GROUPE POC
POLYÈDRES ET OPTIMISATION COMBINATOIRE

cnrs GDR Groupement de recherche
RO Recherche opérationnelle

JPOC 12

JOURNÉES POLYÈDRES ET OPTIMISATION COMBINATOIRE

24 et 25 juin 2021

INVITÉS

Amitabh Basu (USA)
Ricardo Fukasawa (Canada)
Maya Jakobine-Stein (Chili)
Laura Sanita (Canada)

COMITÉ D'ORGANISATION

Zacharie Ales (UMA, ENSTA - CEDRIC)
Mourad Baiou (LIMOS, CNRS)
Fatiha Bendali (LIMOS, Univ. de Clermont-Auvergne)
Ibrahima Diarrassouba (LMAH, Université Le Havre Normandie)
Pierre Fouilhoux (LIPN, Université Sorbonne Paris Nord)
Hela Haj Mohamed (FSM, Université de Monastir)
A. Ridha Mahjoub (LAMSADE, Université Paris-Dauphine)

HTTPS://WWW.LAMSADE.DAUPHINE.FR/˜POC/JPOC12/

EDITO

Nous sommes heureux de vous accueillir, ces 24 et 25 juin 2021, pour la douzième édition des Journées Polyèdres et Optimisation (JPOC12). Les conditions sanitaires nous ont contraintes à réaliser cette journée en distanciel mais cela nous a également fourni l'opportunité d'inviter des orateurs renommés et géographiquement éloignés qui auraient difficilement pu être présent physiquement.

L'optimisation combinatoire est une branche relativement jeune liée à la théorie des graphes, la programmation mathématique, l'informatique théorique (algorithmique et théorie de la complexité) et la recherche opérationnelle. Son importance se justifie d'une part par de nombreuses applications pratiques, pouvant être formulées sous la forme d'un problème d'optimisation combinatoire et d'autre part par la grande difficulté des problèmes d'optimisation. Les approches polyédrales constituent un des outils puissants de cette discipline. Initiées en 1965 par Jack Edmonds en étudiant le problème du couplage, elles sont maintenant de plus en plus utilisées pour résoudre les problèmes d'optimisation combinatoire difficiles. Ces techniques, parfois couplées avec d'autres méthodes comme la génération de colonnes, permettent d'élaborer des algorithmes efficaces de résolution.

L'équivalence établie entre la séparation et l'optimisation, sur un polyèdre d'une part et l'évolution des outils de calcul d'autre part, ont donné un essor important à ces méthodes.

L'optimisation combinatoire est une discipline qui ne cesse de se développer aussi bien sur le plan théorique qu'au niveau des applications. Ces dernières années, des avancées majeures ont été observées en complexité, en algorithmes d'approximation et en performance de résolution de problèmes difficiles de grande taille. Un des objectifs de ces journées est de promouvoir les approches polyédrales et leurs applications en optimisation combinatoire.

Nous avons le plaisir d'accueillir plus de 80 chercheurs, pour se retrouver autour d'exposés sur le thème des polyèdres en optimisation combinatoire. Ces deux journées se déroulent sous la forme d'exposés pléniers de 25 minutes suivis de 5 minutes de questions et de 4 exposés de chercheurs invités de 50 minutes suivis de 10 minutes de questions.

Le comité d'organisation

PROGRAMME

Jeudi 24 juin 2021

8h45 - 9h00 **Ouverture des journées**

9h00 - 9h30
IP formulations for equitable Traveling Salesman Problems
Viet Hung Nguyen, ***Thi Quynh Trang Vo***

9h30 - 10h00
Freight Delivery using Public Transportation Systems : Mixed Integer Programming Approaches for Operational Decision Making
Claudia Archetti and ***Minakshi Punam Mandal***

10h00 - 10h30
Equivalence entre les problèmes de blocker et d'interdiction appliqués au ot max
Isma Bentoumi*, Fabio Furini, A. Ridha Mahjoub, Sebastien Martin*

10h30 - 11h00 PAUSE

11h00 - 11h30
Simulation Monte-Carlo pour le problème stochastique d'optimisation de réseaux
Imen Mejri*, Safa Bhar Layeb*

11h30 - 12h00
Une formulation étendue basée sur les matroïdes pour le problème de l'arbre couvrant budgeté -Projection et décomposition
Charles Nourry*, A. Ridha Mahjoub, Hassene Aissi*

12h00 - 12h30
The resilient virtual network function placement problem
A. Benamiche, Y. Carlinet, ***R. Colares*** *and N. Perrot*

12h30 - 14h00 PAUSE DEJEUNER avec l'outil gather.town

14h00 - 15h00
Concrete complexity bounds for optimizing over integers
Amitabh Basu

15h00 - 15h30
A finite epsilon-convergence algorithm for 0-1 mixed-integer convex two-stage robust optimization with objective uncertainty
Boris Detienne*, Henri Lefebvre, Enrico Malaguti, Michele Monaci*

15h30 - 16h00 PAUSE

16h00 - 16h30
OSPF Weight Setting Problem : extended models and exact algorithms using machine learning
Amal Benhamiche, ***Morgan Chopin****, Sébastien Martin*

16h30 - 17h30
Minimum degrees and tree subgraphs
Maya Jakobine Stein

Vendredi 25 juin 2021

9h00 - 9h30

Variable-Sized Bin Packing Problem with Color Constraints

Said Hanafi, A. Ridha Mahjoub, ***Raouia Taktak****, Christophe Wilbaut*

9h30 - 10h00

Valid Inequalities and Branch-and-Cut-and-Price Algorithm for the Constrained-Routing and Spectrum Assignment Problem

Ibrahima Diarassouba, ***Youssouf Hadhbi****, A. Ridha Mahjoub*

10h00 - 10h30

Le problème de plus court chemin avec inclusion de noeuds

Sébastien Martin, ***Youcef Magnouche****, Corentin Juvigny, Jérémie Leguay*

10h30 - 11h00 PAUSE

11h00 - 11h30

Recognizing box-TDI polyhedra is coNP-complete.

Patrick Chervet, Roland Grappe, Mathieu Lacroix, Rolland Grappe, ***Francesco Pisanu****, Roberto Wolfler Calvo.*

11h30 - 12h00

Formulations pour le probleme de plus grand graphe partiel commun

Etienne de Gastines*, Arnaud Knippel*

12h00 - 12h30

Modele multicritère pour l'optimisation du problème de planification intégrée dans un terminal à conteneurs

Marwa Samrout*, Adnan Yassine, Abdelkader Sbihi*

12h30 - 14h00 PAUSE DEJEUNER avec l'outil gather.town

14h00 - 15h00

On the diameter and the circuit-diameter of polytopes

Laura Sanita

15h00 - 15h30

MIP based approaches for strategic decisions on the Freight on Public Transport problem

Laurent Alfandari, Claudia Archetti, ***Diego Delle Donne*** *and Ivana Ljubic*

15h30 - 16h00 PAUSE

16h00 - 16h30

A Branch-and-Cut algorithm for the Proactive Countermeasures Selection Problem

Ridha Mahjoub, ***M. Yassine Naghmouchi****, Nancy Perrot*

16h30 - 17h30

IP formulations for vehicle routing with stochastic demands

Ricardo Fukasawa

IP formulations for equitable Traveling Salesman Problems

Viet Hung Nguyen[1] et Thi Quynh Trang Vo[2]

1. LIMOS Université Clermont Auvergne. Campus Universitaire des Cézeaux.
1 rue de la Chebarde.TSA 60125 CS 60026. 63178 AUBIERE CEDEX - FRANCE, France
viet_hung.nguyen@uca.fr, thi_quynh_trang.vo@uca.fr

Mots-clefs : traveling salesman problem, equity, ordered weighted averaging, integer linear programming.

1 Introduction and results

We consider several equitable versions of the Traveling Salesman Problem where the equity is based on the cost of the edges taken by the tour. One of these versions is the *balanced traveling salesman problem* defined by Larusic and Punnen (COR, vol. 38, pp 868-875 2010) where the objective is to minimize the difference between the maximum cost and the minimum cost of the edges in the tour. We also consider the OWA (ordered weight average) TSP which favors tours with similar costs on the edges while assuring a degree of efficiency on the total cost. We propose MIP formulations for those problems where the equity constraints are formulated as linear constraints. Moreover, no additional integer variable is needed with respect to the IP formulation of the original TSP. We also present numerical experiments where, in our knowledge, optimal solutions for several instances of the balanced TSP are proved for the first time.

Freight Delivery using Public Transportation Systems: Mixed Integer Programming Approaches for Operational Decision Making

Claudia Archetti[1] and Minakshi Punam Mandal[2]

[1]*ESSEC Business School, archetti@essec.edu*
[2]*ESSEC Business School, b00744633@essec.edu*

June 23, 2021

Abstract

The aim of this study is to explore the potential of using public transportation systems for freight delivery. The idea is to use the spare capacities, particularly during off-peak hours, of the public transportation vehicles like buses, trams, trains, etc. to transport packages within the city, instead of dedicated delivery vehicles. This is aimed towards not only reducing the cost of delivery, but also congestion, traffic, and the resulting negative impacts on the environment. The study contributes towards the growing body of literature on innovative strategies for performing sustainable last-mile deliveries. We study the problem at an operational level, where packages are transported from the Consolidation Distribution Center (CDC) to the pre-determined public vehicle stops by the delivery vehicles, from where they are transported into the city by the public transportation systems. Then the last leg of the delivery of the packages is performed using green vehicles or eco-friendly systems to their respective customers. We propose mixed integer linear programming approaches to study the routing problem of the packages from the CDC to the customers, and provide numerical experiments to study the efficiency and effectiveness of the study.

keywords– *freight delivery, public transportation system, VRPs, sustainable delivery methods*

Équivalence entre les problémes de blocker et d'interdiction appliqués au flot max

Isma Bentoumi[1,2], Fabio Furini[3], A.Ridha Mahjoub[2], Sébastien Martin[1]

1. Huawei Technologies France, Boulogne-Billancourt, France
isma.bentoumi@huawei.com, sebastien.martin@huawei.com

2. LAMSADE, UMR CNRS 7243, Université Paris-Dauphine-PSL, Paris, France
isma.bentoumi@dauphine.eu, ridha.mahjoub@lamsade.dauphine.fr

3. Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", Consiglio Nazionale delle Ricerche (IASI-CNR), Roma, Italy
f.furini@iasi.cnr.it

Mots-clefs : Optimisation combinatoire, optimisation bi-niveau, blocker, interdiction, flot max.

1 Introduction

On s'intéresse à la relation entre les problèmes de blocker et d'interdiction appliqués au flot max. Considérons un graphe $G(V, A)$ où V est l'ensemble des nœuds et A l'ensemble des arcs, contenant un nœud source s et un nœud destination t. Chaque arc $a \in A$ dispose d'une capacité c_a et d'un cout r_a appelé cout d'interdiction.
Le problème d'interdiction appliqué au flot max consiste à trouver un ensemble d'arcs à supprimer dans le réseau de manière à minimiser le flot maximum restant entre s et t. Ces arcs, appelés *arcs interdits*, sont limités par un budget d'interdiction. En d'autres termes, le coût d'interdiction total des arcs interdits doit être inférieur ou égal à une valeur R. On note ce problème par $MFIP(G, c, r, R)$.
Le problème du blocker appliqué au flot max consiste à trouver un ensemble d'arcs dont le cout d'interdiction total est minimum et tel que le flot restant dans le réseau entre s et t est inférieur ou égale à une valeur F, appelé *flot limite*. On note ce problème par $MFBP(G, c, r, F)$.

Le problème d'interdiction appliqué au flot max a été largement étudié dans la littérature ([1], [2]). L'article [1] propose une formulation en nombres entiers pour résoudre MFIP(G, c, r, R).

2 Résultats

Une coupe dans un graphe est une partition des nœuds en deux sous-ensembles $V_s \subset V$ et $V_t = V \backslash V_s$. On définit le cardinal de la coupe $C = (V_s, V_t)$ comme étant l'ensemble des arcs ayant l'extrémité initiale dans l'ensemble V_s et l'extrémité terminale dans l'ensemble V_t. Une coupe séparant deux sommets s et t est une coupe tel que $s \in V_s$ et $t \in V_t$.
La proposition suivante établit une relation entre le probléme de blocker et les coupes dans un graphe.

Proposition 1. *Toute solution optimale du problème de blocker MFBP(G, c, r, F) est contenue dans le cardinal d'une coupe entre s et t dans le graphe G.*

La figure (1) représente la solution d'un problème de blocker appliqué au flot max avec un flot limite $F = 23$. Sur chaque arc sont reportés le coût d'interdiction (au-dessus), le flot passant par l'arc (première valeur en dessous), la capacité (deuxième valeur en dessous, entre parenthèses). Les arcs interdits sont représentés par des traits en pointillés de couleur bleu. Les arcs en rouge représentent le cardinal d'une coupe minimum dans le graphe restant. L'union de ces arcs et des arcs interdits forment le cardinal d'une coupe dans le graphe initial. Remarquons que le flot restant après la suppression des arcs interdits est égal à 18. Cette valeur est bien inférieure au flot limite imposé.

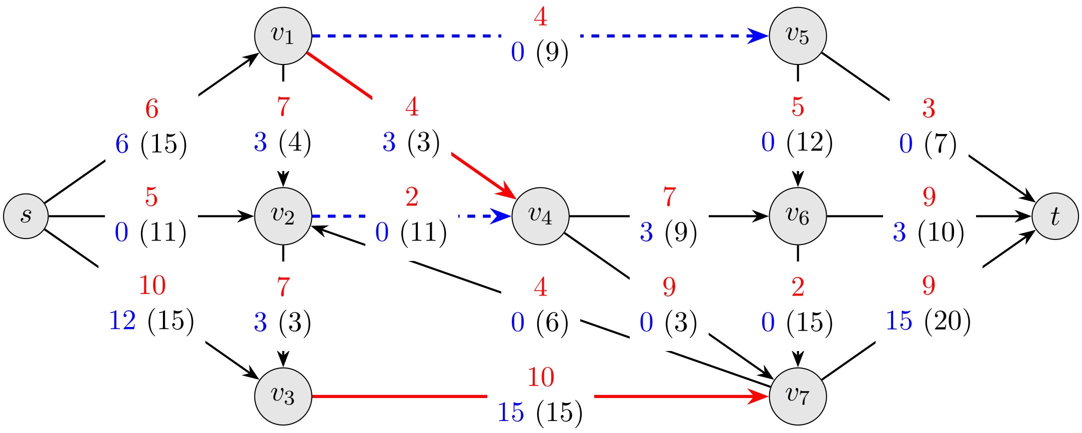


FIGURE 1 – Exemple d'un problème de blocker appliqué au flot max avec un flot limite $F = 23$.

