Journées Polyèdres et Optimisation Combinatoire-JPOC'13

Clermont-Ferrand
28 Juin 2023 — 30 juin 2023

Invités

F. Barahona, IBM New York
S. Boyd, U. d'Ottawa
C. D'Ambrosio, Polytechnique, Paris
F. Foucaud, U. Clermont Auvergne
J. Salazar, U. de La Laguna
A. Sebo, CNRS, Grenoble

Comité d'organisation

M. Baïou
B. Bourdieu
F. Bendali
I. Diarrassouba
P. Fouilhoux
H. Kérivin
V. Limouzy
I. Ljubic
A. R. Mahjoub
J. Mailfert
M. Quinio
H. Toussaint
S. Vanier

EDITO

Nous sommes heureux de vous accueillir à Clermont-Ferrand, du 28 au 30 juin 2023, pour la treizième édition des Journées Polyèdres et Optimisation Combinatoires JPOC13.

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. C'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 et en algorithmes d'approximation, mais aussi en performance de résolution de problèmes pratiques difficiles et de grande taille.

Les approches polyédrales constituent un des outils puissants de l'optimisation combinatoire. Initiées en 1965 par Jack Edmonds dans son étude du problème du couplage, elles sont maintenant de plus en plus utilisées pour résoudre les problèmes difficiles d'aide à la décision. Ces techniques qui s'appuient sur l'évolution des outils informatiques de calcul, couplées avec d'autres méthodes comme la génération de colonnes, permettent d'élaborer des algorithmes de résolution d'une efficacité redoutable.

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 90 chercheurs, pour nous retrouver autour d'exposés dans le domaine de l'optimisation combinatoire. Ces deux journées et demie comprennent six plénières de chercheurs invités, sept exposés semi-pléniers et dix-huit exposés de doctorants.

Merci d'être venu pour les 20 ans de cette conférence !

Le comité d'organisation

PROGRAMME

Mercredi 28 juin 2023

Accueil : 11h00-12h00 et 13h00-14h00

14h00 - 14h30 **Ouverture des journées**
Baiou Mourad, Ridha Mahjoub

14h30 - 15h20
Identification problems in graphs and other discrete structures
Florent Foucaud

15h20 - 15h50
Submodular maximization of concave utility functions composed with a set-union operator with applications to maximal covering location problems
Stefano Coniglio, Fabio Furini, ***Ivana Ljubic***

15h50 - 16h10
Formulation étendue pour le problème de l'arbre couvrant budgeté
Hassene Aissi, A. Ridha Mahjoub, ***Charles Nourry***

16h10 - 16h30
How to cut efficiently in bi-objective Branch&Cut algorithm ?
*Pierre Fouilhoux, Lucas Létocart,****Yue Zhang***

17h00 - 18h30 **Evénement Social**

19h30 - 20h15 **Apéritif de bienvenue**

Jeudi 29 juin 2023

08h30 - 09h20

Mathematical Formulations for Consistent Travelling Salesman Problems

Juan José Salazar Gonzalez

09h20 - 09h50

Sequential matroid-game against greedy

Denis Cornaz, *Michael Lampis, Emiliano Lancini*

09h50 - 10h10

Optimization methods for the multi-commodity flow blocker problem

Isma Bentoumi, *Fabio Furini, A. Ridha Mahjoub, Sebastien Martin*

10h10 - 10h30

Triangulation de Hilbert unimodulaire des cônes simples totalement équimodulaires

Roland Grappe, ***Mathieu Vallée***

10h30 - 11h00 Pause Café

11h00 - 11h50

Perspective Formulations for piecewise convex functions : a theoretical and computational comparison

Claudia D'Ambrosio

11h50 - 12h10

A new hybrid method for unconstrained quadratic programming combined with the techniques of semidefinite programming and branch-and-bound method

Rabih Battikh, *Hassan Alabboud, Jida Bassem, Yassine Adnan*

112h10 - 12h30

Algorithme de Branch&Cut exploitant les symétries du polytope du sac-à-dos matriciel symétrique en poids

Pascale Bendotti, ***Alexandre Heintzmann***, *Cécile Rottner*

12h30 - 14h00 Déjeuner

14h00 - 14h50

The 4/3 Conjecture : Is it true or false ?

Sylvia Boyd

14h50 - 15h20

Online Covering with Multiple Experts

Enicõ Kevi, ***Kim Thang Nguyen***

15h20 - 15h40

An efficient 2-competitive online algorithm for kit update at MSF Logistique

Boris Detienne, ***Mickael Gaury***, *Gautier Stauffer*

15h40 - 16h00

Pickup and Delivery Problem with Cooperative Robots

Jean-Philippe Gayon, Viet Hung Nguyen, ***Chi Thao Nguyen***, *Alain Quilliot, Anh Son Ta*

16h00 - 16h30 Pause Café

16h30 - 17h20

A new algorithm for increasing the weight of minimum spanning trees and hypertrees
Mourad Baïou, ***Francisco Barahona***

17h20 - 17h40

An hypergraph based formulation for an Automatic Storage Design problem
François Clautiaux, Aurélien Froger, ***Luis Marques***

17h40 - 18h00

Nouveaux modèles pour la construction d'arbres de classification optimaux
Zacharie Ales, ***Valentine Huré****, Amélie Lambert*

19h30 - 22h00 **Repas de Gala**

Vendredi 30 juin 2023

08h30 - 08h50

On the star forest polytope for cactus graphs 🗎

*Viet Hung Nguyen,**Thanh Loan Nguyen**, Minh Hieu Nguyen*

08h50 - 09h40

Geometric Packing, Hitting and Representation : the simplest Open Challenges on Geometric Intersection Graphs 🗎

*Marco Caoduro,**Andras Sebo***

09h40 - 10h10

Maximum chordal subgraph problem 🗎

*François Clautiaux,**Pierre Pesneau***

10h10 - 10h30

On the box-total dual integrality of the perfect matching polytope 🗎

*Roland Grappe, Mathieu Lacroix,**Francesco Pisanu**, **Francesco Pisanu**, Roberto Wolfler Calvo*

16h00 - 16h30 Pause Café

11h00 - 11h30

What is the gradient of a Linear Program ? Automatic differentiation on a polytope 🗎

*Léo Baty, Louis Bouvier, **Guillaume Dalle**, Axel Parmentier*

11h30 - 11h50

A branch-and-bound algorithm for two-stage no-wait hybrid flow shop scheduling with interstage flexibility 🗎

*MN. Azaiez, A. Gharbi, Imed Kacem,**Yosra Makhlouf***

11h50 - 12h10

Quantum speed-ups for single-machine scheduling problems 🗎

*Eric Bourreau, **Camille Grange**, Michael Poss, Vincent T'Kindt*

12h10 - 12h30

Cutting Plane and Column Generation Algorithms for the Survivable Constrained-Routing and Spectrum Assignment Problem 🗎

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

12h30 - 14h00 Déjeuner

14h00 - 14h30

Smoothed analysis of the simplex method 🗎

Sophie Huiberts

14h30 - 15h00

The quickest route problem 🗎

Jean-François Maurras

15h00 - 15h20

Formulation étendue pour le polytope des co-2-plexes 🗎

***Alexandre Dupont-Bouillard**, Pierre Fouilhoux, Roland Grappe, Mathieu Lacroix*

15h20 - 15h40

Formulations linéaires pour le problème d'isomorphisme de sous graphes non induits
Etienne De Gastines, *Arnaud Knippel*

15h40 - 16h00

Initial Lagrangian Multipliers Prediction Based on GNNs to Speed Up Bundle Methods
Francesco Demelas, *Mathieu Lacroix, Joseph Le Roux, Axel Parmentier*

16h00 - ∞ **Goûter de clôture des journées**

Identification problems in graphs and other discrete structures

Florent Foucaud, Enseignant-chercheur

Laboratoire LIMOS, UMR CNRS 7024, INP et Université Clermont Auvergne,
Campus Universitaire des Cézeaux.1 rue de la Chebarde.TSA 60125. CS 60026 63178
AUBIERE CEDEX
France
florent.foucaud@uca.fr

Mots-clefs : identification problems on graphs

1 Abstract

We will address selected topics in the area of identification problems on graphs and other discrete structures. In this type of problems, we wish to select a small set of elements (usually vertices, but sometimes edges, paths, etc) with the goal of distinguishing a set of structures (usually, the vertices or the edges of the graph) by means of membership, neighbourhood or distances to the elements in the solution set. We will highlight a number of recent interesting results, problems and conjectures in the area.

Submodular maximization of concave utility functions composed with a set-union operator with applications to maximal covering location problems

Stefano Coniglio and Fabio Furini and Ivana Ljubić

1. University of Bergamo, Bergamo, Italy
stefano.coniglio@unibg.it

2. University of La Sapienza, Rome, Italy
fabio.furini@uniroma1.it

3. ESSEC Business School of Paris, Cergy-Pontoise, France
ljubic@essec.edu

Keywords : Submodular maximization, Benders decomposition, Utility Functions, Mixed-Integer Nonlinear Optimization

1 Introduction

The problem of maximizing a concave, strictly increasing and differentiable utility function $f : \mathbb{R} \to \mathbb{R}$ over a discrete set plays an important role in many investment problems with discrete choices, competitive facility location or combinatorial auctions. In such utility-maximization problems, the function f typically models the decision maker's risk-averse attitude. When some of the coefficients of f are subject to uncertainty, the tools of stochastic optimization are used to maximize it in expectation over a typically discrete set of scenarios.

In this work, we study a generalization of a problem introduced in [1] in which we are given a set N of n *items*, and an additional ground set $\hat{N}$ of $\hat{n}$ *metaitems*. We assume that the two ground sets N and $\hat{N}$ are linked by a *covering relationship* modeled by a bipartite graph $G = (\hat{N} \cup N, E)$ where, for each $j \in N$ and $\ell \in \hat{N}$, $\{j, \ell\} \in E$ if and only if item j is covered by metaitem ℓ. For each item $j \in N$, we define by $\hat{N}(j) \subseteq \hat{N}$ the set of metaitems of $\hat{N}$ by which j is covered and, for each $\ell \in \hat{N}$, we define by $N(\ell) \subseteq N$ the set of items that are covered by ℓ. In order to access an item j of N and benefit from its profit, the decision maker has to select at least one of the metaitems $\ell \in \hat{N}$ covering it.

From a stochastic-optimization perspective, we assume that we are given a finite set M of m discrete scenarios, each occurring with a strictly positive probability π_i, $i \in M$. In addition, let $a_{ij} \in \mathbb{R}_+$, $i \in M, j \in N$, and $d_i \in \mathbb{R}_+$, $i \in M$, be nonnegative real numbers. The values of a_{ij} represent the profit collected under scenario i if item j is "covered". To model the behavior of risk-averse decision makers, we consider a utility function f, such as the negative exponential function

$$f(z) = 1 - \exp\left(- \frac{z}{\lambda} \right), \tag{f}$$

which is frequently used in the literature (see, e.g., [1,5]). Here, $\lambda > 0$ represents the risk-tolerance parameter and, the larger the value of λ and, thus, the closer f is to being linear, the less risk-averse the decision maker is.

The problem we study asks for finding a subset of metaitems (subject to given cardinality or other constraints) which maximizes the expected utility of the sum of the profit of covered items.

After introducing a binary variable $y_\ell \in \{0,1\}$ for each $\ell \in \hat{N}$, equal to 1 if and only if metaitem ℓ is chosen, and the binary variable x_j set to one if and only if the item $j \in N$ is covered, the problem can be cast as the following mixed-integer non-linear program (MINLP):

$$\max_{\substack{x \in \{0,1\}^n \\ y \in Y \cap \{0,1\}^{\hat{n}}}} \quad \sum_{i \in M} \pi_i \, f\Big(\sum_{j \in N} a_{ij} \, x_j + d_i \Big) \tag{1a}$$

$$\text{s.t.} \quad x_j \leq \sum_{\ell \in \hat{N}(j)} y_\ell \qquad j \in N \tag{1b}$$

$$y_\ell \leq x_j \qquad \ell \in \hat{N}, j \in N(\ell), \tag{1c}$$

where Y is a polyhedron encapsulating the constraints imposed on the set of metaitems (such as caradinality or budget constraints) and $Y \cap \{0,1\}^{\hat{n}}$ is the discrete set of feasible choices of metaitems. The objective function corresponds to the expected value of f over the given set of scenarios. For each $j \in N$, Constraints (1b) and (1c) imply that an item $j \in N$ is taken if and only if at least a metaitem ℓ in $\hat{N}$ is chosen from $\hat{N}(j) \subseteq \hat{N}$. Since the objective function of the problem is strictly increasing, Constraints (1c) are automatically satisfied in any optimal solution and can be dropped. Besides being binary, no further constraints are assumed for x.

2 Our Conribution

Throughout the paper, we focus on the case with $n \gg \hat{n}$, i.e., with a number of items orders of magnitude larger than the number of metaitems, which holds in many applications, including those in competitive facility location [3], or influence maximization [4]. We derive a novel exact solution method for solving Problem (1) based on a double-hypograph decomposition. In this approach, the many variables associated with the item set N are projected out and the problem is formulated using the binary y variables and an additional set of n auxiliary variables. Such a decomposition exploits the structural properties of the utility function f and of the set-union operator (which is used to model the covering relationships). The function f is linearized via an outer-approximation method, whereas the set-union operator is linearized in two ways: either (*i*) via a reformulation based on submodular cuts or (*ii*) via a Benders decomposition. Assuming f can be computed in constant time, the inequalities arising from the outer approximation combined with the Benders decomposition can be separated in linear time even for fractional points. We compare the strength of the two resulting mixed integer linear programming reformulations from a theoretical perspective, and also show how to extend them to the case where the utility function is not necessarily increasing. Lastly, we embed the reformulations in two branch-and-cut algorithms, the most efficient of which, according to our computational experiments, allows for solving to optimality instances with up to $n = 20,000$, $\hat{n} = 60$, and $m = 100$ in a short amount of computing time.

The full version of this work has appeared in [2].

