HAL
open science

Un algorithme de branch-and-cut pour la résolution du Problème de Tournées Sélectives

Racha El-Hajj, Duc-Cuong Dang, Aziz Moukrim

To cite this version:

Racha El-Hajj, Duc-Cuong Dang, Aziz Moukrim. Un algorithme de branch-and-cut pour la résolution du Problème de Tournées Sélectives. ROADEF2013, Feb 2013, Troyes, France. hal-00917089

HAL Id: hal-00917089

https://hal.science/hal-00917089v1

Submitted on 12 Dec 2013

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

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

Un algorithme de branch-and-cut pour la résolution du problème de tournées sélectives *

Racha El-Hajj[1,2] Duc-Cuong Dang[1] Aziz Moukrim[1]

[1] Université de Technologie de Compiègne, Département Génie Informatique
Laboratoire Heudiasyc, UMR 7253 CNRS, 60205 Compiègne, France
{duc-cuong.dang, aziz.moukrim}@hds.utc.fr

[2] Université Libanaise, Faculté de Génie, Département Contrôle Industriel
Campus Hadath, Beyrouth, Liban
racha.elhajj@hotmail.com

Mots-clés : *branch-and-cut, graphe d'incompatibilité, inégalités valides.*

Introduction

Nous nous intéressons à la résolution du problème de tournées sélectives (Team Orienteering Problem - TOP). Il s'agit d'une variante du problème de tournées de véhicules (VRP) où a priori il n'est pas possible de servir tous les clients. On considère pour ce problème une flotte de m véhicules, partant tous d'un point d et arrivant tous à un point a, et un ensemble de clients potentiels $\{1, ..., n\}$. A chaque client i est associé un profit P_i pouvant être collecté au plus par un véhicule de la flotte. Le problème est décrit par un graphe $G = (\{1, .., n\} \cup \{d, a\}, E)$ où un temps de trajet C_{ij} est associé à chaque arc (i, j) de E. La résolution du TOP consiste à organiser les visites pour la flotte afin de maximiser la somme des profits collectés tout en respectant un temps de trajet limite L imposé pour chaque véhicule.

A notre connaissance, trois méthodes exactes ont été élaborées pour résoudre ce problème. Butt et Ryan [2] ont proposé un algorithme de *génération de colonnes*. Boussier et al. [1] ont développé un algorithme de *branch-and-price*. Plus récemment, une résolution basée sur *un modèle pseudo-polynomial* a été introduite par Poggi et al. [6]. Ces méthodes ont été capables de résoudre un grand nombre d'instances de Chao et al. [3]. Cependant, plusieurs instances restent encore ouvertes. Dans cet article, nous présentons un schéma de *branch-and-cut* pour résoudre le TOP. Notre approche s'appuie sur un ensemble de propriétés de dominance et d'inégalités valides. Les résultats obtenus sur le benchmark [3] ont montré l'efficacité de notre approche permettant de résoudre à l'optimalité 27 instances encore ouvertes jusque là.

1 Formulation linéaire

Nous utilisons V^-, V^d, V^a et V pour désigner respectivement les ensembles $\{1, ..., n\}$, $V^- \cup \{d\}$, $V^- \cup \{a\}$ et $V^- \cup \{d, a\}$. Le TOP peut être formulé en programmation linéaire en nombres entiers (PLNE) avec les variables de décision y_{ir} et x_{ijr} : $y_{ir} = 1$ si le client i est servi par le véhicule r et 0 sinon ; $x_{ijr} = 1$ si l'arc (i, j) est utilisé par le véhicule r pour visiter le client i puis le client j et 0 sinon.

$$\max \sum_{i \in V^-} \sum_{r \in F} y_{ir} P_i \tag{1}$$

$$\sum_{r \in F} y_{ir} \leq 1 \quad \forall i \in V^- \tag{2}$$

*Ce travail est partiellement soutenu par le Conseil Régional de Picardie et le Fonds Européen de Développement Régional (FEDER), dans le cadre du projet PRIMA.

$$\sum_{j \in V^a} x_{djr} = \sum_{j \in V^d} x_{jar} = 1 \qquad \forall r \in F \tag{3}$$

$$\sum_{i \in V^a \setminus \{k\}} x_{kir} = \sum_{j \in V^d \setminus \{k\}} x_{jkr} = y_{kr} \qquad \forall k \in V^-, \forall r \in F \tag{4}$$

$$\sum_{i \in V^d} \sum_{j \in V^a \setminus \{i\}} C_{ij} x_{ijr} \leq L \qquad \forall r \in F \tag{5}$$

$$\sum_{(i,j) \in U \times U} x_{ijr} \leq |U| - 1 \qquad \forall U \subseteq V^-, |U| \geq 2, \forall r \in F \tag{6}$$

$$\begin{aligned} x_{ijr} &\in \{0,1\} \qquad \forall i \in V, \forall j \in V, \forall r \in F \\ y_{ir} &\in \{0,1\} \qquad \forall i \in V^-, \forall r \in F \end{aligned} \tag{7}$$

La fonction objectif (1) consiste à maximiser la somme des profits collectés. Les contraintes (2) assurent que chaque client est servi au plus par un véhicule, alors que les contraintes (3) et (4) garantissent la connectivité de chaque tournée. La limite imposée sur le temps de trajet est assurée par les contraintes (5), les sous-tours sont interdits par les contraintes (6) et les contraintes (7) traduisent celles d'intégrité.