A notre connaissance, le problème du blocker appliqué au flot max n'a pas été traité dans la littérature. Au cours de nos travaux, nous avons développé une formulation compacte pour le probléme du blocker et avons obtenu des résultats expérimentaux.
Nous avons également identifié une relation entre le problème du blocker MFBP(G, c, r, F) et le problème d'interdiction MFIP(G, c, r, R). Celle-ci est donnée par la proposition suivante.

Proposition 2. *A^I est une solution optimale du problème d'interdiction MFIP(G, c^I, r^I, R) si et seulement si le cardinal de la coupe contenant A^I est une solution optimale du problème du blocker MFBP$(G, c = r^I, R = F)$.*

Références

[1] R. K. Wood. Deterministic network interdiction. *Mathematical and Computer Modeling*, 17(2) :1–18, 1993.

[2] J. Royset and K. R. Wood. Solving the bi-objective maximum flow network interdiction problem. *INFORMS Journal on Computing*, 19(2) :175–184, 2007.

Simulation Monte-Carlo pour le problème stochastique d'optimisation de réseaux

Imen Mejri et Safa Bhar Layeb

Laboratoire de Recherche OASIS, Ecole Nationale d'Ingénieurs de Tunis,
Université de Tunis El Manar, BP 37, Le Belvédère, 1002, Tunis, Tunisie
imen.mejri@enit.utm.tn , safa.layeb@enit.utm.tn

Abstract : Dans ce travail, nous nous intéressons particulièrement à un problème d'optimisation de réseaux avec demande stochastique. Pour résoudre ce problème, nous proposons une formulation stochastique de type arc-chemin qui, à notre connaissance, n'a jamais été utilisée jusqu'à présent pour la résolution des problèmes stochastiques d'optimisation de réseaux. Nous développons une approche de résolution utilisant la méthode Sample Average Approximation (SAA). Les résultats numériques préliminaires sur des instances de la littérature sont assez prometteurs.

Mots-clefs : Problème d'optimisation de réseaux, programmation stochastique, simulation Monte-Carlo, méthode SAA.

1 Introduction

Les problèmes de conception de réseaux, appelés en anglais *Network Design Problems* (NDPs), représentent une catégorie de problèmes combinatoires très étudiés dès le début du vingtième siècle et sont étroitement liés aux problèmes de traitement des données informatiques et à la théorie de la décision. En termes de théorie de graphes, le but est généralement de concevoir un réseau au moindre coût permettant la circulation totale ou partielle d'un flux de matière ou d'information. Ces problèmes sont NP-difficiles [7] et sont souvent modélisés par la programmation linéaire en nombres mixtes. La résolution efficace des NDPs est d'une importance cruciale compte tenu de leur large applicabilité.

Dans ce travail, nous nous intéressons particulièrement à un problème d'optimisation de réseaux avec demande stochastique, noté S-NLP. En effet, Il s'agit d'une version généralisée du problème d'optimisation de réseaux, appelé en anglais *Network Loading Problem* (NLP), qui est une variante du problème de conception de réseaux avec multiflots et capacités discrètes. Le NLP consiste à installer un nombre entier de connexions de manière à pouvoir router simultanément les différentes demandes à travers le réseau, et ce, avec le moindre coût. Ce problème se pose avec grande importance dans le domaine de télécommunications [8]. Pour résoudre ce problème, nous proposons une formulation stochastique de type Arc-Chemin qui, à notre connaissance, n'a jamais été utilisée jusqu'à présent pour la résolution des problèmes stochastiques d'optimisation de réseaux. D'ailleurs, l'un des plus grands défis auquel sont confrontés les experts de réseaux est de garantir que leurs réseaux tels qu'ils sont conçus, restent performants malgré la variabilité des données d'entrée ([2], [14]). Pour faire face à ces incertitudes et plus précisément la stochasticité de la demande, nous développons une approche de résolution utilisant la méthode *Sample Average Approximation* (SAA). Les résultats numériques préliminaires sur des instances de la littérature sont assez encourageants.

2 Présentation du NLP Stochastique

Au cours des dernières décennies et avec le développement rapide du secteur des télécommunications, les problèmes d'optimisation de réseaux (NLPs) ont fait l'objet de nombreuses études scientifiques. Ces études traitent des problèmes réels tels que la conception de réseaux d'accès local avec un ou plusieurs types de technologies et la conception de réseaux en fibre optique de différentes largeurs de bande passante ([1], [6]). Outre le secteur des télécommunications, Les NLPs ont également d'autres applications dans le domaine de la logistique et du transport où ils ont un impact crucial sur la rentabilité des entreprises ([3], [4]).

Bien qu'il existe une littérature abondante sur les NLPs avec des données d'entrée déterministes, les travaux qui ont tenu compte de leur aspect aléatoire réel et complexe [10], sont beaucoup moins nombreux. En effet, les problèmes stochastiques d'optimisation de réseaux (S-NLP) considèrent différents types d'incertitude : (i) une incertitude relative à la variabilité de la capacité abordée par Thapalia et al. [16] lors de la résolution d'un problème d'optimisation d'un réseau de distribution de gaz en tenant compte des perturbations de l'approvisionnement à cause des fuites ou des blocages de canalisations, et (ii) une incertitude relative à la variabilité de la demande de trafic rencontrée souvent dans le domaine des télécommunications. Ce deuxième type d'incertitude est plus étudié dans la littérature ([9], [5], [10]). Ce présent travail s'intéresse également au S-NLP tel que les quantités de demande sont aléatoires et en plus elles peuvent être partiellement acheminées à travers le réseau à concevoir.

Plus précisément, le problème stochastique d'optimisation de réseaux (S-NLP) étudié est défini comme suit. Soit un graphe non orienté $G=(V,E)$, tel que sur chaque arête $e\in E$, il est possible d'installer une connexion caractérisée par un coût d'installation unitaire prédéfini f_e et une capacité entière à déterminer. Soit un ensemble de K demandes/commodités prédéfinies entre des paires de nœuds source-destination, notées respectivement s_k et t_k, k, $k=1...K$. Bien évidemment, K est au plus égal à $n(n-1)/2$. Pour chaque demande k, $k=1...K$, on définit une fonction $d_k(.)$ qui représente la fonction de densité de probabilité de la demande d_k à router du nœud source s_k vers le nœud destination t_k. Ainsi, le vecteur $\tilde{d} = (\tilde{d}_k)_{\mathrm{k=1,..,K}}$ désigne le vecteur de demandes stochastique. Il convient de noter que chaque demande peut être partiellement acheminée à travers plusieurs chemins. Ainsi, un coût unitaire de pénalité γ_k est associé à chaque unité non acheminée de la demande k, $k=1...K$. Le S-NLP consiste alors à installer des capacités entières de manière à pouvoir router partiellement les demandes de trafic tout en minimisant la somme des coûts fixes d'installation et des coûts des pénalités des demandes non acheminées.

3 Modélisation et Résolution

Pour résoudre le S-NLP, nous commençons par proposer une formulation de type Arc-Chemin. Pour toute demande k, $k=1...K$, notons P_k l'ensemble de tous les chemins possibles existants de s_k vers t_k dans le graphe G, et a_{erk} , $e\in E$, $r\in P_k$, une constante binaire qui vaut 1 si l'arête e fait partie du chemin r associé à la demande k, et 0 sinon. Nous considérons les variables de décision suivantes: z_r^k correspond à une variable continue positive représentant la quantité de flot qui circule sur le chemin $r\in P_k$ pour chaque demande k, $k=1,...,K$ entre s_k et t_k ; et y_e correspond à une variable entière représentant la capacité à installer sur chaque arête $e\in E$. Nous introduisons également ρ_k, une variable continue positive qui représente les unités non acheminée de chaque demande.

Il convient de noter qu'un scénario s, $s\in S$, est un vecteur $\tilde{d}^s = (\tilde{d}_1^s,...,\tilde{d}_K^s)$ qui représente une réalisation possible des données stochastiques $\tilde{d}_k$, $k=1,...,K$, avec S est l'ensemble de scénarios possibles. Une formulation stochastique pour le S-NLP peut être établie comme suit :

$$Minimize \sum_{e\in E} f_e\, y_e + \frac{1}{|S|}\sum_{s=1}^{|S|}\sum_{k=1}^{K}\gamma_k \rho_k^s \qquad (1)$$

sous les contraintes :

$$\sum_{r=1}^{|P^k|} z_{rk}{}^s + \rho_k^s \geq \tilde{d}_k^s, \quad k=1,...,K, \quad s\in \quad (2)$$

$$\sum_{k=1}^{K}\sum_{r=1}^{|P^k|} \alpha_{erk}\, z_{rk}{}^s \leq y_e \quad \forall e\in E, \quad s\in \quad (3)$$

$$z_{rk}^{s} \geq 0 \quad , \quad \forall r \in P^{k}, \quad k = 1,\ldots,K, \quad (4)$$

$$\rho_{k}^{s} \geq 0 \quad , \quad \forall k = 1,\ldots,K, \quad s \in S, \quad (5)$$

$$y_{e} \in \mathrm{N}, \quad \forall e \in E. \quad (6)$$

L'objectif (1) est de minimiser la somme des coûts fixes d'installation et des pénalités associées à l'ensemble des demandes non routées. Les contraintes (2) permettent le routage partiel des réalisations possibles du vecteur de demandes $\tilde{d}$. Les contraintes (3) garantissent que la quantité de flot circulant sur chaque arête ne dépasse pas la capacité de la connexion installée sur cette arête. Les contraintes (4) et (5) sont les contraintes de non-négativité des variables *z* et *ρ* et les contraintes (6) garantissent l'intégralité des variables *y*.

Afin de pallier au grand nombre de scénarios, nous utilisons la méthode *Sample Average Approximation* (SAA). C'est une technique d'échantillonnage fondée sur la simulation de Monte-Carlo dans laquelle la valeur prévue de la fonction objectif est remplacée par une valeur approchée en utilisant un ensemble fini de scénarios. Plus précisément, on considère un sous-ensemble réduit *R*, $R \subset S$. Pour chaque scénario *r*, $r \in R$, le problème revient à résoudre un NLP déterministe avec une réalisation particulière de demandes correspondant à $(\tilde{d}_{1}^{r},\ldots,\tilde{d}_{K}^{r})$. Ce problème est résolu à l'optimalité en utilisant une procédure exacte appliquant la décomposition de Benders combinée avec un algorithme de génération de colonnes inspirée des travaux [13] et [12]. Appelons $\tilde{z}(r)$ la valeur de la fonction objectif ainsi trouvé, d'une manière similaire à [11], une borne inférieure valide au S-NLP est alors égale à :

$$Z_{SAA} = \frac{1}{|R|}\sum_{r=1}^{|R|}\tilde{z}(r). \quad (7)$$

Afin d'évaluer la performance de l'approche proposée pour résoudre le S-NLP, une étude expérimentale préliminaire a été implémentée en utilisant le langage de programmation C# couplé au logiciel CPLEX (version 12.5). Les expérimentations ont été menées sur un ensemble initial de topologies de réseaux extraites de la bibliothèque SNDlib [15]. Les résultats préliminaires sont présentés dans le Tableau 1.

Tableau 1 : Résultats Préliminaires de la méthode SAA

Instance	***#nœuds***	***#arêtes***	***#commodités***	Z_{SAA}	T_{SAA} ***(s)***
PDH	11	34	24	502 238 977	186.19
DI-YUAN	11	42	22	1 617 135	82.30
ABILENE	12	15	66	32 817 830 396	359.44
POLSKA	12	18	66	5 477 976	264.06
Moyenne					***222.97***

Globalement, les résultats montrent que les instances sont résolues en un temps moyen raisonnable de l'ordre de 223 secondes, ce qui reflète l'efficacité de la procédure stochastique proposée. En outre, dans ces expérimentations, nous avons testé différentes valeurs pour le nombre de scénarios. L'analyse des temps de calcul révèle que le temps de calcul de la procédure SAA augmente avec le nombre de scénarios générés. Bien évidemment, plus le nombre des scénarios est élevé, plus le résultat obtenu est de meilleure qualité. Une étude expérimentale numérique est en cours afin d'affiner les paramètres de la méthode SAA et d'approfondir les tests sur un ensemble plus grand des instances de la littérature.

4 Conclusion

Dans ce travail de recherche, nous nous sommes intéressés à la résolution d'un problème d'optimisation de réseaux avec des demandes incertaines. Il s'agit d'un problème NP-difficile qui attire l'intérêt des praticiens et des chercheurs pour ses applications pertinentes, notamment dans le domaine des télécommunications.

Pour aborder ce problème, nous avons utilisé une formulation de type Arc-Chemin qui n'a jamais été utilisée jusqu'à présent dans la résolution des problèmes stochastiques d'optimisation de réseaux. Afin de traiter l'incertitude de la demande, nous avons adapté une procédure de simulation Monte-Carlo basée sur la méthode *Sample Average Approximation*. Les résultats numériques préliminaires, fournissant des solutions pour des instances de taille moyenne, sont assez prometteurs et encourageants afin d'affiner et d'approfondir cette étude.

Références

[1] Barahona, F. (1996). *Network design using cut inequalities. SIAM Journal on optimization, 6(3), 823-837.*

[2] Ben Ameur, W., & Kerivin, H. (2005). *Routing of uncertain traffic demands. Optimization and Engineering, 6(3), 283-313.*

[3] Bienstock, D., Chopra, S., Günlük, O., and Tsai, C.Y. (1998), *Minimum cost capacity installation for multicommodity network flows, Mathematical programming, 81(2).*

[4] Chopra, S., Gilboa, I., & Sastry, S. T. (1998). *Source sink flows with capacity installation in batches. Discrete Applied Mathematics, 85(3), 165-192.*

[5] Claßlen, G., Koster, A. M., Kutschka, M., & Tahiri, I. (2015). Robust metric inequalities for network loading under demand uncertainty. Asia-Pacific Journal of Operational Research, 32(05), 1550038.