References

[1] S. Ahmed, A. Atamtürk (2011). Maximizing a class of submodular utility functions. *Mathematical Programming 128(1-2), 149–169.*

[2] S. Coniglio, F. Furini, I. Ljubić (2022). Submodular maximization of concave utility functions composed with a set-union operator with applications to maximal covering location problems. *Mathematical Programming 196(1): 9-56.*

[3] I. Ljubić, E. Moreno (2018). Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *European Journal of Operational Research 266(1), 46-56.*

[4] H. Wu, S. Küçükyavuz (2018). A two-stage stochastic programming approach for influence maximization in social networks. *Computational Optimization with Applications 69(3), 563-595.*

[5] J. Yu, S. Ahmed (2016). Maximizing expected utility over a knapsack constraint. *Operations Research Letters 44(2), 180–185.*

Formulation étendue pour le problème de l'arbre couvrant budgeté

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 polyédrales, formulation étendue, inégalité valide, théorie des graphes

1 Introduction du problème

Le problème de l'arbre couvrant budgeté se situe à l'intersection du problème de l'arbre couvrant de poids minimum et du problème du sac à dos, deux problèmes très connus dans le domaine de la recherche opérationnelle. Il peut se définir de la manière suivante ; étant donné une quantité B d'une ressource et un graphe $G = (V, E)$ où chaque arête $e \in E$ est associée à un coût $c_e \in \mathbb{Q}$ et une consommation en ressource $w_e \in \mathbb{Q}$. Une solution réalisable pour le problème de l'arbre couvrant budgeté correspond donc à un arbre couvrant T satisfaisant la contrainte de budget $\sum_{e \in E(T)} w_e \leq B$. Ce problème peut être retrouvé dans de nombreux cas pratiques, notamment dans le domaine de la télécommunication, mais également en tant que sous-problème dans certaines décompositions. Etant à l'intersection d'un problème polynomial et d'un problème faiblement NP-difficile, le problème de l'arbre budgeté à été montré comme étant également faiblement NP-difficile par [1]. Plusieurs approches ont été utilisées pour résoudre ce problème, notamment des algorithmes d'approximations comme [3,4]. Nous abordons ce problème avec une approche basée sur la programmation mathématique. La contrainte de budget va modifier la structure des polyèdres plus ou moins connus associés aux différentes formulations du problème de l'arbre couvrant. L'objectif principal de notre travail est de comprendre ces nouvelles descriptions, trouver des formulations étendues et les exploiter pour trouver des inégalités valides afin de concevoir des algorithmes de branchements efficaces pour résoudre ce problème de réseau budgété.

2 Formulation étendue du problème

Le problème de l'arbre couvrant classique a été beaucoup étudié, il existe de nombreuses formulations en programme linéaire afin de modéliser ce problème, dont celle basée sur l'élimination des sous-tours, qui est entière. Dans le cas budgeté, on ajoute la contrainte de sac à dos ainsi que des contraintes d'intégrité car l'introduction de cette nouvelle inégalité va faire apparaître des points extrêmes fractionnaires. Dans chacune de ces formulations, on associe à chaque arête $e \in E$ une variable binaire x_e (qui prend la valeur 1 si l'arête appartient à la solution, 0 sinon), on obtient donc le programme :

$$
\begin{aligned}
&\min \sum_{e \in E} c_e \cdot x_e \\
&s.t. \sum_{e \in E(S)} x_e \leq |S| - 1 && \forall S \subseteq V, 2 \leq |S| \leq |V| - 1 \\
&\sum_{e \in E} x_e = |V| - 1 \\
&\sum_{e \in E} w_e \cdot x_e \leq B \\
&x_e \in \{0, 1\} && \forall e \in E
\end{aligned}
$$

Bien que la relaxation de cette formulation soit très proche de la solution optimale, la résolution de ce programme linéaire en nombres entiers peut prendre beaucoup de temps. Nous avons donc cherché des formulations étendues ainsi que des inégalités valides afin de renforcer cette formulation et ainsi améliorer les temps de résolution.
Notre formulation étendue est basée sur un cas particulier de problème de l'arbre couvrant budgeté dans lequel la contrainte de sac à dos peut être associée à un matroïde, et le théorème sur l'intersection de deux matroïdes d'Edmonds ; la formulation ci-dessus est entière et les points extrêmes correspondent aux bases situées à l'intersection des deux matroïdes. Dans le problème de l'arbre couvrant budgeté, on retrouve le matroïde graphique dont les bases correspondent à l'ensemble des arbres couvrant du graphe, et donc dès lors que la contrainte de budget correspond également à un matroïde, on connait un polyèdre entier pour notre problème budgeté. En partant de ce théorème ainsi que des travaux de Balas sur la programmation disjonctive [2], nous avons donné une formulation linéaire qui décrit parfaitement le problème budgeté, mais de taille potentiellement exponentielle. Pour cela, nous proposons une décomposition de la partie sac à dos jusqu'à obtenir des descriptions que l'on peut associer à des matroïdes. Bien que très intéressante théoriquement, une formulation aussi immense ne peut être utilisée directement en pratique, nous avons donc étudié la projection de cette formulation afin d'obtenir des inégalités valides pour le problème, ainsi que des relaxations de cette formulation afin de l'incorporer dans des procédures de séparations efficaces. Nous avons identifier plusieurs familles d'inégalités valides situées à l'intersection du problème de l'arbre couvrant et du sac à dos. Certaines vont venir enrichir la formulation initiale alors que d'autres seront très efficaces dans des procédures de séparation pour couper les solutions fractionnaires. Plusieurs de ces familles d'inégalités sont générées via dualité, en utilisant le lemme de Farkas, elles peuvent donc avoir des patterns ou des coefficients arbitraires, spécifique à l'état du primal. Plusieurs tests ont été réalisés, avec des formulations compactes et de taille exponentielle afin de tester nos inégalités valides et de résoudre le problème le plus vite possible.

3 Conclusions

L'étude de cette formulation étendue entière nous a permis d'identifier différents patterns d'inégalités valides combinant la contrainte du sac à dos ainsi que la structure de l'arbre couvrant. Ces contraintes ont été intégrées dans des algorithmes de branchements afin de tester leurs efficacités dans la résolution du problème de l'arbre couvrant budgeté.

Références

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

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

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

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

How to cut efficiently in bi-objective Branch&Cut algorithm ?

Pierre Fouilhoux, Lucas Létocart, Yue Zhang

LIPN, Université Sorbonne Paris Nord, CNRS UMR 7030, Villetaneuse, France
{pierre.fouilhoux, lucas.letocart, yue.zhang}@lipn.univ-paris13.fr

Keywords : multi-objective integer programming, Branch&Cut, multi-point separation, polyhedral approach.

1 Introduction to multi-objective combinatorial optimization

Many real-world applications are characterized by multiple, usually conflicting objectives and there is often no ideal solution optimizing every objective simultaneously. Consequently, in the multi-objective context, we are interested in *Pareto optimal* [1] also called *efficient* solutions that cannot be improved in any objective without losing advantages in at least one of other objectives. Without loss of generality, consider the following multi-objective integer program (MOIP)

$$\begin{aligned} \min_{x} \quad & z(x) = (z_1(x), \dots, z_p(x)) \\ \text{s.t.} \quad & x \in \mathcal{X} = \{x \in \mathbb{N}^n : Ax \leq b\} \end{aligned}$$

where $p \geq 2$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Denote $\mathcal{Y} = z(\mathcal{X})$ the image of the set of feasible solutions, $\tilde{\mathcal{Y}} = z(\tilde{\mathcal{X}})$ the image of polyhedron $\tilde{\mathcal{X}} = \{x \in \mathbb{R}^n : Ax \leq b\}$ with relaxed integrity, $\mathcal{Y}_N = z(\mathcal{X}_E)$ the *non-dominated set* that is the image of efficient solutions. Our goal is to enumerate the complete non-dominated set $\mathcal{Y}_N$, in order to let deciders choose their preferred solutions among these (potentially exponentially many) efficient solutions [2].

In the literature, exact approaches for multi-objective optimization problems are mainly divided in two categories : the criterion space search scheme and decision space search scheme. Under the criterion space scheme, algorithms explore the objective space and iteratively solve a (mono-objective) IP problem, such as the dichotomy method [3], the ϵ-constraint algorithm [4] and the boxed line method [5]. However this approach might be highly time consuming, since the at each step optimizing an IP problem is generally $\mathcal{NP}$-hard. As to the decision space exploration, the only known framework is the multi-objective Branch&Bound (MOB&B) algorithm [6]. Such a fully implicit enumeration ensures a complete outcome $\mathcal{Y}_N$, iteratively focuses on a easier relaxed subproblem, and can be accelerated by strengthening on the relaxation bound.

In this work, we propose a first generic bi-objective Branch&Cut (BOB&C) algorithm with several alternative ways of adding valid inequalities including multi-point cutting plane algorithm and other improvements.

2 Bi-objective Branch&Bound algorithm

In mono-objective case, the B&B tree search stops when the gap reaches 0 i.e. the lower bound value l equals the upper bound value u. However, the two bounds $l, u \in \mathbb{R}^p$ bounding every non-dominated point (i.e. $l \leqq y \leqq u$, $\forall y \in \mathcal{Y}_N$) are located far away in objective space and never achieve the equality. In multi-objective context, we consider the following definition of bound sets.

Definition 2.1 (Bound sets [7]). Let $\overline{\mathcal{Y}} \subset \mathcal{Y}_N$. A *lower bound set* (LBS) L for $\overline{\mathcal{Y}}$ is an $\mathbb{R}^p_{\geqq}$-closed and $\mathbb{R}^p_{\geqq}$-bounded set $L \subset \mathbb{R}^p$ such that $\overline{\mathcal{Y}} \subset L + \mathbb{R}^p_{\geqq}$ and $L \subset (L + \mathbb{R}^p_{\geqq})_N$. An *upper bound set* (UBS) U for $\overline{\mathcal{Y}}$ is an $\mathbb{R}^p_{\geqq}$-closed and $\mathbb{R}^p_{\geqq}$-bounded set $U \subset \mathbb{R}^p$ such that $\overline{\mathcal{Y}} \subset \mathrm{cl}[(U + \mathbb{R}^p_{\geqq})^c]$ and $U \subset (U + \mathbb{R}^p_{\geqq})_N$.

The bi-objective B&B algorithm [6, 8] is a natural extension of the B&B algorithm for (mixed) integer problems. Each subproblem is evaluated by the two bound sets in Definition 2.1.

The lower bound set is constructed from solving the relaxed subproblem (such as convex relaxation, surrogate relaxation), in this work, we consider the simply BOLP relaxation. First we enumerate every non-dominated extreme points on the relaxed polyhedron $\tilde{\mathcal{Y}}$ in a dichotomic search [3], then the line segments of such extreme points yield exactly the non-dominated boundaries of $\tilde{\mathcal{Y}}$, which is well a valid LBS.

A global upper bound set is built from scratch and is updated whenever a feasible solution is encountered. Subsequently, the current node can be pruned by

- infeasibility
- integrity : when current subproblem has an unique integer solution
- dominance : when the LBS is dominated by current UBS

Otherwise, we continue to branch on free variables or on objective space by Extended Pareto branching (EPB) [9, 10, 11] to split the current problem into distinct subproblems. For instance in Figure 1, the lower bound set $\tilde{\mathcal{Y}}_N$ drawn in green segments is not completely dominated by the upper bound set drawn in red dashed lines, the EPB divides the gray search area into three subproblems respectively bounding by the local nadir points μ_1, μ_2, μ_3. The advantage of EPB rule is that the more non-dominated area split by EPB, the more research area could be discarded.

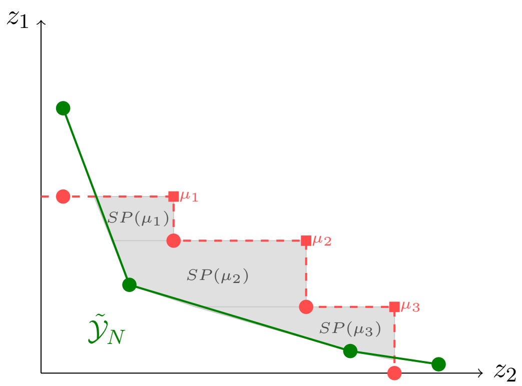


FIGURE 1 – An example of the Extended Pareto branching.

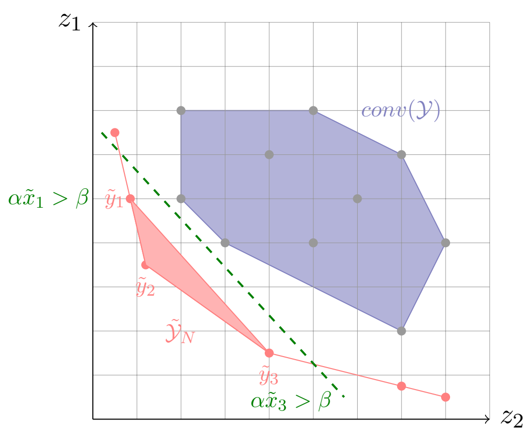


FIGURE 2 – The multi-point separation problem.

3 Bi-objective Branch&Cut algorithm

Multi-point cutting plane scheme To improve the tree search process in BOB&B algorithm, we would like to enhance the lower bound set by polyhedral approaches. Due to the exponential many of lower bounds, rather than applying valid inequalities individually on each lower bound, we prefer to provide the valid inequalities simultaneously separating several points from $conv(\mathcal{Y})$. In bi-objective space, the non-dominated points are lying in a natural order from the left-up to the right-bottom, such a multi-point separation problem is reduced to generate a valid inequality violated only by the two left-most and right-most points in the subset points. In Figure 2, we illustrate the non-dominated parts of valid inequality's image in criteria space separating the convex combination of $\{\tilde{y}_1, \tilde{y}_2, \tilde{y}_3\}$ (on the red LBS) from $conv(\mathcal{Y})$ polytope in blue.

To see the efficiency of our multi-point cutting plane algorithm, we implement an heuristic multi-point separator providing the cover inequality for each knapsack like constraint.

Cutting by commercial solver as black-box Various common valid cuts have been well integrated in the commercial solvers, our subsequent ambition is to make use of the powerful framework (like cut generations, heuristics, preprocessing etc) afforded by modern solvers. While constructing the LBS by solving a series of weighted-sum problems $P_\lambda\{\min \lambda^T z(x),\ x \in \mathcal{X}\}$, the idea is to let the solver treat with the integer problem and get the best fractional solution found so far during a fixed limit. Therefore, each lower bound is obtained independently with the strong help of the solver (different valid cuts, reductions etc). The non-dominated segments of the convex combination of lower bounds mentioned in the previous section is not valid anymore, as the set of lower bounds are not the extreme points on the same polyhedron. We proposed a new algorithm computing a valid LBS constructed as the convex intersection of straight lines respectively passing by lower bound and perpendicular to the parameter normal λ.

Since the LBS construction is the main time-consuming factor in our BOB&C algorithm, we propose the partial re-optimization, the intersection with predecessor's node LBS etc, to avoid the exhaustive computation of LBS.

4 Preliminary experimental results

Our BOB&C algorithm is implemented in JULIA [1] language and under vOPTSLOVER [2] project, which is an open-source software devoted to solving generic MOIP problems. Our experimentation is mainly tested on bi-dimensional knapsack instances collected in the the VOPTLIB [3] library, on set covering instances and on the multi-demand multi-dimensional knapsack instances [12] randomly generated, and run on the `Nvidia Quadro RTX 5000 GPU` accelerators with `16GB` of `RAM` and 40 `CPU` processors.

