré de protection lié au paramètre K . En effet, $K - 1$ arêtes (arcs) de chemins différents peuvent tomber en panne simultanément, sans pour autant que cela aie un impact important sur la qualité du service. Néanmoins, cette contrainte de robustesse n'est pas suffisante car en cas de panne, il se peut que certains de ces K -chemins arête-disjoint soient particulièrement longs, ce qui pourrait se répercuter dans une augmentation du délai de réception de l'information et dès lors dans une diminution de la qualité de service proposé. Dès lors, il est logique d'ajouter une contrainte supplémentaire qui viendrait limiter la longueur des plus courts chemins à utiliser, cette longueur étant ici représentée en termes du nombre d'arêtes (arcs) utilisées par ces chemins. Ces nouvelles contraintes sont appelées "contraintes de sauts". Nous pouvons mettre en exergue trois courants principaux de recherche qui ont intégré ces contraintes de sauts.

Le premier concerne les problèmes de recherche d'arbres de couverture avec contraintes de sauts qui consistent à trouver l'arbre de couverture de coût minimum de telle manière que chaque chemin entre la racine et les feuilles n'utilise pas plus de L sauts. Ce problème a été largement étudié par Gouveia [8], [9] dans lesquels l'auteur propose une formulation de flot multicommodités et discute l'utilisation de relaxation lagrangienne afin d'améliorer la borne inférieure. Dans [10], Gouveia et Requejo proposent une formulation "hop-indexed" basée sur une représentation en couches du graphe d'origine. Cette formulation est exprimée dans un espace de nouvelles variables de flot et non pas dans l'espace original des variables de "design". Dans [3], Dahl propose une étude du polyèdre de ce type de problème lorsque le nombre de sauts est limité à 2.

Le second courant se concentre sur la recherche du plus court chemin entre un noeud d'origine o et une destination d en garantissant l'utilisation d'au maximum L sauts. Dans [4], Dahl et dans [6], Dahl et Gouveia donnent une description complète du polyèdre pour $L \leq 3$ en définissant les "Jump Inequalities" pour lesquelles le problème de séparation peut être résolu en temps polynomial. Dans [5], Dahl, Gouveia et al. étendent leur précédent travail au cas où $L \leq 4$.

Le troisième courant important de recherche concernant les contraintes de sauts, est celui étudié dans le cadre de ce papier et qui traite de la conception de réseau avec K -chemins arête-disjoint qui comporte un nombre maximum de sauts limité au paramètre L . Dans [1], Balakrishnan et al. donnent une formulation mixte en nombres entiers pour le problème où $K = 1$ et dérivent une relaxation lagrangienne pour améliorer la borne inférieure. Dans [13], Pirkul et al. proposent une formulation de flots multicommodités et quelques heuristiques basées sur la relaxation linéaire. Dans [7], Dahl et al. étudient le problème où il y a pour chaque paire d'origine-destination K -chemins arête-disjoint de longueur maximum L et ils prouvent que le problème est NP difficile même lorsque $K = 1$ et $L \leq 2$. Dans [12], Huygens et al. étudient le même problème pour le cas simple commodité. Ils proposent un modèle en nombres entiers pour $K = 2$ et $L \leq 3$ dans l'espace des variables de "design". Dans [11], Huygens et Majhoub proposent de nouvelles inégalités valides pour le problème où $K = 2$ et $L \leq 4$ basées sur les "Jump Inequalities". Dans [2], Bley montre que le problème simple commodité noeud-disjoint avec contraintes de sauts est NP-complet pour de grande valeur de L .

Dans ce papier, nous proposons une formulation étendue, non pas exprimée dans l'espace des variables de "design", mais dans l'espace des variables de flot du graphe en couches proposé par Gouveia [10]. Nous nous servons de l'information supplémentaire livrée par ce graphe pour proposer un modèle simple et performant pour $L \leq 3$ et $K \geq 1$.

3 Formulation étendue et propriétés

Comme énoncé ci-dessus, notre modèle est basé sur la représentation alternative en couches du graphe original proposée par Gouveia [10]. Cette représentation transforme le graphe original en intégrant déjà les contraintes de saut d'une manière particulière. L'idée est de créer un nombre de couches strictement équivalent à la valeur du paramètre L en dupliquant $L + 1$ fois l'ensemble des noeuds comme illustré sur la Figure 1.

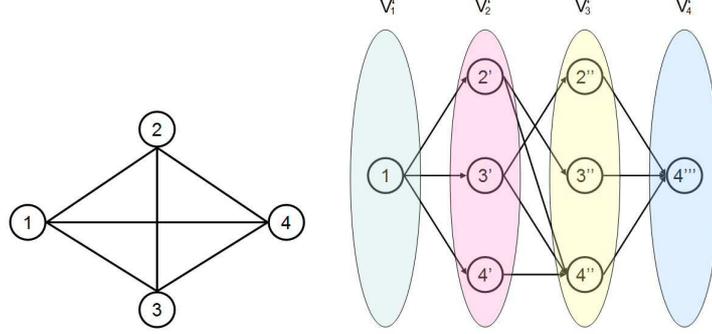


FIG. 1 – Graphe original et sa représentation en couche pour $L \leq 3$.

Plus formellement d'un graphe original non-dirigé $G = (V, E)$, nous créons un graphe alternatif en couches dirigé $G' = (V', A')$ dans lequel V' représente l'ensemble des noeuds et où A' représente l'ensemble des arcs dirigés dans le graphe en couche. $V' = \{V'_1 \cup V'_2 \cup \dots \cup V'_{L+1}\}$ où $V'_1 = \{o\}$, $V'_{L+1} = \{d\}$ et $V'_l = V \setminus \{o\}$, $l = 2, \dots, L - 1$. L'ensemble des arcs est défini par $A' = \{\{(i, j) \mid ij \in E \text{ et } i \in V'_l \text{ et } j \in V'_{l+1}, l = 1, \dots, L\} \cup \{(i, j) \mid ji \in E \text{ et } i \in V'_l \text{ et } j \in V'_{l+1}, l = 1, \dots, L\}\}$. Il est important de noter que chaque chemin réalisable entre l'origine o et la destination d dans le graphe en couches utilise exactement L arcs. Toutes les solutions non réalisables de notre problème utilisant plus de L sauts sont automatiquement rejetées. Associé à ce graphe en couche, nous proposons le modèle suivant pour la version arc-disjoint (notée modèle HopA) :

Model HopA

$$\min \sum_{l \in \{1, \dots, L\}} \sum_{(i, j) \in A'} c_{ij}^l U_{ij}^l \quad (1)$$

subject to

$$\sum_{j: (j, i) \in A'} U_{ji}^l - \sum_{j: (i, j) \in A'} U_{ij}^{l+1} = \begin{cases} -K & \text{if } (i = o) \text{ \& } (l = 0) \\ K & \text{if } (i = d) \text{ \& } (l = L) \\ 0 & \text{else} \end{cases}, \quad \forall i \in N, \forall l \in \{1, \dots, L\}, \quad (2)$$

$$\sum_{l \in \{1, \dots, L\}} U_{ij}^l \leq 1, \quad \forall (i, j) \in A, \quad (3)$$

$$U_{ij}^l \in \{0, 1\}, \quad \forall (i, j) \in A', \forall l \in \{1, \dots, L\} \quad (4)$$

Dans ce modèle, U_{ij}^l représente la quantité d'information envoyée sur l'arc (i, j) lorsque celui-ci est utilisé en position l . Le paramètre c_{ij}^l représente le coût par unité d'information envoyée

sur l'arc (i, j) lorsque celui-ci est utilisé en position l . (1) est la fonction objectif qui minimise le coût total d'installation. (2) sont les contraintes de conservation de flot, notons simplement que pour modéliser les K chemins disjoints, nous envoyons entre l'origine et la destination une quantité égale au paramètre K et nous ajoutons les contraintes (4) qui sont des contraintes de binarité sur les variables de décision. Enfin, nous ajoutons les contraintes (3) qui limitent l'usage d'un même arc par plusieurs chemins. Lorsque $L \leq 3$, il est trivial d'étendre ce modèle au cas arête-disjoint que nous notons model HopE.

Proposition 1 *Pour toute valeur de L et pour $K = 1$, la solution optimale de la relaxation linéaire du model HopA (respectivement du modèle HopE) est toujours entière si et seulement si les deux conditions suivantes sont respectées :*

- i) $\forall (i, j) \in A, \forall l \in \{1, \dots, L\}, c_{ij}^l \leq 0$,
- ii) $\forall (i, j) \in A, \forall (l, q) \in \{1, \dots, L\}^2, c_{ij}^l = c_{ij}^{l+q}$.

Proposition 2 *For $L \leq 3$ and $K \geq 1$:*

1. *Le polyèdre défini par les contraintes de la relaxation linéaire du modèle HopA est entier.*
2. *La solution optimale de la relaxation linéaire du modèle HopE est entière si et seulement si :*
 - i) $\forall ij \in E, \forall l \in \{1, \dots, L\}, c_{ij}^l \geq 0$.

Les preuves de ces propositions se trouvent dans la version complète du papier.

4 Extensions et résultats numériques

Lorsque $L \leq 4$, la relaxation linéaire du modèle HopA (respectivement du modèle HopE) n'est plus entière comme illustré au travers d'un contre-exemple à la Figure 2 ci-dessous.

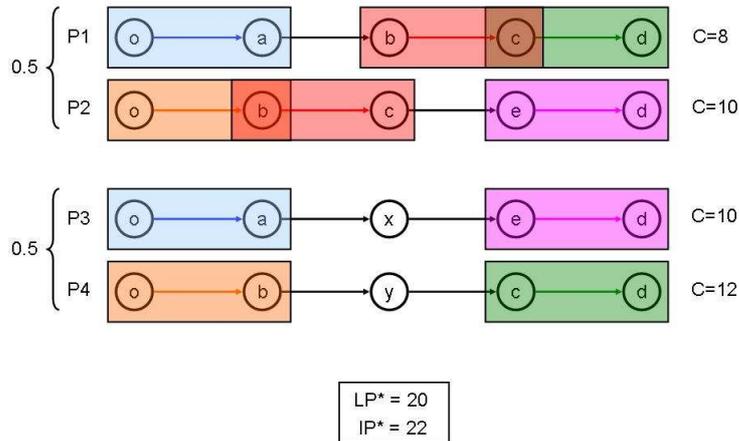


FIG. 2 – Solution optimale fractionnaire pour $L \leq 4$.

Néanmoins la qualité de la borne inférieure de la relaxation linéaire est très bonne. Plus de 28000 tests ont été réalisés pour différentes valeurs des paramètres $L \in [4, 6]$, $K \in [2, 4]$, c_{ij}^l générés aléatoirement ($\in [1, 10], [1, 100], [1, 100]$), sur des graphes complets, pour le cas simple commodité en utilisant Cplex 7.1. sur un Pentium 3, 1.1 Ghz, 1 Gb Ram. Seules 6 instances sur 28840 (soit 0,02%) ont une solution optimale du problème relâché non entière. Les solutions

Instance	L	K	Cost	LP*	IP*	Gap(%)
1	4	3	[1-1000]	583	585	0.34
2	4	4	[1-100]	60.5	61	0.8
3	5	3	[1-1000]	697	700	0.35
4	5	4	[1-100]	127.5	128	0.39
5	6	4	[1-10]	29	29	0.00
6	6	3	[1-100]	132.5	133	0.37

TAB. 1 – Instances fractionnaires et mesure de la borne inférieure.

fractionnaires s'avèrent donc être des cas très pathologiques. Le tableau ci-dessus nous donne une indication de la qualité de la borne inférieure proposée par la relaxation linéaire de notre modèle puisque le "Gap" $\in [0.00\%-0.80\%]$.

5 Conclusion et travaux futurs

Dans ce papier, nous proposons un modèle linéaire pour le problème de conception de réseaux robustes K-arête connexe avec 3 sauts, exprimé dans l'espace des variables de flot associées à la représentation en couches proposée par Gouveia [10]. Nous proposons une description complète du polyèdre pour la version arc-disjoint, ainsi qu'une modélisation garantissant une solution optimale entière pour la version arête-disjoint. Nous étudions la qualité de la borne inférieure proposée par le modèle relâché lorsque $L \geq 4$.

Nous travaillons sur de nouvelles inégalités valides qui coupent les points fractionnaires pour le cas où $L \leq 4$ et ce pour les versions arc-disjoint et arête-disjoint. Une extension vers le cas multicommodités est prévue en proposant l'utilisation de la méthode de décomposition de Benders pour traiter ce type de problème volumineux.

Références

- [1] A. Balakrishnan and K. Altinkemer. Using a hop-constrained model to generate alternative communication network design. *ORSA J Comput*, 4 :192–205, 1992.
- [2] A. Bley. Node-disjoint length-restricted paths. *Masters thesis, TU Berlin*, 1997.
- [3] G. Dahl. The 2-hop spanning tree problem. *Oper Res Lett*, 23 :21–26, 1998.
- [4] G. Dahl. Notes on polyhedra associated with hop-constrained walk polytopes. *Oper Res Lett*, 25 :97–100, 1999.
- [5] G. Dahl, N. Foldnes, and L. Gouveia. A note on hop-constrained walk polytopes. *Oper Res Lett*, 32 :345–349, 2004.
- [6] G. Dahl and L. Gouveia. On the directed hop-constrained shortest path problem. *Oper Res Lett*, 32 :15–22, 2004.
- [7] G. Dahl and B. Johannessen. The 2-path network design. *Networks*, 43(3) :190–199, 2004.
- [8] L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research*, 95 :178–190, 1996.
- [9] L. Gouveia. Using variable redefinition for computing lower bounds for minimum spanning and steiner trees with hop constraints. *INFORMS J Comput*, 10 :180–188, 1998.

- [10] L. Gouveia and C. Requejo. A new lagrangean relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research*, 132 :539–552, 2001.
- [11] D. Huygens and A. R. Mahjoub. Integer programming formulations for the two 4-hop-constrained paths problem. *Networks*, DOI 10.1002(2) :135–144, 2007.
- [12] D. Huygens, A. R. Mahjoub, and P. Pesneau. Two edge-disjoint hop-constrained paths and polyhedra. *SIAM J. Discrete Math.*, 18(2) :287–312, 2004.
- [13] H. Pirkul and S. Soni. New formulations and solution procedures for the hop constrained network design problem. *European Journal of Operational Research*, 148 :126–140, 2003.

Polyèdre des Arbres d'Huffman

Thanh Hai NGUYEN¹, Jean-François MAURRAS et Viet Hung NGUYEN²

1. *Laboratoire d'Informatique Fondamentale,
163 av Luminy 13009 Marseille
{Thanh.Hai.Nguyen, Jean-Francois.Maurras}@lif.univ-mrs.fr*

2. *LIP6 – Université Pierre et Marie Curie – Paris 6,
4 place Jussieu, 75252 Paris Cedex 05.
Hung.Nguyen@lip6.fr*

Mots-clefs : Polyèdre combinatoire, enveloppe convexe, arbre d'Huffman.

1 Introduction

Dans cette présentation, nous étudions l'enveloppe convexe des arbres binaires à racine sur n feuilles. Ce sont des arbres d'Huffman ayant n feuilles. Rappelons tout d'abord qu'un arbre d'Huffman est utilisé principalement pour représenter l'alphabet d'une langue écrite. On attribue aux caractères un certain poids qui est souvent sa fréquence dans la langue en question. La construction de l'arbre d'Huffman consiste tout d'abord à considérer les caractères comme les arbres d'un seul nœud. On remplace ensuite 2 arbres du plus petit poids en créant un nouvel arbre acceptant ces 2 arbres comme fils. Le poids du nouvel arbre est la somme des poids de ses deux fils. On répète ce processus jusqu'à ce qu'il reste un seul arbre. Chaque caractère donc peut-être codé par une séquence, composée de 0s et de 1s, qui représente le chemin unique de la racine vers ce caractère (0si on tourne à gauche, 1 si on tourne à droite). Le codage de Huffman est optimal dans le sens où il minimise la fonction linéaire $\sum_{i=1}^n p_i \times l_i$, où p_i est le poids du caractère i et l_i est la longueur en 0s et en 1s de son codage (ici l_i est variable).

On choisit la façon suivante pour associer un point de l'espace \mathbf{Z}^n à un arbre d'Huffman A donné. Les variables de l'espace dans lequel nous allons décrire ce polyèdre sont indicées par les caractères. À l'arbre A donné, nous associons le point de l'espace dont la coordonnée associée au caractère c est la longueur de son codage déterminé par A . Pour les quatre caractères $\{a, b, c, d\}$, quelques soient les poids attribués, on a 13 arbres Huffman possibles dont les points correspondants sont les suivants :

$(3, 3, 2, 1), (3, 3, 1, 2), (3, 2, 3, 1), (3, 2, 1, 3), (3, 1, 3, 2), (3, 1, 2, 3), (2, 3, 3, 1), (2, 3, 1, 3), (2, 1, 3, 3), (1, 3, 3, 2), (1, 3, 2, 3), (1, 2, 3, 3), (2, 2, 2, 2)$.

Nous notons \mathbb{EH}_n l'ensemble de ces points représentant tous les arbres Huffman possibles pour n caractères.

Nous nous intéressons dans cet exposé à la description de l'enveloppe convexe des points de \mathbb{EH}_n . Nous appelons PAH^n ce polyèdre. Nous rappelons l'algorithme de construction progressive d'un arbre Huffman en dimension $n + 1$ à partir d'un arbre en dimension n , ce qui nous permet d'établir les liens entre les facettes en dimension n et celles en dimension $n + 1$. Nous donnons alors quelques familles d'inégalités du polyèdre d'Huffman. Parmi celles-ci, il y a une famille de facettes dont les coefficients forment une suite de Fibonacci. Les facettes que nous avons obtenues ne constituent qu'une description partielle du polyèdre d'Huffman en dimension n . La contribution principale de ce travail repose essentiellement sur les liens que nous avons établis entre la construction des arbres et celle des facettes, en dimension $n + 1$, à partir des arbres et des facettes en dimension n .