[6] Gendron, B., Potvin, J.Y. and Soriano, P. (2002). *Diversification strategies in local search for a nonbifurcated network loading problem, European Journal of Operational Research, 142(2), 231-241.*

[7] Johnson, D.S., Lenstra, J.K., and Rinnooy Kan, A.H.G. (1978). *The complexity of the network design problem, Networks, 8(4), 279-285.*

[8] Lee, D. N., Medhi, K. T., Strand, J. L., Cox, R. G., & Chen, S. (1989). *Solving large telecommunications network loading problems. AT&T Technical Journal, 68(3), 48-56.*

[9] Mattia, S. (2013). *The robust network loading problem with dynamic routing. Computational Optimization and Applications, 54(3), 619-643.*

[10] Mattia, S., & Poss, M. (2018). *A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation. Computational Optimization and Applications, 69(3), 753-800.*

[11] Mejri, I., Layeb, S. B., Haouari, M., & Mansour, F. Z. (2019). *A simulation-optimization approach for the stochastic discrete cost multicommodity flow problem. Engineering Optimization, 52(3), 507-526.*

[12] Mejri, I., Layeb, S. B., & Mansour, F. Z. (2019). *Enhanced Exact Approach for the Network Loading Problem. In 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 970-975). IEEE.*

[13] Mejri, I., Haouari, M., Layeb, S. B., & Mansour, F. Z. (2019). *An exact approach for the multicommodity network optimization problem with a step cost function. RAIRO-Operations Research, 53(4), 1279-1295.*

[14] Ouorou, A., & Vial, J. P. (2007, October). *A model for robust capacity planning for telecommunications networks under demand uncertainty. In Design and Reliable Communication Networks, 2007. DRCN 2007. 6th International Workshop on (pp. 1-4). IEEE.*

[15] Sorlowski, S., Wessäly, R., Pióro, M., and Tomaszewski, A. (2010). *SNDlib 1.0---survivable network design library, Networks, 55(3), 276-286.*

[16] Thapalia, B. K., Crainic, T. G., Kaut, M., & Wallace, S. W. (2012). *Single-commodity network design with random edge capacities. European Journal of Operational Research, 220(2), 394-403.*

Une formulation étendue basée sur les matroïdes pour le problème de l'arbre couvrant budgeté - Projection et décomposition

Charles Nourry[1], A. Ridha Mahjoub[1] et Hassène Aissi[1]

1. Laboratoire LAMSADE, UMR CNRS 7024, Université Paris-Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
charles.nourry@dauphine.psl.eu
{ridha.mahjoub,hassan.aissi}@lamsade.dauphine.fr

Mots-clefs : optimisation combinatoire, approches polyhédrales, formulation étendue, inégalité valide, théorie des graphes

Dans ces travaux de thèse, nous nous intéressons à la résolution exacte de certains problèmes de conception de réseaux sujet à plusieurs contraintes de budgets. Nous abordons ces problèmes avec une approche basée sur la programmation mathématiques, ces contraintes de budgets vont venir modifier la structure des polyèdres, plus ou moins connus, associés aux problèmes de réseaux. L'objectif principal de notre travail est de comprendre ces nouvelles descriptions, trouver des inégalités valides, facettes et formulations étendues afin de concevoir des algorithmes de branchements efficaces pour résoudre ces problèmes de réseaux budgétés. Considérons le problème de l'arbre couvrant budgété de poids minimum, qui a été montré comme étant faiblement NP-difficile par [1] avec une contrainte de ressource. Ce problème possède de multiples applications, dans la pratique avec par exemple le domaine de la télécommunication, on peut également le retrouver comme sous-problème de certaines décompositions.

Considérons un graphe $G = (V, E)$, où chaque arête $e \in E$ est associé à un vecteur $(c^1, \cdots, c^k) \in \mathbb{Q}^k$, où $k \in \mathbb{N}$ est fixé. On dispose de $k-1$ budgets associé pour chaque ressource $1, \cdots, k-1$, la dernière ressource k va constituer les coefficients de la fonction objectif. Plusieurs approches ont été utilisées pour résoudre ce problème, notamment des algorithmes d'approximations comme [3, 4]. Nous nous intéressons aux approches polyhédrales, avec par exemple la formulation suivante basée sur les contraintes de sous-tours. Dans cette formulation, on associe à chaque arête $e \in E$ une variable binaire x_e (où $x_e = 1$ signifie que l'arête appartient à la solution, et $x_e = 0$ signifie que l'arête n'appartient pas à la solution), on considère le programme linéaire en nombre entier suivant :

$$\min \sum_{e \in E} c_e^k \cdot x_e$$

$$s.t. \sum_{e \in E(S)} x_e \le |S| - 1 \qquad \forall S \subseteq V, 2 \le |S| \le |V| - 1 \tag{1}$$

$$\sum_{e \in E} x_e = |V| - 1 \tag{2}$$

$$\sum_{e \in E} c_e^i \cdot x_e \le b^i \qquad i = 1, ..., k-1 \tag{3}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E \tag{4}$$

Dans un premier temps, nous avons étudié différentes propriétés polyédrales du problème ; cas particuliers, relaxation, inégalités valides et dimension du problème. Et dans un second temps, nous analysons plusieurs formulations étendues, dont une approche basée sur les matroïdes ainsi que les travaux de Balas sur la programmation disjonctive [2]. Nos motivations sur la formulation étendue basée sur les matroïdes viennent d'un cas particulier du problème, qui constitue le cas où le système associé aux contraintes de ressources correspond à un matroïde. Selon Edmonds, le polyhèdre associé à l'intersection de deux matroïdes est entier, or le problème de l'arbre couvrant peut être décrit par le matroïde graphique. Dans la plupart des cas les contraintes de budgets ne correspondent pas à un matroïde. Cependant, en divisant le problème il est possible de se retrouver avec un certains nombre de polyhèdres associés à des intersections de matroïdes. Nous utilisons les travaux de Balas pour retrouver un unique polyhèdre pour notre problème.

Avec notre approche basée sur les matroïdes, nous obtenons une formulation étendue entière pour le problème dans le cas d'une contrainte de budget. A partir de cette formulation, nous avons traité deux approches différentes. Dans un premier temps, nous nous sommes intéressés à la projection de cette formulation dans la dimension des variables d'origines, à savoir que cette projection correspond au polyhèdre entier du problème. Pour cela, nous avons étudié le cône de projection ainsi que ses rayons extrêmes, dans le but de générer des inégalités valides, voir facettes pour le problème. Dans un second temps, nous avons commencé à étudier une approche basée sur les méthodes de décomposition pour notre formulation étendue. Étant donné que le nombre de matroides à considérer peut être de taille exponentielle, on souhaite de ne considérer que des matroides potentiellement intéressants. En partant d'un nombre réduit dans le problème maître, l'idée est de ne générer que des matroïdes potentiellement intéressants via un problème de pricing afin de résoudre le problème de l'arbre couvrant budgeté.

Références

[1] Aggarwal, Vijay and Aneja, Yash P and Nair, KPK. *Minimal spanning tree subject to a side constraint.* Computers & Operations Research, 9(4) :287–296, 1982.

[2] Balas, Egon. Disjunctive programming. *Annals of discrete mathematics*, 5 :3–51, 1979.

[3] Ravi, R., Goemans, M. X. The constrained minimum spanning tree problem. *Scandinavian Workshop on Algorithm Theory*, (pp. 66-75), 1996.

[4] Hassin, R., Levin, A. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2), 261-268, 2004.

The resilient virtual network function placement problem

A. Benamiche, Y. Carlinet, R. Colares and N. Perrot

Orange Labs
46 Avenue de la République, 92320 Châtillon, France
rafael.colaresborgesdeoliveira@orange.com

Keywords: Combinatorial optimization, Valid inequalities, Resilience, Service Function Chain, Virtual Network Function, 5G networks

1 Introduction

In the context of 5G networks, Service Function Chains (SFC) can be seen as origin-destination traffic demands having specific service requirements. These service requirements can be expressed as an ordered set of network functions that must be visited – on the required order – along the SFC's origin-destination route. While the virtualization of network functions allows more flexibility and cost reductions for the service deployment, it also represents a substantial challenge for infrastructure providers since the failure of a single node where a network function is installed causes the crash of the whole SFC. In order to ensure strict Service Level Agreements (SLAs), a common practice is hence to place backup network functions throughout the network. Since the placement of network functions incurs extra costs, a relevant combinatorial problem is to choose where to place such functions in order to minimize the costs while ensuring elevated SFCs availabilities. In this presentation we provide some insights on how to model and solve such combinatorial optimization problem.

2 Problem Definition

In this section, we formally define the Resilient VNF Placement (RVNFP) problem by describing in detail the given input and the desired output to the RVNFP problem.

2.1 Instance description

Let $G = (V, A)$ be a directed, loopless, connected graph. Each node $v \in V$ has a capacity $c_v \in \mathbb{R}$, a unitary processing cost $p_v \in \mathbb{R}$, and an availability $0 < a_v \leq 1$, (*i.e.*, a failure probability of $1 - a_v$). Moreover let $\mathcal{F}$ be the set of VNF types, where each VNF $f \in \mathcal{F}$ has a resource consumption $r_f \in \mathbb{R}$. Finally, let K be set of SFC demands, where each demand $k \in K$ is defined by (i) an origin $o_k \in V$ and a destination $d_k \in V$, (ii) a bandwidth $b_k \in \mathbb{R}$, (iii) a required availability A^k, and (iv) an ordered set of VNFs $F^k \subseteq \mathcal{F}$. A RVNFP instance is hence represented by the triple $(G, \mathcal{F}, K)$.

2.2 Solution Description

For each SFC $k \in K$, let $\mathcal{S}^k = \{S^k_1, \dots, S^k_{|F^k|}\}$ denote the *VNF assignment of SFC k*, where each subset $S^k_i \subseteq V$, for $i = 1, \dots, |F^k|$, represents the set of nodes where the i-th VNF of the

SFC k can be processed. Let $\mathcal{S} = \{\mathcal{S}^k : k \in K\}$ denote the associated *VNF assignment.*

Notice that a node $v \in V$ can only process a given VNF $f \in \mathcal{F}$ if f is *placed* on such node. Therefore, given a VNF assignement $\mathcal{S}$, let $V_f(\mathcal{S}) \subseteq V$ denote, for any $f \in \mathcal{F}$, the set of nodes where f must be placed, that is,

$$V_f(\mathcal{S}) = \left\{ v \in S_i^k : k \in K, i \in \{1, \dots, |F^k|\}, f^{ik} = f \right\},$$

where f^{ik} denotes the i-th element of F^k. Let $\mathcal{V}(\mathcal{S}) = \{V_f(\mathcal{S}) : f \in \mathcal{F}\}$ denote the *VNF placement* induced by $\mathcal{S}$. The cost induced by a VNF placement is

$$\sum_{f \in \mathcal{F}} \sum_{v \in V_f(\mathcal{S})} r_f p_v.$$

Capacity Restrictions. A VNF assignment $\mathcal{S}$ is said to satisfy the network capacity if the amount of resources consumed on each node $v \in V$ is at most the node's capacity c_v, that is,

$$\sum_{\substack{k \in K, i \in \{1,\dots,|F^k|\}: \\ v \in S_i^k}} b^k r_{f_k^i} \le c_v \qquad \forall v \in V. \tag{1}$$

Availability Restrictions. Within a SFC route, VNFs of different types should be placed in series while VNFs of same type (*i.e.*, backup VNFs) should be placed in parallel (see Figure 2). Figure 1 illustrates the VNF assignment of an SFC.

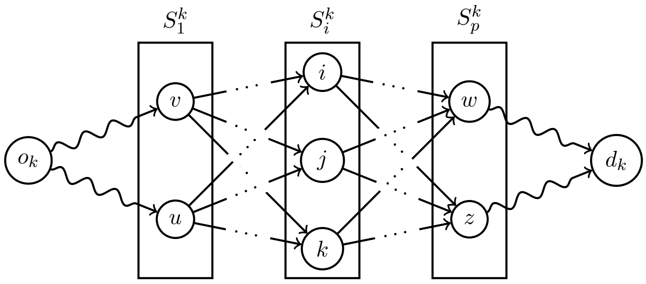
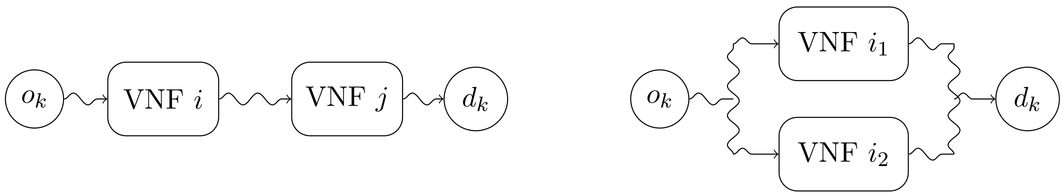


Figure 1: Illustration of the VNF assignment of an SFC k

According to [ETSI GS NFV-REL, 2016], the availability of a system composite C – denoted by a_C – composed of two subcomponents i and j in series (see Figure 2a) is the probability that both subcomponents are operational, that is, $a_C = a_i a_j$. Moreover, the availability of a system composite C built from two parallel subcomponents i and j (see Figure 2b) is the probability that at least one subcomponent is operational, that is, $a_C = 1 - [(1 - a_i)(1 - a_j)]$.



(a) Serial composition

(b) Parallel composition

Figure 2: System compositions

It follows that the availability of an SFC $k \in K$ whose VNF assignment is $\mathcal{S}^k = \{S_1^k, \dots, S_p^k\}$ is given by

$$A(\mathcal{S}^k) = \prod_{i=1}^{p} \left(1 - \prod_{v \in S_i^k} \Big(1 - a_v\Big)\right).$$

A VNF assignment $\mathcal{S}^k$ is said to be resilient for an SFC $k \in K$ if and only if

$$A(\mathcal{S}^k) \geq A^k \qquad \forall k \in K. \tag{2}$$

Feasible Solution. A VNF assignment $\mathcal{S}^k = \{S_1^k, \dots, S_p^k\}$ is said to be feasible if it satisfies both conditions (1) and (2). The RVNFP problem consists hence of finding a feasible VNF assignment $\mathcal{S}^k$ of minimum induced placement cost.