Our first preliminary experiments show the strong efficiency of our multi-point cutting plane algorithm that provides much more valid inequalities cutting several points at the same time than valid cuts violating single point, reduces up to 70% nodes of B&B tree without applying cuts, and significantly improves the computation time with EPB.

Thanks to the advanced commercial solver, the BOB&C algorithm invoking solver's cut generation as black-box (especially with EPB) outperforms among the different version BOB&B&C algorithms and trends to be competitive on large instances compared to ϵ-constraint algorithm.

1. `https://julialang.org`
2. `http://github.com/vOptSolver`
3. `https://github.com/vOptSolver/vOptLib`

Références

[1] Vilfredo Pareto. *Manuel d'économie politique*, volume 38. Giard & Brière, 1909.

[2] Matthias Ehrgott. Approximation algorithms for combinatorial multicriteria optimization problems. *International Transactions in Operational Research*, 7(1) :5–31, 2000.

[3] Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. A recursive algorithm for finding all nondominated extreme points in the outcome set of a multiobjective integer programme. *INFORMS Journal on Computing*, 22(3) :371–386, 2010.

[4] George Mavrotas. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2) :455–465, 2009.

[5] Natashia Boland, Hadi Charkhgard, and Martin Savelsbergh. A criterion space search algorithm for biobjective integer programming : The balanced box method. *INFORMS Journal on Computing*, 27(4) :735–754, 2015.

[6] Gülseren Kiziltan and Erkut Yucaoğlu. An algorithm for multiobjective zero-one linear programming. *Management Science*, 29(12) :1444–1453, 1983.

[7] Matthias Ehrgott and Xavier Gandibleux. Bound sets for biobjective combinatorial optimization problems. *Computers & Operations Research*, 34(9) :2674–2694, 2007.

[8] Anthony Przybylski and Xavier Gandibleux. Multi-objective branch and bound. *European Journal of Operational Research*, 260(3) :856–872, 2017.

[9] Thomas Stidsen, Kim Allan Andersen, and Bernd Dammann. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4) :1009–1032, 2014.

[10] Sune Lauth Gadegaard, Lars Relund Nielsen, and Matthias Ehrgott. Bi-objective branch-and-cut algorithms based on lp relaxation and bound sets. *INFORMS Journal on Computing*, 31(4) :790–804, 2019.

[11] Nicolas Forget, Sune Lauth Gadegaard, Kathrin Klamroth, Lars Relund Nielsen, and Anthony Przybylski. Branch-and-bound and objective branching with three or more objectives. *Computers & Operations Research*, 148 :106012, 2022.

[12] Paola Cappanera and Marco Trubian. A local-search-based heuristic for the demand-constrained multidimensional knapsack problem. *INFORMS Journal on Computing*, 17(1) :82–98, 2005.

Mathematical Formulations for Consistent Travelling Salesman Problems

Juan José Salazar Gonzalez, Professeur

université de La Laguna,
Spania

Mots-clefs : TSP.

1 Abstract

The consistent travelling salesman problem looks for a minimum-cost set of Hamiltonian routes, one for every day of a given time period. When a customer requires service over several days, the service times on different days must differ by no more than a given threshold (for example, one hour). We analyze two variants of the problem, depending on whether the vehicle is allowed to wait or not at a customer location before its service starts. There are three mathematical models in the literature for the problem without waiting times, and this paper describes a new model appropriate to be solved with a branch-and-cut algorithm. The new model is a multicommodity flow formulation on which Benders' Decomposition helps manage a large number of flow variables. There were no mathematical models in the literature for the variant with waiting times, and this paper adapts the four mathematical models to it. We analyze the computational results of the formulations on instances from the literature with up to 100 customers and three days.

Sequential matroid-game against Greedy

D. Cornaz, E. Lancini and M. Lampis

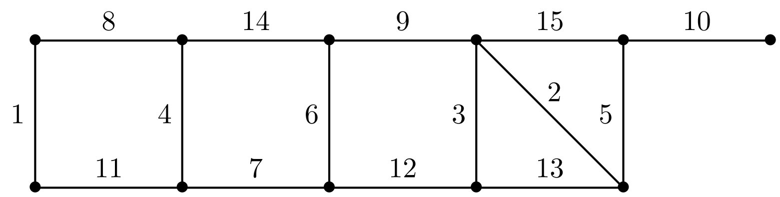
LAMSADE, Université Paris-Dauphine – PSL

Keywords: Sequential game, Transversal matroid

Let $E = \{1, \ldots, n\}$ be the edge-set of a loopless undirected graph G so that $\{1, \ldots, 2r\} \subseteq E$ is an inclusion-wise maximal forest of G. The players pick alternatively edges by respecting the rule of the game, which is that the union of the picked edges has no circuit, that is, at each round, $A \cup B$ must be a forest of the graph, if A denotes the set of edges picked by one player, and B those picked by the other player.

One of the player, denoted GREEDY is totally predictable, and in fact, the ordering $E = \{1, \ldots, n\}$ is also its preference order, and in fact, its strategy. The other, say PLAYER, will try to optimize his set of edges.

For instance, let $A := \emptyset$, $B := \emptyset$, and $E = \{1, \ldots, 15\}$ be the set of elements of the following graph:



Observe that $\{1, \ldots, 10\}$ is a spanning tree of G. Let w be the following vector:

$e =$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$w_e =$	4	3	1	1	1	1	1	1	4	3	1	1	2	5	3

So the game will run as follows: While it is possible, PLAYER chooses an edge $a \in E \setminus (A \cup B)$ so that $A \cup \{a\} \cup B$ is a forest and reset $A := A \cup \{a\}$, and then, GREEDY picks, automatically, the edge b wich is the minimum (with respect to the natural order) over $b \in E \setminus (A \cup B)$ so that $A \cup B \cup \{b\}$ is a forest, and reset $B := B \cup \{b\}$.

PLAYER always plays the first, and, for our example, a possible scenario for the game is as follow :

Round	1	2	3	4	5
PLAYER picks	1	13	15	14	9
GREEDY picks	2	4	6	8	10

Hence, on the given graph with the given weight vector, an optimal strategy for PLAYER ensures him to construct against GREEDY a set A with weight at least $w(A) \geq 4 + 2 + 3 + 5 + 4 = 16$.

This game can be generalized to matroids. We proved that it is NP-hard to find an optimal strategy for graphs, but, using Ghouila-Houri's characterization of totally unimodular matrices, polynomial for laminar matroids.

References

[1] Laurent Gourvès and Jérôme Monnot: *Private communication*, 2015

[2] David A. Kohler and R. Chandrasekaran. A class of sequential games. *Operations Research*, 19(2):270–277, 1971.

[3] Alexander Schrijver: *Combinatorial Optimization*. Springer-Verlag Berlin Heidelberg (2003)

Optimization methods for the multi-commodity flow blocker problem

I. Bentoumi[1], F. Furini[2], A. R. Mahjoub[1] and S. Martin[3]

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

2. Department of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome Via Ariosto 25 00185 Roma Italy
fabio.furini@uniroma1.it
3. Huawei Technologies France.
19 Quai du Point du Jour, 92100 Boulogne-Billancourt, France
sebastien.martin@huawei.com

Keywords : Multi-commodity flow problem, blocker problem, interdiction problem, branch-and-cut, polyhedral analysis.

1 Introduction

We are interested in evaluating the strength of a network by determining the maximum number of failures that it can face. This can be done by solving a multi-commodity flow blocker problem.
Let $G = (V, A)$ be a directed graph where V is the set of vertices and A the set of arcs. Every arc $a \in A$ is associated with a capacity $c_a \in \mathbb{Z}_+$ and a flow cost $p_a \in \mathbb{Z}_+$. Every commodity $k \in K$ is defined as a triplet (s_k, t_k, d_k) where $s_k \in V$ is the source, $t_k \in V$ is the destination and $d_k \in \mathbb{Z}_+$ the bandwidth, i.e., the amount of flow that should pass from s_k to t_k. The multi-commodity flow problem (MCFP) consists in finding a set of arcs in G, called multicommodity flow, routing flow to satisfy all commodities K, with respect to flow conservation constraints and capacity constraints. The MCFP aims to minimize the total flow cost. We study the blocker problem applied to the multi-commodity flow problem, which is called the multi-commodity flow blocker problem (MCFBP). Given an interdiction cost $r_a \in \mathbb{Z}_+$ associated with every arc $a \in A$, the MCFBP consists in finding a set of arcs, with a minimum total interdiction cost, to remove from the graph in such a way that the minimum cost of the multi-commodity flow remaining in the graph is greater than or equal to a given threshold. The threshold is called *target cost value* and it is denoted by Φ. To the best of our knowledge, this problem has not been addressed in the literature. However, many studies have been conducted on a closely related problem known as the multicommodity flow interdiction problem (MCFIP). We refer the interested reader to [2] for further explanations regarding this problem.

We illustrate the features of optimal MCFBP solutions thanks to the graph shown in Figure 1 composed by 8 vertices, 12 arcs and 2 commodities $k_1 = (s_1, t_1, d_1 = 8)$, $k_2 = (s_2, t_2, d_2 = 10)$. We report on each arc two values separated by the symbol " ;" : the first one, in blue, is the capacity of the arc ; the second one, in black, is the cost for sending one unit of flow through the arc. The optimal MCFP solution in this network consists in sending 8 units of flow from s_1 to t_1 through path $\{(s_1, v_1), (v_1, v_3), (v_3, t_1)\}$ and 10 units of flow from s_2 to t_2 through path $\{(s_2, v_2), (v_2, v_4), (v_4, t_2)\}$ with a total cost equal to 268. We assume that all arcs have the same value of interdiction costs, equal to 1, and consider a target cost value equal to 300. An optimal solution of the MCFBP consists in removing the arc (s_2, v_2), represented with a dashed line. The minimum cost of the multi-commodity flow remaining in the graph is equal to 418, which is greater than 300.

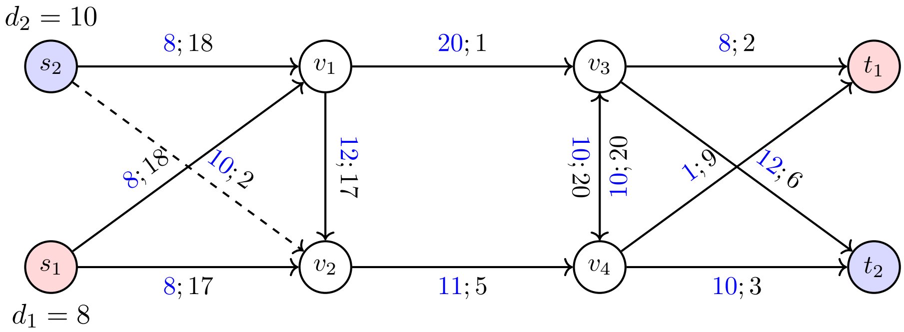


FIGURE 1 – Example of an optimal solution for the MCFBP with a target cost value $\Phi = 300$.

2 Solution approach

We introduce a new IP model for the multi-commodity flow blocker problem. Our goal is to provide a thin and well-understood formulation. This formulation has an exponential number of constraints called cover constraints. It is based on the approach developed in [1] to solve the most vital vertices for the shortest s-t path problem. Let z_a be a binary variable associated with the set of arcs A of a graph G, each variable encoding whether the corresponding arc is removed from G or not. Model (1) solves a multi-commodity flow blocker problem.

$$\min \quad \sum_{a\in A} r_a w_a \tag{1a}$$

$$\sum_{a\in mcf} w_a \geq 1 \qquad mcf \in MCF, \tag{1b}$$

$$w_a \in \{0,1\} \qquad a \in A, \tag{1c}$$

where MCF is the set of all multi-commodity flows in G with a total cost less than the target cost value Φ.

We examine the polyhedral structure of Formulation (1) and give new valid inequalities to strengthen the model. Using this, we develop a Branch-and-cut algorithm to solve Model (1).

References

[1] Magnouche, Y. and S. Martin (2020). Most vital vertices for the shortest s−t path problem : complexity and branch−and−cut algorithm. Optimization Letters 14, 2039–2053.

[2] Lim, C. and J. C. Smith (2007). Algorithms for discrete and continuous multicommodity flow network interdiction problems. IIE Transactions 39 (1), 15–26.

Triangulation de Hilbert unimodulaire des cones simples totalements équimodulaires

R. Grappe[1] et M. Vallée[2]

1. Laboratoire d'Informatique de Paris Nord
Université Sorbonne Paris Nord, CNRS UMR 7030, F-93430
Villetaneuse, France
grappe@lipn.fr

2. Laboratoire d'Informatique de Paris Nord
Université Sorbonne Paris Nord, CNRS UMR 7030, F-93430
Villetaneuse, France
vallee@lipn.fr

Mots clefs : Base de Hilbert, cone, triangulation, totale équimodularité.

Nous nous intéressons aux cones simples générés par les familles de vecteurs totalements équimodulaires. Nous explicitons pour chacun d'eux leur base de Hilbert, en fonction de leurs générateurs, grâce à la structure des matrices totalements équimodulaires de plein rang dans leurs lignes. Nous montrons que tout tel cône C admet une triangulation en sous-cones unimodulaires générés chacun par une partie de la base de Hilbert de C.

Perspective Formulations for piecewise convex functions : a theoretical and computational comparison

Claudia D'Ambrosio, Directrice de Recherche

CNRS,
école polytechnique, France
dambrosio@lix.polytechnique.fr

Mots-clefs : mixed integer nonlinear programs

1 Abstract

In this talk, we focus on mixed integer nonlinear problems with piecewise convex inequalities. This class of mixed integer nonlinear programs (MINLP) arise as subproblems solved at each iteration by the Sequential Convex MINLP method. We generalize the classic formulation for piecewise linear functions to the case of piecewise convex ones and strengthen them through perspective reformulation. Moreover, we compare the different formulations and we show that they are not equivalent, contrarily to what has been shown in the literature for the piecewise linear formulations.

A new hybrid method for unconstrained quadratic programming combined with the techniques of semidefinite programming and branch and bound method

Battikh Rabih [1], Alabboud Hassan [2], Jida Bassem [3] and Yassine Adnan [1,4]

1. Normandie Univ., Univ.Le Havre, LMAH, FR CNRS 3335, 76600 Le Havre, France
r_ battikh@hotmail.com

2. Lebanese University, Laboratory of Informatics and Applications LIA, Doctoral School in Science and Technology DSST, Mitein street Tripoli, Lebanon
3. Université Toulouse II, 5 Allée Antonio Machado, 31058 Toulouse CEDEX 9, France
4. Normandie Univ., Univ.Le Havre, ISEL, 76600 Le Havre, France

keywords : Binary unconstrained Quadratic Programming, metaheuristic algorithms, Branch and Bound method, hybrid algorithm, semidefinite programming, fixing criteria.