2 Polyèdre PAH^n

Théorème 2.0.1. *Les points dans \mathbb{EH}_n sont tous les sommets extrêmes de PAH^n .*

2.1 Facettes de Fibonacci

Théorème 2.1.1. *Soit α_i tels que $\alpha_1 = \alpha_2 = \alpha_3 = 1$, $\alpha_i = \alpha_{i-1} + \alpha_{i-2}$, $\forall i \geq 4$, l'inégalité suivante définit une facette de PAH^n :*

$$\sum_{i=1}^n \alpha_i \mathbf{x}_i \geq \alpha_{n+4} - 3. \quad \forall n \geq 3$$

Par permutation des coefficients croissants, on obtient une famille de $\frac{n!}{6}$ facettes de Fibonacci.

2.2 Facettes de Fibonacci 1-saut

Théorème 2.2.1. *Étant donnée la facette Fibonacci 0-saut de PAH^n , $\forall n \geq 4$:*

$$\sum_{i=1}^n \alpha_i \mathbf{x}_i \geq b_n = \alpha_{n+4} - 3.$$

où $\alpha_1 = \alpha_2 = \alpha_3 = 1$, $\alpha_i = \alpha_{i-1} + \alpha_{i-2}$, $\forall i \geq 4$.

– En dimension $n + 1$ l'inégalité :

$$f_{f1s}(\mathbf{x}) = \sum_{i=1}^{n-1} \alpha_i \mathbf{x}_i + \alpha_{n+1} \mathbf{x}_n + \alpha_{n+1} \mathbf{x}_{n+1} \geq$$

définit une facette de PAH^{n+1} .

$$b_{n+1} = \alpha_{n+5} + 2\alpha_{n-1} - 3.$$

2.3 Facettes de Fibonacci 1-saut Fibonacci

Théorème 2.3.1. *Étant donnée la facette Fibonacci 1-saut de PAH^n , $\forall n \geq 5$:*

$${}^n f(\mathbf{x}) = \sum_{i=1}^{n-2} \alpha_i \mathbf{x}_i + \alpha_n \mathbf{x}_{n-1} + \alpha_n \mathbf{x}_n \geq \alpha_{n+4} + 2\alpha_{n-2} - 3.$$

où $\alpha_1 = \alpha_2 = \alpha_3 = 1$, $\alpha_i = \alpha_{i-1} + \alpha_{i-2}$, $\forall i \geq 4$.

– En dimension $n + k$, $k \geq 1$, l'inégalité :

$${}^{n+k} f(\mathbf{x}) = \sum_{i=1}^{n-2} \alpha_i \mathbf{x}_i + \alpha_n \mathbf{x}_{n-1} + \alpha_n \mathbf{x}_n + \alpha_n (\alpha_4 \mathbf{x}_{n+1} + \dots + \alpha_{k+3} \mathbf{x}_{n+k})$$

définit une facette de PAH^{n+k} .

$$\geq \alpha_{n+k+4} + (\alpha_{k+5} - 1)\alpha_{n-2} - 3.$$

3 Polyèdre Dominant des arbres d'Huffman

3.1 Facettes 0-lifting de Fibonacci

Théorème 3.1.1. *Étant donnée la facette Fibonacci de PAH^n :*

$$\sum_{i=1}^n \alpha_i \mathbf{x}_i \geq b_n = \alpha_{n+4} - 3. \quad \forall n \geq 3$$

où $\alpha_1 = \alpha_2 = \alpha_3 = 1$, $\alpha_i = \alpha_{i-1} + \alpha_{i-2}$, $\forall i \geq 4$.

1 La famille de facettes Fibonacci de PAH^n est celle de $DPAH^n$

2 La facette Fibonacci de $DPAH^n$ est aussi celle de $DPAH^{n+k}$ avec $b_{n+k} = b_n + 1 = \alpha_{n+4} - 2$ et $\alpha_{n+k} = \alpha_{n+k-1} = \dots = \alpha_{n+1} = 0$.

Ces facettes de la famille Fibonacci nous permettent de **caractériser les sommets** ($\langle n - 1, n - 1, n - 2, n - 3, \dots, 3, 2, 1 \rangle$). Ce sont les sommets générés par l'arbre le plus profond.

4 Conclusion et Perspectives

Nous avons obtenu certains types de facettes qui permettent décrire partiellement le polyèdre PAH^n , et complètement le polyèdre $PAHt$, PAH^5 , PAH^6 , PAH^7 . Nous avons aussi trouver la génération de facettes en dimension $n + 1$ à partir de la dimension n . Nous espérons l'exploiter pour trouver un moyen de décrire complètement PAH^n vu que optimiser sur ce polyèdre peut se faire en temps polynômial.

Références

- [1] T. Christof, <http://www.zib.de/Optimization/Software/Porta/>, Logiciel Porta.
- [2] K. Fukuda, http://www.ifor.math.ethz.ch/fukuda/cdd_home/, Logiciel cdd, cdd+.
- [3] R. G. Gallager, *Information theory and reliable communications*, J. Wiley, 1968, pp. 45.
- [4] D. A. Huffman, *A method for the construction of minimum redundancy codes*, Proc. IRE, 40 (1951), pp. 1098 – 1101.
- [5] M. M. Kapranov, *The permutoassociahedron, Mac Lane's coherence theorem and asymptotic zones for the KZ equation*, (J. of Pure and Applied Algebre 85 (1993)). pp. 119 – 142.
- [6] Carl W. Lee, *The Associahedron and Triangulations of the n-gon*, (Europ. J. Combinatorics, 10 (1989)). pp. 551 – 560.
- [7] J. F. Maurras, *Programmation linéaire, complexité*, (Springer, 2002).
- [8] D. S. Parker et P. Ram, *The construction of Huffman codes is a submodular ('convex') optimization problem over a lattice of binary trees*, SIAM. J. Comput. 28(5), 1999, p. 2 – 5.
- [9] M. Pouzet, K. Reuter, I. Rival and N. Zaguia. *A generalized permutahedron*, Algebra Universalis, 34 (1995), pp. 496 – 509.
- [10] N. J. A Sloane, S. Plouffe, *The encyclopedia of integer sequences*, Academic Press, 1995.

Sur la qualité des colonnes générées dans un schéma de génération de colonnes

N. Touati¹, L. Létocart¹ et A. Nagih²

1. LIPN, UMR CNRS 7030, Institut Galilée, Université Paris 13,
av. J-B Clément, 93430 Villetaneuse
nora.touati@lipn.univ-paris13.fr
lucas.letocart@lipn.univ-paris13.fr
2. LITA, Université Paul Verlaine,
Ile du Saulcy 57045, Metz Cedex 1.
anass.nagih@univ-metz.fr

Mots-clefs : Génération de colonnes, diversification, stabilisation, problème de tournées de véhicules avec fenêtres de temps.

Il est bien connu que l'intensification des solutions dans un schéma de Génération de Colonnes (GC) [2] qui consiste à rajouter au Problème Maître (PM) plusieurs colonnes lors de chaque itération, permet de diminuer le nombre d'itérations. Malheureusement, cette intensification surcharge rapidement le PM alors que la base optimale ne contient qu'une très faible proportion de l'ensemble des colonnes générées. On rajoute alors au PM un grand nombre de colonnes inutiles. À cet effet, nous nous sommes proposés d'étudier les caractéristiques des colonnes générées afin d'établir un critère de sélection de bonnes colonnes pour diminuer la taille des problèmes maîtres, sans augmenter significativement le nombre d'itérations, dans le but de diminuer le temps de résolution global. Nous validerons notre étude sur le Problème de Tournées de Véhicules avec Fenêtres de Temps (PTVFT).

1 Intensification des solutions dans un schéma de génération de colonnes

Afin d'accélérer la résolution par la GC, il est avantageux d'insérer lors de chaque itération plusieurs colonnes sous-optimales par rapport au Sous-Problème (SP) dans le PM. Cette procédure est appelée Génération de Colonnes avec Intensification des Solutions (GCIS). Les expérimentations menées dans ce travail portent sur 4 instances de Solomon (dont le nombre de sommets varie entre 25 et 200) et sur 40 instances générées aléatoirement (dont le nombre de sommets varie entre 100 et 160) du PTVFT [3]. Les résultats exposés représentent les moyennes obtenues sur toutes ces instances. Les sous-problèmes sont résolus avec la programmation dynamique [1], elle permet de calculer plusieurs solutions réalisables.

Des résultats préliminaires sur les instances de test résolues avec la GCIS ont révélé deux phénomènes intéressants. D'un côté, le pourcentage de colonnes sous-optimales dans la base optimale est supérieur à 93%. L'ajout prématuré de colonnes au PM permet ainsi d'accélérer le calcul d'une base optimale, et donc de diminuer le nombre d'itérations. D'un autre côté, plus de 99% de l'ensemble des colonnes générées en moyenne, n'appartiennent pas à la base optimale. Nous calculons donc une grande proportion de colonnes inutiles. On peut déduire que parmi toutes les colonnes sous-optimales générées, il existe de bonnes colonnes (qui vont contribuer à la base optimale) et des colonnes inutiles qui ralentissent la résolution.

Afin d'éviter une rapide augmentation de la taille du PM, nous rajoutons généralement au PM les $k\%$ meilleures solutions lors de chaque itération, k étant un paramètre à fixer. Le

cas précédent est associé à $k = 100$. Nous appelons cette procédure la Génération de Colonnes $k\%$ Intensifiée (GC $k\%$ I). La figure 3 montre une étude comparative des procédures GC $k\%$ I, avec $k = 10, 20, 50$ et 100 . Le nombre total de colonnes générées augmente avec l'augmentation du paramètre k , alors que le nombre d'itérations et le temps de résolution global diminuent. La génération des $k\%$ solutions de meilleurs coûts réduits augmente considérablement le nombre d'itérations et le temps de résolution global lorsque k est petit.

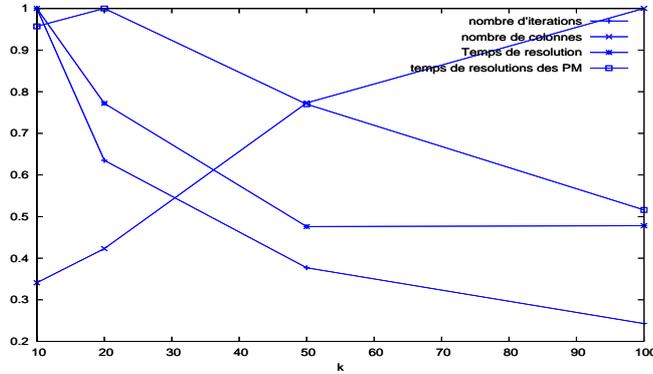


FIG. 3 – Intensification des solutions

1.1 Caractéristiques des colonnes sous-optimales

L'analyse des structures des colonnes sous-optimales obtenues en chaque itération des procédures ci-dessus, a montré que les colonnes sous-optimales d'un SP de meilleurs coûts réduits sont très voisines par rapport à la solution optimale, i.e. elles couvrent presque les mêmes sommets. La figure 4-(a) (resp. 4-(b)) illustre l'approximation du domaine réalisable du SP $Conv(X_B)$

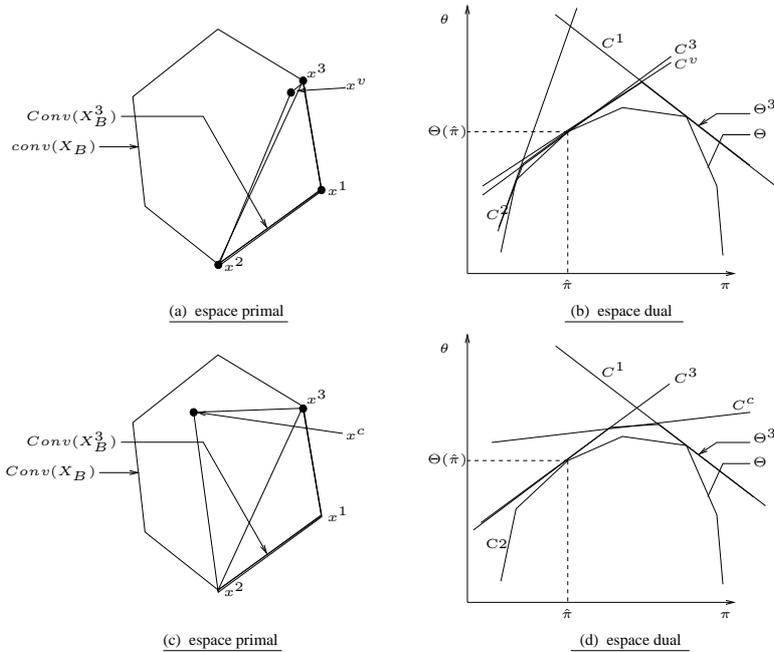


FIG. 4 – Caractéristiques des colonnes

(resp. la fonction duale Θ) noté $Conv(X_B^3)$ (resp. Θ^3), obtenue par les colonnes x^1, x^2 et x^3 (resp. C^1, C^2 et C^3). Soit x^v une solution voisine à x^3 . L'ajout de x^v à ce modèle n'apporte pas

d'informations pertinentes en plus de celles obtenues par x^3 . Du point de vue dual, la coupe ajoutée (C^v) est redondante.

Nous nous sommes ainsi intéressés à l'étude des caractéristiques de colonnes complémentaires i.e qui ne possèdent aucun sommet interne (i.e. à l'exception des sommets source et puits) en commun. La figure 4-(c) (resp. 4-(d)) illustre la même situation que la figure 4-(a) (resp. 4-(b)), mais au lieu d'ajouter une colonne voisine, nous rajoutons une solution complémentaire par rapport à x^3 , notée x^c . La complémentarité (resp. déviation) de la solution x^c (resp. la coupe C^c) par rapport à x^3 (resp. C^3) permet d'améliorer l'approximation $Conv(X_B^3)$ (resp. Θ^3). Dans le cadre de l'intensification des solutions, nous avons besoin de générer moins de colonnes, mais de bonne qualité, afin d'améliorer significativement l'approximation de la fonction duale. À cet effet, au lieu de sélectionner des colonnes à rajouter au PM en fonction de leur coût réduit (qui revient à rajouter des coupes redondantes), nous rajoutons des colonnes complémentaires. Nous présentons dans la suite l'impact de l'ajout de ces solutions sur les performances du schéma de GC.

1.2 Méthodes de diversification et résultats expérimentaux

Soit T^l l'ensemble des colonnes de coût réduit négatif calculées à l'itération l de la GC. La Génération de Colonnes avec Diversification par Sélection (GCDS), consiste de manière itérative, à sélectionner une colonne de coût réduit négatif x dans l'ensemble T^l , puis à supprimer dans cet ensemble toutes les colonnes possédant au moins un sommet interne dans x . Cette procédure est répétée jusqu'à ce qu'il n'existe aucun chemin de la source au puits. La figure 5 montre une étude comparative des différentes procédures. La GCDS réduit considérablement le nombre total de colonnes générées, nous ne générons avec cette méthode que 31% (resp. 61% et 39%) des colonnes générées par la GCIS (resp. GC_20%_I et GC_50%_I). Les PM sont plus petits et leur résolution plus facile. En moyenne, le gain obtenu sur le temps de résolution des PM par rapport à la GCIS est de 98%. Le nombre de colonnes sous-optimales générées lors de chaque itération avec la GCDS est très petit comparé à la GCIS, alors que leur coût est élevé. Même si cette méthode réduit le nombre de colonnes générées et le temps de résolution des PM, le nombre d'itérations augmente, afin de le diminuer, nous nous trouvons dans le besoin d'intensifier la génération de colonnes complémentaires.

méthode	GC_20_I	GC_50_I	GCIS	GCDS	GCDR
nombre d'itérations	95	56	36	68	60
nombre de colonnes	7466	11714	15147	4565	3976
temps global de résolution	75.82'	46.77'	46.97'	36.77'	35.07'
temps de résolution des PM	874.2''	673.2''	450.0''	7.6''	5.3''

FIG. 5 – Complémentarité des solutions dans la génération de colonnes

Nous proposons à cet effet la Génération de Colonnes avec Diversification par Résolution (GCDR), qui consiste itérativement à résoudre le SP sur le graphe courant (initialement le graphe de départ) et supprimer du graphe courant tous les sommets internes appartenant à la solution optimale obtenue. Cette procédure s'arrête lorsqu'il n'existe aucun chemin de la source au puits. La figure 5 montre que l'intensification de solutions complémentaires permet de réduire le nombre total de colonnes générées, nous générons avec la GCDR en moyenne 87% (resp. 26%) de celles de la GCIS (resp. GCIS). Cela diminue aussi le nombre d'itérations et le temps de résolution des PM ainsi que le temps de résolution global.

Références

- [1] R. Bellman, Dynamic programming, Princeton university press, NJ, USA, 1957.
- [2] G. Desaulniers, J. Desrosiers and M.M. Solomon, Column generation, Springer, New York, 2005.
- [3] G. Desrosiers, F. Soumis and M. Desrochers, Routing with time windows by column generation, Networks, 14 :545-565, 1984.

Meilleur et pire optimum pour des programmes linéaires avec second membres incertains

Nabila REMLI

*Université Paris Dauphine, LAMSADE
Place du Maréchal de Lattre de Tassigny,
F-75775 Paris Cedex 16, France
nabila.remli@lamsade.dauphine.fr*

Mots-clefs : meilleur optimum, pire optimum, incertitude, programme linéaire

Face à des problèmes contenant des paramètres incertains, il est nécessaire de développer des méthodologies qui intègrent les incertitudes dans le processus d'optimisation. Plus précisément, pour des programmes linéaires dont les coefficients sont incertains, il n'existe pas de solution optimale au sens habituel, et il faut définir et déterminer des solutions adéquates, que l'on nommera *solutions robustes* à ces programmes.