3 Problem Formulation

A natural formulation for RVNFP problem is described below. The binary variable x_v^{ik} indicates whether or not node v can process the i-th VNF of SFC k (*i.e.*, if $x_v^{ik} = 1$ then $v \in S_i^k$, otherwise $v \notin S_i^k$). For each node $v \in V$ and VNF $f \in \mathcal{F}$, the variable y_v^f indicates whether or not VNF f is placed on node v (*i.e.*, if $y_v^f = 1$ then $v \in V_f(\mathcal{S})$, otherwise $v \notin V_f(\mathcal{S})$).

$$\min \sum_{v \in V} \sum_{f \in \mathcal{F}} (r_f p_v) y_v^f \tag{3}$$

subject to

$$\sum_{v \in V} x_v^{ik} \geq 1 \qquad \forall k \in K, i \in \{1, \dots, |F^k|\}, \tag{4}$$

$$\sum_{k \in K} \sum_{\substack{i \in 1, \dots, |F^k|: \\ f_k^i = f}} x_v^{ik} \leq M y_v^f \qquad \forall v \in V, f \in F, \tag{5}$$

$$\sum_{k \in K, i \in I^k} b^k r_{f_k^i} x_v^{ik} \leq C_v \qquad \forall v \in V, \tag{6}$$

$$\prod_{i=1}^{|F^k|} \left(1 - \prod_{v \in V} \Big(1 - a_v x_v^{ik}\Big)\right) \geq A^k \qquad \forall k \in K, \tag{7}$$

$$x_v^{ik} \in \{0, 1\} \qquad \forall v \in V, k \in K, i \in \{1, \dots, |F^k|\}, \tag{8}$$

$$y_v^f \in \{0, 1\} \qquad \forall v \in V, f \in \mathcal{F}. \tag{9}$$

The objective function (3) gives the total VNF placement cost. The assignment constraints (4) ensure that at least one VNF should be assigned to each SFC for each VNF type requested. The VNF placement constraints (5) impose that a VNF can only be assigned to an SFC if it is already placed. The capacity constraints (6) guarantee that the node's capacity is not violated. Availability constraints (7) guarantee the resilience of each SFC. Finally, constraints (8) and (9) settle the domains of variables.

4 Contributions

Due to constraints (7), formulation (3)-(9) is clearly non-linear. In order to linearize it, we replace such non-linear constraints by the following linear ones,

$$\sum_{(v,i)\in(V\times\{1,\ldots,|F^k|\})\setminus S} x_v^{ik} \geq 1 \qquad \forall k \in K, S \subseteq V \times \{1,\ldots,|F^k|\} : A(S) < A^k, \tag{10}$$

which states that for any non-resilient assignment, at least one more VNF should be assigned to the given SFC.

Such procedure allows us to model the RVNFP problem as an ILP. However, the obtained formulation is non-compact as constraints (10) appear in exponential number. Moreover, their associated separation problem is NP-Hard. Indeed, when demands require only one VNF type to be visited (*i.e.*, $|F^k| = 1$ for all $k \in K$), such constraints can be seen as cover inequalities which are known for being NP-Hard to separate (see [Klabjan et al., 1998]). Fortunately, when the solution is integer, the separation problem can be solved exactly in linear time. For this reason, we propose to solve the separation problem heuristically whenever the solution is fractional and exactly otherwise.

Our first experimental results revealed very poor performances since a tremendous amount inequalities (10) were added via the exact separation, and as a consequence, too much time was being spent trying to find an optimal solution on a branch that should be pruned way earlier. In order to deal with such bad performances, we investigate different families of valid inequalities that are capable of considerably reinforcing the given ILP formulation. Our experimental results confirm such reinforcement.

References

[ETSI GS NFV-REL, 2016] ETSI GS NFV-REL (2016). Virtualisation, network functions; reliability; report on models and features for end-to-end reliability. *ETSI GS NFV-REL 003 V1. 1.2 (2016-07).*

[Klabjan et al., 1998] Klabjan, D., Nemhauser, G. L., and Tovey, C. (1998). The complexity of cover inequality separation. *Operations Research Letters*, 23(1-2):35–40.

Concrete complexity bounds for optimizing over integers

Amitabh Basu

Johns Hopkins University, USA

We discuss the complexity of the two main ingredients in integer optimization algorithms : cutting planes and branch-and-bound. We prove upper and lower bounds on the efficiency of these algorithms, when efficiency is measured in terms of complexity of the LPs that are solved. More precisely, we focus on the sparsity of the LPs and the number of LPs as measures of complexity. We also give general conditions under which combining branching and cutting into a branch-and-cut algorithm can do exponentially better than pure cutting planes or pure branch-and-bound. Time permitting, some connections to mathematical logic and proof complexity will be discussed.

A finite ε-convergence algorithm for 0-1 mixed-integer convex two-stage robust optimization with objective uncertainty

Boris Detienne[1], Henri Lefebvre[2], Enrico Malaguti[2], Michele Monaci[2]

1. Université de Bordeaux, IMB UMR CNRS 5251, Inria Bordeaux Sud-Ouest
boris.detienne@u-bordeaux.fr

2. Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi", Università di Bologna, Italy {henri.lefebvre,enrico.malaguti,michele.monaci}@unibo.it

Mots-clefs : uncertainty, two-stage robustness, convex optimization

1 Introduction

In this work, we study the wide class of optimization problems where some costs are not known at decision time and the decision flow is modeled as a two-stage process. We show how two-stage robust models for this class of problems can be solved by means of a branch-and-price algorithm where one may branch on continuous values so as to tighten the optimality gap. Our approach generalizes a recent result from the literature which addressed the linear case and was only applicable in presence of linking constraints involving binary variables [1], and extends the associated results to problems with convex constraints and general mixed-integer linking constraints. The convergence of the method is proven to ε-optimality.

2 Problem statement

More formally, we consider uncertain optimization problems where a two-stage decision flow is required. Mixed-integer variables $\boldsymbol{x}$ are used to model the decisions to be taken here and now (i.e., before the realization of the uncertainty), whereas mixed-integer variable $\boldsymbol{y}$ model the decisions to be taken in a wait-and-see phase (i.e., once the uncertainty is revealed). Uncertainty is modeled by continuous variables $\boldsymbol{\xi}$ that belong to a bounded convex set Ξ.

Our problem of interest then reads :

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{\xi}\in\Xi} \min_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} f(\boldsymbol{x},\boldsymbol{\xi},\boldsymbol{y}) \tag{2SRO-P}$$

Here, $\mathcal{X}$ denotes the first-stage feasible set while, for any $\bar{\boldsymbol{x}} \in \mathcal{X}$, $\mathcal{Y}(\bar{\boldsymbol{x}})$ denotes the set of admissible second-stage decisions. The continuous relaxation of $\mathcal{X}$ and $\mathcal{Y}(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}$ is assumed to be convex, closed and bounded. In particular, denoting by X^* the set of optimal first-stage solutions of (2SRO-P), we assume that one is able to find a hyper rectangle $[\boldsymbol{l},\boldsymbol{u}]$ such that $X^* \cap [\boldsymbol{l},\boldsymbol{u}] \neq \emptyset$. Function f is assumed to fulfill mild assumptions, ensuring the validity of the following reformulation of (2SRO-P) :

$$\min_{\boldsymbol{x}\in\mathcal{X}\cap[\boldsymbol{l},\boldsymbol{u}]} \max_{\boldsymbol{\xi}\in\Xi} \min_{(\boldsymbol{t},\boldsymbol{y})\in\mathcal{Y}'(\boldsymbol{x})} \sum_{i\in Q} w_i(\boldsymbol{\xi})t_i \tag{2SRO-R}$$

where $\mathcal{Y}(\boldsymbol{x}) = \text{proj}_{\boldsymbol{y}}(\mathcal{Y}'(\boldsymbol{x}))$ and $\overline{\mathcal{Y}}(\boldsymbol{x})$ is convex, closed and bounded for all $\boldsymbol{x}$ taken in the continuous relaxation of $\mathcal{X}$ and functions $(w_i)_{i\in Q}$ are concave functions.

3 Main contribution

Our main contribution is to propose a novel solution approach for problem (2SRO-P) by means of a branch-and-price algorithm where some of the first-stage continuous variables may be selected for branching throughout the execution. Our algorithm relies on the following two proposition.

Proposition 1 (Lower-bounding problem)**.** *The following optimization problem's optimal objective value is always a lower bound on the the optimal objective value of* (2SRO-P) *:*

$$\begin{array}{lll} \underset{\boldsymbol{x},\boldsymbol{y},(\boldsymbol{v}^i)_{i\in Q},\boldsymbol{\xi}}{\text{minimize}} & \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i\in Q} t_i w_{i*}\left(\frac{\boldsymbol{v}^i}{t_i}\right) & \\ \text{subject to} & \boldsymbol{x} \in \mathcal{X} \cap [\boldsymbol{l}, \boldsymbol{u}] & \\ & (\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) \in \text{conv}\,(S) & \\ & \sum_{i\in Q} \boldsymbol{v}^i = \boldsymbol{\xi} & \\ & \boldsymbol{v}^i \in \mathbb{R}^{|U|} & \forall i \in Q \\ & \boldsymbol{\xi} \in \mathbb{R}^{|U|} & \end{array} \tag{P}$$

where $\delta^*(\cdot|X)$ *denotes the convex conjugate of the indicator function of set* X*,* w_{i*} *denotes the concave conjugate of function* w_i *for* $i \in Q$ *and* $\text{conv}\,(S)$ *is the convex hull of the following set :*

$$S = \left\{ (\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \begin{array}{ll} l_j \leq x_j \leq u_j & \forall j \in I \\ x_j \in \{0, 1\} & \forall j \in I_B \\ (\boldsymbol{t}, \boldsymbol{y}) \in \mathcal{Y}'(\boldsymbol{x}) & \end{array} \right\} \tag{1}$$

Proposition 2 (Tightness condition)**.** *Denoting by* $v(\cdot)$ *the optimal objective value of problem* $\cdot$*, the following implication holds :*

$$X^* \cap \text{vert}\,([\boldsymbol{l}, \boldsymbol{u}]) \neq \emptyset \Rightarrow v(\text{P}) = v(\text{2SRO-P})$$

where $\text{vert}\,([\boldsymbol{l}, \boldsymbol{u}])$ *denotes the set of extreme points of* $[\boldsymbol{l}, \boldsymbol{u}]$*.*

We then apply our new methods to two problems. The first one is a variant of the Capacity Facility Location problem where the unitary travel costs are not known at decision time. The opening of facilities must be decided before knowing the exact travel costs while the underlying routing problem is delayed to the second stage. Note that each arc between an opened facility and a customer is also associated to a fixed activation cost one has to pay when using the arc. The uncertain travel costs are assumed to follow a normal probability distribution and the uncertainty set is therefore modeled as an ellipsoidal set. The second problem is a two-stage capital budgeting problem where a budget may be invested in a set of projects with uncertain profits. The profits are computed according to dependent or independent unknown risk factors. The decision maker may choose to invest here and now or after having observed the risk factors. However, early investments enjoy a first-mover advantage whereas a postponed investment only generates a fraction of the profit. We also study the impact of possible loans in the investment plan. For both problems we report encouraging computational results on a large benchmarks derived from instances in the literature.

Références

[1] Ayşe N Arslan and Boris Detienne. Decomposition-based approaches for a class of two-stage robust binary optimization problems. *INFORMS Journal on Computing*, 2021.

OSPF Weight Setting Problem: extended models and exact algorithms using machine learning

Amal Benhamiche[1], Morgan Chopin[1], Sébastien Martin[2]

1. Orange Labs, DATA&IA,
Chatillon, France
{*amal.benhamiche,morgan.chopin*}*@orange.com*

2. Huawei Technologies & co.
20 Quai du Point du Jour, Boulogne Billancourt, France
sebastien.martin@huawei.com

Mots-clefs : Traffic engineering, column generation, dynamic programming, neural networks

Introduction. The arising of Network Virtualization will be a key enabler for the deployment of virtualized components capable of performing efficient path computation on behalf of the routers, thus allowing the optimization of operational IP networks. This perspective change draws again the traffic engineering community's attention to classical problems related to IP network optimization and raises the question of finding effective algorithms allowing to solve those problems for large scale networks. In particular, traffic routing in IP networks still draws heavily on shortest paths based routing protocols, such as open shortest path first (OSPF), and finding a set of link weights that induce shortest paths while also minimizing the network congestion is one of the key issues for the design of efficient IP networks.

In this context, we address the *Minimum Unsplittable Shortest Path Routing* (MinUSPR) problem. This problem is formally defined as follows. We consider given a bidirected graph $G = (V, A, c)$ representing an IP network where V is the set of nodes and A is the set of arcs. Every node $v \in V$ corresponds to a router and every arc $a \in A$ corresponds to a link connecting two routers with a given capacity $c_a \geqslant 0$. We are also given a set of commodities K representing a traffic matrix where each commodity $k \in K$ is characterized by an origin (router) node s^k, a destination (router) node t^k and a volume of traffic $D^k \geqslant 0$ (measured in Mbit/s) to be routed between s^k and t^k. The MinUSPR problem consists then in finding a set of weights to assign to the arcs of G such that (i) there is a unique shortest path in G for each commodity between its origin and destination according to the identified weights and (ii) the network congestion is minimum.

The MinUSPR problem has been widely studied in the literature for both its *splittable* and *unsplittable* versions. Several works propose algorithms to solve this problem using meta-heuristic, approximate and exact approaches (see [1] and the references therein). Our work extends the results proposed in [1] and aims at investigating hybrid approaches based on OR/ML methodologies for the problem.

Contributions. Our contributions can be summarized as follows

Path-based formulation. We first propose a non-compact MILP formulation for the problem obtained by applying a Dantzig-Wolfe decomposition to the compact (arc-flow) formulation presented in [2]. The resulting model contains a exponential number of (path) variables and we manage to solve its linear relaxation using a column generation procedure. More precisely, we solve iteratively the problem with a subset of columns. The remaining formulation is called *Restricted Master Problem* (RMP), and new columns (path variables) with negative reduced cost are generated dynamically though a shortest path-based pricing subproblem then added to the RMP.

Routing-path formulation. This model has also an exponential number of variables based on routing. A *routing* is an assignment of one path for each commodity. Let R be the set of all possible routings. The decision variables are the variables x_r that takes the value 1 if the routing r is used and 0 otherwise, for each $r \in R$. There is a variable $r_u^v \in \mathbb{R}^+$ for each $u, v \in V$ equal to the potential of node $u \in V$, that is the distance between router $u \in V$ and router $v \in V$. A variable $u_a^t \in \{0, 1\}$ for every $a \in A, t \in T$ equal to 1 if the link $a \in A$ belongs to a shortest path towards destination $t \in T$ and 0 otherwise. Finally, a variable $w_a \in \mathbb{Z}^+$ corresponding to the weight assigned to the arc a for each $a \in A$.

$$\min \quad \sum_{r \in R} x_r L_r \tag{1}$$

$$\lambda : \sum_{r \in R} x_r = 1 \tag{2}$$

$$\lambda_a^t : \sum_{r \in R} \alpha_{ra}^t x_r = u_a^t \qquad \forall a \in A, \forall t \in T, \tag{3}$$

$$w_{uv} - r_u^t + r_v^t \geq 1 - u_{uv}^t \qquad \forall a \in A \quad \forall t \in T \tag{4}$$

$$w_{uv} - r_u^t + r_v^t \leq M(1 - u_{uv}^t) \qquad \forall a \in A \quad \forall t \in T \tag{5}$$

where M is a big value. Inequality (2) ensures that only one routing is selected. Inequalities (3) make the link between variables x and u. Inequalities (4) and (5) ensure a feasible weight on each link. The pricing problem consists in finding a routing r (multi-commodity flow) such that $\lambda + \sum_{a \in A} \sum_{t \in T} \lambda_a^t y_a^t > L_r$ where y_a^t is equal to 1 if at least one commodity with the destination t use the arc a and 0 otherwise. It can be solved by a variant of the classical multi-commodity flow path based model.