Abstract

In this work we propose to solve the unconstrained quadratic problem (UQP) using a hybrid and exact methods, this problem consists in minimizing a quadratic function a binary variables (0-1 variables). This combinatorial optimization problem is known to be NP-hard, There are many applications that lead to (UQP): machine learning [14, 15], transportation [6], Financial Services [7], Asset Exchange Problems [8], Clustering Problems [13]. Moreover, many combinatorial optimization problems pertaining to graphs such as determing maximum cliques, maximum cut, minimum coverings, maximum vertex packing, and maximum independent sets are known to be capable of being formulated by the (UQP) problem. Several heuristic approaches have been proposed for solving (UQP). We propose in this work a new hybrid algorithm (HA) based on some procedures. Our procedures are very efficient and fast, but unfortunately sometimes they are stuck in a local minimum, to overcome this handicap, we combined them with a simulated annealing algorithm[5, 9, 10]. To get rid for like cases, after that we repeated these procedures several times to obtaining the best solution by our hybrid algorithm (HA). Then we integrate this hybrid method with a semidefinite relaxation of (UQP) problem [2, 3, 4] in a branch and bound strategy [11, 12] to speed up and facilitate obtaining an exact optimal solution of reasonable sizes problems. In order to solve more easily the (UQP) problem we propose to use the fixing criteria, where through these criteria we can decrease the dimension of the (UQP) problem when it is possible and sometimes we can solve this problem completely by using a repeat loop based on these criteria [1].
Numerical results are presented to consolidate the demonstrated theoretical results and prove effectiveness and performance in speed and quality of our new approach.

References

[1] Hassan Alabboud. La programmation semi-définie combinée et comparée avec d'autres problèmes d'optimisation. Thèse de doctorat. Université de Havre, 2007.

[2] Steven J. Benson and Yinyu Ye. DSDP5: Software For Semidefinite Programming. Technical Report ANL/MCS-P1289-0905, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 2005. Submitted to ACM Transactions on Mathematical Software.

[3] Steven J. Benson and Yinyu Ye. DSDP5 User Guide — Software for Semidefinite Programming. Technical Report ANL/MCS-TM-277, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 2005. http://www.mcs.anl.gov/ benson/dsdp.

[4] Steven J. Benson, Yinyu Ye, and Xiong Zhang. Solving Large-Scale Sparse Semidefinite Programs for Combinatorial Optimization. SIAM Journal on Optimization, 10(2): 443—461, 2000.

[5] Vladimír Černỳ. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of optimization theory and applications, 45: 41—51, 1985.

[6] James Clark, Tristan West, Joseph Zammit, Xiaohu Guo, Luke Mason, and Duncan Russell. Towards real time multi-robot routing using quantum computing technologies. In Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, 111—119, 2019.

[7] Jeffrey Cohen, Alex Khan, and Clark Alexander. Portfolio optimization of 60 stocks using classical and quantum algorithms. arXiv preprint arXiv: 2008.08669, 2020.

[8] Fred Glover, Gary Kochenberger, Moses Ma, and Yu Du. Quantum Bridge Analytics II: Network Optimization and Combinatorial Chaining for Asset Exchange. arXiv preprint arXiv: 1911.03036, 2019.

[9] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. science, 220(4598): 671—680, 1983.

[10] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6): 1087—1092, 1953.

[11] LG Mitten. Branch-and-bound methods: General formulation and properties. Operations Research, 18(1): 24—34, 1970.

[12] David R Morrison, Sheldon H Jacobson, Jason J Sauppe, and Edward C Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optimization, 19: 79—102, 2016.

[13] Christian FA Negre, Hayato Ushijima-Mwesigwa, and Susan M Mniszewski. Detecting multiple communities using quantum annealing on the D-Wave system. Plos one, 15(2): e0227538, 2020.

[14] Jennifer Sleeman, John Dorband, and Milton Halem. A hybrid quantum enabled RBM advantage: convolutional autoencoders for quantum image compression and generative learning. In Quantum information science, sensing, and computation XII, 11391: 23—38. SPIE, 2020.

[15] Dennis Willsch, Madita Willsch, Hans De Raedt, and Kristel Michielsen. Support vector machines on the D-Wave quantum annealer. Computer physics communications, 248: 107006, 2020.

Algorithme de Branch & Cut exploitant les symétries du polytope du sac-à-dos matriciel symétrique en poids

A. Heintzmann[1,2], P. Bendotti[1] et C. Rottner[1]

1. EDF Lab Paris-Saclay, 7 Bd. Gaspard Monge, 91120 Palaiseau, France
{alexandre.heintzmann,pascale.bendotti,cecile.rottner}@edf.fr

2. LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
alexandre.heintzmann@laas.fr

Mots-clefs : étude polyédrale, branch & cut, symétries.

1 Introduction

En tant que cœur combinatoire de nombreux problèmes, le problème de sac-à-dos (KP) et ses variantes ont été largement étudiées dans la littérature [**hojny2020knapsack**]. Nous considérons la variante suivante introduite dans [**heintzmann2022polyhedral**]. Soit M et N des entiers positifs. Soit un KP avec N groupes de M objets. On note (i, j) l'objet j du groupe i. Dans chaque groupe, les contraintes d'ordre sont telles que (i, j) peut être sélectionné uniquement si $(i, j - 1)$ est sélectionné. L'objet (i, j) a une valeur v_{ij} et un poids w_j, ce qui implique que (i, j) et (i', j) ont le même poids, d'où la symétrie en poids du sac-à-dos. Le problème de sac-à-dos matriciel symétrique en poids (SMKP) consiste à maximiser la valeur totale des objets sélectionnés, dont le poids ne peut pas dépasser la capacité C du sac-à-dos. L'intérêt de cette variante est que le SMKP est le cœur combinatoire du Hydro Unit Commitment, le problème de gestion de production des centrales hydroélectriques.

Soit x_{ij} une variable binaire telle que $x_{ij} = 1$ si (i, j) est sélectionné, $x_{ij} = 0$ sinon. Une formulation du SMKP est comme suit.

$$
\begin{aligned}
\max \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} v_{ij} & \\
s.c. \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} w_j \leq C & \\
x_{ij} \leq x_{ij-1} & \qquad \forall i, \forall j > 1 \\
x_{ij} \in \{0, 1\} & \qquad \forall i, \forall j
\end{aligned}
$$

On observe que le SMKP présente une symétrie de faisabilité par rapport au groupes. En effet, au sein de chaque groupe les contraintes d'ordre sont les mêmes, les poids sont identiques d'un groupe à l'autre et la contrainte de capacité s'applique à tous les groupes à la fois. Cependant, les valeurs v_{ij} des objets ne sont pas symétriques, il n'y a donc pas de symétrie pour la valeur des solutions.

Les contraintes d'ordre étant un cas particulier des contraintes de précédence, le SMKP est par définition un cas particulier du KP avec contraintes de précédence [**boyd1993**]. Il est aussi possible de formuler le SMKP avec des contraintes disjonctives plutôt qu'avec des contraintes d'ordre, faisant du SMKP un cas particulier du KP avec contraintes disjonctives [**salem2018optimization**]. De plus, la formulation du SMKP avec des contraintes d'ordre et celle avec des contraintes de disjonction mènent à la même relaxation linéaire [**croxton2003comparison**]. Le SMKP est aussi une généralisation du KP avec variables entières non-bornées [**pochet1995integer**]. Le KP avec variables entières étant NP-difficile, le SMKP est aussi NP-difficile. Malgré la présence de plusieurs groupes d'objets, le SMKP n'est pas un KP multiple [**ferreira1996solving**]. En effet, le SMKP n'a qu'une seule contrainte de capacité s'appliquant à tous les groupes, et non une contrainte par groupe.

En Section 2, nous rappelons les résultats présentés dans [**heintzmann2022polyhedral**]. En Section 3, nous décrivons de nouveaux algorithmes permettant de séparer efficacement les inégalités introduites.

2 Inégalités de pattern

Nous considérons des instances du SMKP pour lesquelles le polyèdre est de pleine dimension. Il est montré dans [**heintzmann2022polyhedral**] que toute instance du SMKP peut être modifiée pour que le polyèdre associé soit de pleine dimension, sans changer l'ensemble des solutions réalisables.

A partir de l'enveloppe convexe du SMKP, on distingue trois types d'inégalités : les inégalités de la formulation (borne, ordre,...), des inégalités avec des coefficients 0-1 dites binaires, et des inégalités avec des coefficients entiers positifs dites entières. Certaines inégalités de la formulation définissent des facettes. Dans ces travaux, nous nous consacrons aux conditions nécessaires et suffisantes pour que les inégalités binaires définissent des facettes. En effet, on peut voir sur des exemples que les familles d'inégalités du sac-à-dos [**balas1975facets**] et du sac-à-dos avec contraintes de précédence [**boyd1993**] ne contiennent pas les inégalités binaires du SMKP.

En raison de la nature symétrique du SMKP, lorsqu'une inégalité est facette, n'importe quelle permutation des indices des groupes mène à une autre facette. Par conséquent, pour chaque facette du SMKP, il existe jusqu'à un nombre exponentiel de facettes symétriques. Pour réduire le nombre d'inégalités qu'on manipule, nous définissons les patterns.

Définition 1 (Pattern $\mathcal{P}$)**.** *Un pattern est un ensemble de N ensembles $S_i \subseteq \{1, \ldots, M\}$, $i \leq N$. Chaque ensemble S_i contient les indices j des objets d'un même groupe i.*

Comme les ensembles ne sont pas ordonnés, un pattern peut représenter toutes les permutations de groupes.

Définition 2 (Ensemble de variables $\mathcal{X}$ associé à $\mathcal{P}$)**.** *Un ensemble de variables $\mathcal{X}$ est associé au pattern $\mathcal{P}$ s'il existe une permutation π de $\{1, \ldots, N\}$ telle que $x_{ij} \in \mathcal{X} \Leftrightarrow j \in S_{\pi(i)} \in \mathcal{P}$.*

On note $\mathbb{X}(\mathcal{P})$ l'ensemble de tous les ensembles de variables $\mathcal{X}$ associé à $\mathcal{P}$.

Définition 3 ($rank(\mathcal{P})$)**.** *Le rang de $\mathcal{P}$, noté $rank(\mathcal{P})$, est une borne supérieure valide pour la somme des variables de n'importe quel ensemble $\mathcal{X}$ associé à $\mathcal{P}$.*

Définition 4 (Inégalités de pattern)**.** *Les inégalités d'un pattern $\mathcal{P}$ sont définies comme suit :*

$$\sum_{x_{ij} \in \mathcal{X}} x_{ij} \leq rank(\mathcal{P}) \qquad \forall \mathcal{X} \in \mathbb{X}(\mathcal{P})$$

Définition 5 (k-intersection)**.** *Soient $\mathcal{X}$ et $\mathcal{Y}$ deux ensembles de variables. L'ensemble $\mathcal{Y}$ est une k-intersection de $\mathcal{X}$ si $|\mathcal{Y} \cap \mathcal{X}| = k$; $\forall x_{ij} \in \mathcal{Y}$ si $x_{ij'} \in \mathcal{X}$ avec $j' \leq j$ alors $x_{ij'} \in \mathcal{Y}$; et l'inégalité suivante est vérifiée*

$$\sum_{i=1}^{N} \sum_{j=1}^{\max(j'|x_{ij'} \in \mathcal{Y})} w_j \leq C$$

Théorème 1. *Soit un pattern $\mathcal{P}$ de rang p et contenant un ensemble S_i de cardinalité au plus 1. Les conditions suivantes sont nécessaires et suffisantes pour que les inégalités de $\mathcal{P}$ définissent des facettes, avec $\mathcal{X} \in \mathcal{X}(\mathcal{P})$.*

$$|S_i| \geq 1 \qquad \forall S_i \in \mathcal{P} \qquad \text{(i)}$$

$$\exists \mathcal{Y} \subseteq \mathcal{V} \text{ } k\text{-intersection de } \mathcal{X} \text{ avec } x_{iM} \in \mathcal{Y} \qquad \forall i \leq N \qquad \text{(ii)}$$

$$\exists \mathcal{Y} \subseteq \mathcal{V} \text{ } k\text{-intersection de } \mathcal{X} \text{ avec } x_{ij-1} \in \mathcal{Y} \text{ et } x_{ij'} \notin \mathcal{Y}, \forall j' \geq j \qquad \forall x_{ij} \in \mathcal{X} \qquad \text{(iii)}$$

La condition (i) indique qu'aucun ensemble S_i de $\mathcal{P}$ ne peut être vide. La condition (ii) stipule que pour n'importe quel groupe i, il existe une solution réalisable vérifiant l'inégalité de pattern de $\mathcal{P}$ à l'égalité, avec $x_{iM} = 1$. La condition (iii) est similaire à (ii), mais avec $x_{i,j-1} = 1$ pour tout $x_{i,j} \in \mathcal{X}$ et $x_{i,j-1} \notin \mathcal{X}$. Pour un pattern $\mathcal{P}$, les conditions (i) (ii) et (iii) peuvent être vérifiées en temps polynomial par programmation dynamique.

Les conditions (i) (ii) et (iii) sont nécessaires et suffisantes dans le cas d'un pattern avec un ensemble de cardinalité au plus 1. De plus, ces conditions restent nécessaires pour un pattern quelconque, et permettent de garantir une dimension minimum pour leurs inégalités. Comme ces conditions peuvent aussi être efficacement implémentées, nous nous focalisons sur ces trois conditions pour développer des algorithmes dédiées dans le cadre d'un schéma de Branch & Cut. Pour la suite, on appelle pattern-flexible un pattern vérifiant ces trois conditions.

3 Algorithme de Branch & Cut en deux phases

Grâce à la définition de patterns et de conditions associées, il est possible de définir un schéma de Branch & Cut en deux phases pour gagner en efficacité pour résoudre le SMKP. La première phase consiste à générer des patterns-flexibles. La seconde phase consiste, à partir des patterns générés en pré-traitement, de séparer des inégalités de pattern au cours du Branch & Cut.

Génération de patterns-flexibles Pour la première phase, nous cherchons à générer des pattern-flexibles. Premièrement, on choisi un entier aléatoire p. On initialise un pattern $\mathcal{P}$ aléatoire contenant p indices, dont au moins l'indice M dans chaque groupe, ainsi $\mathcal{P}$ vérifie (i). Ensuite, tant qu'il existe $j \in S_i \in \mathcal{P}$ qui ne vérifie pas (ii), j est remplacé par $j - 1$. Similairement, s'il existe $S_i \in \mathcal{P}$ ne vérifiant pas (iii), on ajoute M à S_i. Le pattern $\mathcal{P}$ vérifie alors (i) (ii) et (iii). Si $\mathcal{P}$ est aussi de rang p, alors il est pattern-flexible, sinon il est écarté.