Dans cette étude, nous nous intéressons aux programmes linéaires dont les second membres sont incertains et modélisés par des intervalles d'incertitudes. Aucune distribution de probabilité n'est connue sur les paramètres incertains. Ces problèmes sont répandus en recherche opérationnelle. Nous pouvons citer en exemple, les problèmes de gestion de stock et de transport avec demandes incertaines, et des problèmes de flots avec capacités incertaines. La plupart des travaux existants sont dédiés à résolution de programmes linéaires incertains, notamment quand les coefficients de la fonction objectif ne sont pas connus avec précision et modélisé par des intervalles d'incertitude ou des scénarios discrets. Les approches utilisées s'inspirent de la théorie de la décision, utilisant le critère du pire cas ou le critère du regret maximum pour la détermination des solutions robustes [1]. Cependant, peu de travaux ont été, à notre connaissance, consacrés aux programmes linéaires admettant des second membres incertains. Nous citerons Chinneck et Ramadan [2] qui ont développé des algorithmes pour le calcul du meilleur et pire optimum.

Nous proposons dans cette étude une modélisation par intervalle des second membres incertains dans un programme linéaire. Les solutions du meilleur optimum et du pire optimum pour ces problèmes y sont présentés. Nous étudions également la complexité des problèmes obtenus. En admettant que pour toute valeur fixée des paramètres incertains (appelé scénario), le domaine réalisable est non vide, nous démontrons que le problème du calcul du meilleur optimum est polynomial quelque soit le type du programme linéaire, et que la recherche du pire optimum est un problème facile à résoudre dans le cas de contraintes d'inégalité ou quand il n'y a qu'une solution réalisable unique. Enfin, le calcul du pire optimum devient NP-difficile pour les programmes linéaires avec contraintes d'égalité.

Références

- [1] J.W. Chinneck and K. Ramadan (2000). *Linear programming with interval coefficients*. The Journal of the Operational Research Society, 51(2) :209-220.
- [2] P. Kouvelis and G. Yu. (1997). *Robust discrete optimization and its applications. non convex optimization and its applications*. Kluwer Academic Publishers.

Sur le polytope des *st*-chaînes

Viet Hung Nguyen

LIP6, Université Pierre et Marie Curie, Paris 6
Hung.Nguyen@lip6.fr

Mots-clefs : chemin non-orienté, chaîne, enveloppe convexe, facettes.

1 Introduction

Soit $G = (V, E)$ un graphe non-orienté où les arêtes sont pondérées par $w \in \mathbb{Q}^E$. Considérons les chaînes élémentaires dans G entre deux sommets spécifiques s et t . Ces chaînes seront appelées *st*-chaînes. Nous considérons le problème de la plus courte *st*-chaîne (par rapport à w). Ce problème est *NP*-difficile dans le cas général mais quand les circuits de G sont de poids non-négatifs, il est polynomial. On peut le ramener dans ce cas à un problème de couplage parfait de poids minimum dans un autre graphe dérivé de G [1]. Dans ce travail, nous nous intéressons à l'enveloppe convexe des vecteurs caractéristiques des *st*-chaînes que nous appelons sous le nom de *polytope des st-chaînes*. Notre contribution consiste essentiellement en deux points :

1. Présenter une large classe d'inégalités définissant des facettes du polytope des *st*-chaînes.
2. Montrer qu'un très petit sous-ensemble de cette classe d'inégalités suffit pour définir le programme linéaire "minimal" formulant le cas polynomial du problème de la plus courte *st*-chaîne.

2 *T*-joints

Un *T*-joint avec $T \subset V$ est un sous-graphe de G dont les sommets de degré impair sont exactement ceux de T . Quand $T = \{s, t\}$, les *T*-joints sont composés d'une *st*-chaîne et éventuellement de circuits supplémentaires. Lorsque les circuits de G sont de poids non-négatifs, on peut toujours trouver un *T*-joint de poids minimum qui est une *st*-chaîne. Le problème de trouver un *T*-joint de poids minimum est polynomial. On connaît une description complète de l'enveloppe convexe des vecteurs caractéristiques des *T*-joints [1], ce polytope est décrit complètement par les inégalités suivantes :

$$x(\delta(U) \setminus F) - x(F) \geq 1 - |F| \text{ où } U \subseteq V, F \subseteq \delta(U), \text{ et } |U \cap T| + |F| \text{ impair} \quad (1)$$

$$0 \leq x_e \leq 1 \quad (2)$$

L'inégalité (1) est appelé *inégalité de blossom*. Le problème de séparation des inégalités de blossom peut être résolu en temps polynomial [2][3][4]. Le problème de la plus courte *st*-chaîne dans un graphe sans circuit négatif peut donc être résolu en temps polynomial en résolvant le programme linéaire suivant :

$$\min w^t x$$

s.c.

$$x(\delta(U) \setminus F) - x(F) \geq 1 - |F| \text{ où } U \subseteq V, F \subseteq \delta(U), \text{ et } |U \cap \{s, t\}| + |F| \text{ impair} \quad (3)$$

$$0 \leq x_e \leq 1 \quad (4)$$

On peut obtenir une solution optimale issue de cette résolution qui pourrait contenir des circuits de poids zéro. En les enlevant on obtient la plus courte *st*-chaîne.

3 Facettes du polytope des *st*-chaînes

Les inégalités (3) sont donc valides pour le polytope des *st*-chaînes mais définissent-elles toutes, des facettes? Dans cet exposé, nous montrons que ces inégalités définissent des facettes du polytope des *st*-chaînes si et seulement si l'ensemble F est un couplage.

Nous montrons en outre que le système défini par les inégalités (3) et (4) n'est pas minimal pour résoudre le problème de la plus courte *st*-chaîne quand G ne comporte pas des circuits négatifs. On peut en effet utiliser seulement (4) et les inégalités (3) dont les ensembles F sont soit l'ensemble vide soit un singleton, c.à.d. une arête.

4 Applications

Notre travail est motivé par certains problèmes difficiles dont les solutions admettent les *st*-chaînes comme sous-structures. Un algorithme de résolution de type plan-coupant nécessite une recherche d'inégalités valides ainsi que des procédures de séparation. Une inégalité valide pour les *st*-chaînes est donc valide dans la projection des solutions sur le sous-espace des variables constituant une *st*-chaîne. Cette inégalité peut être liftée pour devenir une inégalité valide dans l'espace des variables du problème initial. On peut aussi "lifter" l'algorithme de séparation. Si on utilise les résultats bien connus des *T*-*joins* avec $T = \{s, t\}$, alors on peut appliquer ce principe aux inégalités (3). Évidemment on souhaiterait générer plusieurs inégalités violées à la fois, on risque que ces inégalités soient dominées par d'autres inégalités violées. On peut donc se limiter à générer seulement des inégalités (3) violées avec F couplage et éviter par conséquent de générer les inégalités superflues qui alourdissent le programme linéaire. Nous avons appliqué nos résultats avec succès sur le problème Ring-Star qui recherche un réseau de coût minimum dont le backbone est une *st*-chaîne et les autres noeuds sont des clients connectés à un des noeuds appartenant à cette *st*-chaîne [5].

Références

- [1] Schrijver, A. : Combinatorial Optimization : Polyhedra and Efficiency. Springer-Verlag, Berlin Heidelberg (2003).
- [2] Padberg, M.W. and Rao, M.R. : Odd minimum cut-sets and b-matchings. Math. Oper. Res. 7 (1982) 67-80.
- [3] Grötschel, M. and Holland, O. : A cutting plane algorithm for minimum perfect 2- matching. Computing 39 (1987) 327-344.
- [4] Letchford A.N., Reinelt G., and Theis, D.O. : A Faster Exact Separation Algorithm for Blossom Inequalities. D. Bienstock and G. Nemhauser (Eds.) : IPCO 2004, LNCS 3064, 196-205.
- [5] Nguyen, V.H. and Kedad-Sidhoum, S. : A New formulation for the Ring-Star problem. 2007. Submitted.

Sur les facettes des programmes en nombres entiers mixtes avec deux variables entières

G. Cornuejols^{1,2}

¹ *LIF, Marseille*

² *Tepper School of Business, Carnegie Mellon University, Pittsburgh PA, USA*
gerard.cornuejols@lif.univ-mrs.fr

Mots-clefs : Facettes, Programmes en nombres entiers mixtes

Ces résultats ont été obtenus en collaboration avec François Margot. Nous considérons une relaxation infinie du programme en nombres entiers mixte avec deux variables entières libres et plusieurs variables continues nonnégatives, et nous donnons une caractérisation complète de ses facettes. Nous dérivons une caractérisation analogue pour le programme en nombres entiers mixte sous-jacent.

Optimisation robuste et polyèdres

O. Klopfenstein^{1,2}

1. France Télécom R&D, 38-40 rue du gl Leclerc, 92130 Issy-les-Moulineaux
2. Université de Technologie de Compiègne, Laboratoire Heudiasyc UMR CNRS 6599,
60205 Compiègne Cedex
olivier.klopfenstein@orange-ftgroup.com

Mots-clefs : optimisation robuste, polyèdres

On considère un ensemble F d'événements pouvant survenir, et on lui associe l'ensemble $X(F)$ des solutions réalisables pour tout événement de F . L'ensemble $X(F)$ est donc l'ensemble des solutions *robustes* à tous les événements de F . La réalisabilité d'une solution x pour un événement ω est caractérisée par une fonction $g(x, \omega)$: si cette fonction est positive en (x, ω) , alors la décision x est réalisable pour l'événement ω .

On s'intéresse alors au problème d'optimisation robuste générique suivant :

$$\min_{x \in X(F)} \max_{\omega \in F} f(x, \omega)$$

où f désigne le coût de la solution x lorsque l'événement ω survient.

Lors de cette présentation, on tâchera de clarifier les liens existant entre un ensemble F d'événements et l'ensemble $X(F)$ des solutions correspondantes, sous différentes hypothèses sur g et f . On s'attardera notamment sur le cas des polyèdres. On décrira des cas où F polyédrique implique $X(F)$ polyédrique ; on mettra aussi en évidence des réciproques, permettant d'affirmer l'existence d'un ensemble polyédrique d'événements auquel un polyèdre de solutions est robuste.

Linear and Integer Programming vs Linear Integration and Counting

J.-B. Lasserre

LAAS-CNRS

7 Avenue du Colonel Roche, 31 077 Toulouse Cedex 4

lasserre@laas.fr

Mots-clefs : Linear and Integer Programming, Linear Integration and Counting

In this talk we analyze and develop some striking analogies between four related problems, namely linear programming and integer programming on the one hand and linear integration and counting on the other hand. The two former optimization problems are the max-plus algebra analogues of the two latter "integration" problems. The linear integration and counting problems have well-defined duals (although not qualified as such in the literature) obtained from the Laplace and Z-transforms, the analogues of the Fenchel transform to obtain the LP dual. A careful analysis of beautiful results by Barvinok, Brion and Vergne, sheds some interesting light on the links and analogies between the four problems, and in particular on duality for integer programs.

Coupes disjonctives pour la programmation mixte sous contraintes quadratiques

A. Saxena¹¹ P.Bonami²² et J. Lee³

1. Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA
anureets@andrew.cmu.edu

2. Laboratoire d'Informatique Fondamentale de Marseille,
CNRS-Université de Marseille, France
pierre.bonami@lif.univ-mrs.fr

3. IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA
jonlee@us.ibm.com

Mots-clefs : Optimisation Globale, Programmation non-linéaire en nombres entiers, Programmation entière sous contraintes quadratiques, Programmation disjonctive

Pour un résumé long de ces travaux le lecteur est invité à se reporter à [8].

Nous étudions des programmes mixtes sous contraintes quadratiques de la forme :

$$\begin{array}{ll}
 \min & a_0^T x \\
 \text{s.t.} & \\
 & x^T A_i x + a_i^T x + b_i \leq 0, \quad i = 1 \dots m; \\
 & x_j \in \mathbb{Z}, \quad j \in N_I; \\
 & l \leq x \leq u,
 \end{array} \tag{MIQCP'}$$

où N ($n = |N|$) est l'ensemble des variables, N_I l'ensemble des variables entières, A_i ($i=1, \dots, m$) sont des matrices symétriques $n \times n$ (en général non semi-défini positives), a_i ($i = 0, \dots, m$) l et u sont des vecteurs de \mathbb{R}^n et b_i ($i = 1, \dots, m$) sont des scalaires.

Les problèmes du type de (MIQCP') sont très difficiles car ils combinent deux types de non-convexité : des variables entières et des contraintes quadratiques non-convexes. Dans ces travaux nous proposons de construire des relaxations fortes de (MIQCP') en utilisant des techniques de programmation disjonctive et la méthodologie du lift-and-project.

Une approche standard pour dériver des relaxation convexes de (MIQCP') et d'introduire des variables représentant le produit de x_i par x_j : $y_{ij} = x_i x_j$. On obtient ainsi la formulation étendue suivante :

$$\begin{array}{ll}
 \min & a_0^T x \\
 \text{s.t.} & \\
 & A_i \cdot Y + a_i^T x + b_i \leq 0, \quad i = 1 \dots m; \\
 & x_j \in \mathbb{Z}, \quad j \in N_I; \\
 & l \leq x \leq u; \\
 & Y = x x^T.
 \end{array} \tag{MIQCP}$$

¹¹Ces travaux ont été effectués en partie lorsque le premier auteur visitait IBM T.J. Watson Research Center à Yorktown Heights qui est remercié pour son soutien. Aussi soutenu par la grant NSF DMI-0352885 et par la grant ONR N00014-03-1-0133

²²La recherche du second auteur a été effectuée en partie lorsque celui-ci travaillait au centre IBM T.J. Watson. Le second auteur est soutenu par le contrat ANR BLAN06-1-138894

Notons que la seule contrainte non-linéaire (et non-convexe) de **(MIQCP)** est $Y = xx^T$. Celle-ci peut être reformulée comme un paire de contraintes SDP : $Y - xx^T \succcurlyeq 0$ et $xx^T - Y \succcurlyeq 0$. La première de ces contraintes est une contrainte SDP convexe et peut être prise en compte efficacement par des méthodes de programmation semi-définie. La seconde contrainte, non-convexe, constitue l'objet principal de nos recherches.

Un aspect important des programmes non-convexes est que ceux-ci se composent souvent de contraintes non-convexes de structure simple liées par un ensemble de contraintes convexes. Dans de telles problèmes convexifier les contraintes non-convexes indépendamment les unes des autres n'amène pas à des relaxations fortes. Pour obtenir de telles relaxations, il est nécessaire d'utiliser des approches qui utilisent les contraintes liant les différentes sources de non-convexité. Ici une relaxation convexe classique de **(MIQCP)** est obtenue en relâchant les contraintes d'intégrité et en remplaçant la contrainte $y_{ij} = x_i x_j$ par son enveloppe convexe donnée par les inégalités de McCormick ou Sherali-Adams[7, 9]. Cette relaxation peut être renforcée par l'ajout de la contrainte SDP convexe $Y - xx^T \succcurlyeq 0$ [6, 10, 3, 1]. La relaxation ainsi obtenue est relativement faible car elle prend en compte chaque non-convexité séparément sans prendre en compte les contraintes liant ces non-convexités. Pour renforcer cette relaxation, nous proposons un algorithme de génération d'inégalités valides basé sur la programmation disjonctive de Balas [2]. Cette approche permet de dériver des inégalités prenant en compte la totalité du problème de manière à capitaliser sur les interaction entre les différentes contraintes.

Pour appliquer la programmation disjonctive, nous devons traduire les sources de non-convexités de **(MIQCP)** sous la forme de disjonctions. Pour les contraintes d'intégrité sur les variables x_j ($j \in N_I$) nous utilisons les méthodes qui ont été développées au cours des 30 dernières années. Notre principale contribution est de dériver aussi des disjonctions valides à partir de la contrainte $xx^T - Y \succcurlyeq 0$. Pour cela, nous analysons les valeurs propres et vecteur propres de la matrice $\hat{Y} - \hat{x}\hat{x}^T$. Ceci nous permet de dériver des inégalités simples de la forme $Y.c.c^T \leq (c^T x)^2$ qui sont à la base des disjonctions que nous utilisons. Nous proposons plusieurs méthodes pour dériver les inégalités du type $Y.c.c^T \leq (c^T x)^2$, en particulier nous introduisons la notion de *largeur* d'une disjonction et montrons que les disjonctions de petite largeur sont à même de donner des bonnes coupes pour les problèmes quadratiques. Basé sur cette observation nous proposons un modèle nous permettant de trouver de meilleures disjonctions.

Ainsi nous proposons 3 algorithmes permettant d'obtenir des relaxations de plus en plus fortes :

1. le premier algorithme résout la relaxation classique de **(MIQCP)** en approchant la contrainte SDP $Y - xx^T \succcurlyeq 0$ par des contraintes quadratiques convexes ;
2. Le second ajoute au premier des coupes disjonctives issues des valeurs propres strictement positives de la matrice $\hat{Y} - \hat{x}\hat{x}^T$;
3. le troisième ajoute au second des disjonctions diversifiées et de largeur minimale.

Les méthodes que nous développons ont été implémentées puis testées sur trois types d'instances : une sélection des problème de la GlobalLib[4], des exemples de problèmes proposés par Grossmann et Lee[5] et des instances de problèmes quadratiques non-convexes sous contraintes de bornes[11]. Nous présentons des résultats de calcul sur ces trois types d'instances.

Références

- [1] K. M. Anstreicher, Semidefinite Programming versus the Reformulation-Linearization Technique for Nonconvex Quadratically Constrained Quadratic Programming. Pre-print. Optimization Online, May 2007.
- [2] E. Balas, Disjunctive programming : properties of the convex hull of feasible points. *Disc. Appl. Math.*, 89 (1-3), 1998, 3-44.
- [3] S. Burer et D. Vandembussche. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. To appear in *Math. Programming*.
- [4] GLOBALlib, www.gamsworld.org/global/globallib/globalstat.htm
- [5] S. Lee et I. E. Grossmann. A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. *Computers and Chemical Engineering*. 25, 2001, 1675-1697.
- [6] S. Kim et M. Kojima. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optim. Methods and Software*. 15, 2001, 201-204.
- [7] G. P. McCormick. Computability of global solutions to factorable nonconvex programs : Part I Convex underestimating problems. *Math. Prog.*, 10, 1976, 147-175.
- [8] A. Saxena, P. Bonami et J. Lee. Disjunctive Cuts for Non-Convex Mixed Integer Quadratically Constrained Programs. IPCO2008, mai 2008, Bologne.
- [9] H. D. Sherali et W. P. Adams. A reformulation-linearization technique for solving discrete and continuous nonconvex problems. Kluwer, Dordrecht 1998.
- [10] H. D. Sherali et B.M.P. Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *J. Global Optim.* 22, 2002, 233-261.
- [11] D. Vandembussche and G.L. Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Math. Prog.*, 102(3), 2005, 559-575.