Neural networks. In the column generation loop we add a neural network to find a heuristic solution at each iteration. Let r be the routing added by one loop of the column generation. The neural network is composed by an input vector that corresponds to the utilization of each link given by the routing r and an output vector that corresponds to the weights (w_a). Thus the input vector and output vector have the same size. We also apply our neural network to the linear relaxation of path variables of the path-based formulation. To train the neural network, we generate on the same input graph a dataset with few commodities that can be solved optimally using exact methods.

Generating dataset through dynamic programming. A dynamic programming algorithm based on a *tree decomposition* for solving the MinUSPR problem was proposed in [2]. More precisely, the algorithm computes an optimal set of weights in time $2^{O(\omega^8 |K|)} \cdot n$ where ω is the treewidth of the input graph. Recall, that the "treewidth" is a graph invariant that measures how close a graph is of being a tree. For instance, trees have the smallest possible treewidth value 1 while clique graphs of n vertices have the largest treewidth value n. Thus, the algorithm is particularly suited to solve the problem on "tree-like" (sparse) networks. A promising research direction would be to use this algorithm to quickly generate a dataset for our neural network approach with examples of sparse networks and "small" number of commodities. Computing such dataset may seem restrictive at first, but most of the real life networks are sparse. Furthermore, we think that learning on a small, but carefully chosen, set of demands is enough to help the neural network to generalize well.

Future works include the integration of these approaches into a branch-and-price algorithm and an experimental study to assess its effectiveness on state of the art instances.

Références

[1] Andreas Bley, Bernard Fortz, Éric Gourdin, Kaj Holmberg, Olivier Klopfenstein, Michal Pióro, Artur Tomaszewski and Hakan Ümit. *Optimization of OSPF Routing in IP Networks.* Graphs and Algorithms in Communication Networks 2010 : 199-240

[2] Amal Benhamiche, and Morgan Chopin. Toward Scalable Algorithms for the Unsplittable Shortest Path Routing Problem. *CoRR abs/2006.04324*, *https ://arxiv.org/abs/2006.04324*, 2020.

Minimum degrees and tree subgraphs

Maya Jakobine Stein

Universidad de Chile, Chili

This talk is a survey of the history and recent developments on questions related to tree subgraphs of graphs having large minimum degree. The starting point is the following easy observation : Any graph of minimum degree at least k contains any trees on k+1 vertices as a subgraph. This can be shown by just greedily embedding the tree in a connected way. Several attempts have been made to relax the conditions in different ways. For instance, it is possible to drop the minimum degree condition to 2k/3 if a maximum degree condition is imposed at the same time, or if certain other conditions on the host graph, or on the tree are added. We will also consider recent extensions of this type of question to directed graphs and hypergraphs.

Variable-Sized Bin Packing Problem with Color Constraints

Saïd Hanafi[1], A. Ridha Mahjoub[2], Raouia Taktak[3] and Christophe Wilbaut[1]

1. Laboratoire LAMIH, UMR CNRS 8201, Université Polytechnique Hauts-de-France
Le Mont Houy, 59313 Valenciennes Cedex 9, France
{Said.Hanafi,Christophe.Wilbaut}@uphf.fr

2. Laboratoire LAMSADE, UMR CNRS 7024, Université Paris-Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
ridha.mahjoub@lamsade.dauphine.fr

3. Laboratoire Smarts/CRNS, Université de Sfax
Technopole de Sfax, Cité el Ons 3021, Tunisie
raouia.taktak@isims.usf.tn

Mots-clefs : Variable-Sized Bin Packing, color constraints, mixed integer linear programming

1 Introduction, assumptions and notations

The Variable-sized Bin Packing Problem with Color Constraints (VSBPP-CC) is a generalization of the classical one-dimensional Bin Packing Problem known to be NP-hard. The VSBPP-CC can be introduced with the following notations. Let $\mathcal{N}$ be a set of n items. Then, let $\mathcal{C}$ be a set of p colors such that each item $i \in \mathcal{N}$ is characterized by a weight a_i and a color $c_i \in \mathcal{C}$. Finally, let $\mathcal{B}$ denotes the set of q possible used bins, $1 \leq q \leq n$, and $M = \{b_1, \ldots, b_k, \ldots, b_m\}$ a set of m different available capacities to be assigned to the bins. The VSBPP-CC consists in assigning to each used bin from $\mathcal{B}$ a capacity from M. The objective is to minimize the residual capacity in the used bins such that : (i) each item is assigned to exactly one type of the bins, (ii) the total weights of items assigned to each bin does not exceed its capacity, and (iii) no more than two colors appear in each used bin. The latter condition may be seen as a maximum color capacity for each bin.

The VSBPP-CC was introduced by Dawande et al. in [1]. Authors were motivated by a real problem from the steel industry since it arises as a model for the slab design problem in the production planning process of a steel plant. Further works have been interested to the problem. In [2], Gangani and Refalo propose an efficient constraint programming model and use a large neighborhood search associated with a specific strategy for variable and value selection within a depth-first search algorithm. In [3], Heinz et al. propose a column-generation based algorithm that solves different variants of the problem.

Without loss of generality, we assume that :
- The number q of possible used bins satisfies $1 \leq q \leq n$;
- The number m of bins' capacities is such that $m > 1$;
- The number p of colors satisfies $2 < p \leq n$;
- The items are sorted in non-decreasing order : $a_i \leq a_{i+1}$, $i = 1, \ldots, n-1$;
- The types of bin are sorted in non-decreasing size : $b_k \leq b_{k+1}, k = 1, \ldots, m-1$;
- All items can be packed, *i.e.*, $\max\{a_i : i \in \mathcal{N}\} \leq b_m$, otherwise the problem is infeasible ;
- All items cannot be packed in one bin, *i.e.*, $\sum_{i \in \mathcal{N}} a_i > b_m$, otherwise the problem is trivial.

— All data are non negative integers, *i.e.*, $a_i, c_i \in \mathbb{Z}$ for all $i \in \mathcal{N}$, and $b_k \in \mathbb{Z}, k \in M$. We also use the following notation. Let $\mathcal{N}^c = \{i \in \mathcal{N} : c_i = c\}$ denotes the set of items with color $c \in \mathcal{C}$. Let $\mathcal{M} = \{0, 1, \ldots, m\}$ denotes the index set of available capacities including *dummy* capacity $b_0 = 0$ indexed by 0.

2 Integer Linear Programming Formulations

One straightforward integer linear formulation is obtained by introducing the following binary variables. $x_{ij} = 1$ if item $i \in \mathcal{N}$ is assigned to bin $j \in \mathcal{B}$, 0 otherwise. $y_{kj} = 1$ if capacity $b_k, k \in \mathcal{M}$ is assigned to bin $j \in \mathcal{B}$, 0 otherwise. $z_{cj} = 1$ if color $c \in \mathcal{C}$, is used in bin $j \in \mathcal{B}$, 0 otherwise.

VSBPP-CC can hence been formulated by the following natural formulation.

$$\min \sum_{j \in \mathcal{B}} \sum_{k \in \mathcal{M}} b_k y_{kj} - \sum_{i \in \mathcal{N}} a_i \tag{1}$$

$$\sum_{j \in \mathcal{B}} x_{ij} = 1 \qquad \forall i \in \mathcal{N} \tag{2}$$

$$\sum_{k \in \mathcal{M}} y_{kj} = 1 \qquad \forall j \in \mathcal{B} \tag{3}$$

$$\sum_{i \in \mathcal{N}} a_i x_{ij} \leq \sum_{k=0}^{m} b_k y_{kj} \qquad \forall j \in \mathcal{B} \tag{4}$$

$$\sum_{c \in \mathcal{C}} z_{cj} \leq 2 \qquad \forall j \in \mathcal{B} \tag{5}$$

$$x_{ij} \leq z_{c_i j} \qquad \forall i \in \mathcal{N}, j \in \mathcal{B} \tag{6}$$

$$x_{ij}, y_{kj}, z_{cj} \in \{0, 1\} \qquad \forall i \in \mathcal{N}, j \in \mathcal{B}, k \in \mathcal{M}, \forall c \in \mathcal{C}. \tag{7}$$

The objective function (1) consists in minimizing the total residual capacity in the used bins. Constraints (2) ensure that each item is packed in exactly one bin. Constraints (3) express the fact that to each bin we assign only one capacity. Constraints (4) are the capacity constraints, they guarantee that the total weight of items does not exceed the capacity of the bin they are packed in. Constraints (5) and (6) both ensure that the total number of colors of items packed in a bin does not exceed 2. Finally, constraints (7) are the binary constraints on variables.

We propose other ILP and MILP formulations for the problem. We address the special case where to each color is assigned only one item. A substantial experimental study will also be presented.

References

[1] Dawande, M., and Kalagnanam, J. (1998). The multiple knapsack problem with color constraints. *Yorktown Heights, NY : IBM TJ Watson Research Center.*

[2] Gargani, A., and Refalo, P. (2007, September). An efficient model and strategy for the steel mill slab design problem. *In International Conference on Principles and Practice of Constraint Programming (pp. 77-89).* Berlin, Heidelberg.

[3] Heinz, S., Schlechte, T., Stephan, R., and Winkler, M. (2012). Solving steel mill slab design problems. *Constraints*, 17(1), 39-50.

Valid Inequalities and Branch-and-Cut-and-Price Algorithm for the Constrained-Routing and Spectrum Assignment Problem

Ibrahima Diarassouba[1], Youssouf Hadhbi[2] et Ali Ridha Mahjoub[3]

1. Laboratoire LMAH, FR CNRS 3335, Université Le Havre Normandie
Place Robert Schuman, 76600 Le Havre, France
Email : diarrasi@univ-lehavre.fr

2. Laboratoire LIMOS, UMR CNRS 6158, Université Clermont Auvergne
1 rue de la Chebarde, 63178 Aubière Cedex, Clermont Ferrand, France
Email : youssouf.hadhbi@uca.fr

3. Laboratoire LAMSADE, UMR CNRS 7024, Université Paris-Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
Email : ridha.mahjoub@lamsade.dauphine.fr

Mots-clefs : réseaux FlexGrid, routage, affectation spectrale, programmation linéaire en nombres entiers, inégalité valide, branchement, génération de colonnes, génération de coupes, formulation étendue, séparation, heuristique primale, méta-heuristique.

Nous nous intéressons dans ce travail au problème du routage contraint et affectation spectrale, appelé "Constrained-Routing and Spectrum Assignment" (C-RSA) selon la terminologie anglaise. Il est connu comme un problème clé d'une importance cruciale d'un point de vue théorique, ainsi que sur le plan pratique pour le dimensionnement et la gestion de réseaux optiques élastiques FlexGrid. Il peut être énoncé comme suit. Nous considérons un réseau FlexGrid sous forme d'un graphe non orienté, sans boucle et connexe $G = (V, E)$, qui est spécifié par un ensemble de noeuds V, et un ensemble de liens E. On associe à chaque lien $e \in E$, une longueur $\ell_e \in \mathbb{R}_+$ (en km), un coût $c_e \in \mathbb{R}_+$, et un spectre $\mathbb{S}$ tel que le spectre de chaque lien $e \in E$ est divisé en $\bar{s} \in \mathbb{N}_+$ slots numérotés de 1 jusqu'au $\bar{s}$. Étant donné un ensemble K de demandes tel que chaque demande $k \in K$ est spécifiée par un noeud d'origine $o_k \in V$, un noeud de destination $d_k \in V \setminus \{o_k\}$, un nombre de slots $w_k \in \mathbb{Z}_+$, et une longueur maximale $\bar{l}_k$ (en km). Le problème C-RSA consiste à déterminer pour chaque demande $k \in K$, un chemin du routage p_k d'une longueur inférieure à $\bar{l}_k$ (contrainte de longueur maximale), et un seul intervalle de slots contigus $S_k \subset \mathbb{S}$ de largeur égale à w_k (contrainte de continuité et contguité) tel que $S_k \cap S_{k'} = \emptyset$ pour chaque paire de demandes $k, k' \in K$ $(k \neq k')$ avec $E(p_k) \cap E(p_{k'}) \neq \emptyset$ (contrainte de non-chevauchement) de sorte que la somme totale de longueurs de chemins utilisés pour acheminer les demandes K soit minimisée (c.a.d., $\sum_{k \in K} \sum_{e \in E(p_k)} l_e$).
Le C-RSA est connu pour être NP-difficile [1]. L'objectif principal de ce travail est d'exploiter les propriétés théoriques du C-RSA, et de concevoir un algorithme exact en combinant les algorithmes de branchement, de génération de colonnes et de coupes pour résoudre le problème.
Dans un premier temps, nous avons introduit une nouvelle formulation mathématique étendue en termes de programme linéaire en nombres entiers basée sur des variables dites chemins. Cette formulation contient un nombre très grand de variables, voir même exponentiel au pire des cas. Nous utilisons donc un algorithme de génération de colonnes pour résoudre sa relaxation linéaire. Pour ce faire, nous avons d'abord étudié notre problème de "pricing" tel que nous avons montré que l'ajout de chaque colonne (variable) dans un programme maitre restreint, se ramène au problème de plus court chemin élémentaire avec contraintes de ressources (PCCECR). Ce dernier est déjà connu pour être NP-difficile [2]. Pour cela, nous avons développé un algorithme