Ce processus de génération peut échouer à trouver un pattern-flexible. Cependant, ce processus ne fait intervenir que des algorithmes polynomiaux, car vérifier si la condition (ii) ou (iii) peut se faire en temps polynomial. L'idée est alors de répéter ce processus plusieurs fois pour générer des patterns. A noter aussi que par son aspect aléatoire, ce processus permet d'obtenir des patterns différents pour un même rang.

Séparation d'inégalités de pattern Pour la seconde phase, nous utilisons un algorithme de Branch & Cut, avec un algorithme de séparation s'appuyant sur les patterns générés en première phase. L'idée de la séparation est d'utiliser le point fractionnaire $\hat{x}$ afin d'obtenir, pour un pattern $\mathcal{P}$ généré, la permutation menant à l'inégalité la plus violée. Pour ce faire, on peut résoudre un problème de matching maximal (MMP) défini comme suit. Soit $\mathcal{H}$ un graphe biparti pondéré. Résoudre le MMP revient à trouver un ensemble A d'arcs, tel qu'au plus un arc de A est incident à un nœud de $\mathcal{H}$, maximisant la somme des poids de A. Pour appliquer le MMP au cas de la permutation de $\mathcal{P}$, on construit $\mathcal{H}$ avec d'un côté N nœuds correspondant aux N groupes du SMKP, et de l'autre côté N nœuds correspondant aux N ensembles du pattern. Entre chaque groupe i et chaque ensemble $S_{i'}$, il existe un arc de valeur $\sum_{j \in S_{i'}} \hat{x}_{i,j}$.

Proposition 1. *Résoudre le MMP permet d'obtenir la permutation maximisant le membre de gauche de l'inégalité du pattern $\mathcal{P}$.*

L'intérêt de résoudre un MMP est que ce dernier peut être résolu en temps polynomial par rapport au nombre de nœuds [**edmonds1972theoretical**]. Dans notre cas, $\mathcal{H}$ possède $2 \cdot N$ nœuds. Ainsi résoudre le MMP est en temps polynomial par rapport au nombre de groupes N, et est indépendant du nombre d'objets par groupe M. Cette séparation est répétée pour obtenir une permutation pour chaque pattern généré en première phase.

Résultats numériques Afin de mesurer l'efficacité de cet algorithme de Branch & Cut en deux phases, nous comparons trois configurations d'utilisation de CPLEX 12.8. La première est CPLEX par défaut ; la seconde est CPLEX sans coupes avec séparation des inégalités de patterns ; la troisième est CPLEX par défaut avec séparation des inégalités de patterns. Pour ce faire, un jeu d'instances variées a été généré avec 5 à 10 objets et 20 à 30 groupes. Parmi ces instances sont retenues que celles nécessitant au moins une minute pour être résolues par CPLEX sans coupes. Cette comparaison montre que l'algorithme de Branch & Cut en deux phases permet d'accélérer la résolution d' un grand nombre d'instances du SMKP. Sur 15 instances, 10 sont mieux résolues uniquement avec les inégalités de patterns. Dans ce cas le temps nécessaire de résolution peut être divisé par un facteur 10, et le nombre de nœuds développés par un facteur 10^3 par rapport à CPLEX par défaut. Une des instances est aussi résolue au nœud racine, alors qu'elle nécessite plus de 200 secondes pour être résolue par CPLEX par défaut. Parmi les 5 instances restantes, 3 sont mieux résolues par CPLEX par défaut avec séparation des inégalités de patterns, dont une résolue au nœud racine alors qu'elle n'est pas résolue en une heure par CPLEX par défaut. Seuls dans ces deux derniers cas, CPLEX par défaut est la configuration la plus performante.

On constate par ces résultats que notre algorithme en deux phases permet d'efficacement prendre en compte les symétries de faisabilité du SMKP, par le biais des patterns. Les perspectives seraient d'étendre ces travaux au cas pratique du 1-HUC, et ce schéma de Branch & Cut en deux phases à d'autres problèmes présentant des symétries de faisabilité.

The 4/3 Conjecture : Is it true or false ?

Sylvia Boyd, Professeure émérite

Université d'Ottawa,
Canada
sboyd@uottawa.ca

Mots-clefs : TSP

1 Abstract

The well-known 4/3 Conjecture states that the integrality ratio of the subtour linear programming relaxation for the metric Travelling Salesman Problem is at most 4/3. This conjecture has been around for almost 40 years and has been the focus of much research, yet it remains unresolved. In this talk we survey results that support the conjecture, as well as discuss properties that a possible counter example might possess.

Online Covering with Multiple Experts

K. T. Nguyen[1] et E. Kevi[2]

1. Laboratoire d'Informatique de Grenoble.
Bâtiment IMAG, 700 Av. Centrale, 38401 Saint-Martin-d'ères
kim.thang.nguyen@univ-grenoble-alpes.fr

2. Laboratoire d'Informatique de Grenoble.
Bâtiment IMAG, 700 Av. Centrale, 38401 Saint-Martin-d'Hères
eniko.kevi@univ-grenoble-alpes.fr

Mots-clefs : Online algorithms, learning, combinatorial optimization

1 Abstract

Designing online algorithms with machine learning predictions is a recent technique beyond the worst-case paradigm for various practically relevant online problems (scheduling, caching, clustering, ski rental, etc.). While most previous learning-augmented algorithm approaches focus on integrating the predictions of a single oracle, we study the design of online algorithms with multiple experts. We propose a new dynamic benchmark (linear combinations of predictions that change over time) to go beyond the popular benchmark of a static best expert in hindsight.

In the talk, I will present a connection between combinatorial optimization and machine learning that leads to a competitive algorithm in the new dynamic benchmark with a performance guarantee of O(log K), where K is the number of experts, for 0-1 online optimization problems. Furthermore, our multiple-expert approach provides a new perspective on how to combine in an online manner several online algorithms - a long-standing central subject in the online algorithm research community.

An efficient 2-competitive online algorithm for kit update at MSF Logistique

B. Detienne[1], M. Gaury[1] et G. Stauffer[2]

1. Université de Bordeaux,
351 cours de la Libération CS10004, 3405 Talence CEDEX, France
mickael.gaury@u-bordeaux.fr, boris.detienne@math.u-bordeaux.fr

2. Faculty of Business and Economics (HEC Lausanne)
Quartier de Chamberonne, CH-1015 Lausanne, Suisse
gautier.stauffer@unil.ch

Mots-clefs : 2-competitive online algorithm, kit update, greedy policy

1 Introduction

MSF Logistique (MSF-Log) is responsible for supplying medical and non-medical items to the field when needed by MSF or other partner NGOs, such as the Red Cross. Our focus here is on emergency response. As the first 72 hours after a disaster are crucial for an effective intervention [1], MSF has designed kits containing the necessary medical and non-medical items, which should be stored in appropriate quantities at MSF-Log and ready for shipment within 24 hours. The medical items in the kits should have a minimum shelf life of k months (typically 6 or 12 months). Due to the presence of medical products with expiration dates and the minimum shelf life requirement, the kits need regular updates to maintain their validity and readiness for shipment. This entails substituting medical components with new ones that have a longer shelf life. However, updating the kits is a time-consuming process. When a kit needs updating, it must be retrieved from the storage location, brought to the update area, opened, expired items disposed of, replaced with new ones (also sourced from the storage location), resealed, returned to its designated storage location, and accompanied by necessary paperwork. Additionally, expired items are usually destroyed (unless they can still be used for other operations), incurring costs.

This paper addresses the problem of updating a single kit “efficiently”. We consider two key performance indicators (KPIs) : updating time and destruction costs. These KPIs are combined into a single economic measure by converting updating time to cost, using the hourly labor cost of the person responsible for the update. It is important to note that there are set-up costs associated with bringing, opening, closing, and returning the kits to the storage location, as well as handling costs and destruction costs when an item is substituted. Since we do not know when the kit will be needed, we aim to minimize the total cost involved over a “long” time horizon, which serves as a proxy for the average per-period cost over an infinite horizon. A component in a kit must be updated at least k months before its expiration but can be updated earlier. We refer to the underlying problem as the kit-update problem, and a formal definition is provided in the next section. We consider both deterministic and adversarial versions of the problem.

Considering the set-up costs when opening a kit, it may be beneficial to substitute an item before the last moment to save on updating time. The optimal policy typically exploits this feature. These types of problems are known as opportunistic replacement problems in the literature. There are simple (suboptimal) policies to address such problems. For instance, one may choose to update a kit only when an item no longer meets the shelf life requirement, substituting only the item(s) in that situation. This policy is referred to as the age policy [2]. It is worth mentioning that the age policy was initially used at MSF-Log when initiating this project.

The opportunistic replacement problem (ORP) was first introduced by McCall et al. [3] and has been studied in various fields, including wind farms by Ding and Tian [4]. Some variants of the opportunistic replacement problem are known to be NP-hard [5]. Moreover, authors have identified connections between the opportunistic replacement problem and the joint replenishment problem [6], leading to the development of models and algorithms that exploit this similarity.

In the following sections, we provide a precise definition of the MSF-Log problem, provide an efficient 2-competitive online algorithm for the problem, and we illustrate the performances of the corresponding algorithm on a case study at MSF-Log.

2 Problem definition

We can capture the essence of the kit-update problem outlined earlier by separating the set-up costs, which are independent of the number of items being updated in the kit (including bringing the kit from the storage location, opening it, completing paperwork, etc.), from the additional costs associated with item substitution (such as destruction costs and handling costs). We formulate the problem as follows :

Problem. KIT-UPDATE

input : a set $\{0, 1, ..., T\}$ *of time periods,* m *different components numbered* $1, ..., m$*, (fixed) costs* K_i*, for* $i = 0, 1, ..., m$*, and, for each* $i \in [m] := \{1, ..., m\}$*, a set* I_i *of intervals, of the form* $[u, v]$ *for* $u \leq v \in \{0, 1, ..., T\}$*, such that* $\cup_{I \in I_i} I \cap \{0, 1, ..., T\} = \{0, 1, ..., T\}$*.*

output : a subset of (update) periods $S_i \subseteq \{1, \ldots, T\}$ *for each* $i = 1, \ldots, m$ *(we call this a schedule) that satisfy :*

($\mathcal{P}_i$) for each pair (u, v) *in* $S_i \cup \{0\} \times S_i \cup \{T+1\}$ *with* $u < v$ consecutive*, there exists an interval of* I_i *covering* $[u, v-1]$ *(*u *and* v *are said to be consecutive if* $u < v$ *and* $S_i \cap (u+1, ..., v-1) = \emptyset$*) and that minimise* $|\cup_{i=1}^{n} S_i|.K_0 + \sum_{i=1}^{n} |S_i|.K_i$ *(over such sets).*

Remark 1. *We assume that* $\cup_{I \in I_i} I \cap \{0, 1, ..., T\} = \{0, 1, ..., T\}$ *otherwise the problem is infeasible. The intervals in* I_i *represent batches of components* i *that become available at the beginning of period* l *and that expires at the end of period* u*.*

Remark 2. *Observe that we do not pay for the "update" in period* 0 *as the (w.l.o.g. unique) interval of* I_i *starting in 0 corresponds to the remaining shelf-life of the component* i *currently in the kit.*

The KIT-UPDATE problem is closely related to the JRP-D (Joint Replenishment Problem with Deadline). As a result, we can leverage the proofs presented by Nonner and Souza [7] which establish a polynomial-time reduction from vertex cover to KIT-UPDATE for JRP-D and derive the following conclusion.

Lemma 1. *KIT-UPDATE is NP-HARD and actually APX-HARD.*

3 A simple 2-competitive algorithm

We now sketch a very simple 2-approximation for the KIT-UPDATE problem and explain why it can be leverage to design a 2-competitive online algorithm.

We consider $m+1$ relaxations $P_0, P_1, ..., P_m$ of our original problem. Problems P_i for $i = 1, ..., m$ is the problem limited to component i only and where we pay fixed cost K_i only. Problem P_0 is the original problem but where we consider only the fixed cost K_0 (i.e. $K_1 = K_2 = ... = K_m = 0$). All problems $P_0, P_1, ..., P_m$ can be solved easily by postponing the updates as late as possible. More formally, this means that in problem $P_i, i = 1, ..., m$, we can find the optimal solution S_i iteratively as follows :

Set $t = 0$ and $S_i = \{\}$;
while $t < T+1$ **do**
  $t = \max\{t' > t : \exists I \in I_i \text{ s.t. } [t, t'-1] \subseteq I\}$;
  $S_i = S_i \cup \{t\}$;
end

In problem P_0, similarly, we can build the optimal solution S_0 (we update all product in each period of S_0 as it does not cost anything additional, that is $S_i = S_0$ for all $i = 1, ..., m$) by looking at the earliest expiration time over all components at each step. Hence we can set S_0 iteratively as follows :

Set $t = 0$ and $S_0 = \{\}$;
while $t < T+1$ **do**
  $t = \max\{t' > t : \exists I \in I_i \text{ s.t. } [t, t'-1] \subseteq I, \text{ for all } i = 1, ..., m\}$;
  $S_0 = S_0 \cup \{t\}$;
end

For a period t in $\{1, ...T\}$ we define $p(t)$ to be the max u in $S_0 \cup \{0\}$ with $u \leq t$ and $s(t)$ to be the min u in $S_0 \cup \{T+1\}$ with $u \geq t$. We can create a feasible solution to our original problem by considering, for all $i = 1, ...m$, $S'_i := \cup_{t \in S_i} \{p(t), s(t)\} \setminus \{T+1\}$ (note that by construction no $p(t)$ is equal to 0 for t in S_i). Indeed, consider two consecutive values $t < t'$ in $S'_i \cup \{T+1\}$. There exists an interval I of I_i covering $[t, t'-1]$ because either t is $p(t'')$ for some $t \leq t'' < t'$ in S_i and then $t' = s(t'')$ and I exists by the feasibility of S_0 ; or t is $s(t'')$ for some $t'' < t$ and then $t' = p(t''')$ for t''' consecutive to t'' in S_i (or $t' = T+1$) and I exists by the feasibility of S_i. The cost of the corresponding solution is by construction no more than twice the sum of the costs of the solution to the problems $P_0, P_1, ..., P_m$. It is not difficult to prove that the sum of the latter costs is a lower bound on the optimal value. This yields a simple 2-approximation.

The above algorithm only use information available at time t to take decisions regarding which component to update. It can thus be used as a 2-competitive online (polytime) algorithm for the online version of the problem where each interval $I = [u, v]$ of I_i is revealed at time u and the horizon T is adversarial too.