2 Élimination des sous-tours

Soit S un ensemble de sommets, nous introduisons l'ensemble $\delta(S)$ pour désigner les arcs connectant les sommets de S avec ceux en dehors et l'ensemble $\gamma(S)$ pour désigner les arcs connectant les sommets de S entre eux. A la place de (6), les contraintes suivantes, appelées GSEC (Generalized Subtour Elimination Constraints) et ajoutées au modèle au besoin, assurent que chaque client servi par le véhicule r est connecté aux dépôts.

Propriété 2.1 *GSEC :*

$$\sum_{(u,v) \in \delta(S)} x_{uvr} \geq 2y_{ir}, \forall S \subset V, \{d,a\} \subseteq S, \forall i \in V \setminus S, \forall r \in F \tag{8}$$

Propriété 2.2 *Contraintes équivalentes aux GSEC :*

$$\sum_{(u,v) \in \gamma(S)} x_{uvr} \leq \sum_{i \in S} y_{ir} - y_{jr}, \forall S \subset V, \{d,a\} \subseteq S, \forall j \in V \setminus S, \forall r \in F \tag{9}$$

$$\sum_{(u,v) \in \gamma(U)} x_{uvr} \leq \sum_{i \in U} y_{ir} - y_{jr}, \forall U \subseteq V^-, \forall j \in U, \forall r \in F \tag{10}$$

3 Propriétés de dominance

Soit X une instance du TOP avec m véhicules, nous utilisons X_I pour désigner la même instance dans laquelle le profit de chaque client est considéré égal à 1. Nous définissons également X^g comme étant l'instance X dans laquelle le nombre de véhicules est réduit à g ($g \leq m$). D'autre part, nous utilisons $LB(X)$ (resp. $UB(X)$) pour désigner une borne inférieure (resp. supérieure) du TOP sur l'instance X. Nous avons alors les inégalités valides suivantes.

Propriété 3.1 *Elimination de la symétrie en se basant sur les profits :*

$$\sum_{i \in V^-} y_{i(r+1)} P_i - \sum_{i \in V^-} y_{ir} P_i \leq 0, \forall r \in F \setminus \{m\} \tag{11}$$

Propriété 3.2 *Bornes sur le profit :*

$$\sum_{r \in H} \sum_{i \in V^-} y_{ir} P_i \leq UB(X^{|H|}), \quad \forall H \subset F \tag{12}$$

$$\sum_{r \in H} \sum_{i \in V^-} y_{ir} P_i + UB(X^{m-|H|}) \geq LB(X), \quad \forall H \subseteq F \tag{13}$$

Propriété 3.3 *Bornes sur le nombre de clients qu'on peut servir :*

$$\sum_{r \in H} \sum_{i \in V^-} y_{ir} \leq UB(X_I^{|H|}), \quad \forall H \subset F \tag{14}$$

$$\sum_{i \in V^-} y_{ir} \geq LB(\bar{X}_I^1), \quad \forall r \in F \tag{15}$$

Les valeurs de LB dans (13) peuvent être calculées à partir d'une heuristique [4]. Les valeurs de UB dans (12), (13) et (14) sont obtenues à partir de la résolution de $X^{|H|}$ et $X^{m-|H|}$ en utilisant par exemple la PLNE. De même que pour les bornes inférieures LB_z nous cherchons seulement des bornes supérieures UB de qualité satisfaisante. Dans (15), $LB(X_I^1)$ est une borne inférieure obtenue en procédant à la résolution du modèle linéaire en nombres entiers décrit ci-dessus où il s'agit de minimiser la fonction objectif en rajoutant les contraintes (12) et (13) avec $|H| = 1$.

4 Graphes d'incompatibilité et coupes des cliques

Soit S un sous-ensemble de sommets de V^- (ou d'arcs de E), nous désignons par $MinTraj(S)$ la longueur du plus court chemin de d à a contenant tous les sommets (ou les arcs) de S. Le graphe d'incompatibilité entre les clients est défini par : $G_{V^-}^{Inc} = (V^-, E_{V^-}^{Inc})$ avec $E_{V^-}^{Inc} = \{[i,j] | i \in V^-, j \in V^-, MinTraj(\{i,j\}) > L\}$. De la même manière, le graphe d'incompatibilité $G_E^{Inc} = (E, E_E^{Inc})$ est tel que $E_E^{Inc} = \{[u,v] | u \in E, v \in E, MinTraj(\{u,v\}) > L\}$. Les propriétés suivantes sont valides.