L'algorithme CCB : comment améliorer l'efficacité de la méthode de Benders

M. Minoux

*Université Paris-6
michel.minoux@lip6.fr*

Mots-clefs : Algorithme CCB

La méthode de Benders est une technique classique de résolution de problèmes mixtes difficiles, utilisable dans de nombreux contextes d'application : optimisation de réseaux, problèmes de logistique divers (par exemple combinant transport, localisation et stockage) etc. On propose une technique générale fondée sur le concept de 'Covering Cut Bundle' (CCB) pour rendre plus efficace la mise en oeuvre de cette méthode, en particulier lorsque la résolution du programme maître restreint, à chaque itération, est coûteuse en temps de calcul (ce qui est fréquent). Des résultats de calcul sur divers exemples d'application sont présentés.

Preuve constructive d'une relation min-max multicoupe/multiflot dans les graphes série-parallèles

D. Cornaz¹

1. Université Blaise Pascal - Clermont Ferrand II,
Complexe scientifique des Cézeaux, 63177 Aubière cedex
cornaz@isima.fr

Mots-clefs : Condition de coupe, circuits, relation min-max, série-parallèle, matroïdes.

1 Introduction

Soit $G = (V, E)$ un graphe.

Un ensemble d'arêtes $D \subseteq E$ est une *multicoupe* de G si c'est l'ensemble de toutes les arêtes dont les extrémités appartiennent à deux classes différentes d'une certaine partition de V . Si G est muni d'un vecteur de poids $w \in \mathbb{Z}^E$ sur les arêtes, le poids d'une multicoupe D est $w(D) = \sum_{e \in D} w_e$. On note $\max\text{-multicut}(G, w)$ le poids maximum d'une multicoupe de G avec les poids w .

Notons que le vecteur w induit une partition $\{R, E \setminus R\}$ de E où R est l'ensemble des arêtes dont le poids n'est pas négatif. Si l'on considère le sous-graphe $H = (V, E \setminus R)$ comme un réseau avec des capacités $|w_f|$ pour chaque arête $f \in E \setminus R$, alors l'ensemble R peut être considéré comme un ensemble de demande : pour chaque arête $e = uv \in R$, il y a une demande de flot entier de valeur w_e entre u et v . Bien sûr, il se peut que toute la demande ne puisse pas être acheminée dans le réseau. Mais il existe toujours un sous-ensemble $R' \subseteq R$ tel que les demandes dans $R \setminus R'$ puissent toutes être acheminées. Pour un tel vecteur R' , on dit que $\sum_{e \in R'} w_e$ est une *perte (de multiflot)*. On note $\min\text{-flowloss}(G, w)$ la perte minimum (de multiflot). Remarquons que $\min\text{-flowloss}(G, w)$ est égale à la somme des demandes moins le multiflot maximum.

2 L'algorithme

Nous donnons un algorithme qui, étant donné un graphe série-parallèle $G = (V, E)$ avec des coûts $w_e \in \mathbb{Z}$ pour chaque arête e , donne en temps polynomial une multicoupe de poids maximum et la perte minimum de multiflot. Cet algorithme fournit une preuve du fait que $\max\text{-multicut}(G, w) = \min\text{-flowloss}(G, w)$ pour tout $w \in \mathbb{Z}^E$ si et seulement si G est série-parallèle.

3 Condition de coupe et matroïdes

Le résultat ci-dessus généralise le fait que la condition de coupe implique l'existence d'un multiflot entier dans les séries-parallèles.. Ceci nous amène à considérer un troisième paramètre associé à (G, w) : La condition de coupe est qu'aucune multicoupe n'a un poids strictement positive. Bien sûr cela n'est pas toujours satisfait par (G, w) mais il existe toujours un sous-ensemble $R' \subseteq R$ dont la suppression nous donne que la condition de coupe est vérifiée. Notons $\min\text{-cutcond}(G, w)$ le minimum de $\sum_{e \in R'} w_e$ pour un tel R' . On a alors pour tout (G, w) les relations $\max\text{-multicut}(G, w) \leq \min\text{-cutcond}(G, w) \leq \min\text{-flowloss}(G, w)$.

Nous établissons que ces relations sont vraies pour tous les matroïdes (y compris ceux qui ne sont pas binaires).

4 Polyèdres et TDIness

La relation min-max obtenue grâce à l'algorithme nous donne un système linéaire TDI et permet de redémontrer le résultat de Chopra sur le polytope des multicoupes. Au niveau des matroides, on peut obtenir facilement une description du polytope des multicoupes des matroides uniformes (où multicoupe veut dire ici une union de cocircuits).

5 Conclusion

Les perspectives de ces travaux concernent les graphes signés. En effet, la condition de coupe implique l'existence d'un multiflot entier dans les graphes signés sans mineur K_4 impair. Ceci est impliqué par la caractérisation des hypergraphes binaires Mengeriens de Seymour. Mais peut-on étendre l'égalité max-multicut = min-flowloss aux graphes signés (G, R) sans mineur K_4 impair ?

Références

- [1] F. Barahona and A. R. Mahjoub, "On the cut polytope", *Mathematical programming* 36 (1986) 157-173.
- [2] F. Barahona and M. Grötschel, "On the cycle polytope of a binary matroid", *Journal of Combinatorial Theory, Series B* 40 (1986) 40-62.
- [3] S. Chopra, "The Graph Partitioning Polytope on Series-Parallel and 4-Wheel Free Graphs", *SIAM Journal on Discrete Mathematics* 7, No. 1 (1994), 16-31.
- [4] J. Edmonds and E.L. Johnson, "Matching, Euler tours and the Chinese postman", *Math. Programming* 5 (1973) 88-124.
- [5] M. Fischetti, J. J. Salazar, "Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control", *Mathematical programming* 84(2) (1999) 283-312.
- [6] M. Grötschel, *Cardinality homogeneous set system, cycles in matroids, and associated polytopes*, in *The Sharpest Cut*, MPS-SIAM Series on Optimization (2004), Ed. Grötschel.
- [7] A. Schrijver, *Combinatorial Optimization*, Springer-Verlag Berlin Heidelberg (2003).
- [8] P.D. Seymour, "Sums of circuits", in : *Graph Theory and related topics* Academic Press, New-York (1979) 341-355.
- [9] P.D. Seymour, "Matroids and multicommodity flows", *European Journal of Combinatorics* 2 (1981) 257-290.

Approximation de problèmes de couverture de tâches en transport ferroviaire

J. Sadki¹, L. Alfandari^{1,2}, A. Nagih³ et A. Plateau⁴

1. LIPN, UMR-CNRS 7030, Université Paris 13,
99 Av. J-B Clément 93430 Villetaneuse,
{jalila.sadki}{laurent.alfandari}@lipn.univ-paris13.fr

2. ESSEC,
Av. Bernard Hirsch B.P. 50105 95105 Cergy
alfandari@essec.fr

3. LITA, Université Paul Verlaine-Metz,
Ile du Saulcy 57045 Metz Cedex 1
anass.nagih@univ-metz.fr

4. CEDRIC, CNAM,
292 Rue Saint-Martin 75141 Paris
aplateau@cnam.fr

Mots-clefs : Programmation linéaire en nombres entiers, problème de couverture, approximation, planification ferroviaire.

1 Introduction

Le sujet de la communication est le problème ferroviaire dit d'affectation de locomotives (*locomotive assignment* [3]). Étant donné un ensemble de locomotives de différents types, et un ensemble de tâches de traction, le problème consiste à déterminer un enchaînement de tâches par locomotive de façon à couvrir l'ensemble des tâches de traction tout en minimisant le coût du déploiement des locomotives. Ce problème d'optimisation est NP-Difficile étant donné que le problème de couverture d'ensemble (*Set Covering Problem*) en est un cas particulier. Cette complexité est confirmée, d'un point de vue pratique, par les résultats d'expérimentations effectuées sur des jeux de données réels résolus avec la méthode de génération de colonnes, dont le problème maître se modélise comme un problème de couverture connu dans la littérature sous le nom *Covering Integer Programming* (CIP). Dans ce contexte, nous nous intéressons à la résolution approchée de ce problème ferroviaire via des heuristiques d'approximation basées sur une reformulation du problème maître.

2 Présentation du problème ferroviaire d'affectation de locomotives

Soit N l'ensemble des tâches à couvrir (trains à tracter), et Ω^k l'ensemble des chemins réalisables par type $k \in K$ de locomotives d'un réseau ferroviaire (un chemin réalisable est un enchaînement de tâches par une locomotive respectant certaines contraintes de ressources). Chaque train i nécessite un nombre b_i d'unités de puissance pour être tracté, de même chaque chemin p est associé à une locomotive d'un certain type k qui est caractérisé par un nombre w^k d'unités de puissance de traction. Un train n'est supposé couvert que si la puissance totale de l'assemblage de locomotives (*consist*) qui lui sont affectées est au moins égale à sa puissance de traction requise. L'objectif, dans ce problème est de couvrir toutes les tâches de traction

par des locomotives à moindre coût. Le problème maître issu de la décomposition de Dantzig-Wolfe, dans une approche de résolution par génération de colonnes, est donné par la formulation suivante :

$$(\mathcal{PM}) \begin{cases} \min & \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k y_p^k \\ \text{s.c} & \sum_{k \in K} \sum_{p \in \Omega^k} w^k a_{ip} y_p^k \geq b_i \quad \forall i \in N \\ & y_p^k \geq 0 \text{ entiers} \quad \forall k \in K, \forall p \in \Omega^k \end{cases}$$

La variable du problème y_p^k désigne le nombre de locomotives de type k empruntant le chemin p , c_p^k représente le coût d'un chemin réalisable $p \in \Omega^k$ et a_{ip} vaut 1 si la tâche i appartient au chemin p et 0 sinon.

Des expérimentations numériques comparatives ont été faites entre la méthode de génération de colonnes et une résolution directe par Cplex. Outre la difficulté liée à la taille des problèmes, elles montrent que les temps de résolution ainsi que les sauts d'intégrité (*gaps*) sont intrinsèquement liés aux contraintes de couverture pondérées, et justifient l'utilisation complémentaire d'heuristiques pour (\mathcal{PM}) qui est un cas particulier du problème (CIP) décrit dans la section suivante.

3 Présentation du problème générique de *Covering Integer Programming* (CIP)

Le problème (CIP) qui généralise le problème (\mathcal{PM}) ci-dessus est le suivant. Soit $\mathcal{C} = \{1, \dots, n\}$ un ensemble de n éléments, et $\mathcal{S} = \{S_1, \dots, S_m\}$ un ensemble de m sous-ensembles de \mathcal{C} . On associe à chaque sous-ensemble S_j de \mathcal{S} un coût $c_j \in \mathbb{N}$, à chaque élément i de \mathcal{C} une demande $b_i \in \mathbb{N}$ à couvrir, et à chaque couple (i, S_j) un poids $a_{ij} \in \mathbb{N}$.

Le problème *Covering Integer Programming* consiste à sélectionner une famille de sous-ensembles de \mathcal{S} , chaque sous-ensemble pouvant être pris plusieurs fois, de façon à couvrir à coût minimal chaque élément de \mathcal{C} . Il se formule comme suit :

$$(CIP) \begin{cases} \min & \sum_{j=1}^m c_j y_j \\ \text{s.c} & \sum_{j=1}^m a_{ij} y_j \geq b_i \quad \forall i \in \{1..n\} \\ & y_j \in \mathbb{N} \end{cases} \quad (1)$$

La variable y_j désigne le nombre de fois où un sous-ensemble S_j est sélectionné dans la solution optimale. La contrainte (1) assure la couverture de la demande de chaque élément i .

4 Approximation

L'heuristique gloutonne de Dobson [2] présente le meilleur rapport d'approximation possible pour la résolution du CIP, soit $H(\max_j(\sum_i a_{ij}))$ où H désigne la série harmonique ($H(n) \leq 1 + \ln(n)$). Elle consiste à sélectionner à chaque itération le sous-ensemble qui minimise le ratio coût sur couverture résiduelle, les poids a_{ij} et b_i sont mis à jour au fur et à mesure des itérations, et l'algorithme s'arrête dès que tous les éléments sont entièrement couverts. Cette heuristique généralise l'heuristique de Chvátal [1] pour le cas non pondéré.

Nous revisitons l'analyse au pire des cas de cette heuristique via une reformulation du CIP

en un problème de couverture d'ensemble particulier.

La résolution par cette heuristique du problème maître ferroviaire nous amène à résoudre à chaque itération r un problème auxiliaire associé à chaque sous-graphe $G^k = (N^k, A^k)$ du réseau de transport. Ce problème noté (\mathcal{SP}_k) vise à trouver un plus court chemin qui minimise la fonction ratio formulée comme suit :

$$(\mathcal{SP}_k) \begin{cases} \min & \frac{\sum_{(i,j) \in A^k} c_{ij} x_{ij}}{\sum_{(i,j) \in A^k, i \in N^k} \min(w^k, b_i^r) x_{ij}} \\ \text{s.c} & \\ & x \in \Omega^k, x \in \{0, 1\}^{|A^k|} \end{cases}$$

avec $x = (x_{ij})_{(i,j) \in A^k}$ et b_i^r la puissance résiduelle de la tâche i à l'itération r .

D'une façon générale, les problèmes combinatoires de minimisation de fonctions fractionnaires ($\min \frac{f(x)}{g(x)}$ où f et g sont des fonctions linéaires) sont NP-difficiles [6]. Il est connu que si le dénominateur $g(x)$ est positif alors le problème fractionnaire peut être résolu en temps polynomial si le problème linéaire paramétré associé ($\min f(x) - \lambda g(x)$ où λ est un paramètre réel) est solvable en temps polynomial [5]. Ce principe s'applique à notre problème auxiliaire de plus court chemin fractionnaire (*Min Ratio Path Problem*) dans le cas sans contrainte de ressource. Le problème ferroviaire est par conséquent approximable par l'heuristique de Dobson [2] au même rapport logarithmique soit $H(\max_{p \in \Omega^k} (\sum_{i \in p} \min(w^k, b_i)))$ [4]. D'autre part, Hashizume [7] ayant montré que, si le problème linéaire paramétré n'est pas polynomial mais approximable à rapport $1 + \epsilon$ alors le problème fractionnaire associé est aussi approximable au même rapport $1 + \epsilon$, nous en déduisons que, dans le cas avec une contrainte de ressource de type capacité ($\sum_{(i,j) \in A} v_{ij} x_{ij} \leq B$), le problème fractionnaire admet un schéma d'approximation en temps polynomial dérivé d'un schéma d'approximation du problème linéaire de plus court chemin contraint [8].

Références

- [1] V.Chvátal, A greedy-heuristic for the Set-Covering Problem. *Mathematics for Operations Research* 4 (1979) 233-235.
- [2] G.Dobson, Worst-case analysis of greedy heuristics for integer programming with non-negative data. *Mathematics for Operations Research* 7 :4 (1982) 515-531.
- [3] K.Ziarati, F.Soumis, J.Desrosiers, S.Gélinas, A.Saintonge. Locomotive assignment with heterogeneous consists at CN North America. *European Journal of Operational Research* 97 (1997) 281-292.
- [4] L.Alfandari, V. Paschos. Master slave strategy and polynomial approximation. *Computational optimization and Applications* 16 :3 (2000) 231-245.
- [5] N.Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research* 4 :4 (1979) 414-424.
- [6] A.Nagih. Problèmes fractionnaires : Tour d'horizon sur les applications et méthodes de résolutions. *RAIRO Operations Research*. 33 (1999) 383-419.
- [7] S.Hashizume, M.Fukushima, N.Katoh, T.Ibaraki. Approximation algorithms for combinatorial fractional programming problems. *Mathematical programming* 37 (1987) 255-267.
- [8] R.Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research* 17 :1 (1992) 36-42.

Testing Cut Generators for ILP

F. Margot

*Tepper School of Business, Carnegie Mellon University
Pittsburgh PA, USA
fmargot@andrew.cmu.edu*

Mots-clefs : Cut generators

This talk describes recent work on the testing of cut generators for integer linear programming. The goal of this work is to build experiments that can be used for comparing slight variants of a cut generator. Cut generators should be compared along several dimensions, for example the precision (what is the likelihood of a feasible solution to be cut?), the strength (are the cuts obtained stronger or weaker?), the speed of the generator, and the speed of the reoptimization after cuts are added.

While generator and reoptimization speeds are relatively easy to test, both precision and strength are more difficult to evaluate. Results for testing several variants of Gomory cut generators and several variants of reduce-and-split cut generators are presented.

The approach followed is in the spirit of the empirical analysis of algorithms advocated by Hooker and McGeoch, among others. The object of empirical analysis of algorithms is to use experiments to suggest hypotheses about the behavior of algorithms. Hypotheses are then tested using controlled experiments, well-grounded in statistical analysis.

Polytope des cliques circulaires et calcul du nombre d'indépendance des graphes quasi-adjoints

A. Pêcher¹

1. *Université de Bordeaux (LaBRI, INRIA),
351 cours de la Libération, 33405 Talence, France
arnaud.pecher@labri.fr*

Mots-clefs : graphes parfaits, colorations circulaires, graphes sans griffes

Un résultat majeur de la théorie des graphes est que le nombre chromatique d'un graphe parfait [1] est calculable en temps polynomial (Grötschel, Lovász et Schrijver [2, 3]). Le nombre circulaire chromatique est un raffinement du nombre chromatique usuel d'un graphe. Xuding Zhu a mis en évidence que les cliques circulaires sont le pendant naturel des cliques pour les colorations circulaires. Ceci lui a permis de définir élégamment les graphes circulaires-parfaits [5], une famille de graphes contenant strictement les graphes parfaits. La complexité du calcul du nombre circulaire chromatique d'un graphe circulaire parfait est inconnue.