pseudo-polynomial basé sur la programmation dynamique adaptée pour le "pricer".
En outre, nous avons identifié plusieurs classes d'inégalités valides pour améliorer la qualité de la relaxation linéaire de notre formulation. Certaines de ces inégalités sont obtenues en définissant des graphes de conflits issus de contraintes de non-chevauchement et de longueur maximale : inégalités de clique, inégalités de cycles impairs sans cordes et leurs généralisations utilisant certaines techniques du "lifting", ainsi que des inégalités de couverture minimale issues de certaines contraintes de capacité liées d'une manière indirecte à notre problème. De plus, nous avons utilisé la procédure de Chvatal-Gomory en combinant certaines inégalités valides pour générer des nouvelles classes d'inégalités valides plus généralisées. Nous avons décrit ensuite leurs procédures de séparation en se basant sur des algorithmes exactes, des algorithmes gloutonnes, ainsi que des heuristiques pour les inégalités valides dont le problème de séparation associé est NP-difficile.
En se basant sur les résultats obtenus, nous élaborons deux algorithmes exacts : "Branch-and-Price" (B&P) et "Branch-and-Cut-and-Price" (B&C&P) pour résoudre le problème. Ces deux algorithmes ont été implémenté en C++ sous Linux en utilisant le framework SCIP 7.0 utilisant CPLEX 12.9 pour la résolution de la relaxation linéaire dans chaque noeud dans l'arbre de B&P et B&C&P. Ces derniers ont été testés sur des serveurs de haute performance du LIMOS avec une taille mémoire limitée à 64 gb tout en bénéficiant du parallélisme en activant 8 "threads", et avec un temps de CPU limité à 5 heures.
Enfin, nous avons proposé une étude comparative approfondie sur le comportement de nos deux algorithmes en utilisant deux types d'instances : aléatoires et réalistes, et 14 graphes (topologies). Ils sont composés de deux types de graphes : réels, et d'autres réalistes de SND-Lib avec un nombre de liens $21 \leq |E| \leq 166$, et un nombre de nœuds $14 \leq |V| \leq 161$. Noter bien que nous avons testé 4 instances pour chaque triplet $(G, K, \bar{s})$ avec $|K|$ qui varie entre 10 jusqu'à 300, et $\bar{s}$ jusqu'à 320 slots. Les résultats montrent que notre algorithme B&C&P est capable de fournir des solutions optimales pour plusieurs instances, ce qui n'est pas le cas de l'algorithme B&P. De plus, nous avons remarqué que le fait d'ajouter nos inégalités valides, nous a permis de réduire le nombre moyen de noeuds dans l'arbre de B&C&P, la moyenne de gap d'optimalité, ainsi que le temps moyen de la résolution en comparant avec B&P. D'autre part, nous avons étudié l'impact de l'ajout de chaque inégalité valide. Les résultats ont montré aussi que certaines d'entre elles, en particulier les inégalités de clique et de couverture minimale, sont beaucoup plus efficaces que d'autres. D'autre part, nous avons boosté encore nos deux algorithmes à l'aide d'un algorithme de démarrage à chaud basé sur certaines méta-heuristiques adaptées pour notre problème : de recuit simulé, de recherche tabou, et algorithmes génétiques, qui permettent de fournir une solution intégrale "réalisable" dans la racine de l'arbre de nos deux algorithmes, et également une heuristique primale basée sur une méthode hybride entre un algorithme glouton et un algorithme de recherche locale permettant donc de construire une solution intégrale "réalisable" à partir d'une solution fractionnaire donnée dans chaque noeud de l'arbre de nos deux algorithmes.
Notre prochaine étape consiste à améliorer nos procédures de séparation, gérer la symétrie, ainsi que d'étudier différentes stratégies de branchement de Ryan and Foster afin d'améliorer l'efficacité de nos deux algorithmes face à certaines instances difficiles à résoudre à l'optimum.

Références

[1] K. Christodoulopoulos, I. Tomkos et E. Varvarigos. "Elastic Bandwidth Allocation in Flexible OFDM-Based Optical Networks". Dans le journal de "Lightwave Technology", vol : 29, pp. 1354 – 1366, 10 Mars 2011.
[2] M. Dror. "Note on the Complexity of the Shortest Path Models for Column Generation in VRPTW". Dans le journal de " Operations Research", vol : 42, pp. 977 – 978, 1 Octobre 1994.

Le problème de plus court chemin avec inclusion de nœuds

Sébastien Martin[1],Youcef Magnouche[1],Corentin Juvigny[2] et Jérémie Leguay[1]

Huawei Technologies & co, 18 Quai du Point du Jour, 92100 Boulogne-Billancourt, France
{prénom.nom}@huawei.com[1], *cojuvigny@gmail.com*[2]

Mots-clefs : Le problème de plus court chemin, Complexité, Programme linéaire en nombres entiers, Génération de colonnes

1 Introduction

Le problème de plus court chemin est un problème d'optimisation très connu qui a été étudié de façon approfondie depuis des décennies [3]. Il existe des algorithmes, tels que Dijkstra ou Bellman-Ford, qui peuvent résoudre ce problème en un temps polynomiale. Cependant, trouver le plus court chemin peut ne pas suffire pour certains cas d'utilisation. En effet, il est parfois nécessaire de passer par des nœuds spécifiques avant d'atteindre la destination. Dans cet article, nous considérons une variante du problème classique de plus court chemin qui consiste, étant donné un graphe $G = (V \cup \{s,t\}, A)$, un sous-ensemble de nœuds ordonnés $I = \{v_1, \ldots, v_k\} \subseteq V$ et une fonction $L : A \to \mathbb{R}$, à trouver un chemin simple entre s et t de longueur minimum dans G visitant, dans l'ordre, tous les nœuds de I.

Une version générique a été considérée dans la littérature appelée le problème du plus court parcours [4] qui consiste à trouver un plus court chemin d'un nœud d'origine à un nœud de destination de telle sorte que le chemin doit visiter, dans l'ordre, une séquence d'ensembles disjoints de noeuds $T_1, ..., T_k$ et chaque arc ne peut être utilisé plus d'une fois.

Soit $G = (V \cup \{s,t\}, A)$ un graphe orienté où V est l'ensemble des sommets et A est l'ensemble des arcs. Pour chaque arc $(u,v) \in A$, $c_{(u,v)}$ représente la distance entre u et v. Pour chaque nœud $v \in V$, soit $\delta^+(v)$ et $\delta^-(v)$ les ensembles d'arcs entrants et sortants de v, respectivement.

Théorème 1. *Le problème de plus court chemin avec inclusion de nœuds est NP-Difficile*

2 Formulation compacte

Cette première formulation est une extension de la formulation classique du problème de plus court chemin présenté dans l'article de L.Taccari [3].

Considérons les variables suivantes :

$$x_a^i = \begin{cases} 1 & \text{if } v = v_i \\ -1 & \text{if } v = v_{i+1} \\ 0 & \text{sinon} \end{cases} \qquad \forall a \in A, \forall i \in I \setminus \{v_k\}.$$

Le problème de plus court chemin avec inclusion de nœuds est équivalent au MILP suivant

$$
\begin{aligned}
\min \quad & \sum_{i \in I \setminus \{v_k\}} \sum_{a \in A} c_a x_a^i \\
\sum_{a \in \delta^+(v)} x_a^i - \sum_{a \in \delta^-(v)} x_a^i = & \begin{cases} 1 & \text{if } v = v_i \\ -1 & \text{if } v = v_{i+1} \\ 0 & \text{else} \end{cases} && \forall v \in V,\ \forall i \in I \setminus \{v_k\} \quad \textbf{(C1)} \\
\sum_{j \in I \setminus \{i\}} \sum_{a \in \delta^+(v_j)} x_a^i + \sum_{j \in I \setminus \{i+1\}} \sum_{a \in \delta^-(v_j)} x_a^i = 0 & && \forall i \in I \setminus \{v_k\}, \quad \textbf{(C2)} \\
\sum_{i \in I \setminus \{v_k\}} x_a^i \leq 1 & && \forall a \in A \quad \textbf{(C3)} \\
x_a^i \in \{0,1\} & && \forall a \in A,\ \forall i \in I \setminus \{v_k\}
\end{aligned}
$$

où les contraintes (C1) représentent les contraintes de conservation de flots requises par chaque chemin entre deux nœuds d'inclusion, les contraintes (C2) assurent que l'ordre des nœuds d'inclusion est respecté et (C3) garantissent que chaque arc est utilisé au plus une fois.

3 Formulation étendue

Soit P_i l'ensemble de tous les chemins possibles entre les nœuds d'inclusion v_i et v_{i+1} ne contenant aucun autre nœud d'inclusion de $I \setminus \{v_i, v_{i+1}\}$. Dans cette section, nous introduisons une formulation étendue pour le problème où les variables permettent de sélectionner exactement un chemin de P_i pour tout $i \in I \setminus \{v_k\}$. Cette formulation a un nombre exponentiel de variables définies comme suit

$$
\lambda_p^i = \begin{cases} 1 & \text{if } p \in P_i, \\ 0 & \text{sinon.} \end{cases} \quad \text{for all } i \in I \setminus \{v_k\} \text{ and } p \in P_i
$$

Le problème de plus court chemin avec inclusion de nœuds est équivalent au programme linéaire en nombres entiers suivant

$$
\begin{aligned}
\min \quad & \sum_{i \in I \setminus \{v_k\}} \sum_{p \in \mathcal{P}_i} \sum_{a \in p} c_a \lambda_p^i \\
\sum_{p \in \mathcal{P}_i} \lambda_p^i = 1 & && \forall i \in I \setminus \{v_k\} \quad (\textbf{C1 :}\ \beta_i) \\
\sum_{i \in I \setminus \{v_k\}} \sum_{p \in P_i : a \in p} \lambda_p^i \leq 1 & && \forall a \in A \quad (\textbf{C2 :}\ \alpha_a) \\
\lambda_p^i \in \{0,1\} & && \forall i \in I \setminus \{v_k\}, \forall p \in P_i
\end{aligned}
$$

Les contraintes (C1) représentent les contraintes de convexité qui assurent la sélection d'un seul chemin de P_i et les contraintes (C2) garantissent que chaque arc appartient à au plus un chemin sélectionné.

Références

[1] Daniele Ferone, Paola Festa, and Francesca Guerrier (2020). An efficient exact approach for the constrained shortest path tour problem. *Optimization Methods and Software*, 35(1) :1–20.

[2] Daniele Ferone, Paola Festa, Francesca Guerriero, and Demetrio Lagana (2016). Solving the shortest path tour problem. *Computers & Operations Research*, 74 :64–77.

[3] Leonardo Taccar (2016). Integer programming formulations for the elementary shortest path problem. *European Journal of Operational Research*.

[4] Paola Festa, Francesca Guerriero, Demetrio Lagana, and Roberto Musmanno (2013). The constrained shortest path tour problem. *European Journal of Operational Research*, 230(3) :464–474.

[5] Rafael Castro de Andrade and Rommel Dias Saraiva (2018). An integer linear programming model for the constrained shortest path tour problem. *Electronic Notes in Discrete Mathematics*, 69 :141–148.

Formulations pour le problème de plus grand graphe partiel commun

Etienne de Gastines[1] et Arnaud Knippel[1]

1. Laboratoire LMI, INSA Rouen Normandie
685 avenue de l'Université - BP 08 76 801 Saint Etienne du Rouvray Cedex
{etienne.mace_de_gastines,arnaud.knippel}@insa-rouen.fr

Mots-clefs : plus grand graphe partiel commun, polyèdres, programmation linéaire 0-1

1 Introduction

Le problème du plus grand graphe partiel commun (Maximum Common Edge Subgraph - MCES) a été introduit en premier par Bokhari dans [4], pour résoudre le problème d'affectation de tâches à des processeurs tout en maximisant les demandes de communications entre tâches. Ce problème donne également une mesure de similarité entre deux graphes et est utilisé comme outil pour comparer des molécules.

Il peut être énoncé comme suit :

Soit $G = (V_G, E_G)$ et $H = (V_H, E_H)$ deux graphes. $S = (V_S, E_S)$ est un graphe partiel de G si $V_S \subseteq V_G$ et $E_S \subseteq V_G$. Si un graphe partiel de G est isomorphe à un graphe partiel de H, alors on le qualifie de graphe partiel commun à G et H. On recherche un graphe partiel commun à G et H maximal en terme de nombre d'arêtes. Nous considérons ici les graphes G et H comme étant simples et non-orientés. Le voisinage d'un sommet u est noté $N(u)$

Le problème est $\mathcal{NP}$-difficile et généralise les problèmes d'isomorphisme de sous-graphe, de clique maximale et de chemins hamiltoniens.

La première formulation en nombres entiers a été proposée par Marenco et Loiseau ([6], [7]). Elle introduit les variables $x_{u,v}$ $\forall u \in V_G, v \in V_H$ valant 1 si le sommet u est associé au sommet v, 0 sinon, et les variables y_{u_1,u_2} $\forall u_1u_2 \in E_G$ valant 1 si l'arête u_1u_2 appartient au sous graphe commun, 0 sinon. Une deuxième formulation a été proposée par Bahiense, Manić, Piva, et de Souza ([2], [3]), qui utilise les mêmes variables x, mais introduit pour sa part les variables $c_{u_1u_2,v_1v_2}$ $\forall u_1u_2 \in E_G, v_1v_2 \in E_H$, valant 1 si l'arête u_1u_2 est associée à l'arête v_1v_2.

2 Nouvelles formulations

La nouvelle formulation que nous proposons effectue un choix encore différent des précédentes, nous introduisons des variables $z_{u,v}$ $\forall u \in V_G, v \in V_H$ valant 1 si le sommet u est associé à un voisin du sommet v dans le graphe partiel commun, et réciproquement. Cela nous permet de proposer la formulation suivante (formulation réduite) :

$$\begin{aligned}
\max \quad & 1/2 \sum_{u\in V_G, v\in V_H} z_{u,v} & (1)\\
\text{s.t} \quad & \sum_{v\in V_H} x_{u,v} = 1, \quad \forall u \in V_G & (2)\\
& \sum_{u\in V_G} x_{u,v} = 1, \quad \forall v \in V_H & (3)\\
& z_{u_1,v_2} \leq \sum_{v_1\in N(v_2)} x_{u_1,v_1}, \quad \forall u_1 \in V_G, v_2 \in V_H & (4)\\
& z_{u_1,v_2} \leq \sum_{u_2\in N(u_1)} x_{u_2,v_2}, \quad \forall u_1 \in V_G, v_2 \in V_H & (5)\\
& x_{u,v} \in \{0,1\}, \quad \forall u \in V_G, v \in V_H & (6)\\
& z_{u,v} \in \{0,1\}, \quad \forall u \in V_G, v \in V_H. & (7)
\end{aligned}$$

Nous proposons aussi une version symétrisée de la formulation proposée par [2], qui donne une meilleure relaxation.

Nous avons testé numériquement les nouvelles formulations obtenues avec les deux anciennes formulations sur deux bases de données, une base de données de graphes issus de problèmes rassemblés par Marenco et Loiseau ([7]) et provenant essentiellement d'applications pratiques, et une partie de la base de donnés ARG ([5], [1]) composées de graphes générés aléatoirement selon divers modèles. Nous observons que la formulation réduite la plus compétitive des formulations.