4 Case study and Perspective

The study was carried out between January 2020 and October 2021, focusing on a specific type of kit comprising 8 perishable items. During this period, a total of 65 kits were updated 227 times, using the age policy implemented by MSF-Log. To assess the costs involved, we obtained estimates for set-up, handling, and destruction costs through discussions and screenings with the operators and the head of supply chain management at MSF-Log.

We compared the performance of their policy with our 2-competitive algorithm and provide a summary of the results below. Additionally, we established a lower bound on the best possible policy by solving the problem retrospectively using a mixed-integer programming formulation similar to the one employed in [6]. This analysis not only demonstrates that our online algorithm achieves a savings of 33% in this particular case but also highlights that the performance of the online algorithm is significantly better than worst-case scenarios (deviating by less than 1.33% from the optimal solution in this instance).

	nb of product update	nb of kit update	total time	total cost
MSF policy	329	227	56h45	1611,50 €
online algorithm	353	112	28h	1090,80 €
"a posteriori" optimal policy	311	112	28h	1076,20 €

As part of our ongoing research, we are developing additional instances to evaluate the practical and worst-case performance of the online algorithm. These investigations will serve as the subject of our future research, allowing us to further analyze and understand the algorithm's effectiveness in various scenarios.

References

[1] Salmerón J, Apte A. Stochastic optimization for natural disaster asset prepositioning., Production and operations management, 2010

[2] Glasser GJ. The age replacement problem. Technometrics, 83-91, 1967

[3] Ab-Samat H, Kamaruddin S. Opportunistic maintenance (OM) as a new advancement in maintenance approaches : A review., Journal of Quality in Maintenance Engineering, 20(2), 98-121, 2014

[4] Ding F, Tian Z, Opportunistic maintenance for wind farms considering multi-level imperfect maintenance thresholds. Renewable Energy, 45, 175- 182, 2012

[5] Almgren T, Andréasson N, Patriksson M, Strömberg AB, Wojciechowski A, Önnheim M, The opportunistic replacement problem : theoretical analyses and numerical tests. Mathematical Methods of Operations Research, 76(3), 289-319, 2012

[6] Gustavsson E, Patriksson M, Strömberg AB, Wojciechowski A, Önnheim M. Preventive maintenance scheduling of multi-component systems with interval costs., Computers Industrial Engineering, 76, 390-400, 2014

[7] Nonner, T., Souza, A. (2009). Approximating the joint replenishment problem with deadlines. Discrete Mathematics, Algorithms and Applications, 1(02), 153-173.

Pickup and Delivery Problem with Cooperative Robots

Chi Thao Nguyen[1], Jean-Philippe Gayon[1], Viet Hung Nguyen[1], Alain Quilliot[1] et Anh Son Ta[2]

1. Laboratoire LIMOS, UMR CNRS 7024, INP et Université Clermont Auvergne, Campus Universitaire des Cézeaux.1 rue de la Chebarde.TSA 60125. CS 60026 63178 AUBIERE CEDEX -FRANCE
chi_ thao.nguyen@uca.fr,j-philippe.gayon@uca.fr, viet_ hung.nguyen@uca.fr, alain.quilliot@isima.fr

2. School Of Applied Mathematics and Informatics, Hanoi University of Science and Technology, 01 Dai Co Viet Street - HANOI -VIETNAM
taanhson123@gmail.com

Mots-clefs : Vehicle Synchronization ; Robots Cooperating ;Pickup and Delivery Problem ; Vehicle Routing Problem

1 Introduction

A new interest emerged among automated vehicles inside warehouse problems, that is cooperative robots. i.e. when a package is too large, it requires several robots to cooperate in carrying the package from its pick-up point to the destination. We call this the Pickup and Delivery Problem with Cooperative Robots (PDP-CR). The goal of PDP-CR is to minimize the total operation time consisting of total traveling and waiting time or the makespan. The cooperating between robots feature is what makes this problem original and interesting.
Some ideas can be taken from other types of problems like pick-up delivery problems and vehicle routing problems with synchronization to address this. The primary goal of this paper is to develop and test several mathematical models for the PDP-CR. The next section will describe the problem. In the third section, we will propose two types MILP formulations, one based on VRP and another based on the flow model. The numerical result and performance comparison between the two models is presented in section four.

2 Problem Definition

We have a fleet of I identical robots and J total tasks. Every task j is characterized by its pick-up location p_j, its destination d_j, its process time c_j and its number of robots requires n_j. The early robots that visit the task must wait for all the robots required to arrive, then they can start to process the task from its pick-up point to its destination.

There are two special cases of tasks : 1. *at-place task* has $p_j = d_j$. It may correspond to a manufacturing operation that requires n_j robots to be processed within time c_j ; and 2. *Traveling task*, this kind of task has $c_i = c_{p_i d_i}$. This may refer to a task that requires multiple robots to carry a package from p_i to d_i. Traveling task j is considered done only if there are n_j robots moving from p_j to d_j *simultaneously*, which means, after all the robots required for task j are at p_j, they start moving from p_j to d_j and reach d_j at the same time.

The problem instance can be solved on a graph $G = (V, E)$. V contains the location of the depot, pick-up location, and destination of all tasks. Node 0 represents the depot. All the robots

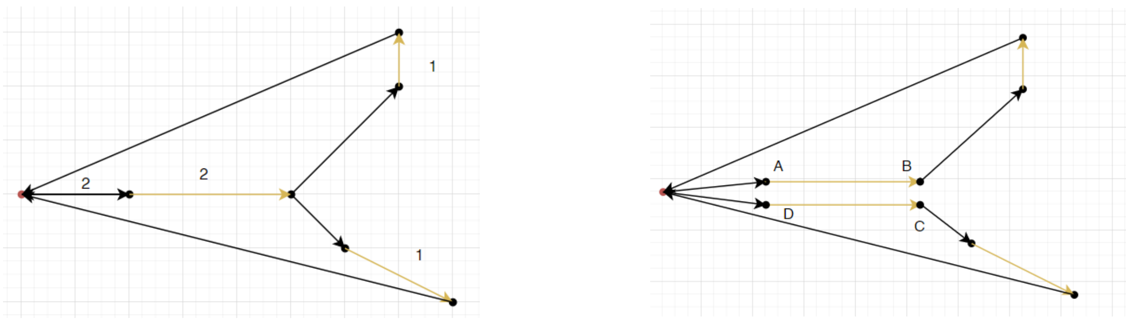


FIGURE 1 – Left : One sample solution on the original graph and Right : The same solution on the expanded graph, note that two yellow edges in the middle are at the same location as the yellow edge in the first figure (which denotes a traveling task that requires 2 robots). They are drawn separately for the simplicity of illustration.

start and end their path at the depot. c_{ij} is the time for any robot to move from task i to task j on the graph i.e. the time for any robot to go from d_i to p_j. The goal of the problem is to minimize the sum of costs, which is the total traveling time and waiting time of all robots ; or the makespan which is the time the last robot returns to the depot.

3 Formulations

In this section, we will propose 2 types of MIP formulation, one is based on Vehicle Routing Problem, another is based on flow formulation.

3.1 VRP Based Formulation

3.1.1 Expanded Graph

We can model the demand of n_j robots for task j by replacing it with n_j *expanded task*. Every expanded task requires exactly one robot, and its locations are at the same place as its original task. This means we replace pick up and destination location (p_j, d_j) with n_j pairs of *expanded* pick up - destination location $(p_j^1, d_j^1), (p_j^2, d_j^2), \dots (p_j^{n_j}, d_j^{n_j})$. The expanded pick-up nodes $p_j^1, p_j^2, \dots p_j^{n_j}$ are at the same location as p_j, and the expanded destination nodes are also at the same location as d_j.

n_j robots operating synchronously from one task's pick-up location to its destination is equivalent to there being n_j robots, each moving from one of the task's expanded pick-up node simultaneously, and they have to arrive at their corresponding destination node, also simultaneously.

We also introduce new sets :
- J' is the expanded tasks set.
- $J_j \in J'$: Set of all expanded tasks that its origin task is j. In other words, if expanded tasks i' and j' in J_j then their origin task is j.

The problem now can be solved on the expanded graph $G' = (V', E')$, with V' is the set of all the expanded pick-up and destination nodes.

An example can be seen in figure 1.
subsubsectionMILP Model With the Expanded graph being established, we will present a MILP model for the problem in this section.

Variables :
— $x_{ij} = 1$ if there is a robot going from expanded task i to expanded task j, 0 otherwise.
— $y_i \in \mathbb{R}$ is the waiting time of the robot at task i.
— $z_i \in \mathbb{R}$: The time robot start working on task i.
Model to minimize the sum of costs :

$$\begin{aligned}
\min \quad & \sum_{i,j \in J', i \neq j} c_{ij} x_{ij} + \sum_{i \in J'} y_i, & (1)\\
\text{s.t.} \quad & \sum_{j \in J'} x_{0j} = I, & (2)\\
& \sum_{i \in J'} x_{i0} = I, & (3)\\
& \sum_{i \in J', i \neq j} x_{ij} = 1 \qquad \forall j \in J', & (4)\\
& \sum_{j \in J', j \neq i} x_{ij} = 1 \qquad \forall i \in J', & (5)\\
& z_j + M(1 - x_{ij}) \geq z_i + c_i + c_{ij} + y_j \quad \forall i, j \in J', i \neq j & (6)\\
& z_j - M(1 - x_{ij}) \leq z_i + c_i + c_{ij} + y_j \quad \forall i, j \in J', i \neq j & (7)\\
& z_i = z_j \qquad \forall j' \in J, \forall i, j \in J_{j'}, & (8)\\
& x_{ij} \in \{0, 1\} \qquad \forall i, j \in J', i \neq j & (9)\\
& y_i \geq 0 \qquad \forall i \in J' & (10)\\
& z_i \geq 0 \qquad \forall i \in J' & (11)
\end{aligned}$$

Explanation :
— (1) We are trying to minimize the total traveling time and waiting time.
— (2), (3) make sure that there are M robots go in and go out the depot.
— (4), (5) : every task must be visited exactly once.
— (6), (7) are the linearization of the following constraint :

$$x_{ij} = 1 \implies z_j = z_i + c_{ij} + c_i + y_j \qquad \forall i, j \in J', i \neq j$$

The above constraint ensures that for each robot, if it moves from expanded task i to j then the time it reaches j must be equal to its start doing task i plus operating time of i plus traveling time between i and j plus its waiting time at j. M is a sufficiently large number.
— (8) is the constraint that makes sure the synchronization happens. If expanded task i and expanded task j have the same original task, then they must start at the same time.

Model to minimize the makespan :

$$\begin{aligned}
min \quad & C_{\max} & (12)\\
s.t. \quad & (2) - (5) & \\
& z_j + M(1 - x_{ij}) \geq z_i + c_i + c_{ij} \qquad \forall i, j \in J', i \neq j & (13)\\
& z_i \leq C_{\max} \qquad \forall i \in J' & (14)\\
& (8), (10), (11) &
\end{aligned}$$

In the makespan objective formulation, we don't need to know the waiting time of the robots for any task.

We will also introduce the flow-based model for the makespan variants in the paper.

4 Numerical Result

Using CPLEX as a solver, we implemented the MIP formulations and tested them on randomly generated data.

5 Conclusion

In this paper, the Pickup and Delivery with Cooperative Robots Problem is presented. We've also presented the MILP model for it. The numerical result is presented to approximate the maximum size that can be solved using OPL of both formulations as well as comparison the effectiveness between the two.

Références

[1] A. Cerqueus, *Bi-objective branch-and-cut algorithms applied to the binary knapsack problem.* PhD thesis, université de Nantes, 2015.

[2] P. Hansen, "Bicriterion path problems," in *Multiple criteria decision making theory and application*, pp. 109–127, Springer, 1980.

[3] A. M. Geoffrion, "Proper efficiency and the theory of vector maximization," *Journal of mathematical analysis and applications*, vol. 22, no. 3, pp. 618–630, 1968.

[4] X. Gandibleux, H. Morita, and N. Katoh, "The supported solutions used as a genetic information in a population heuristic," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 429–442, Springer, 2001.

[5] B. Villarreal and M. H. Karwan, "An interactive dynamic programming approach to multi-criteria discrete programming," *journal of mathematical analysis and applications*, vol. 81, no. 2, pp. 524–544, 1981.

[6] M. Ehrgott and X. Gandibleux, "Bound sets for biobjective combinatorial optimization problems," *Computers & Operations Research*, vol. 34, no. 9, pp. 2674–2694, 2007.

[7] A. Przybylski, X. Gandibleux, and M. Ehrgott, "A recursive algorithm for finding all non-dominated extreme points in the outcome set of a multiobjective integer programme," *INFORMS Journal on Computing*, vol. 22, no. 3, pp. 371–386, 2010.

[8] Y. P. Aneja and K. P. Nair, "Bicriteria transportation problem," *Management Science*, vol. 25, no. 1, pp. 73–78, 1979.

[9] Y. Haimes, "On a bicriterion formulation of the problems of integrated system identification and system optimization," *IEEE transactions on systems, man, and cybernetics*, vol. 1, no. 3, pp. 296–297, 1971.

[10] G. Mavrotas, "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems," *Applied mathematics and computation*, vol. 213, no. 2, pp. 455–465, 2009.

[11] G. Kiziltan and E. Yucaoğlu, "An algorithm for multiobjective zero-one linear programming," *Management Science*, vol. 29, no. 12, pp. 1444–1453, 1983.

[12] T. Vincent, *Caractérisation des solutions efficaces et algorithmes d'énumération exacts pour l'optimisation multiobjectif en variables mixtes binaires.* PhD thesis, Nantes, 2013.

[13] S. L. Gadegaard, M. Ehrgott, and L. R. Nielsen, "A cut-and-branch approach for a class of bi-objective combinatorial optimization problems," in *23rd International Conference on Multiple Criteria Decision Making*, vol. 43, 2015.

[14] E. L. Ulungu and J. Teghem, "The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems," *Foundations of computing and decision sciences*, vol. 20, no. 2, pp. 149–165, 1995.

[15] A. Przybylski and X. Gandibleux, "Multi-objective branch and bound," *European Journal of Operational Research*, vol. 260, no. 3, pp. 856–872, 2017.

[16] M. Ehrgott and X. Gandibleux, "Bounds and bound sets for biobjective combinatorial optimization problems," in *Multiple criteria decision making in the new millennium*, pp. 241–253, Springer, 2001.

[17] M. Ehrgott, *Multicriteria optimization*, vol. 491. Springer Science & Business Media, 2005.

[18] V. Pareto, *Manuel d'économie politique*, vol. 38. Giard & Brière, 1909.