Nous introduisons le polytope des cliques circulaires, un polytope dont les points extrémaux sont en correspondance avec les cliques circulaires induites d'un graphe, et nous montrons à l'aide de celui-ci et le résultat de Grötschel, Lovász et Schrijver déjà mentionné, que le nombre (circulaire) chromatique d'un graphe fortement circulaire-parfait est calculable en temps polynomial (les graphes fortement circulaire-parfaits étant une famille intermédiaire entre les graphes parfaits et les graphes circulaire-parfaits).

Une autre application, plus inattendue, de ce polytope est qu'en le combinant avec la caractérisation des facettes du polytope des stables d'un graphe proche-biparti de Shepherd [4], nous avons pu établir que le nombre d'indépendance d'un graphe quasi-adjoint est calculable en temps polynomial. Il s'agit de la première preuve de ce résultat de nature polyédrale, mais elle est limitée au cas non-pondéré. Pour la version pondérée, une compréhension plus fine du polytope des stables reste nécessaire.

Références

- [1] C. Berge, *Färbungen von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind*, *Wiss. Zeitschrift der Martin-Luther-Universität Halle-Wittenberg* 10 (1961) 114–115.
- [2] M. Grötschel, L. Lovász et A. Schrijver, *The Ellipsoid Method and its Consequences in Combinatorial Optimization*. *Combinatorica* 1 (1981) 169-197.
- [3] M. Grötschel, L. Lovász et A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag (1988).
- [4] F.B. Shepherd. *Applying lehman's theorem to packing problems* *Math. Program.* 71 (1995) 353–367
- [5] X. Zhu, *Circular Perfect Graphs* *J. of Graph Theory* 48 (2005) 186–209

Optimisation en présence de contraintes non holonomes

Witold Respondek

*INSA-Rouen, Laboratoire de Mathématique,
Place Emile Blondel BP 08
76131 Mont-Saint-Aignan Cédex
witold.respondek@insa-rouen.fr*

Mots-clefs : contraintes non holonomes, minimisation de l'action, approche variationnelle, approche mécanique

Le but de cet exposé est de discuter les systèmes mécaniques soumis à contraintes non holonomes. Nous allons définir et discuter d'abord la notion de contraintes non holonomes, ensuite le problème de commandabilité et principalement nous allons étudier comment établir les équations du mouvement. Nous allons illustrer les notions et les résultats par des exemples et nous allons aussi donner quelques remarques sur les systèmes non holonomes discrets.

Considérons un système mécanique dont l'espace des configurations (c'est à dire, des positions) Q , qui est un ouvert de \mathbb{R}^n (ou une variété différentiable de dimension n), soumis aux contraintes

$$\varphi(q) = 0,$$

où $\varphi : Q \rightarrow \mathbb{R}^k$. Ces contraintes ne dépendent que des configurations $q \in Q$. On dit alors qu'elles sont *holonomes* et on peut restreindre le système à

$$C = \{q \in Q : \varphi(q) = 0\},$$

qui, sous l'hypothèse de régularité de φ , est une sous-variété de Q .

Soit $TQ = \bigcup_{q \in Q} T_qQ$, où T_qQ est l'espace tangent à Q en q , qui est formé par toutes les paires (q, v_q) , où v_q sont les vecteurs de toutes les vitesses attachées en q . Considérons une application $\Phi : TQ \rightarrow \mathbb{R}^k$ et les contraintes définies par

$$\Phi(q, v) = 0$$

qui donnent alors un sous-ensemble

$$\mathcal{C} = \{(q, v) \in TQ : \Phi(q, v) = 0\}$$

de TQ qui, en général, dépend des configurations et des vitesses. Ces contraintes sont dites *non holonomes* si justement elles ne peuvent pas être représentées par une application qui dépendrait seulement des configurations.

Très souvent l'application Φ est linéaire par rapport aux vitesses $\Phi(q, v) = \Omega(q)v$. Dans ce cas en chaque point $q \in Q$ les vitesses admissibles forment un sous-espace linéaire $\mathcal{D}(q)$ de T_qQ définie par $\mathcal{D}(q) = \ker \Omega(q)$ et une trajectoire $q(t)$ est admissible (c'est à dire : respecte les contraintes $\Omega(q(t)) \cdot \dot{q}(t) = 0$) si et seulement si

$$\dot{q}(t) \in \mathcal{D}(q(t)).$$

Le champ des sous-espaces $\mathcal{D}(q)$, noté par \mathcal{D} , est appelé une distribution. Choisissons des champs de vecteurs g_1, \dots, g_m sur Q , où $m = n - k$, qui (localement) engendrent \mathcal{D} . Il est clair qu'une trajectoire $q(t)$ respecte les contraintes si et seulement si elle satisfait

$$\dot{q}(t) = \sum_{i=1}^k u_i(t)g_i(q(t)),$$

pour des fonctions $u_i(t)$ (appelées contrôles). Les contraintes non holonomes ne changent pas alors l'espace des configurations (qui reste toujours Q) mais réduisent l'espace des vitesses et, en conséquence, l'espace des trajectoires admissibles : en effet seules les trajectoires satisfaisant $\dot{q}(t) \in \mathcal{D}(q(t))$ sont admissibles.

Considérons un système mécanique sur Q et supposons qu'aucune contrainte ne soit imposée. Le principe de Hamilton énonce que le système se déplace de telle façon que sa trajectoire $\gamma = q(t)$ minimise la fonctionnelle d'action

$$J(\gamma) = \int_{T_0}^{T_1} L(q(t), \dot{q}(t), t) dt \longrightarrow \min,$$

où le *lagrangien* $L(q, \dot{q}) = E(q, \dot{q}) - V(q)$ est la différence entre l'énergie cinétique E et l'énergie potentielle V .

Les trajectoires minimisant J (plus généralement, les points critiques de la différentielle dJ) sont des solutions de l'équation d'Euler-Lagrange :

$$(EL) \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0.$$

La question que nous allons aborder est : comment décrire le mouvement du système s'il se déplace sous des contraintes non holonomes ?

Le premier sous-problème est le suivant : étant donné deux points dans Q , un point initial q_0 et un point final q_1 , existe-il toujours une trajectoire admissible joignant q_0 et q_1 ? La réponse est positive si les champs de vecteurs g_1, \dots, g_m et tous leurs crochets de Lie engendrent l'espace tangent $T_q Q$, en chaque point $q \in Q$. Supposons alors que cette condition, dite condition du rang de Lie, soit satisfaite.

Le deuxième sous-problème est donc : parmi toutes les trajectoires joignant q_0 et q_1 trouver celle qui réalise le mouvement physique. Deux approches sont envisageables : mécanique et variationnelle.

Il est connu que si un système mécanique, sans contraintes non holonomes, se déplace en présence d'une force extérieure F_{ext} sa trajectoire est donnée par la modification suivante de l'équation d'Euler-Lagrange :

$$(EL)_{ext} \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = F_{ext}.$$

On peut dire qu'un système non holonome se déplace sous l'action d'une force extérieure F_{ext} qui est la réaction des contraintes et alors $F_{ext} = \sum_{i=1}^k \lambda_i \omega^i$ où les ω^i sont les lignes de la matrice Ω définissant les contraintes. On obtient donc les équations du mouvement suivantes :

$$(MNH - EL) \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \sum_{i=1}^k \lambda_i \omega^i.$$

Cette formulation d'équations de mouvement sera appelée *le problème mécanique non holonome* et c'est le principe de d'Alembert qui conduit à ce système d'équations.

La deuxième approche est basée sur la minimisation de la fonctionnelle d'action J sur l'espace des trajectoires admissibles. En suivant la méthode de multiplicateurs de Lagrange nous formons le lagrangien étendu

$$\mathcal{L}(q, \dot{q}, t) = L(q, \dot{q}, t) - \sum_{i=1}^k \mu_i \omega^i \dot{q}$$

(comme dans le problème de la minimisation de $F : Q \longrightarrow \mathbb{R}$ sur $C = \{q \in Q : g^i(q) = 0\}$ nous considérons $\mathcal{F} = F - \sum_i \mu_i g^i$). Si une trajectoire $q(t)$ minimise le lagrangien étendu \mathcal{L} (plus généralement, est un point critique de sa différentielle) alors elle satisfait

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = 0;$$

Ce qui est équivalent à

$$(VNH - EL) \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \sum_{i=1}^k \dot{\mu}_i \omega^i - \mu_i (d\omega^i \dot{q}).$$

Ceci sera appelé *le problème variationnel non holonome*.

Il est alors naturel de se poser les trois questions suivantes :

- Les problèmes (MNH-EL) et (VNH-EL) ont-ils les mêmes solutions ?
- Sinon, quel est le problème dont les solutions décrivent les systèmes physiques (réels) ?
- Quelles sont les interprétations géométriques de (MNH-EL) et (VNH-EL) et de leurs solutions ?

La réponse à la première question est négative : en effet les dérivées $\dot{\mu}_a$ des multiplicateurs sont impliqués dans (VNH-EL) contrairement à (MNH-EL), où ils entrent de façon statique.

La deuxième question attire l'attention des chercheurs depuis la fin du XIX^{ème} siècle. Nous devons à D. Korteweg (1899) et même plus encore à H. Hertz (1894) une claire distinction des deux approches et la compréhension du fait que c'est l'approche mécanique non holonome (MNH-EL) qui décrit les systèmes physiques réels et que l'approche variationnelle n'est pas applicable aux systèmes mécaniques sous contraintes non holonomes. C'est Hertz aussi qui a proposé les termes *holonome* et *non holonome* en mécanique en s'inspirant des mot grecs $\acute{o}\lambda\omicron\varsigma$ (entier, intégral) et $\nu\omicron\mu\omicron\varsigma$ (nom, principe). Notons que l'approche variationnelle, même étant inapplicable en mécanique des systèmes non holonomes, a une place très importante en contrôle optimal et en géométrie sous-Riemannienne.

Pour répondre à la troisième question, supposons (pour la simplicité) l'absence de l'énergie potentielle V et alors le lagrangien L est égal à l'énergie cinétique E . S'il n'y a pas de contraintes non holonomes on peut donner deux interprétations aux solutions de l'équation d'Euler-Lagrange (EL) : elles sont les courbes les plus courtes (minimisant la distance défini par la métrique riemannienne correspondant à l'énergie cinétique) et elles sont aussi les courbes les plus droites (le long desquelles le système n'accélère pas). En présence de contraintes non holonomes ces deux propriétés ne coïncident plus. Justement les solutions de (VNH-EL) sont "les plus courtes" et les solutions de (MNH-EL) sont "les plus droites". Ce sont ces dernières qui réalisent le mouvement physique, et la mécanique non holonome est un des rares domaines de la physique qui n'obéit pas à un principe variationnel.

Références

[1] A.M. Bloch, (2003). *Nonholonomic mechanics and Control*. Springer-Verlag, New York-Heidelberg-Berlin.

[2] F. Bullo et A.D. Lewis, (2004). *Geometric Control of Mechanical Systems*. Springer-Verlag, New York-Heidelberg-Berlin.

Ordonnancement en Juste-à-Temps sur une Seule Machine

M. Rebai et I. Kacem¹

1. ICD-LOSI CNRS FRE, Université de Technologie de Troyes, France.

Mots-clefs : Ordonnancement, Méthode exacte.

Ce papier propose deux formulations linéaires différentes pour le problème d'ordonnancement d'un ensemble de N tâches sur une seule machine pour lequel, la minimisation du coût de retard et de l'avance est étudiée comme critère d'évaluation.

1 Première formulation mathématique

Nous présentons cette modélisation du problème en deux étapes. Dans la première étape, nous définissons les variables de décision ainsi que la fonction objectif et l'ensemble des contraintes liées au problème (modèle standard). Dans la deuxième étape, nous ajoutons un autre type de contraintes sous forme de coupes valides au modèle développé dans la première étape (modèle avec coupes).

➤ Modèle standard

w_i : pénalité de retard, h_i : pénalité d'avance, d_i : date due et p_i : durée opératoire

di^+ : déviation positive de la tâche i par rapport à sa date due ou également le retard

di^- : déviation négative de la tâche i par rapport à sa date due ou également l'avance

C_i : date d'achèvement de la tâche i

$$x_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ est affectée à la position } j \\ 0 & \text{sinon} \end{cases}$$

$$\text{Minimiser } Z = \sum_{i=1}^N w_i di^+ + h_i di^-$$

S/C

$$C_i + d_i^- - d_i^+ = d_i \quad i=1, \dots, N \quad (1)$$

$$\sum_{j=1}^N x_{ij} = 1 \quad i=1, \dots, N \quad (2)$$

$$\sum_{i=1}^N x_{ij} = 1 \quad j=1, \dots, N \quad (3)$$

$$C_i - M(1 - x_{ij}) \leq \sum_{k=1}^j \sum_{l=1}^N p_l x_{lk} \quad i=1, \dots, N \quad j=1, \dots, N \quad (4)$$

$$C_i + M(1 - x_{ij}) \geq \sum_{k=1}^j \sum_{l=1}^N p_l x_{lk} \quad i=1, \dots, N \quad j=1, \dots, N \quad (5)$$

$$x_{ij} \in \{0, 1\} \quad i=1, \dots, N \quad j=1, \dots, N$$

$$C_i, d_i^-, d_i^+ \in \mathbb{N} \quad i=1, \dots, N$$

(1) Détermine la déviation optimale de chaque tâche par rapport à sa date due.

(2) Exprime qu'une tâche ne peut occuper qu'une seule position.

- (3) Exprime qu'une position ne peut recevoir qu'une seule tâche.
- (4) Permet de déterminer une valeur de C_i . Si $x_{ij} = 1$, elle permet de majorer C_i par la somme des durées opératoires des jobs ordonnancés dans les positions $1, 2, \dots, j$. Sinon, elle est redondante.
- (5) Permet de déterminer une valeur de C_i . Si $x_{ij} = 1$, elle permet de minorer C_i par la somme des durées opératoires des jobs ordonnancés dans les positions $1, 2, \dots, j$. Sinon, elle est redondante.
- (2) et (3) Assurent la non interruption des tâches lors de leurs exécutions sur la machine.
- (4) et (5) permettent de déterminer une valeur de $C_i = \sum_{k=1}^{j-1} p_{[k]} + p_i$ lorsque $x_{ij} = 1$ (avec $p_{[k]}$ la durée opératoire du job ordonnancé à la position k). Dans le cas où $x_{ij} = 0$ ces contraintes deviennent redondantes.

Cette formulation a permis la résolution de quelques problèmes de petite taille en temps raisonnable (10 tâches). De plus, sa relaxation linéaire conduit dans la plupart des cas à une borne inférieure relativement faible. Pour ces raisons, nous avons pensé à introduire des coupes valides.

➤ **Modèle avec coupes valides**

Les coupes proposées sont sous forme d'inégalités basées sur des propriétés spécifiques au problème $I \parallel \sum w_i C_i$.

Proposition 1 (Kacem [4])

Soit $w = (w_1, w_2, \dots, w_N)$ un vecteur de N entiers positifs et soit $(t_i)_{1 \leq i \leq N}$ le temps de début d'exécution de la tâche i dans une séquence quelconque. Alors l'inégalité suivante est vérifiée : $\sum_{i=1}^N w_i (t_i + p_i) \geq WF_1(p, w)$

ou encore $\sum_{i=1}^N w_i C_i \geq WF_1(p, w)$ avec $WF_1(p, w)$ la valeur minimale du flowtime pondéré obtenue en appliquant la règle *WSPT* (proposée par Smith) au problème $I \parallel \sum w_i C_i$.

Proposition 2

Soit $w' = (w'_1, w'_2, \dots, w'_N)$ un vecteur de N entiers positifs et soit $P = \sum_{i=1}^N p_i$. On pose C_i le temps d'achèvement de la tâche i dans la séquence. Alors l'inégalité suivante est vérifiée : $\sum_{i=1}^N w'_i C_i \leq \sum_{i=1}^N w'_i (P + p_i) - WF_1(p, w')$.

Proposition 3

Soit $w = (w_1, w_2, \dots, w_N)$ un vecteur de N entiers positifs et soit $(t_i)_{1 \leq i \leq N}$ le temps de début d'exécution de la tâche i dans une séquence quelconque. Alors l'inégalité suivante est vérifiée : $\sum_{i=1}^N w_i (t_i + p_i) \leq WF_2(p, w)$

ou encore $\sum_{i=1}^N w_i C_i \leq WF_2(p, w)$ avec $WF_2(p, w)$ la valeur maximale du flowtime pondéré obtenu en appliquant la règle *WLPT* au problème $I \parallel \sum w_i C_i$.

Proposition 4

Soit $w' = (w'_1, w'_2, \dots, w'_N)$ un vecteur de N entiers positifs et soit $P = \sum_{i=1}^N p_i$.

On pose C_i le temps d'achèvement de la tâche i dans la séquence. Alors l'inégalité suivante est vérifiée : $\sum_{i=1}^N w'_i C_i \geq \sum_{i=1}^N w'_i (P + p_i) - WF_2(p, w')$.