Références

[1] ARG Database, https ://mivia.unisa.it/datasets/graph-database/arg-database/

[2] Bahiense, L. and Piva, B. and de Souza, C., *A branch&cut algorithm for the maximum common edge subgraph problem*, Electronic Notes in Discrete Mathematics **35** (2009), 47–52.

[3] Bahiense, L. and Manić, G and Piva, B. and de Souza, C., *The maximum common edge subgraph problem : A polyhedral investigation*, Discrete Applied Mathematics **160** (2012), 2523–2541.

[4] Bokhari, *On the Mapping Problem*, IEEE Transactions on Computers **3** (1981), 207–214.

[5] De Santo, M., Foggia, P., Sansone, C. and Vento, M., *A large database of graphs and its use for benchmarking graph isomorphism algorithms*, Pattern Recognition Letters **24** (2003), 1067–1079.

[6] Marenco, Javier and Loiseau, Irene, "A branch&cut algorithm for a problem arising in parallel programming environments ", Universidade de Buenos Aires, Departamento de Computación,2000.

[7] Marenco, Javier and Loiseau, Irene, "Un algoritmo branch-and-cut para el problema de mapping ", Master thesis, Universidade de Buenos Aires, Departamento de Computación,1999.

Modèle multicritère pour l'optimisation du problème de planification intégrée dans un terminal à conteneurs

Marwa Samrout[1], Adnan Yassine[1] et Abdelkader Sbihi[2]

1. Laboratoire LMAH, EA 3821, Université Le Havre Normandie
25 rue Philippe Lebon, BP 1123, 76063 Le Havre Cedex France
marwa.al-samrout@etu.univ-lehavre.fr
adnan.yassine@univ-lehavre.fr
2. Départment de Finance, Opérations et Marketing, Brest Business School
2 Avenue de Provence, 29200 Brest, France
abdelkader.sbihi@univ-paris1.fr

Mots-clefs : Terminaux à conteneurs, postes d'amarrage, grues de quai, optimisation multi-objectif, ordonnancement, planification intégrée, transbordement

1 Introduction

Le développement des terminaux portuaires à conteneurs constitue un des vecteurs du commerce mondial. A ce titre, une stratégie efficace d'ordonnancement des postes à quai et d'optimisation du temps de service (déchargement/chargement) est d'une importance cruciale pour augmenter le trafic et la compétitivité portuaires et conduire à une hausse des revenus, tout en optimisant la gestion environnementale et la satisfaction des clients. Cette étude a vocation à proposer un modèle mathématique multicritère capable de traiter de manière exhaustive les tâches de transbordement direct et indirect ainsi que l'allocation des postes d'amarrage (BAP) et l'affectation des grues de quai aux navires (QCAP) dans la pratique quotidienne du terminal. L'objectif est de minimiser le délai de séjour des navires au port en tenant compte des contraintes de disponibilité des ressources. Nous validerons le modèle multi-objectif de planification intégrée et nous proposerons des approches de résolution algorithmiques. Ensuite, nous réaliserons des simulations numériques sur des données réelles et d'autres générées aléatoirement afin de montrer l'efficacité des solutions obtenues et la robustesse du modèle mathématique développé. Enfin, nous dégagerons des indicateurs de performance afin de montrer que la planification intégrée des opérations est une réelle alternative qui améliore nettement les mesures d'efficacité d'un terminal portuaire.

2 Modélisation mathématique

En arrivant au terminal, un navire doit répondre à plusieurs tâches. Généralement, une partie de ses marchandises doit être transbordée directement dans un autre navire, une autre partie doit être transbordée d'une façon indirecte c.à.d transférée entre le moyen de transport terrestre et le navire afin qu'elle soit stockée temporairement dans la cour du terminal tandis que le reste de la cargaison peut être déchargé et stocké sans transbordement. Il existe de nombreuses combinaisons possibles de scénarios, qui peuvent être générées en modifiant la séquence des ordres de service pour chaque navire et chaque sous-scénario entraîne un temps d'attente différent pour ces navires. Dans ce modèle, le quai est partitionné en un nombre déterminé B de postes de taille égale. Chaque poste peut être occupé par un navire au plus à la fois. Et puisque le nombre de conteneurs emportés par un méga-navire est élevé, le temps de traitement requis pour traiter un tel navire peut dépasser une journée. Par conséquent, nous choisissons de discrétiser aussi le temps en utilisant des périodes relativement longues de 3 ou 4 heures [1]. On s'intéresse ici à un ensemble $\mathcal{V}$ de navires dont les dates d'arrivées sont connues, avec $\mathcal{V} = \mathcal{M} \cup \mathcal{F}$ où $\mathcal{M}$ et $\mathcal{F}$ représentent respectivement l'ensemble des navires mères et l'ensemble des navires "feeder". Pour chaque navire $i \in \mathcal{V}$, On définit : l_i longueur du navire i, p_i :durée de traitement du navire i, a_i :date d'arrivée du navire i et d_i : la date de départ estimée du navire i. Étant donné le vecteur d'informations $\{l_i, p_i, a_i, d_i\}$ d'un navire i arrivant au terminal, le problème d'optimisation consiste alors à déterminer la position d'accostage b_i de ce navire ($1 \leq b_i \leq B-1$ où B est l'ensemble de postes à quai), la date d'accostage bt_i, développer la séquence des ordres de service et attribuer l'heure de début de chaque opération de manutention à un niveau tactique pour chaque navire i. Le problème est adapté dans les hypothèses suivantes :

• Le navire doit être servi sans rupture dès son arrivée au terminal jusqu'à son départ
• Les opérations de transbordement des conteneurs entre les navires n'existent qu'entre les navires mères et les navires "feeder" et elles peuvent être effectuées entre plusieurs navires en même temps. Ainsi, les navires mères peuvent échanger des conteneurs avec six navires "feeders" au max, tandis que les navires "feeders" peuvent être affectés à trois navires-mères au max
• Les navires entrants peuvent accoster au quai sans aucune restriction physique et ils ont tous une capacité connue de conteneurs.
Les ensembles, les paramètres et les variables de décision utilisés sont les suivants :

$\mathcal{F}$ (resp. $\mathcal{M}$) : ensemble de navires "feeders" (resp. mères)
$\mathcal{Q}$: ensemble de grues de quai
$\mathcal{T}$: ensemble de périodes de temps
$\mathcal{S}$: ensemble de "shifts" de l'horizon de planification
O_i : ensemble d'ordres de services de navire i
F_i : ensemble de navires "feeder" ayant une tâche de transbordement avec le navire-mère i
M_i : ensemble de navires mères ayant des conteneurs à transférer au navire "feeder" i
e_i : date à partir de laquelle le navire i peut partir (date d'achèvement)
β_{ijk} : date de début de l'opération de transbordement ayant l'ordre de service k entre les navires i et j, avec $k \in O_i$
β_{ik} : date de début de l'opération de déchargement des conteneurs (sans transbordement) de navire i à l'ordre de service k, $k \in O_i$
t_i : durée nécessaire pour terminer le déchargement de tous les conteneurs de non-transbordement du navire i
t_{ji} : durée nécessaire pour traiter tous les conteneurs à transborder entre le navire "feeder" j et le navire mère i
Q_s^{max} : variable entière, nombre maximum de QCs utilisées pendant le "shift" de travail s
Q_s^{min} : variable entière, nombre minimum de QCs utilisées pendant le "shift" de travail s

$$r_{ij} = \begin{cases} 1 & \text{si le navire j accoste après le départ du navire i} \\ 0 & \text{sinon} \end{cases} \tag{1}$$

$$z_{ij} = \begin{cases} 1 & \text{si le navire j accoste complètement au-dessus du navire i sur le diagramme temps-espace} \\ 0 & \text{sinon} \end{cases} \tag{2}$$

$$DT_{ijk} = \begin{cases} 1 & \text{si le navire i gère les conteneurs de transbordement avec le navire j à l'ordre de service } k \\ 0 & \text{sinon} \end{cases} \tag{3}$$

$$NT_{ik} = \begin{cases} 1 & \text{si le navire i manipule des conteneurs sans transbordement à l'ordre de service } k \\ 0 & \text{sinon} \end{cases} \tag{4}$$

$$x_{itq} = \begin{cases} 1 & \text{si QC q est occupée par le navire i à l'intervalle de temps t} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Le premier terme de la fonction objectif (6) est la somme des temps de séjour d'un navire, y compris les deux temps en attente d'être à quai et le temps de traitement. Le deuxième terme est une pénalité accumulée par les navires en retard. Notant que la notation z^+ est égale à $max(z; 0)$. La deuxième fonction objectif (7) sert à minimiser la gamme de grues de quai maximale et minimale utilisées pour économiser les ressources.

$$min \sum_{k \in \mathcal{F} \cup \mathcal{M}} (c_k - a_k) + \sum_{k \in \mathcal{F} \cup \mathcal{M}} f_k.(c_k - d_k)^+ \tag{6}$$

$$min \sum_{s \in \mathcal{S}} (Q_s^{max} - Q_s^{min}) \tag{7}$$

tels que

$$r_{ij} + r_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i, j \in \mathcal{F} \cup \mathcal{M} \quad and \quad i < j \tag{8}$$

$$r_{ij} + r_{ji} \leq 1 \quad \forall i, j \in \mathcal{F} \cup \mathcal{M} \quad and \quad i < j \tag{9}$$

$$z_{ij} + z_{ji} \leq 1 \quad \forall i, j \in \mathcal{F} \cup \mathcal{M} \quad and \quad i < j \tag{10}$$

$$b_{tj} \geq e_i + (r_{ij} - 1).M \quad \forall i, j \in \mathcal{F} \cup \mathcal{M} \quad and \quad i \neq j \tag{11}$$

$$b_j \geq b_i + e_i + (z_{ij} - 1).M \quad \forall i, j \in \mathcal{F} \cup \mathcal{M} \quad and \quad i \neq j \tag{12}$$

$$b_{ti} \geq a_i \quad \forall i \in \mathcal{F} \cup \mathcal{M} \tag{13}$$

$$e_i \geq b_{ti} + p_i \quad \forall i \in \mathcal{F} \cup \mathcal{M} \tag{14}$$

$$b_i \leq B - (l_i + 1) \quad \forall i \in \mathcal{F} \cup \mathcal{M} \tag{15}$$

$$b_i \geq 1 \quad \forall i \in \mathcal{F} \cup \mathcal{M} \tag{16}$$

$$\sum_{k \in O_i} DT_{ijk} = \sum_{k' \in O_j} DT_{ijk'} = 1 \quad \forall i \in \mathcal{M}, \forall j \in F_i \tag{17}$$

$$\sum_{j \in F_i \cup M_i} DT_{ijk} + NT_{ik} = 1 \quad \forall i \in \mathcal{F} \cup \mathcal{M}, \forall k \in O_i \tag{18}$$

$$\sum_{k \in O_i} NT_{ik} = 1 \quad \forall i \in \mathcal{F} \cup \mathcal{M} \tag{19}$$

$$\sum_{j \in F_i \cup M_i} \beta_{ij1}.DT_{ij1} + \beta_{i1}.NT_{i1} \geq a_i \quad \forall i \in \mathcal{F} \cup \mathcal{M} \tag{20}$$

$$\sum_{j \in F_i \cup M_i} (\beta_{ij,k-1} + t_{ji}).DT_{ijk-1} + (\beta_{i,k-1} + t_i).NT_{i,k-1} \leq \sum_{j \in F_i \cup M_i} \beta_{ijk}.DT_{ijk} + \beta_{ik}.NT_{ik} \quad \forall i \in \mathcal{F} \cup \mathcal{M}, \forall k \in O_i, k > 1 \tag{21}$$

$$\sum_{k \in O_i} \beta_{ijk}.DT_{ijk} = \sum_{k' \in O_j} \beta_{ijk'}.DT_{ijk'} \quad \forall i \in \mathcal{M}, \forall j \in F_i \tag{22}$$

$$\sum_{i \in \mathcal{F} \cup \mathcal{M}} x_{itq} \leq 1 \quad \forall t \in \mathcal{T}, q \in \mathcal{Q} \tag{23}$$

$$\sum_{q\in\mathcal{Q}} x_{itq} \leq |A_i| \quad \forall i \in \mathcal{F}\cup\mathcal{M}, t \in \mathcal{T} \tag{24}$$

$$q \leq q' + |Q|.(2 - x_{itq} - x_{jtq'}) \quad \forall i \in \mathcal{F}\cup\mathcal{M}, t \in \mathcal{T}, y_{ij} = 1, r_{ij} = r_{ji} = 0 \tag{25}$$

$$Q_s^{max} \geq \sum_{i\in\mathcal{F}\cup\mathcal{M}} \sum_{q\in\mathcal{Q}} x_{itq} \quad \forall t \in T_s, s \in \mathcal{S} \tag{26}$$

$$Q_s^{min} \leq \sum_{i\in\mathcal{F}\cup\mathcal{M}} \sum_{q\in\mathcal{Q}} x_{itq} \quad \forall t \in T_s, s \in \mathcal{S} \tag{27}$$

Les contraintes (8) à (10) garantissent que les bateaux ne se chevauchent.
Les contraintes (11) et (12) garantissent que les heures d'accostage et les positions d'accostage sélectionnées sont cohérentes avec les définitions de r_{ij} et z_{ij}, où M est un entier positif grand.
Les contraintes (13) et (14) forcent l'heure d'accostage à se produire au plus tôt à l'heure d'arrivée et l'heure de départ à se produire au plus tôt à l'heure de fin du service.
Les contraintes (15) et (16) garantissent que tous les navires tiennent sur le quai.
La contrainte (17) indique que les conteneurs de transbordement entre deux navires doivent être manipulés une seule fois pour tous les navires.
La contrainte (18) garantit que les navires entrants doivent travailler avec une seule opération de manutention au même ordre de service.
La contrainte (19) impose que les conteneurs de non-transbordement pour chaque navire ne doivent être entretenus qu'une seule fois.
La contrainte (20) garantit que chaque navire manipulant une des opérations assignées au premier ordre de service doit commencer après son arrivée au terminal.
La contrainte (21) garantit que l'heure de début de l'une des opérations de manutention affectée à un certain navire en tant que k^{eme} (lorsque $k > 1$) ordre de service, ne doit pas être inférieure à l'heure d'achèvement de son prédécesseur.
Éq. (22) garantit que les dates de début de l'opération de transbordement direct doivent être les mêmes pour les deux navires (j, i), malgré l'ordre de service prévu pour chaque navire.
Éq. (23) signifie qu'une QC ne doit pas être utilisée par plus d'un navire à la fois.
Éq. (24) garantit que le nombre de QCs utilisées par un navire ne doit pas dépasser son max.
Éq. (25) est utilisé pour empêcher le croisement des QCs.
Eq. (26) et (27) donnent la définition du nombre max et min de QCs utilisées.