[19] G. B. Dantzig and M. N. Thapa, *Linear programming 1 : introduction.* Springer Science & Business Media, 2006.

[20] L. G. Khachiyan, "A polynomial algorithm in linear programming," in *Doklady Akademii Nauk*, vol. 244, pp. 1093–1096, Russian Academy of Sciences, 1979.

[21] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302–311, 1984.

[22] R. V. Book, "Michael r. garey and david s. johnson, computers and intractability : A guide to the theory of *np*-completeness," *Bulletin (New Series) of the American Mathematical Society*, vol. 3, no. 2, pp. 898–904, 1980.

[23] M. Ehrgott, "Approximation algorithms for combinatorial multicriteria optimization problems," *International Transactions in Operational Research*, vol. 7, no. 1, pp. 5–31, 2000.

[24] J. G. Ecker and I. Kouada, "Finding all efficient extreme points for multiple objective linear programs," *Mathematical Programming*, vol. 14, no. 1, pp. 249–261, 1978.

[25] H. P. Benson, "An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem," *Journal of Global Optimization*, vol. 13, no. 1, pp. 1–24, 1998.

[26] M. W. Savelsbergh, "Preprocessing and probing techniques for mixed integer programming problems," *ORSA Journal on Computing*, vol. 6, no. 4, pp. 445–454, 1994.

[27] T. Achterberg, "Constraint integer programming (ph. d. thesis)," *TU Berlin*, 2007.

[28] S. S. Brito and H. G. Santos, "Preprocessing and cutting planes with conflict graphs," *Computers & Operations Research*, vol. 128, p. 105176, 2021.

[29] M. R. Garey and D. S. Johnson, *Computers and intractability*, vol. 174. freeman San Francisco, 1979.

[30] R. Borndörfer, *Aspects of set packing, partitioning, and covering.* PhD thesis, 1998.

[31] I. Méndez-Díaz and P. Zabala, "A cutting plane algorithm for graph coloring," *Discrete Applied Mathematics*, vol. 156, no. 2, pp. 159–179, 2008.

[32] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, vol. 2. Springer Science & Business Media, 2012.

[33] S. Rebennack, "Stable set problem : Branch & cut algorithms.," 2009.

[34] M. W. Padberg, "On the facial structure of set packing polyhedra," *Mathematical programming*, vol. 5, no. 1, pp. 199–215, 1973.

[35] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem," in *50 Years of Integer Programming 1958-2008*, pp. 77–103, Springer, 2010.

[36] V. Chvátal, "Edmonds polytopes and a hierarchy of combinatorial problems," *Discrete mathematics*, vol. 4, no. 4, pp. 305–337, 1973.

[37] H. Crowder, E. L. Johnson, and M. Padberg, "Solving large-scale zero-one linear programming problems," *Operations Research*, vol. 31, no. 5, pp. 803–834, 1983.

[38] H. P. Benson, "Existence of efficient solutions for vector maximization problems," *Journal of Optimization Theory and Applications*, vol. 26, no. 4, pp. 569–580, 1978.

[39] A. Przybylski, X. Gandibleux, and M. Ehrgott, "A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives," *Discrete Optimization*, vol. 7, no. 3, pp. 149–165, 2010.

[40] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," in *50 Years of Integer Programming 1958-2008*, pp. 105–132, Springer, 2010.

[41] R. M. Ramos, S. Alonso, J. Sicilia, and C. González, "The problem of the optimal biobjective spanning tree," *European Journal of Operational Research*, vol. 111, no. 3, pp. 617–628, 1998.

[42] F. Sourd and O. Spanjaard, "A multiobjective branch-and-bound framework : Application to the biobjective spanning tree problem," *INFORMS Journal on Computing*, vol. 20, no. 3, pp. 472–484, 2008.

[43] K. Florios, G. Mavrotas, and D. Diakoulaki, "Solving multiobjective, multiconstraint knapsack problems using mathematical programming and evolutionary algorithms," *European Journal of Operational Research*, vol. 203, no. 1, pp. 14–21, 2010.

[44] T. Vincent, F. Seipp, S. Ruzika, A. Przybylski, and X. Gandibleux, "Multiple objective branch and bound for mixed 0-1 linear programming : Corrections and improvements for the biobjective case," *Computers & Operations Research*, vol. 40, no. 1, pp. 498–509, 2013.

[45] G. Mavrotas and D. Diakoulaki, "A branch and bound algorithm for mixed zero-one multiple objective linear programming," *European Journal of Operational Research*, vol. 107, no. 3, pp. 530–541, 1998.

[46] G. Mavrotas and D. Diakoulaki, "Multi-criteria branch and bound : A vector maximization algorithm for mixed 0-1 multiple objective linear programming," *Applied mathematics and computation*, vol. 171, no. 1, pp. 53–71, 2005.

[47] P. Belotti, B. Soylu, and M. M. Wiecek, "A branch-and-bound algorithm for biobjective mixed-integer programs," *Optimization Online*, 2013.

[48] U.-E. Lukata and J. Teghem, "Solving multi-objective knapsack problem by a branch-and-bound procedure," in *Multicriteria analysis*, pp. 269–278, Springer, 1997.

[49] C. Bazgan, H. Hugot, and D. Vanderpooten, "Solving efficiently the 0–1 multi-objective knapsack problem," *Computers & Operations Research*, vol. 36, no. 1, pp. 260–279, 2009.

[50] J. Jorge, *Nouvelles propositions pour la résolution exacte du sac à dos multi-objectif unidimensionnel en variables binaires*. PhD thesis, Université de Nantes, 2010.

[51] T. Stidsen and K. A. Andersen, "Optimized parallel branch and cut algorithm for biobjective tsp," in *Workshop Recent Advances in Multi-Objective Optimization*, vol. 43, 2015.

[52] M. Fischetti and A. Lodi, "Optimizing over the first chvátal closure," *Mathematical Programming*, vol. 110, no. 1, pp. 3–20, 2007.

[53] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM review*, vol. 33, no. 1, pp. 60–100, 1991.

[54] N. Boland, H. Charkhgard, and M. Savelsbergh, "A criterion space search algorithm for biobjective integer programming : The balanced box method," *INFORMS Journal on Computing*, vol. 27, no. 4, pp. 735–754, 2015.

[55] T. Perini, N. Boland, D. Pecin, and M. Savelsbergh, "A criterion space method for biobjective mixed integer programming : The boxed line method," *INFORMS Journal on Computing*, vol. 32, no. 1, pp. 16–39, 2020.

[56] P. Serafini, "Some considerations about computational complexity for multi objective combinatorial problems," in *Recent advances and historical development of vector optimization*, pp. 222–232, Springer, 1987.

[57] T. Stidsen, K. A. Andersen, and B. Dammann, "A branch and bound algorithm for a class of biobjective mixed integer programs," *Management Science*, vol. 60, no. 4, pp. 1009–1032, 2014.

[58] S. L. Gadegaard, L. R. Nielsen, and M. Ehrgott, "Bi-objective branch-and-cut algorithms based on lp relaxation and bound sets," *INFORMS Journal on Computing*, vol. 31, no. 4, pp. 790–804, 2019.

[59] N. Forget, S. L. Gadegaard, K. Klamroth, L. R. Nielsen, and A. Przybylski, "Branch-and-bound and objective branching with three or more objectives," *Computers & Operations Research*, vol. 148, p. 106012, 2022.

[60] P. Cappanera and M. Trubian, "A local-search-based heuristic for the demand-constrained multidimensional knapsack problem," *INFORMS Journal on Computing*, vol. 17, no. 1, pp. 82–98, 2005.

A new algorithm for increasing the weight of minimum spanning trees and hypertrees

Mourad Baïou[1] and Francisco Barahona[2]

1. Laboratoire LIMOS,
Clermont-Ferrand, France
mourad.baiou@limos.fr

2. IBM Thomas J. Watson Research Center
New york, USA
barahon@us.ibm.com

Mots-clefs : Combinatorial optimization

We propose an algorithm to increase the weight of all minimum spanning trees in a graph under the assumption of a linear cost for each edge's weight increase and a fixed budget. We consider a graph with n nodes and m edges. Frederickson & Solis-Oba proposed an algorithm that requires the same asymptotic complexity as n^2m applications of Goldberg & Tarjan's push-preflow algorithm. By establishing a connection with Network Reinforcement, we develop an algorithm with the same asymptotic complexity as nm applications of the push-preflow algorithm, which improves the time complexity by a factor of n.

Additionally, we address the question of increasing the strength of a network, defined as the maximum number of disjoint spanning trees. We give an algorithm to increase the strength under a limited budget.

An hypergraph based formulation for an Automatic Storage Design problem

Luis Marques[1], François Clautiaux[1], Aurélien Froger[1]

1. Univ. Bordeaux, CNRS, Inria, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
luis.lopes-marques@inria.fr

Keywords : Hypergraphs, Dynamic programming, Integer linear programming, Arc-flow models, Cutting and Packing, Two stage guillotine knapsack, Temporal knapsack

Arc-flow formulation

Arc-flow formulations are increasingly popular in the field of integer programming (see [de Lima et al., 2022] for a recent survey). These formulations, often built from transition graphs of dynamic programs (DP), generally have strong linear relaxations, and can be solved directly by general purpose MIP solvers.

We first give a general overview on the link between dynamic programming and network-flow formulations. In the following, we express our dynamic programs using a backward recursive formulation. We note s the initial state, t the terminal state and f the function associating to a state v the minimum cost of going from state s to v. We also note $c_{u,v}$ the cost of going from state u to state v. The dynamic program at state v can be defined as follows, where $\Gamma^-(v)$ is the set of states preceding v.

$$f(v) = \begin{cases} 0 & \text{if } v = s \\ \min_{u \in \Gamma^-(v)} \{c_{u,v} + f(u)\} & \text{otherwise} \end{cases} \quad (1)$$

Finding the optimal solution $f(t)$ to the dynamic program is equivalent to solving a minimum cost flow problem in the transition graph associated with the DP. Let $G = (V, A)$ be the transition graph associated with the dynamic program. For every vertex $v \in V$, we respectively note $A^-(v)$ and $A^+(v)$ the set of arcs leaving and the set of arcs entering v. We keep the notation s (resp. t) for the vertex corresponding to the initial (resp. terminal) state.

An advantage of the MIP formulation is that one can include additional constraints (called *side* constraints in the remainder). These constraints may be added to the arc flow formulation in the form of resource constraints. Let $\mathcal{R}$ be a set of additional resources. For every resource $r \in \mathcal{R}$, we note $A(r)$ the set of arcs in A that consume the resource r. We also note $b_{a,r}$ the consumption of resource r associated to arc a and q_r the available amount of resource r. The corresponding arc flow formulation is as follows.

$$\begin{aligned}
\text{minimize} \quad & \sum_{a \in A} c_a x_a & & (2)\\
\text{subject to} \quad & \sum_{a \in A^-(v)} x_a - \sum_{a \in A^+(v)} x_a = 0 & v \in V \setminus \{s,t\} \quad & (3)\\
& \sum_{a \in A^-(t)} x_a = 1 & & (4)\\
& \sum_{a \in A(r)} b_{a,r} x_a \leq q_r & r \in \mathcal{R} \quad & (5)\\
& x_a \in \mathbb{N} & a \in A \quad & (6)
\end{aligned}$$

Constraints (3) and (4) represent the flow problem with a single source and a single sink. Note that without loss of generality, we can create a dummy node and link it to all the sinks to obtain a single source and single sink problem. (5) are resource constraints and (6) define the domain of the variables.

We study the extension of arc flow formulations to DP that can be expressed with hypergraphs. Such formulations in classical graphs usually provide strong linear relaxations. If resources constraints (5) are relaxed, the constraints matrix is totally unimodular, meaning the solution to the linear relaxation of the problem is feasible and optimal for the integer problem (see e.g. [Wolsey and Nemhauser, 1999]). In hypergraphs, this property does not hold, as the constraints matrix is not totally unimodular. However, since the transition hypergraphs are acyclic, the formulations is totally dual integral (TDI), as stated in [Martin et al., 1990]. Another difference between arc flow formulations in hypergraphs and in graphs is the size of the solution space. In transition graphs, by flow conservation there is at most one unit of flow going through an arc. This is not the case in flow formulations in hypergraphs, increasing the size of the solution space and the branch-and-bound tree. Furthermore, hyperarcs where more than one tail coincide with the same state allow flow to be created, resulting in a degradation of the linear relaxation's quality. Figure 1 shows an example of one unit of flow being created with half a unit of flow.

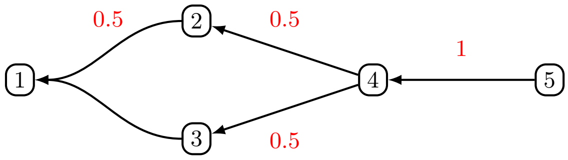
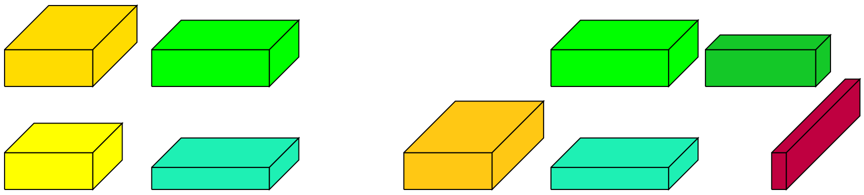


FIGURE 1 – Example of half a unit of flow creating one unit of flow

Application to automatic storage design

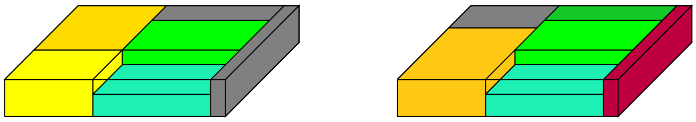
We use a problem called *Automatic Storage Design* (ASD) as an example of hypergraph modelling. In the ASD problem, given a set of items and the dimensions of a large storage device, the goal is to design a set of shelves in a box, design a set of compartments in every shelf and assign items to the compartments, where deciding to assign an item is associated with a profit. The objective is to maximize the total profit. There are two types of decisions to make : the design of the device, and the assignment of items in the compartments. The design parts are defined on a set of packing constraints, i.e., the sum of the shelves' heights must not exceed the box's height and the sum of compartments' widths in every shelf must not exceed the box's width.

The assignment part is defined on a set of temporal knapsack constraints, i.e., the items must fit with respect to their height, width, length and time windows. For an item to be a candidate for assignment in a compartment, its height must be less than or equal to the compartment's height and its width must be equal to the compartment's width. The length dimension represents the knapsack constraint, i.e., the sum of items' lengths with respect to their time windows must not exceed the compartment's length. This problem generalizes the temporal knapsack problem, and the three-dimensional three-stage guillotine cutting problem.