En ajoutant au modèle proposé les coupes ci dessus nous obtenons la formulation suivante :

$$\text{Minimiser } Z = \sum_{i=1}^N w_i d_i^+ + h_i d_i^-$$

S/C

$$C_i + d_i^- - d_i^+ = d_i \quad i=1, \dots, N$$

$$\begin{aligned}
\sum_{j=1}^N x_{ij} &= 1 & i=1, \dots, N \\
\sum_{i=1}^N x_{ij} &= 1 & j=1, \dots, N \\
C_i - M(1 - x_{ij}) &\leq \sum_{k=1}^j \sum_{t=1}^N p_t x_{tk} & i=1, \dots, N \quad j=1, \dots, N \\
C_i + M(1 - x_{ij}) &\geq \sum_{k=1}^j \sum_{t=1}^N p_t x_{tk} & i=1, \dots, N \quad j=1, \dots, N \\
\sum_{i=1}^N w_i C_i &\geq WF_1(p, w) \\
\sum_{i=1}^N w'_i C_i &\leq \sum_{i=1}^N w'_i (P + p_i) - WF_1(p, w') \\
\sum_{i=1}^N w_i C_i &\leq WF_2 \\
\sum_{i=1}^N w'_i C_i &\geq \sum_{i=1}^N w'_i (P + p_i) - WF_2(p, w') \\
x_{ij} &\in \{0, 1\} & i=1, \dots, N \quad j=1, \dots, N \\
C_i, d_i^-, d_i^+ &\in \mathbb{N} & i=1, \dots, N
\end{aligned}$$

2 Deuxième formulation mathématique

L'idée de base de cette formulation consiste à discrétiser l'horizon de traitement des tâches en unités de temps (Dans notre cas l'horizon de traitement des tâches est $P = \sum_{i=1}^N p_i$). En choisissant la variable binaire X_{it} comme une variable de décision telle que :

$$X_{it} = \begin{cases} 1 & \text{si le traitement de tâche } i \text{ commence à l'instant } t \\ 0 & \text{sinon} \end{cases}$$

d_i^+ la déviation positive de la tâche i par rapport à sa date due,

d_i^- la déviation négative de la tâche i par rapport à sa date due,

w_i la pénalité de retard et h_i la pénalité d'avance.

Nous pouvons écrire le modèle linéaire suivant :

$$\text{Minimiser } Z = \sum_{i=1}^N w_i d_i^+ + h_i d_i^-$$

$$\begin{aligned}
&S/C \\
&\sum_{t=1}^{P-p_i+1} X_{it} = 1 & \text{pour tout } i = 1 \dots N \quad (1)
\end{aligned}$$

$$\sum_{i=1}^N \sum_{s=\text{Max}(t-p_i+1, 1)}^t X_{is} \leq 1 & \text{pour tout } t = 1 \dots P \quad (2)$$

$$\sum_{t=1}^{P-p_i+1} (t + p_i) \times X_{it} - d_i^+ + d_i^- = d_i + 1 & \text{pour tout } i = 1 \dots N \quad (3)$$

$$P = \sum_{i=1}^N p_i ; X_{it} \in \{0, 1\} ; d_i^+, d_i^- \in \mathbb{N} ;$$

(1) Chaque tâche i doit avoir un seul instant de l'intervalle $[1, P-p_i + 1]$ comme date de début.

(2) Pour exprimer qu'à un instant donné une et une seule tâche soit en cours de traitement sur la machine.

(3) Pour déterminer la déviation optimale de la tâche i par rapport à sa date due.

3 Résultats numériques et conclusions

Les instances faisant l'objet des tests sont de tailles respectives : 10, 15, 20, 25 et 30 tâches. Elles sont générées de la façon suivante : Pour chaque tâche i , nous affectons à la durée p_i , à la pénalité de retard w_i et à la pénalité d'avance h_i des entiers aléatoires de l'intervalle uniforme $[1,10]$. Ensuite, nous déterminons pour chaque tâche i sa date due d_i qui est un entier aléatoire de l'intervalle $[P(1 - T - R/2), P(1 - T + R/2)]$ où $P = \sum_{i=1}^N p_i$, $T \in \{0, 0.2, 0.6, 0.8, 1\}$ et $R \in \{0.2, 0.4, 0.6, 0.8\}$. T désigne le facteur du retard moyen alors que R représente l'écart relatif à la date due.

Analyse en fonction du facteur T

T	0,2	0,4	0,6	0,8	1
Ecart 1 (%)	72,6225	92,52	100	92,39	87,0475
Ecart 2 (%)	31,475	56,9375	99,1075	61,08	39,61
Ecart 3 (%)	0,4925	5,085	74,94	16,1975	1,24

Ecart 1 : Ecart moyen entre la borne inférieure obtenue par la relaxation linéaire du modèle sans coupes et la solution optimale.

Ecart 2 : Ecart moyen entre la borne inférieure obtenue par la relaxation linéaire du modèle avec coupes et la solution optimale.

Ecart 3 : Ecart moyen entre la borne inférieure obtenue par la relaxation linéaire du deuxième modèle et la solution optimale.

Au vue de ces résultats, nous pouvons faire les conclusions suivantes :

- L'introduction des coupes au premier modèle améliore la borne inférieure.
- L'amélioration paraît importante lorsque $T \neq 0,6$.
- La borne linéaire du second modèle est meilleure que celle du premier modèle.

Ces modèles ont été ensuite exploités dans un algorithme de branch and bound. Nous avons utilisé en plus de plusieurs bornes (certaines d'entre elles sont issues de la littérature et d'autres sont nouvelles). Les résultats préliminaires sont encourageants. La prochaine étape de cette étude sera l'amélioration du deuxième modèle et l'exploration des propriétés faciale du problème.

Références

- [1] Azizoglu M, Kondakci S, Kirca O. Bicriteria scheduling problem involving total tardiness and total earliness penalties. International Journal of Production Economics 1991; Vol 23; pp17–24.
- [2] Baker, K.R & Scudder, G.D. Sequencing with earliness and tardiness penalties: a review, Operation research 1990, Vol 38; pp 22-36
- [3] Held M, Karp RM. A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied Mathematics 1962; vol 10; pp 196–210.
- [4] Kacem I. Lower bounds for tardiness minimization on a single machine with family setup times. International journal of Operations Research, 2007; Vol. 4 No. 1; pp 18-31.
- [5] Liaw CF. A branch and bound algorithm for the single machine earliness and tardiness scheduling problem. Computers & Operations Research 1999; vol 26; pp 679 – 693.
- [6] Sakib A. M, Anup K. S. Single machine weighted earliness/tardiness penalty problem with a common due date. Computers & Operation Research 2001; vol 28; pp 649 – 669.
- [7] M'Halla R. Minimising total earliness and tardiness on a single machine using a hybrid heuristic. Computers & Operation Research 2006; vol 34; pp 3126 – 3142.
- [8] Ow, P.S. and Morton, E.T. The single machine early/tardy problem. Management science 1989; vol 35; pp 331-342.
- [9] Li G. Single machine earliness and tardiness scheduling. European Journal of Operational Research 1997; pp 546–558.

Sur le partitionnement des graphes en composantes unicycliques avec contraintes

W. Ben-Ameur¹, M. Hadji² et A. Ouorou³

1. *Institut National des Télécommunications,
9 rue Charles Fourier, 91011 Evry
Walid.Benameur@it-sudparis.eu*

2. *Institut National des Télécommunications,
9 rue Charles Fourier, 91011 Evry
Makhlouf.Hadji@it-sudparis.eu*

3. *France télécom R&D, Issy- les - moulineaux
Adam.Ouorou@orange-ftgroup.com*

Mots-clefs : Graphe unicyclique, Matroïdes, Couplage, Polyèdres, Optimisation des réseaux

Étant donné un graphe pondéré G , nous souhaitons dans un premier temps partitionner les sommets de G en plusieurs composantes connexes unicycliques de coût total minimum. Pour cela, on développe deux algorithmes polynômiaux, le premier se base sur la théorie des matroïdes alors que le deuxième est un algorithme de couplage maximum de poids minimum. On généralise ce problème en y ajoutant différentes contraintes (taille des cycles, contraintes d'appartenance d'un certain nombre de sommets aux cycles des composantes connexes, nombre maximum des composantes connexes). Une étude polyédrale est alors proposée. Plusieurs inégalités valides sont mise en évidence, et les faces induites par ces inégalités sont étudiées.

1 Deux algorithmes polynômiaux pour le partitionnement en composantes unicycliques

Étant donné un graphe pondéré et non orienté $G = (V, E)$, on souhaite répartir les sommets de G en plusieurs composantes unicycliques (contenant au plus un cycle), avec un coût total minimum. On propose alors deux algorithmes polynômiaux, le premier se base sur la théorie des matroïdes. Il est en effet facile de prouver que $M = (E, \mathcal{F})$, où

$$\mathcal{F} = \{I \subseteq E, I \text{ les composantes de } I \text{ contiennent au plus un cycle}\}$$

est un matroïde. Ceci suggère immédiatement un algorithme glouton qui nous donne la décomposition souhaitée.

L'autre approche consiste à construire d'abord un nouveau graphe biparti dont les sommets seront l'union de E et V . Les arêtes de ce nouveau graphe se définissent ainsi :

il existe une arête entre le sommet i et le sommet x , si et seulement si i est une extrémité de l'arête x . Le poids de l'arête ix , est le même que celui de l'arête initiale x dans G .

Une fois le graphe biparti construit, on calcule un couplage maximum de poids minimum. Le résultat est une décomposition en composantes unicycliques de coût total minimum.

2 Graphes unicycliques avec contraintes sur le cycle

On généralise le problème précédent en y rajoutant plusieurs types de contraintes : contraintes de taille des cycles (on interdit les cycles de taille inférieure à p), des contraintes d'appartenance d'un certain nombre de sommets aux cycles des composantes, ainsi qu'une contrainte sur le nombre maximum de composantes connexes.

Ce problème a des applications dans le domaine des télécommunications. Il a aussi des liens avec d'anciens problèmes rencontrés. En effet, deux cas particuliers sont à signaler. Par exemple, lorsque la taille minimale des cycles est $p = |V| - 1$, on obtient alors le problème du voyageur de commerce. Ceci implique alors clairement que notre problème est aussi NP-difficile. L'autre cas est lorsqu'on impose que chaque composante soit exactement un cycle, on obtient alors le problème de 2-factors, qui est un problème facile.

Pour résoudre ce problème, plusieurs classes d'inégalités valides sont mises en évidence, et les faces induites par ces dernières sont aussi étudiées. Des algorithmes de séparation ont été mis en oeuvre pour les séparer, et certains parmi eux sont polynômiaux. Au final, on propose un algorithme à plans coupants pour résoudre ce problème pour des instances de tailles moyennes.

Branchement local pour le problème de sac-à-dos multidimensionnel à choix multiples

Nawal Cherfi¹ et Mhand Hifi^{1,2}

1. CERMSEM, MSE, Université Paris 1, 106-112, Bd de l'Hôpital, 75013 Paris, France.
Nawal.Cherfi@malix.univ-paris1.fr

2. LaRIA, Université de Picardie Jules Verne, 5 rue du Moulin Neuf, 80000 Amiens, France.
hifi@laria.u-picardie.fr

Mots-clefs : problème de sac à dos, génération de colonnes, optimisation, branchement local.

1 Introduction

De nombreux problèmes d'optimisation combinatoire issus des applications réelles se modélisent comme des problèmes linéaires en nombres entiers (Wolsey [8]) ou des problèmes linéaires mixtes. La plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles, ce qui complique leur résolution de manière exacte et laisse place aux méthodes de résolution approchée par des heuristiques ou des méthaheuristiques.

Dans cet exposé, nous nous intéresserons à la résolution approchée du problème de sac-à-dos généralisé à choix multiple (noté MMKP) qui est une version plus générale du fameux sac-à-dos unidimensionnel (Balas and Zemel [2]). Le MMKP a plusieurs applications pratiques, comme l'allocation de ressources dans un système multimédia avec de multiples sessions (Akbar *et al.* [1], Hifi *et al.* [4], Khan [6], Moser *et al.* [7]). Une instance du problème est caractérisée par la donnée d'un vecteur capacité $R = (R^1, \dots, R^k, \dots, R^m)$, et d'un ensemble $J = (J_1, \dots, J_i, \dots, J_n)$ de variables divisées en n classes disjointes, où chaque classe J_i , $i = 1, \dots, n$, est composée d'un nombre fini d'objets $l_i = |J_i|$. Chaque objet j appartenant à une classe J_i est représenté par un profit v_{ij} et par un vecteur poids $r = (r_{ij}^1, \dots, r_{ij}^k, \dots, r_{ij}^m)$. Toutes les valeurs v_{ij} et r_{ij}^k sont positives.

Le but du MMKP consiste à attribuer au sac exactement un seul élément de chaque classe avec le maximum de profit sans violer les contraintes de capacités. Formellement, le MMKP peut être décrit sous la forme suivante :

$$\text{(MMKP)} \left\{ \begin{array}{l} \text{Maximiser} \quad U(x) = \sum_{i=1}^n \sum_{j=1}^{l_i} v_{ij} x_{ij} \\ \text{s.c.} \quad \sum_{i=1}^n \sum_{j=1}^{l_i} x_{ij} r_{ij}^k \leq R^k \quad k = 1, \dots, m \quad (1) \\ \sum_{j=1}^{l_i} x_{ij} = 1 \quad i = 1, \dots, n \quad (2) \\ x_{ij} \in \{0, 1\} \quad i = 1, \dots, n, j = 1, \dots, l_i. \end{array} \right.$$

D'un point de vue expérimental, la formulation originale du MMKP n'est pas souhaitable à utiliser ; spécialement, pour des instances du problème de taille large. En effet, le nombre des variables et des contraintes s'accroît rapidement suivant, respectivement, l'augmentation du nombre de classes et du nombre des objets de chacune d'elles. Ceci nous a ramené à introduire la technique de génération de colonnes dans notre méthode de résolution, notée (PA), qui a pour idée principale de faire coopérer une méthode d'arrondi et un algorithme exact. Elle s'appuie sur deux phases :

- (i) La première phase consiste à appliquer, de manière itérative, la génération de colonnes et arrondir la solution optimale de la relaxation en continu obtenue pour fixer une partie des variables
- (ii) La deuxième phase est appliquée pour résoudre de façon exacte la partie restreinte du problème.

Par la suite, nous combinons la procédure (PA) avec la méthode de branchement local. Cette dernière est une méthode de recherche par voisinage, proposée par Fischetti et Lodi dans [3] où les voisinages sont obtenus en introduisant des inégalités linéaires non-valides. La linéarité de ces contraintes ajoutées au problème offre la possibilité d’explorer les voisinages déduits en utilisant des solveurs commerciaux tel que “Cplex”. À l’origine le branchement local visait à améliorer l’aspect heuristique des solveurs des problèmes linéaires mixtes.

Dans notre technique de résolution nous introduisons la procédure (PA) dans un schéma de branchement local qu’on notera (BL), où elle sera utilisée comme boîte noire pour explorer les voisinages (fournir les solutions).

Enfin, nous présentons une version augmentée de l’approche (BL) dont l’idée essentielle est d’imiter une méthode de Branch and Bound (B&B) mais sans garantir l’optimalité des solutions obtenues. Cette version de l’approche consiste globalement à appliquer l’heuristique (BL) proposée à un ensemble de nœuds sélectionnés d’un arbre de branch-and-bound trancaturé.

2 Méthodologie

Dans cet exposé, nous proposons un algorithme de branch-and-bound trancaturé dans lequel certains nœuds de l’arbre développé sont résolus en utilisant la procédure de branchement local (BL). L’approche proposée peut se résumer dans les étapes suivantes :

1. Démarrer avec une solution réalisable.
2. Générer un ensemble de nœuds selon les critères de la séparation dans un B&B.
3. Appliquer l’heuristique (BL) à un ensemble de nœuds élites (par sélection).

L’heuristique que nous utilisons est basée sur la méthode du branchement local dans lequel une procédure d’arrondi combinée avec la génération de colonnes est utilisée pour produire des solutions réalisables. Elle peut se résumer dans les étapes suivantes :

1. Soit η un nœud actif en cours d’exploration contenant un programme linéaire restreint (ILP) associé au MMKP.
2. Soit \bar{x} une solution réalisable de (ILP) tel que $\bar{S} = \{j \in S : \bar{x}_j = 1\}$ alors effectuer une partition de l’espace de recherche en deux disjonctions :

$$\sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \in N \setminus \bar{S}} x_j \leq k$$
 (nœud gauche) ou $\sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \in N \setminus \bar{S}} x_j \geq k + 1$ (nœud droit)
3. Appliquer la procédure de génération de colonnes au problème (ILP) du nœud gauche.
4. Appliquer une procédure d’arrondi au problème (ILP) pour fixer une partie des variables, notée I1, et compléter la deuxième partie, notée I2, en utilisant un algorithme exact. Soit \bar{x}_2 la solution obtenue.
5. Mettre à jour la solution obtenue :
Si $(U(\bar{x}_2) > U(\bar{x}))$ **alors**
 $\bar{x} \leftarrow \bar{x}_2$, répéter les étapes 2-5 jusqu’à la vérification du test d’arrêt.
sinon utiliser une technique de diversification.

6. Créer de nouveaux nœuds en utilisant des stratégies spéciales de branchement et supprimer les nœuds dont la valeur objectif est inférieure ou égale à la meilleure solution obtenue.
7. Sortir avec la meilleure solution si le test d'arrêt est vérifié ou lorsque la liste des nœuds est vide ; répéter les étapes 2-6 dans le cas contraire.

La méthode est ensuite testée sur des instances de littérature et comparée aux résultats du solveur "Cplex" et une méthode de recherche locale réactive [4]. L'approche proposée améliore les résultats de la majorité de ces instances en un temps de calcul raisonnable.

Références

- [1] M. M. Akbar, M. W. H. Sadid, M. S. Rahman, M. A. H. Newton and M. Kaykobad. A Parallel Heuristic Algorithm for Multiple-Choice Multi-Dimensional Knapsack Problem, *Working Paper*, Department of CSE, BUET, Dhaka-1000, Bangladesh, 2005.
- [2] E. Balas and E. Zemel. An algorithm for large zero-one knapsack problem, *Operations Research*, vol. 28, pp.1130-1154, 1980.
- [3] M.Fischetti and A.Lodi.'Local branching', *Mathematical Programming*, Vol. 98, pp.23-47, 2003
- [4] M. Hifi, M. Michrafy and A. Sbihi, A reactive local search-based algorithm for the multiple-choice multi-dimensional knapsack problem. *Computational Optimization and Applications*, to appear.
- [5] H. Kellerer, U. Pferschy, D. Pisinger, *Knapsack Problems*, Springer Verlag, ISBN 3- 540-40286-1.
- [6] S. Khan, Quality adaptation in a multi-session adaptive multimedia system : model and architecture, *PhD Thesis*, Department of Electronical and Computer Engineering, University of Victoria, May 1998.
- [7] M. Moser, D. P. Jokanović and N. Shiratori. An algorithm for the multidimensional multiple-choice knapsack problem, *IEECE Transactions on Fundamentals of Electronics*, vol. 80, No 3., pp.582-589, 1997.
- [8] L. A. Wolsey, *Integer Programming*, Wiley-Interscience, 1998.