Propriété 4.1 *Soit K (resp. Q) une clique dans $G_{V^-}^{Inc}$ (resp. G_E^{Inc}) :*

$$\sum_{i \in K} y_{ir} \leq 1, \forall r \in F \tag{16}$$

$$\sum_{(u,v) \in Q} x_{uvr} \leq 1, \forall r \in F \tag{17}$$

Les deux graphes $G_{V^-}^{Inc}$ et G_E^{Inc} sont construits par un prétraitement et sauvegardés pour chaque instance. Pour les inégalités (16) et (17), l'utilisation des cliques maximales est souhaitée. Dans notre approche, les cliques sont générées grâce à des décompositions des graphes d'incompatibilité. L'usage de ces décompositions est détaillé dans la prochaine section.

5 Algorithme de branch-and-cut

Prétraitements : Les bornes inférieures et supérieures utilisées dans les inégalités (12)-(15) sont calculées durant un temps limité à 5% du temps total de résolution. D'autre part, une décomposition du graphe $G_{V^-}^{Inc}$ en cliques maximales est effectuée en utilisant [5] pour ajouter au modèle les contraintes (16) associées.

Notre algorithme : De même que [1, 6], notre règle de branchement porte d'abord sur y_{ir} puis sur x_{ijr}. Lors de la première itération, le modèle linéaire contenant les contraintes (2)-(5), (7), (11)-(15) et (16) est résolu à l'optimalité pour obtenir une première solution. Si cette solution ne contient aucun sous-tour, alors elle est optimale et la résolution est terminée. Sinon, les contraintes (8), (9) et (10) sont ajoutées au modèle. De plus, les décompositions en cliques maximales des sous-graphes de $G_{V^-}^{Inc}$ et G_E^{Inc} induits par les sommets et arcs de la solution sont générées et les contraintes correspondantes (16) et (17) sont ajoutées au modèle.

6 Résultats numériques

Notre approche a été testée sur l'ensemble des instances de Chao et al. [3]. Cet ensemble contient 387 instances réparties sur 7 classes dont le nombre de clients et de véhicules peut atteindre respectivement 100 et 4. Notre algorithme est écrit en C++ et les calculs sont réalisés sur une machine AMD Opteron 2.60 GHz. Nous avons utilisé CPLEX 12.4 pour résoudre les modèles PLNE. De façon identique à [1, 6], le temps de résolution est limité à 2h.

Le tableau suivant montre le nombre d'instances résolues à l'optimalité après l'activation des différents composants de notre algorithme (propriétés ci-dessus). Ces composants ont été activés successivement de sorte que l'algorithme final est obtenu à la dernière activation. Nous remarquons que le nombre d'instances résolues augmente alors que le temps CPU diminue chaque fois qu'un nouveau composant est ajouté à l'algorithme. Notre méthode a pu résoudre 276 instances au total. En comparant à l'approche de [1] qui est capable de traiter 270 instances du benchmark, notre méthode est clairement efficace. De plus, nous avons pu résoudre à l'optimalité 27 instances encore ouvertes jusque là.

Composants activés	Instances résolues	CPU (en secondes)
Modèle standard	177/387	198.9
+ Élimination de sous-tours	233/387	61.75
++ Élimination de la symétrie	243/387	14.32
+++ Bornes sur le profit/nombre de clients	265/387	13.16
++++ Coupes des cliques	276/387	3.42

TAB. 1 – Performance de notre algorithme sur les instances de Chao et al. [3].

Conclusion et perspective

Nous avons présenté dans cet article un algorithme de branch-and-cut pour la résolution du TOP. Plusieurs propriétés permettant de renforcer la formulation linéaire classique ont été proposées. Elles sont basées sur l'élimination de la symétrie et des sous-tours ainsi que l'introduction de bornes sur le profit/nombre de clients et l'utilisation de coupes exploitant les cliques de graphes d'incompatibilité. Les expérimentations réalisées sur l'ensemble des instances de [3] montrent que notre algorithme est capable de résoudre un grand nombre d'instances de tailles variées. L'algorithme proposé permet également de fermer plusieurs instances. Ces résultats montrent l'efficacité de notre méthode pour la résolution du TOP.

Références

[1] S. Boussier, D. Feillet, and M. Gendreau. An exact algorithm for team orienteering problems. *4OR*, 5(3) :211–230, 2007.

[2] S. E. Butt and D. M. Ryan. An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers & OR*, 26 :427–441, 1999.

[3] I.-M. Chao, B. Golden, and E. A. Wasil. The team orienteering problem. *European Journal of Operational Research*, 88 :464–474, 1996.

[4] D.-C. Dang, R. N. Guibadj, and A. Moukrim. A PSO-inspired algorithm for the team orienteering problem. Submitted preprint, 2012.

[5] D.-C. Dang and A. Moukrim. Subgraph extraction and metaheuristics for the maximum clique problem. *Journal of Heuristics*, 18 :767–794, 2012.

[6] M. Poggi de Aragão, H. Viana, and E. Uchoa. The team orienteering problem : Formulations and branch-cut and price. In *ATMOS*, pages 142–155, 2010.