Conclusion

Nous avons travaillé sur l'optimisation combinatoire avec un intérêt initial pour la programmation par contrainte et l'ordonnancement. Et par conséquent, un modèle mathématique multi-objectif de planification intégrée était construit et validé par l'intermédiaire d'un l'outil informatique d'optimisation commercialisé, le logiciel NetBeans IDE 8.2. Le programme est écrit en Java avec une bibliothèque CPLEX. Des simulations numériques sont menées afin de montrer la robustesse du modèle.

Références

[1] Ak, A. (2008). Berth and quay crane scheduling : problems, models and solution methods (Doctoral dissertation, Georgia Institute of Technology).

On the diameter and the circuit-diameter of polytopes

Laura Sanita

TU Eindhoven, Netherlands

The diameter of a polytope P is the maximum length of a shortest path between a pair of vertices of P, when one is allowed to walk on the edges (1-dimensional faces) of P. Despite decades of studies, it is still not known whether the diameter of a d-dimensional polytope with n facets can be bounded by a polynomial function of n and d. This is a fundamental open question in discrete mathematics, motivated by the (still unknown) existence of a polynomial pivot rule for the Simplex method for solving Linear Programs. A generalized notion of diameter, recently introduced in the literature, is that of circuit-diameter, defined as the maximum length of a shortest path between two vertices of P, where the path can use all edge directions (called circuits) that can arise by translating some of the facets of P. In this talk, I will discuss some algorithmic and complexity results related to the diameter and the circuit-diameter of polytopes, highlighting important open questions.

MIP based approaches for strategic decisions on the Freight on Public Transport problem

Laurent Alfandari, Claudia Archetti,
Diego Delle Donne and Ivana Ljubic

ESSEC Bussines School of Paris, France.

Freight transportation is a key issue within big cities, involving a significant amount of activities with undesirable consequences on the traffic, the environment and subsequently on human health. Most big cities in the world are facing significant challenges related to the congestion and pollution generated by the number of vehicles which need to travel within urban areas. Freight transportation vehicles represent a significant portion of this volume and their number is expected to continue increasing due to new business models. Recent studies have shown the viability of innovative strategies for performing sustainable last-mile delivery operations by integrating existing public transportation services (PTS) into city logistics related to freight delivery. The idea is to use PTS to carry parcels between some predetermined points in the city, replacing usual freight transportation vehicles, and completing the last part of the routes with small, "green"vehicles. Many challenges arise on strategical, tactical and operational aspects of the proposed methodologies. In this work, we focus on the strategical stage, which asks to select a subset of city resources (i.e., bus lines, bus stops) to maximize the potential freight flow through the city restricted to strategical and budget constraints. We propose different Mixed Integer Programming approaches for this problem and perform an exhaustive computational experimentation to test the practical contribution of the proposed models.

A Branch-and-Cut algorithm for the Proactive Countermeasures Selection Problem

A. Ridha Mahjoub[1], M.Yassine Naghmouchi[1,2], Nancy Perrot[2]

1. Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016, Paris, France
{mahjoub,mohamed-yassine.naghmouchi}@lamsade.dauphine.fr

2. Orange Labs, France
nancy.perrot@orange.com

Mots-clefs : Polyhedra, Branch-and-Cut, countermeasure selection.

1 Introduction

We consider the Proactive Countermeasures Selection Problem, defined in [1]. An instance of the PCSP is given by a triplet (G, K, D). Here, G = (V, A) is a directed graph called the Risk Assessment Graph [2], where $V = S \cup T$, $S \cap T = \emptyset$. Each arc (i, j) of A has a weights $w_{ij} \in \mathbb{R}_+$. $K = \{(t, k) : k \in K_t, t \in T\}$ is a set of available countermeasures such that K_t is the set of countermeasures associated with t. The placement of k on t has a positive cost $c_k^t \in \mathbb{R}+$, and it increases the weight of t-ongoing arcs by a factor $\alpha_t^k \in \mathbb{R}_+$. $D = (d_t^s)_{s \in S, t \in T} \in \mathbb{R}_+$ is a positive security threshold vector. The PCSP consists in selecting a set of countermeasures, at minimal cost, such that for each $(s, t) \in S \times T$ the length of the $s - t$ shortest path is at least d_s^t. A bilevel model, as well as two compact and extended formulations, were introduced in [1] to solve the PCSP.

In this abstract, we give polyhedral results and a Branch-and-Cut algorithm developed for solving the extended formulation as well as some numerical results showing the efficiency of the algorithm.

2 Polyhedral investigation and Branch-and-Cut algorithm

We study the polytope associated to the path formulation. We characterize the dimension of the polytope by considering *the essential countermeasures set*, i.e., the countermeasures such that if we remove at least one of them, the PCSP does not have a solution. We then introduce several classes of valid inequalities, namely *the path covering inequalities*, *the countermeasures path inequalities* and *the essential-by subsets removing-countermeasure inequalities*. We discuss when these inequalities define facets. We also study the optimality conditions for the PCSP.

The polyhedral results are used within a Branch-and-Cut algorithm. We develop a preprocessing phase considering the essential countermeasures equations and the optimality condition inequalities. We devise separation routines for the basic and valid inequalities. In particular, we propose exact separation algorithms for both the basic inequalities and the countermeasures path inequalities.We also prove that the separation problems of path covering inequalities and essential-by subsets removing-countermeasure inequalities are NP-Complete requiring then the use of heuristics to separate them. In addition, we have provided a primal heuristic in order to reduce the size of the Branch-and-Cut tree and accelerate the resolution of the problem.

3 Numerical Results

We conduct extensive experimentations on random and realistic instances of the PCSP problem. In this section, we present some numerical tests of the compact formulation and the path formulation. The computational study shows the efficiency of the polyhedral results. The optimality condition inequalities and valid inequalities play an important role in the resolution of the problem as we can see in Table 1. The entries of the table are the following:

$\lvert S\rvert$	:	number of access points;
$\lvert T\rvert$	:	number of asset-vulnerability nodes;
$I\_\lvert S\rvert, \lvert T\rvert$	:	name of the instance;
$\lvert A\rvert$	:	number of arcs;
$\lvert K\rvert$	:	number of countermeasures;
N	:	number of nodes in the Branch & Cut tree;
$NOpt$	:	the number of instances solved to optimality / total number of instances;
Gap	:	the relative error between the best upper bound (the optimal solution if the problem has been solved to optimality) and the lower bound obtained at the root,
CPU	:	total CPU time (in the format hh:mm:ss).

			Compact formulation				Branch and Cut			
$Name$	$\lvert A\rvert$	$\lvert K\rvert$	N	Gap	CPU	$NOpt$	N	Gap	CPU	$NOpt$
I_10,100	139.2	201	34	0.07	0:1: 5	5/5	43.6	0.06	0:01:2	5/5
I_20,200	500.2	667.2	70.8	0.09	1:38:2	5/5	153	0.08	0:12:2	5/5
I_30,300	1103.6	1100.8	116	0.16	4:10:5	1/5	177	0.08	0:49:3	5/5
I_40,400	1864.8	981.8	120.2	0.39	-	0/5	63.8	0.09	3:49:3	3/5
I_50,500	2676.4	1674.4	138	0.23	3:41:1	1/5	157.2	0.09	3:22:3	4/5
I_60,600	3276.2	1613	88.2	0.41	-	0/5	79.8	0.10	1:52:3	3/5
I_70,700	3988.4	1876.2	57.8	0.49	-	0/5	104	0.09	2:57:6	3/5
I_80,800	4694	2854.4	68.2	0.51	-	0/5	92.6	0.13	3:52:7	2/5
I_90,900	6204.6	2765.4	54	0.45	-	0/5	79	0.12	3:19:5	3/5
I_100,1000	10866.4	3596	-	-	-	0/5	107	0.19	4:42:3	1/5
I_110,1100	14444.6	4987.2	-	-	-	0/5	103	0.23	4:49:5	1/5
I_120,1200	19546.6	5503.6	-	-	-	0/5	69	0.37	-	0/5

Table 1: Comparison with the compact formulation for a family of random instances

References

[1] A. R. MAHJOUB, M. Y. NAGHMOUCHI, and N. PERROT. *A Bilevel Programming Model for Proactive Countermeasure Selection in Complex ICT Systems.* Electronic Notes in Discrete Mathematics, 2018, vol. 64, p. 295-304.

[2] M. Y. NAGHMOUCHI, N. PERROT, N. Kheir,A. R. MAHJOUB, and J. P. Wary, *A new risk assessment framework using graph theory for complex ICT systems.* Managing Insider Security Threats (pp. 97-100). CCS, ACM.

IP formulations for vehicle routing with stochastic demands

Ricardo Fukasawa

University of Waterloo, Canada

The deterministic vehicle routing problem consists on finding the cheapest set of routes to serve a set of customers, each with a fixed demand, subject to respecting capacity constraints on each route. In the stochastic variant, demands are considered random variables. Most approaches for such problems assume independence of such random variables. We discuss progress in formulating the problem as an integer program without such assumptions.

Recognizing box-TDI polyhedra is coNP-complete

Francesco Pisanu

Mots-clefs : Box-TDI polyhedron, linear system, NP-hardness

1 Abstract

Box-totally dual integral (box-TDI) systems are linear systems which yield strong min-max relations, such as the MaxFlow-MinCut theorem. In 2008, Ding, Feng, and Zang [1] proved that deciding whether a linear system is box-TDI is coNP-complete.

A polyhedron that can be described by such a system is called a box-TDI polyhedron. Surprisingly, these polyhedra admit matricial and geometric characterizations that do not depend on the system chosen to describe them. In particular, the complexity of recognizing such polyhedra does not follow from the result of Ding, Feng, and Zang.

We prove that recognizing box-TDI polyhedra is coNP-complete.

This is a joint work with Patrick Chervet, Roland Grappe, Mathieu Lacroix, and Roberto Wolfler Calvo.

Références

[1] Ding, G., Feng, L. Zang, W. The complexity of recognizing linear systems with certain integrality properties. Math. Program. 114, 321–334 (2008).

Index

LISTE DES PARTICIPANTS

Hassene Aissi
Université Paris Dauphine

Goyal Akshit
University Of Minnesota

Arenas Vasco Alejandro
Universidad Eafit

Zacharie Ales
UMA, ENSTA Paris - CEDRIC

Carlos Alejandro Alfaro
Banxico

Lamia Aoudia
USTHB

Zohra Aoudia
Universitè de Bejaia

Fatiha Bendali
Université Clermont Auvergne-LIMOS

Pascale Bendotti
EDF R&D - LIP6

Isma Bentoumi
Lamsade, Université Paris-Dauphine PSL

Matthias Brugger
TU Munich

François Clautiaux
Université de Bordeaux

Rafael Colares
Orange

Danai Deligeorgaki
KTH

Diego Delle Donne
ESSEC

Onur Demiray
EPFL

Aloïs Duguet
LAAS

Alexandre Dupont-Bouillard
LIPN - Université Sorbonne Paris Nord

Ayoub El-Omary
Université Polytechnique Hauts-De-France

Foivos Fioravantes
Université Cote D'azur, INRIA, CNRS

Jonathan Fontaine
CEA - List

Pierre Fouilhoux
LIPN - Université Sorbonne Paris Nord

Estéban Gabory
Université de Paris

Paolo Gianessi
Mines Saint-Étienne

Natividad González Blanco
University Of Seville

Rodolphe Griset
EDF

Evgeny Gurevsky
Université de Nantes

Youssouf Hadhbi
LIMOS

Youssouf Hadhbi
LIMOS

Khadija Hadj Salem
INESC TEC, Porto, Portugal

Hela Haj Mohamed
FSM Université Monastir

Alexandre Heintzmann
LAAS-CNRS

Himanshu Himanshu Gupta
University Of Delaware

Cédric Joncour
Université Le Havre Normandie, Normandie Université, LMAH

Herve Kerivin
Uca - Limis

Muluk Komal
Rwth Aachen University, Germany

Yerlan Kuzbakov
ESSEC

Craig Larson
Virginia Commonwealth University

Arnaud Laurent
INRIA Lille

Safa Layeb
National Engineering School Of Tunis

Yuanyuan Li
ESSEC Business School

Tom Liebling
EPFL

Pedro Liguori
BCG Gamma

Ivana Ljubic
ESSEC

A. Ridha Mahjoub
LAMSADE - Université Paris Dauphine

Jean Mailfert
LIMOS - Clermont-Ferrand

Amin Mallek
Zetem O2c/Bremen Universität

Ibrahim Mamane Souleye
Université Abdou Moumouni

Sébastien Martin
Huawei

Ota Matheus Jun
University Of Waterloo

Mouna Medimagh
Faculté Des Sciences de Monsatir

Imen Mejri
National Engineering School Of Tunis

Guillaume Michel
LIRMM

Asma Miled
Faculté Des Sciences de Monastir

Mehdi Naima
Université Sorbonne Paris Nord

Salima Nait Belkacem
Universite

José Neto
Télécom Sudparis, Institut Polytechnique De Paris

Alantha Newman
G-SCOP

Viet Hung Nguyen
LIMOS - Clermont Auvergne Inp

David Nizard
Université Paris Saclay

Charles Nourry
Université Paris-Dauphine

Pawan Pawan Aurora
Iiser Bhopal

Mercedes Pelegrín
École Polytechnique

Pierre Pesneau
IMB - Université de Bordeaux

Thibault Prunet
Ecole Des Mines de Saint Etienne

Maurice Queyranne
UBC Sauder School Of Business

Shaghayegh Ramezanpour Shalmani
ESSEC Business School

Atefi Reza
Ferdowsi University Of Mashhad

Cécile Rottner
EDF R&D

Charbel Salameh
Accenta

Emna Salhi
Faculté Des Sciences de Monastir

Phillippe Samer
Universitetet I Bergen

Marwa Samrout
LMAH Université Le Havre Normandie

Mariam Sangare
Laboratoire D'informatique, de Robotique Et De Microélectronique de Montpellier

Andreas S. Schulz
TU Munich

Raouia Taktak
Sfax University, Tunisia

Sophie Toulouse
Université Paris 13

Manuel Trotta
Université Clermont Auvergne

Sonia Vanier
Université Paris1 Panthéon-Sorbonne

Thi Quynh Trang Vo
LIMOS, Universite Clermont - Auvergne

Naghmouchi Yassine
ENSTA