Une Approche Lagrangienne Appliquée aux Problèmes d'Alignement dans la Bioinformatique

S. Balev

*Université du Havre, LITIS EA 4108
BP 540, 76058 Le Havre - FRANCE
stefan.balev@univ-lehavre.fr*

Mots-clefs : programmation en variables binaires, relaxation lagrangienne, bioinformatique

1 Introduction

Une grande partie des problèmes bioinformatiques sont des problèmes d'alignement. Les problèmes d'alignement de deux séquences sont faciles. Ils sont traités par des algorithmes de programmation dynamique. Mais les problèmes d'alignement d'une séquence et une structure ou de deux structures sont NP-durs et nécessitent des algorithmes spécialisés de résolution exacte ou approchée.

Ici nous nous intéressons à un problème particulier d'alignement séquence-structure. La résolution de ce problème fait partie d'une classe de méthodes de prédiction de la structure des protéines à partir de leur séquence d'acides aminés. Nous présenterons une approche basée sur une formulation en variables binaires et une relaxation lagrangienne. Cette approche peut être appliquée sur d'autres problèmes d'alignement séquence-structure ou structure-structure.

2 Formulation du problème

Une structure est une description abstraite d'une famille structurale de protéines. C'est un ensemble ordonné de m blocs. Le bloc i a une longueur fixe l_i et représente une séquence contiguë d'acides aminés. Les blocs correspondent aux éléments conservés de la structure secondaire (hélices α et feuillets β). La structure contient également une description des interactions entre des paires d'acides aminés appartenant aux blocs. Ces informations sont regroupées dans un graphe de contact avec des sommets représentant les blocs et un ensemble d'arêtes E représentant des interactions entre eux.

Un alignement entre une séquence requête de longueur N acides aminés et une structure est une couverture de zones de la requête par les blocs de la structure. Les blocs doivent préserver leur ordre et ne peuvent pas se chevaucher (cf Fig. 6). Un alignement est complètement déterminé par les positions de début des blocs. Pour simplifier nos notations, nous utiliserons des positions relatives. Si la position absolue du bloc i est j , sa position relative est $j - \sum_{k=1}^{i-1} l_k$. Ainsi, les positions possibles de chaque bloc sont entre 1 et $n = N + 1 - \sum_{i=1}^m l_i$ et l'ensemble d'alignements possibles est $T = \{\pi = (\pi_1, \dots, \pi_m) : 1 \leq \pi_1 \leq \dots \leq \pi_m \leq n\}$. Il est facile de démontrer que le nombre d'alignements possibles est $|T| = \binom{m+n-1}{m}$, ce qui est un nombre énorme même pour des instances de petite taille.

Le score d'un alignement est défini par un ensemble de coefficients c_{ijkl} , $(i, k) \in E$, $1 \leq j \leq l \leq n$, où c_{ijkl} est le score généré par l'interaction entre les blocs i et k quand le bloc i est placé à la position j et le bloc k occupe la position l . Ainsi, le but est de trouver l'alignement de score minimum parmi tous les alignements de T .

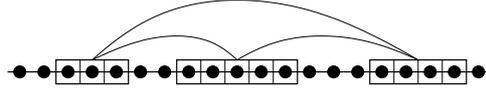


FIG. 6 – Exemple d’alignement d’une séquence requête de longueur 20 et une structure contenant 3 blocs de longueurs 3, 5 et 4

Soient y_{ij} , $i = 1, \dots, m$, $j = 1, \dots, n$ des variables binaires, telles que $y_{ij} = 1$ ssi le bloc i occupe la position j . Soit Y le polytope défini par les contraintes suivantes :

$$\sum_{j=1}^n y_{ij} = 1 \quad i = 1, \dots, m \quad (1)$$

$$\sum_{l=1}^j y_{il} - \sum_{l=1}^j y_{i+1,l} \geq 0 \quad i = 1, \dots, m-1, j = 1, \dots, n-1 \quad (2)$$

$$y_{ij} \geq 0 \quad i = 1, \dots, m, j = 1, \dots, n \quad (3)$$

Ainsi, $Y \cap B^{mn}$ décrit exactement l’ensemble des alignements possibles. Il est intéressant de noter que tous les sommets du polytope Y sont entiers [5] et donc on peut optimiser toute fonction linéaire des variables y sans imposer une contrainte d’intégralité de celles-ci.

Malheureusement, notre fonction de score ne peut pas être exprimée comme une fonction linéaire des variables y . En fait, il s’agit d’une fonction quadratique et non-convexe $f(y) = \sum c_{ijkl} y_{ij} y_{kl}$. Afin de linéariser cette fonction, nous introduisons des variables binaires $z_{ijkl} = y_{ij} y_{kl}$. Les variables z sont reliées aux variables y par les contraintes suivantes :

$$y_{ij} = \sum_{l=k}^n z_{ijkl} \quad (i, k) \in E, j = 1, \dots, n \quad (4)$$

$$y_{kl} = \sum_{j=1}^l z_{ijkl} \quad (i, k) \in E, l = 1, \dots, n \quad (5)$$

Avec ces contraintes supplémentaires le polytope correspondant n’est plus intégral et la relaxation linéaire ne donne pas la solution optimale du problème.

Différents modèles de programmation en variables binaires, plus ou moins proches de celui que nous venons de décrire ont été proposés [4, 1]. La résolution de ces modèles par la méthode de branch-and-bound standard utilisant des relaxations linéaires est relativement efficace. Typiquement l’arbre de recherche est très réduit, souvent ne contenant qu’un seul noeud. Mais pour des instances de grande taille le nombre de variables et de contraintes est tellement grand que même la relaxation linéaire devient très difficile à résoudre. C’est la raison pour laquelle nous utiliserons une relaxation lagrangienne qui est à la fois plus facile à résoudre et fournit des meilleures bornes.

3 Relaxation lagrangienne

Comme nous l’avons vu, les contraintes « difficiles » et donc candidats de relaxation sont (4) et (5). Si l’on relaxe toutes ces contraintes (cf Fig. 7b), la relaxation devient très facile à résoudre. Pour chaque interaction $(i, k) \in E$, il suffit de choisir des positions j et l qui minimisent le coût associé à z_{ijkl} et indépendamment de choisir les meilleurs positions de blocs par rapport

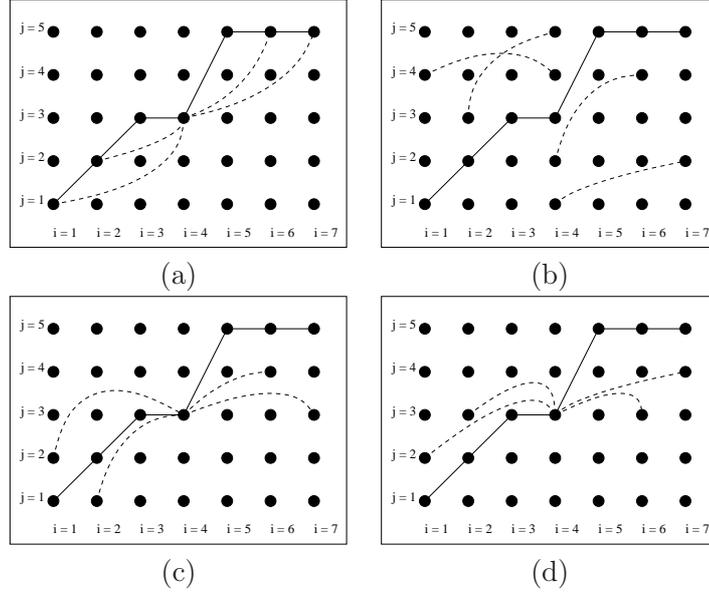


FIG. 7 – Exemple d’une instance du problème d’alignement avec $m = 7$ blocs et $n = 5$ positions libres. Les colonnes correspondent aux blocs et les lignes aux positions. L’ensemble d’interactions entre blocs est $E = \{(1, 4), (2, 4), (4, 6), (4, 7)\}$. Les lignes solides relient les points représentant les positions des blocs telles qu’elles sont définies par les variables y . Les lignes pointillées relient les points représentant les positions des blocs telles qu’elles sont définies par les variables z . (a) une solution satisfaisant toutes les contraintes. (b) les contraintes (4) et (5) sont relaxées. (c) seulement la moitié des contraintes (4) et (5) est relaxée (l’une des extrémités de chaque interaction $(i, k) \in E$). (d) comme (c), mais avec des contraintes de non-chevauchement des blocs sur les variables z .

aux variables y . La complexité d’une telle procédure est $O((m + |E|)n^2)$. On peut démontrer que dans ce cas la borne fournie par le dual lagrangien est égale à celle fournie par la relaxation linéaire.

Afin de serrer cette borne, nous ne relaxons qu’une partie des contraintes (4) et (5). Pour chaque interaction $(i, k) \in E$, nous choisissons l’une des extrémités (soit i , soit k) et nous la « détachons » en relaxant soit (4), soit (5) (cf Fig. 7c). Cette relaxation n’est pas plus difficile à résoudre que la précédente. En effet, considérons un bloc i placé à la position j . Par une simple procédure de programmation dynamique on peut trouver les positions optimales de toutes les blocs k , tels que (i, k) ou (k, i) est une interaction avec extrémité détachée k . En plus on peut imposer des contraintes de non-chevauchement sur ces blocs, comme le montre Fig. 7d. En prenant en compte ces contributions de score pour chaque point (i, j) , il est facile de trouver les positions optimales des blocs par rapport aux variables y . Pour une explication plus détaillée de l’algorithme de résolution de la relaxation lagrangienne, voir [2]. Cette relaxation est nettement meilleure que la précédente. En plus, la complexité de résolution reste la même, $O((m + |E|)n^2)$. Dans le pire des cas $|E|$ est d’ordre m^2 , mais pour les problèmes pratiques $|E|$ est d’ordre m , ce qui donne une complexité $O(mn^2)$.

Nous résolvons le dual lagrangien par une méthode d’optimisation sous-gradient standard [3] limitée à 500 itérations. À chaque itération t nous résolvons la relaxation lagrangienne avec les valeurs courantes λ^t des multiplicateurs lagrangiens. Pour l’itération suivante, $\lambda^{t+1} = \lambda^t + \theta_t s^t$, où s^t est le sous-gradient de la fonction objective pour $\lambda = \lambda^t$ et θ_t est le pas qui diminue à chaque itération.

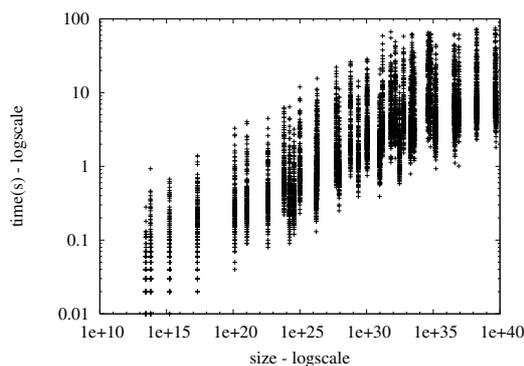


FIG. 8 – Temps d’exécution pour 9 136 instances du problème d’alignement en fonction de la taille de l’espace de recherche.

4 Résultats expérimentaux

Nous avons implémenté un algorithme branch-and-bound qui utilise les bornes fournies par le dual lagrangien et nous l’avons testé sur des problèmes d’alignement de taille différente. Fig. 8 montre le temps d’exécution de 9 136 alignements en fonction de la taille de l’espace de recherche $|T|$.

La solution optimale a été trouvée en moins d’une minute pour toutes les instances sauf 34. Pour 32 de celles dernières l’optimum a été trouvé en moins de 4 minutes et seulement pour deux instances l’optimum n’a pas été trouvé au bout d’une heure. Cependant, pour ces deux instances notre algorithme a fourni en une minute une solution sous-optimale avec un gap prouvé de la valeur objective de moins de 0,1%. Il est intéressant de noter que pour 79% des instances la solution optimale a été trouvée dans la racine de l’arbre de branch-and-bound. Autrement dit, la solution produite par le dual lagrangien est réalisable pour le problème original. On observe le même phénomène quand on utilise une relaxation linéaire. L’avantage de notre méthode est que le dual lagrangien est résolu beaucoup plus rapidement par notre algorithme dédié que la relaxation linéaire par la méthode de simplexe. En titre de comparaison, le temps moyen de résolution de la relaxation linéaire par CPLEX pour une instance de taille d’ordre 10^{38} est plus d’une heure, alors que le dual lagrangien est résolu en environ 15 secondes pour des instances de la même taille.

Références

- [1] R. Andonov, S. Balev, and N. Yanev. Protein threading : From mathematical models to parallel implementations. *INFORMS Journal on Computing*, 16(4) :393–405, 2004.
- [2] S. Balev. Solving the protein threading problem by lagrangian relaxation. In *Proceedings of WABI 2004 : 4th Workshop on Algorithms in Bioinformatics*, volume 3240 of *LNCS/LNBI*, pages 182–193. Springer-Verlag, 2004.
- [3] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.
- [4] J. Xu, M. Li, G. Lin, D. Kim, and Y. Xu. RAPTOR : optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology*, 1(1) :95–118, 2003.
- [5] N. Yanev, R. Andonov, P. Veber, and S. Balev. Lagrangian approaches for a class of matching problems in computational biology. *Computers and Mathematics with Applications*, 55 :1054–1067, 2008.

Formulations linéaires en nombres entiers pour des problèmes d'isomorphisme exact et inexact

P. Le Bodic^{1,2}, S. Adam², P. Héroux², A. Knippel¹ et Y. Lecourtier²

1. LMI/INSA de Rouen, Département Génie Mathématique
{pierre.le-bodic, arnaud.knippel}@insa-rouen.fr

2. Laboratoire d'Informatique, Traitement de l'Information et des Systèmes (LITIS),
Université de Rouen, Saint-Etienne-du-Rouvray
{sebastien.adam, pierre.heroux, yves.lecourtier}@univ-rouen.fr

Mots-clefs : isomorphisme de sous-graphe, programmation linéaire 0-1

1 Introduction

De nombreuses applications requièrent l'évaluation d'une mesure de similarité entre objets. En particulier, ces objets peuvent être des images, des textes, des molécules. Selon les cas, on peut vouloir les reconnaître, les classifier, ou effectuer des recherches parmi ceux-ci. Les applications sont par exemple la reconnaissance d'écriture manuscrite, l'analyse d'images et de vidéos, l'identification biométrique.

Les graphes sont une manière puissante de représenter ces objets. Dans cette représentation, les sommets sont généralement eux-mêmes des objets, tandis que les arcs ou les arêtes correspondent à des relations entre ceux-ci. Ainsi, les algorithmes identifiant des isomorphismes entre graphes permettent de mesurer la similarité de deux objets représentés par des graphes.

On distingue plusieurs problèmes d'isomorphisme de graphe. Dans tous les cas, les graphes peuvent être orientés, leurs arcs ou sommets étiquetés. On peut vouloir identifier un isomorphisme entre deux graphes, entre un graphe et un sous-graphe, ou trouver le plus grand sous-graphe commun à deux graphes. Pour une présentation des techniques et des applications relatives à l'isomorphisme de graphe, nous conseillons [1] au lecteur. Des formulations mathématiques ont déjà été proposées pour des problèmes proches : un programme linéaire 0-1 a par exemple été utilisé pour le calcul d'une distance d'édition entre deux graphes [2]. Une interprétation probabiliste d'une formulation quadratique continue a permis de traiter le problème d'isomorphisme de sous-graphe [3]. À notre connaissance, aucune formulation linéaire en nombre entiers n'a été proposée pour le problème de l'isomorphisme de sous-graphe.

On note $V_{\mathcal{X}}$ et $E_{\mathcal{X}}$ les ensembles de sommets et d'arcs d'un graphe \mathcal{X} . Étant donnés deux graphes $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ et $\mathcal{S} = (V_{\mathcal{S}}, E_{\mathcal{S}})$, le problème de l'isomorphisme de sous-graphe consiste à trouver un sous-graphe $\mathcal{G}' = (V_{\mathcal{G}'}, E_{\mathcal{G}'})$ de \mathcal{G} isomorphe à \mathcal{S} , avec $V_{\mathcal{G}'} \subseteq V_{\mathcal{G}}$ et $E_{\mathcal{G}'} \subseteq \{e = (v_1, v_2) \in E_{\mathcal{G}} / v_1, v_2 \subseteq V_{\mathcal{G}'}\}$.

Pour de nombreuses applications, des étiquettes sont associées aux sommets ou aux arcs. On cherche alors le sous-graphe \mathcal{G}' le plus proche de \mathcal{S} , au sens de la minimisation des différences d'étiquettes entre sommets associés et arcs associés. Nous présentons ici deux formulations sous forme de programmes linéaires 0-1 pour ce problème. Dans la variante de l'isomorphisme inexact, on recherche un sous-graphe \mathcal{G}' qui n'a pas nécessairement la même structure que \mathcal{S} . Nous exposons également une formulation autorisant ces différences de structure, les pénalisant au même titre que les différences d'étiquettes.

2 Deux formulations pour l'isomorphisme de sous-graphe

On considère deux graphes orientés \mathcal{G} et \mathcal{S} , dont les arcs et les sommets sont valués. Les formulations suivantes permettent de trouver un isomorphisme de sous-graphe, inexact au sens des étiquettes. On note $d(.,.)$ la distance entre deux sommets ou deux arcs, l'un pris dans \mathcal{S} , l'autre dans \mathcal{G} ; les distances sont des données du problème.

On utilise des variables de décision pour les sommets et les arcs. $x_{i,k} = 1$ signifie qu'on apparie le sommet $k \in V_{\mathcal{G}}$ au sommet $i \in V_{\mathcal{S}}$. De manière similaire, $y_{ij,kl} = 1$ si l'arc $kl \in E_{\mathcal{G}}$ est associé à l'arc $ij \in E_{\mathcal{S}}$.

(F1)

$$\text{Minimiser } \sum_{i \in V_{\mathcal{S}}} \sum_{k \in V_{\mathcal{G}}} d(i, k) * x_{i,k} + \sum_{ij \in E_{\mathcal{S}}} \sum_{kl \in E_{\mathcal{G}}} d(ij, kl) * y_{ij,kl} \quad (1)$$

$$\text{s.c. } \sum_{k \in V_{\mathcal{G}}} x_{i,k} = 1 \quad \forall i \in V_{\mathcal{S}} \quad (2)$$

$$\sum_{kl \in E_{\mathcal{G}}} y_{ij,kl} = 1 \quad \forall ij \in E_{\mathcal{S}} \quad (3)$$

$$\sum_{i \in V_{\mathcal{S}}} x_{i,k} \leq 1 \quad \forall k \in V_{\mathcal{G}} \quad (4)$$

$$y_{ij,kl} \leq x_{i,k} \quad \forall ij \in E_{\mathcal{S}}, \forall kl \in E_{\mathcal{G}} \quad (5)$$

$$y_{ij,kl} \leq x_{j,l} \quad \forall ij \in E_{\mathcal{S}}, \forall kl \in E_{\mathcal{G}} \quad (6)$$

$$y_{ij,kl} \geq x_{i,k} + x_{j,l} - 1 \quad \forall ij \in E_{\mathcal{S}}, \forall kl \in E_{\mathcal{G}} \quad (7)$$

$$x_{i,k} \in \{0, 1\} \quad \forall i \in V_{\mathcal{S}}, \forall k \in V_{\mathcal{G}} \quad (8)$$

$$y_{ij,kl} \in \{0, 1\} \quad \forall ij \in E_{\mathcal{S}}, \forall kl \in E_{\mathcal{G}} \quad (9)$$

Notons $n_{\mathcal{S}} = |V_{\mathcal{S}}|$, $n_{\mathcal{G}} = |V_{\mathcal{G}}|$, $m_{\mathcal{S}} = |E_{\mathcal{S}}|$, et $m_{\mathcal{G}} = |E_{\mathcal{G}}|$. (F1) utilise $n_{\mathcal{S}}n_{\mathcal{G}} + m_{\mathcal{S}}m_{\mathcal{G}}$ variables de décision binaires, et $n_{\mathcal{S}} + m_{\mathcal{S}} + 3m_{\mathcal{S}}m_{\mathcal{G}}$ contraintes.

On souhaite minimiser les distances entre chaque sommet (ou arc) du graphe \mathcal{S} et le sommet (respectivement arc) de \mathcal{G} associé à celui-ci. La fonction objectif (1) minimise donc la somme de ces distances. La contrainte (2) permet de s'assurer qu'à chaque sommet de \mathcal{S} est bien associé un et un seul sommet de \mathcal{G} . De la même manière, la contrainte (3) permet de s'assurer que chaque arc de \mathcal{S} est bien apparié à un et un seul arc de \mathcal{G} . La contrainte (4) contraint chaque sommet de \mathcal{G} à être associé à au plus un sommet de \mathcal{S} . La contrainte (5) implique que si un arc de \mathcal{S} est apparié avec un arc de \mathcal{G} , leurs sommets d'origine doivent également être appariés. Il en va de même avec leurs sommets d'arrivée, dont la contrainte (6) impose l'association. La contrainte (7) impose que deux arcs soient appariés si leurs extrémités sont appariées deux à deux.

On peut exprimer différemment les contraintes liant les variables de décision des sommets et celles des arcs :

$$\sum_{ij \in E_{\mathcal{S}}} y_{ij,kl} = x_{i,k} \quad \forall i \in V_{\mathcal{S}}, \forall kl \in E_{\mathcal{G}}$$

Cette relation se comprend de la manière suivante : pour un arc kl de $E_{\mathcal{G}}$ et pour un sommet i de $V_{\mathcal{S}}$, si i et k sont appariés, alors il y a exactement un arc issu de i apparié à kl . Si i et k ne sont pas appariés, alors aucun arc issu de i ne peut être associé à kl . Les relations suivantes

sont bâties selon le même principe :

$$\begin{aligned} \sum_{ij \in E_S} y_{ij,kl} &= x_{j,l} \quad \forall j \in V_S, \forall kl \in E_G \\ \sum_{kl \in E_G} y_{ij,kl} &= x_{i,k} \quad \forall k \in V_G, \forall ij \in E_S \\ \sum_{kl \in E_G} y_{ij,kl} &= x_{j,l} \quad \forall l \in V_G, \forall ij \in E_S \end{aligned}$$

Ces relations nous permettent de proposer la formulation suivante :

(F2)

$$\text{Minimiser } \sum_{i \in V_S} \sum_{k \in V_G} d(i,k) * x_{i,k} + \sum_{ij \in E_S} \sum_{kl \in E_G} d(ij,kl) * y_{ij,kl} \quad (10)$$

$$\text{s.c. } \sum_{k \in V_G} x_{i,k} = 1 \quad \forall i \in V_S \quad (11)$$

$$\sum_{kl \in E_G} y_{ij,kl} = 1 \quad \forall ij \in E_S \quad (12)$$

$$\sum_{i \in V_S} x_{i,k} \leq 1 \quad \forall k \in V_G \quad (13)$$

$$\sum_{kl \in E_G} y_{ij,kl} = x_{i,k} \quad \forall k \in V_G, \forall ij \in E_S \quad (14)$$

$$\sum_{kl \in E_G} y_{ij,kl} = x_{j,l} \quad \forall l \in V_G, \forall ij \in E_S \quad (15)$$

$$x_{i,k} \in \{0, 1\} \quad \forall i \in V_S, \forall k \in V_G \quad (16)$$

$$y_{ij,kl} \in \{0, 1\} \quad \forall ij \in E_S, \forall kl \in E_G \quad (17)$$

(F2) utilise $n_S n_G + m_S m_G$ variables de décision binaires, et $n_S + 2m_S n_G$ contraintes.

Soit \mathcal{P}_1 (respectivement \mathcal{P}_2) le polyèdre des solutions de la relaxation continue de (F1) (respectivement (F2)).

Proposition *Nous démontrons que $\mathcal{P}_2 \subseteq \mathcal{P}_1$ et que cette inclusion est stricte au moins dans certains cas.*

Les premiers résultats numériques confirment la supériorité de la formulation (F2).

3 Isomorphisme inexact de sous-graphe

On propose ici une formulation robuste à l'absence dans \mathcal{G} de sommets ou d'arcs pouvant être associés à ceux du graphe \mathcal{S} . On procède par l'ajout direct dans \mathcal{G} des éléments non appariés de \mathcal{S} . Cet ajout est effectué à l'aide de variables de décision plutôt qu'en modifiant les données du problème.

On introduit les variables binaires $u_i \forall i \in V_S$ et $e_{ij} \forall ij \in E_S$. u_i est associée au sommet i de \mathcal{S} de la manière suivante : $u_i = 0$ si et seulement si un sommet k de \mathcal{G} est associé à i . Dans le cas contraire, pour pouvoir identifier un isomorphisme de \mathcal{S} dans \mathcal{G} , on considère que le sommet i se trouve également dans \mathcal{G} , ce qu'on traduit par $u_i = 1$.

Les variables e_{ij} revêtent la même signification pour les arcs que les variables u_i pour les sommets. Ils indiquent si un arc de \mathcal{S} a été assigné à un arc de \mathcal{G} ou si il a été ajouté dans \mathcal{G} .

L'ajout d'un sommet et d'un arc engendre un sur-coût. On note $c(\cdot)$ ce coût de création, pour les sommets et les arcs de \mathcal{S} .

On introduit (F1a), une formulation basée sur (F1) utilisant les variables u_i et e_{ij} :

(F1a)

$$\text{Minimiser } \sum_{i \in V_S} \sum_{k \in V_G} d(i, k) * x_{i,k} + \sum_{ij \in E_S} \sum_{kl \in E_G} d(ij, kl) * y_{ij,kl} + \sum_{i \in V_S} c(i) * u_i + \sum_{ij \in E_S} c(ij) * e_{ij} \quad (18)$$

$$\text{s.c. } \sum_{k \in V_G} x_{i,k} + u_i = 1 \quad \forall i \in V_S \quad (19)$$

$$\sum_{kl \in E_G} y_{ij,kl} + e_{ij} = 1 \quad \forall ij \in E_S \quad (20)$$

$$\sum_{i \in V_S} x_{i,k} \leq 1 \quad \forall k \in V_G \quad (21)$$

$$y_{ij,kl} \leq x_{i,k} \quad \forall ij \in E_S, \forall kl \in E_G \quad (22)$$

$$y_{ij,kl} \leq x_{j,l} \quad \forall ij \in E_S, \forall kl \in E_G \quad (23)$$

$$y_{ij,kl} \geq x_{i,k} + x_{j,l} - 1 \quad \forall ij \in E_S, \forall kl \in E_G \quad (24)$$

$$x_{i,k} \in \{0, 1\} \quad \forall i \in V_S, \forall k \in V_G \quad (25)$$

$$y_{ij,kl} \in \{0, 1\} \quad \forall ij \in E_S, \forall kl \in E_G \quad (26)$$

$$u_i \in \{0, 1\} \quad \forall i \in V_S \quad (27)$$

$$e_{ij} \in \{0, 1\} \quad \forall ij \in E_S \quad (28)$$

La fonction objectif ainsi que les contraintes de (F1a) correspondent à celles de (F1), adaptées si besoin est à l'aide des variables u_i et e_{ij} . Les contraintes de (21) à (26) sont identiques à celles de (F1).

La fonction objectif (18) reprend sur la première ligne les termes de la fonction objectif (1) de la formulation (F1). La deuxième ligne correspond aux coûts de création des éléments de \mathcal{S} qui n'ont été associé à aucun élément de \mathcal{G} . La contrainte (19) implique que si l'on n'associe pas un sommet k à chaque sommet i , alors il faut créer i dans \mathcal{G} . La contrainte (20) joue le même rôle pour les arcs ij .

De façon similaire, nous étendons la formulation (F2) pour prendre en compte les différences structurelles. Nous comparons ces formulations à partir de résultats numériques et envisageons les améliorations possibles.

Références

- [1] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004.
- [2] Derek Justice. A binary linear programming formulation of the graph edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8) :1200–1214, 2006. Fellow-Alfred Hero.
- [3] C. Schellewald and C. Schnorr. Probabilistic subgraph matching based on convex relaxation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 171–186, 2005.

Index

Adam S.	59
Alfandari L.	38
Balev S.	55
Ben-Ameur W.	50
Bonami P.	32
Botton Q.	13
Cherfi N.	52
Cornaz D.	36
Cornuejols G.	29
Fortz B.	13
G�rard Y.	9
Gouveia L.	13
H�roux P.	59
Hadji M.	50
Hifi M.	52
Kacem I.	46
Klopfenstein O.	30
Knippel A.	59
L�tocard L.	22
Lasserre J.-B.	31
Le Bodic P.	59
Lecourtier Y.	59
Margot F.	41
Maurras J.-F.	19
Michel S.	10
Minoux M.	35
Nagih A.	22, 38
Nguyen T. H.	19
Nguyen V. H.	19, 27
Ouorou A.	50
P�cher A.	42
Plateau A.	38
Rebai M.	46
Remli N.	26
Respondek W.	43
Sadki J.	38
Saxena A.	32
Touati N.	22
Vanderbeck F.	10

LISTE DES PARTICIPANTS

Absi Nabil

Ecole des Mines, St Etienne
absi@emse.fr

Aguilera Jorge

Ecole des Mines, Alès
jorge-antonio.aguilera-cabanas@ema.fr

Auger David

Télécom ParisTech
auger@enst.fr

Balev Stefan

LITIS, Le Havre
stefan.balev@univ-lehavre.fr

Ben-Ameur Walid

Institut TELECOM & Manag. SudParis, Evry
walid.benameur@it-sudparis.eu

Bendali Fatiha

LIMOS, Clermont-Fd
bendali@isima.fr

Bonami Pierre

LIF, Marseille
Pierre.Bonami@lif.univ-mrs.fr

Borne Sylvie

LIPN, Paris 13
sylvie.borne@lipn.univ-paris13.fr

Botton Quentin

CORE, UCL, Louvain-la-neuve
quentin.botton@uclouvain.be

Bouallagui Sarra

LMI/INSA Rouen
sarra.bouallagui@gmail.com

Cao Xiaokang

UTC, Compiègne
caoxiaok@utc.fr

Casier Aurélie

ULB, Bruxelles
aciasier@ulb.ac.be

Cherif Nawal

Paris 1
nawcherfi@yahoo.fr

Cornaz Denis

LIMOS, Clermont-Fd
cornaz@isima.fr

Cornuejols Gérard

LIF Marseille, Carnegie Mellon University
gerard.cornuejols[at]lif.univ-mrs.fr

De Melo Rafael

UCL, Louvain-la-neuve
melo@core.ucl.ac.be

Diarrassouba Ibrahima

LIMOS, Clermont-Fd
diarrass@isima.fr

Didi Biha Mohamed

Université d'Avignon
mohamed.didi-biha@univ-avignon.fr

Fonlupt Jean

ECP6, Paris 6
Jean.Fonlupt@math.jussieu.fr

Fouilhoux Pierre

LIP6, Paris 6
pierre.fouilhoux@lip6.fr

Gay Jean-Christophe

LIMOS, Clermont-Fd
gay@isima.fr

Gérard Yann

LAIC, Clermont-Fd
gerard@laic.u-clermont1.fr

Gravot David

ROSTUDEL
dgravot@rostudel.com

Gueye Sérigne

LMAH, Le Havre
serigne.gueye@univ-lehavre.fr

Hadjali Abdelkader

LMI/INSA, Rouen
hadjaliam@insa-rouen.fr

Hadji Makhoul

Institut Telecom SudParis, Evry
Makhoul.Hadji@it-sudparis.eu

Herrera Carlos

CRAM, Nancy
carlos.herrera@cram.uhp-nancy.fr

Hifi Mhand

UPJV, Amiens
hifi@u-picardie.fr

Hijazi Hassan

Orange Labs, LIF, Marseille
hassan.hijazi@yahoo.com

Ho Vincent

ULB, Bruxelles
vincent.ho@ulb.ac.be

Huynh Tuong Nguyen

Laboratoire d'Informatique de Tours
nguyen.huynh@etu.univ-tours.fr

Itmi Mhamed

LITIS/INSA, Rouen
Mhamed.Itmi@insa-rouen.fr

Janaqi Stefan

Ecole des Mines, Alès
Stefan.Janaqi@ema.fr

Joncour Cédric

INRIA , Univ Bordeaux 1
cedric.joncour@math.u-bordeaux1.fr

Kacem Imed

UTT, Troyes
imed.kacem@utt.fr

Klopfenstein Olivier

Orange Labs, Paris
olivier.klopfenstein@orange-ftgroup.com

Knippel Arnaud

INSA, Rouen
Arnaud.Knippel@insa-rouen.fr

Kone Oumar

LAAS, Toulouse
okone@laas.fr

Lacroix Mathieu

LIMOS, Clermont-Fd
lacroix@isima.fr

Lannez Sébastien

LAAS, Toulouse
sebastien.lannez@sncf.fr

Lasserre Jean-Bernard

LAAS, Toulouse
lasserre@laas.fr

Le Bodic Pierre

INSA, Rouen
pierre.le-bodic@insa-rouen.fr

Lourimi Ali

PRINCE, Sousse, Tunisie
lourimi_ali@yahoo.fr

Mahjoub A. Ridha

LAMSADE, Paris Dauphine
mahjoub@lamsade.dauphine.fr

Margot François

Carnegie Mellon University
fmargot@andrew.cmu.edu

Martin Sébastien

LAMSADE, Paris Dauphine
martin@lamsade.dauphine.fr

Meurdesoif Philippe

INRIA, Univ Bordeaux 1
Philippe.Meurdesoif@math.u-bordeaux1.fr

Michel Sophie

INRIA Bordeaux / LMAH, Le Havre
michels@univ-lehavre.fr

Michrafy Mustapha

CES, Laboratoire Cermsem, Paris
michrafy@yahoo.fr

Minoux Michel

LIP6, Paris 6
michel.minoux@lip6.fr

Naudin Edith

EQUITIME, Gières
enaudin@equitime.fr

Nguyen Viet Hung

LIP6, Paris 6
Hung.Nguyen@lip6.fr

Nguyen Thanh Hai

LIF, Marseille
thanh.hai.nguyen@lif.univ-mrs.fr

Niu Yi-Shuai

LMI/INSA, Rouen
niuys@insa-rouen.fr

Ould Ahmed Mounir

Université d'Amiens
mohamed.ould@u-picardie.fr

Pêcher Arnaud

LABRI/INRIA, Bordeaux
arnaud.pecher@labri.fr

Pesneau Pierre

IMB/INRIA, Univ Bordeaux 1
pesneau@math.u-bordeaux1.fr

Rebaï Maher

UTT, Troyes
maher.rebai@utt.fr

Remli Nabila

LAMSADE, Paris Dauphine
nabila.remli@dauphine.fr

Respondek Witold

INSA, Rouen
witold.respondek@insa-rouen.fr

Sadki-Fenzar Jalila

LIPN, Paris 13
jalila.sadki@lipn.univ-paris13.fr

Skoda Alexandre

ECP6, Paris 6
askoda@math.jussieu.fr

Thabti Boulbaba

PRINCE, Sousse, Tunisie
b_thabti@yahoo.fr

Thiao Mamadou

LMI/INSA, Rouen
thiaoma@insa-rouen.fr

Touati Nora

LIPN, Paris 13
nora.touati@lipn.univ-paris13.fr

Vanier Sonia

Université Paris1
vanier@univ-paris1.fr

Zaourar Lilia

G-SCOP, Grenoble
lilia.zaourar@g-scop.inpg.fr