(a) $t = 0$ (b) $t = 1$

FIGURE 2 – Instance example with two time steps



(a) $t = 0$ (b) $t = 1$

FIGURE 3 – Example of solution for the instance in Figure 2

We first introduce a compact model. Each binary variable z_i is equal to one if one shelf in the box m is allocated with item $i \in \mathcal{I}$, meaning that the shelf takes the same height as i, and zero otherwise. Each binary variable $y_{i,j}$ is equal to one if one compartment in the shelf allocated by $i \in \mathcal{I}$ is allocated by $j \in \mathcal{I}$, meaning that the compartment takes the same width as j. Finally, each binary variable $x_{i,j,k}$ is equal to one if item $k \in \mathcal{I}$ is assigned to the compartment allocated by $j \in \mathcal{I}$ and zero otherwise.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i \in \mathcal{I}} \sum_{\substack{j \in \mathcal{I} \\ j \geq i}} \sum_{\substack{k \in \mathcal{I} \\ k \geq j}} p_i x_{i,j,k} & \\
\text{subject to} \quad & \sum_{i \in \mathcal{I}} z_i h_i \leq H & \\
& \sum_{\substack{j \in \mathcal{I} \\ j \geq i}} y_{i,j} w_j \leq W & i \in \mathcal{I} \\
& \sum_{\substack{k \in \mathcal{I} \\ k \geq j \\ s_k \leq t \leq f_k}} l_k x_{i,j,k} \leq L & i, j \in \mathcal{I}, j \geq i, t \in \mathcal{T} \\
& \sum_{\substack{i,j \in \mathcal{I} \\ i \leq j \leq k}} x_{i,j,k} \leq 1 & k \in \mathcal{I} \\
& x_{i,j,k} \leq y_{i,j} & i, j, k \in \mathcal{I}, k \geq j \geq i \\
& y_{i,j} \leq z_i & i, j \in \mathcal{I}, j \geq i \\
& z_i \in \{0, 1\} & i \in \mathcal{I} \\
& y_{i,j} \in \{0, 1\} & i, j \in \mathcal{I}, j \geq i \\
& x_{i,j,k} \in \{0, 1\} & i, j, k \in \mathcal{I}, k \geq j \geq i
\end{aligned}
$$

This problem can be reformulated as a maximum cost flow problem with resource constraints in a directed acyclic hypergraph. We propose a dynamic program to create a transition hypergraph for the automatic storage design problem with resource constraints relaxed. From the transition hypergraph we construct an arc flow formulation to solve the maximum cost flow problem with resource constraints.

First, we say that a *stage 1* (resp. *stage 2*) decision corresponds to placing a shelf (resp. compartment) in a box (resp. shelf) and we say that *stage 3* decisions are what items are assigned to the compartments. Figure 4 illustrates an example of design decisions. We note by $\alpha^1(h, w, l)$ the maximum profit obtained from a stage 1 state of dimensions h, w, l, by $\alpha^2(h, w, l)$ the maximum profit obtained from a stage 2 state of dimensions h, w, l and by $\alpha^3(h, w, l)$ the maximum profit obtained by assigning items to a compartment of dimensions h, w and l.

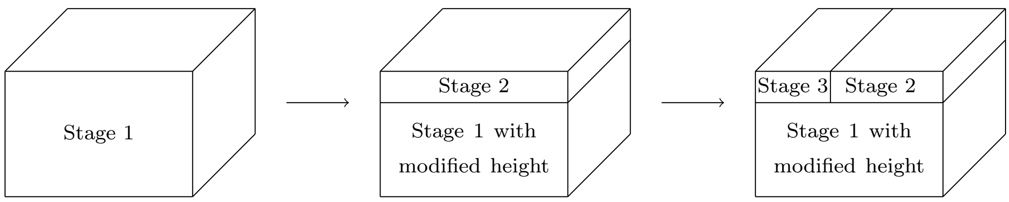


FIGURE 4 – Example of design decisions

Stage 1 and stage 2 decisions cut the box horizontally then vertically as in a two-stage guillotine cut problem. Similarly to [Clautiaux et al., 2018], the recursions $\alpha^1(h, w, l)$ and $\alpha^2(h, w, l)$ define two subproblems and create an hyperarc in the transition hypergraph. By noting $\mathcal{H}$ (resp. $\mathcal{W}$) the set of different heights (resp. widths), $\alpha^1(h, w, l)$ and $\alpha^2(h, w, l)$ are defined as follows.

$$
\alpha^1(h, w, l) = \max_{h' \in \mathcal{H}, h' \leq h} \{\alpha^2(h', w, l) + \alpha^1(h - h', w, l)\}
$$

$$\alpha^2(h, w, l) = \max_{w' \in \mathcal{W}, w' \leq w} \{\alpha^3(h, w', l) + \alpha^2(h, w - w', l)\}$$

The recursion $\alpha^3(h, w, l)$ is defined as in [Clautiaux et al., 2021]. The main idea is that items are decomposed into two events, one for the beginning of its time window, one for the ending. At each state, we decide if we accept the event or not. For an event related to the beginning of an item's time window, it relates to assigning the item to the knapsack or not. For an event relate to the end of an item's time window, it relates to updating the capacity if the item is in the knapsack or not. This recursion creates only classical arcs with one head and one tail. Figure 5 illustrates an example of the transition hypergraph in which states are noted $(h, w, l)^s$ where h, w, l and s are respectively the height, width, length and stage of the state.. In this figure, the compartments are not made explicit as they are similar to the transition graph generated by the dynamic program in [Clautiaux et al., 2021].

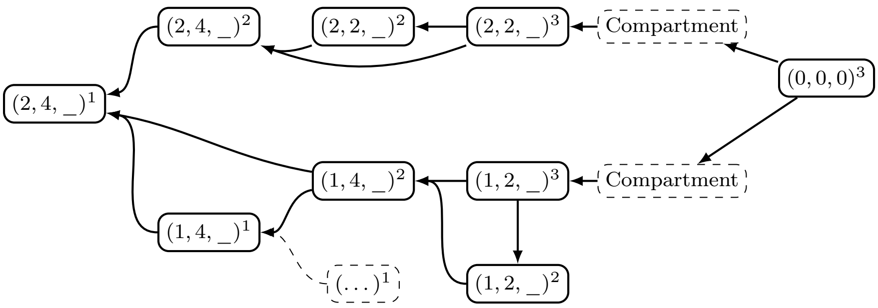


FIGURE 5 – Example of hypergraph

We introduce several preprocessing ideas useful to reduce the size of the hypergraph. Among them we reduce symmetries by enforcing an order on guillotine cuts, we aggregate equivalent states using methods from [Iori et al., 2021] and we introduce methods detect trivial subproblem and replace their states by another set of states where some constraints are relaxed.

Extensive numerical studies are performed on a set of 70 randomly generated instances to assess the efficiency of our methods and compare them with each other.

Références

[Clautiaux et al., 2021] Clautiaux, F., Detienne, B., and Guillot, G. (2021). An iterative dynamic programming approach for the temporal knapsack problem. *European Journal of Operational Research*, 293(2) :442–456.

[Clautiaux et al., 2018] Clautiaux, F., Sadykov, R., Vanderbeck, F., and Viaud, Q. (2018). Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem. *Discrete Optimization*, 29 :18–44.

[de Lima et al., 2022] de Lima, V. L., Alves, C., Clautiaux, F., Iori, M., and de Carvalho, J. M. V. (2022). Arc Flow Formulations Based on Dynamic Programming : Theoretical Foundations and Applications. *European Journal of Operational Research*, 296(1) :3–21.

[Iori et al., 2021] Iori, M., De Lima, V. L., Martello, S., Miyazawa, F. K., and Monaci, M. (2021). Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289(2) :399–415.

[Martin et al., 1990] Martin, R. K., Rardin, R. L., and Campbell, B. A. (1990). Polyhedral Characterization of Discrete Dynamic Programming. *Operations Research*, 38(1) :127–138.

[Wolsey and Nemhauser, 1999] Wolsey, L. A. and Nemhauser, G. L. (1999). *Integer and combinatorial optimization*, volume 55. John Wiley & Sons.

Nouveaux modèles pour la construction d'arbres de classification optimaux

Z. Ales[1,2], V. Huré[1] et Amélie Lambert[1]

1. Laboratoire CEDRIC, CNAM,
valentine.hure@lecnam.net, amelie.lambert@cnam.fr

2. Laboratoire UMA, ENSTA,
zacharie.ales@ensta-paris.fr

Mots-clefs : optimisation combinatoire, arbres de classification optimaux, PLNE, QP.

1 Arbres de classifications

On considère un jeu de données $(X_i, y_i)_{i\in\mathcal{I}}$ où $X_i \in \mathbb{R}^{|\mathcal{J}|}$ est le vecteur d'attributs de la donnée i et $y_i \in \mathcal{K}$ est sa classe. La classification supervisée consiste à construire une fonction de classification $f : \mathbb{R}^{|\mathcal{J}|} \mapsto \mathcal{K}$ (ou classifieur) à partir d'un jeu données.

Les arbres de décision constituent une famille de classifieurs qui attribuent une classe aux données en fonction de leur parcours dans un arbre. Les nœuds de branchement de cet arbre appliquent des règles de la forme $aX < b$ qui orientent chaque donnée vers un de leurs deux nœuds fils. Les règles sont dites univariées quand a est unitaire et multivariées sinon. Les feuilles attribuent une classe aux données qui les atteignent.

Déterminer un arbre de décision minimisant le nombre d'erreurs de classification est un problème NP-complet [5]. L'heuristique CART [3] est l'algorithme de la littérature le plus utilisé, mais son approche gloutonne fournit généralement des solutions sous-optimales. Récemment, plusieurs modèles ont été introduits pour résoudre à l'optimalité ce problème notamment via la PLNE [1, 2] ou la programmation dynamique [4]. Certains modèles se cantonnent à la construction d'arbres de décision pour des données binaires (*i.e.* $X_i \in \{0,1\}^{|\mathcal{J}|}$) [1, 4]. D'autres modèles permettent de créer des arbres de décisions à partir de données continues (*i.e.* $X_i \in \mathbb{R}^{|\mathcal{J}|}$) [2, 6]. Le modèle (OCT) [2] permet notamment de considérer des règles de branchement univariées ou multivariées.

Nos travaux se sont concentrés d'abord sur la résolution optimale du problème avec une modélisation quadratique inspirée du modèle linéaire (OCT) et une extension du modèle de [1] aux données continues. Nous avons aussi travaillé sur l'algorithme paramétrant les arbres, réduisant ainsi le temps de calcul de la construction d'arbre de classification performants.

La réduction de temps de calcul n'était pas assez significative pour envisager la prise en compte de jeux de données de plus d'une dizaine de milliers de données. Nous développons donc actuellement une approche permettant de réduire la taille des modèles résolus afin de pouvoir construire des arbres de classification pour de plus gros jeux de données.

2 Modélisations directes du problème

Modélisation quadratique

Le modèle (OCT) utilise trois familles de variables pour dénombrer les erreurs de classification.

Nous proposons une formulation quadratique ($QOCT$) permettant de supprimer ces variables. Nous avons linéarisé cette formulation de deux manières et montré que ses linéarisations possèdent toute une meilleure relaxation linéaire que (OCT). De plus, nous observons expérimentalement une réduction du temps de résolution par rapport à (OCT).

Généralisation du modèle de flots aux données continues
Ce modèle [1] associe à chaque donnée correctement classée un flot unitaire représentant son parcours dans l'arbre. Nous introduisons une extension (F) de ce modèle permettant de classifier des données continues. Dans le cas multivarié, nous démontrons que notre formulation permet des séparations non autorisées par (OCT). Enfin, nous observons expérimentalement que notre formulation est significativement plus rapide que (OCT).

Paramétrage des arbres
Les formulations considérées permettent de construire un arbre pour une profondeur et un nombre de nœuds de branchement fixés. Pour obtenir le meilleur arbre il faut donc résoudre un nombre conséquent de programmes. L'approche classique consiste à tester tous les couples de profondeurs et de nombre de nœuds de branchement. Nous définissons un algorithme permettant de réduire significativement le nombre de programmes à résoudre.

3 Modélisations utilisant des regroupements de données

Le nombre de variables des modélisations dépend en partie de la taille du jeu de données. Ainsi, pour réduire d'avantage le temps de calcul, nous proposons de réduire la taille des jeux de données considérés via des regroupements de données.

Regroupements de données
Le but du regroupement de données est de remplacer un groupe de données par un unique point. Il faut donc regrouper ensemble des données ayant de grandes chances d'atteindre la même feuille. La Figure 1 est un exemple de regroupement, à gauche nous avons le jeu de données initial et à droite une simplification de celui-ci comportant uniquement 9 données au lieu de 30.

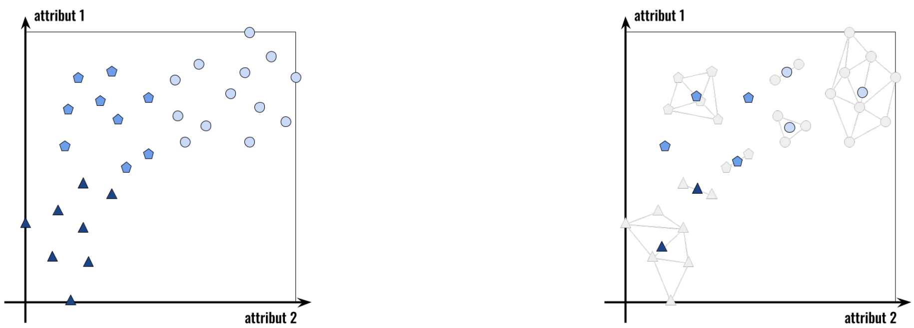


FIGURE 1 – Jeu initial : 30 données, 3 classes (gauche), et jeu fusionné : 9 données (droite)

Modélisations adaptées aux regroupements et algorithmes de résolution
Pour adapter n'importe quelle modélisation directe (P) aux regroupements, nous avons envisager deux approches. Dans la première (P_1), le représentant de chaque groupe est son barycentre.

L'objectif de la modélisation est adapté à considérer des données ayant un poids. Par exemple dans la Figure 1, le représentant du groupe en haut à droite aura un plus gros poids que le représentant du groupe en haut à gauche car il représente plus de données. Dans la seconde adaptation (P_2), les représentants ne sont pas prédéterminés, un point du groupe sera déterminé comme représentant lors de la résolution du programme. La première adaptation permet de réduire le nombre de contraintes et de variables tandis que la seconde ne permet de réduire que le nombre de variables. Cependant, nous démontrons que la seconde adaptation garantie l'optimalité de l'arbre sous certaines conditions

Pour un regroupement R donné, l'arbre obtenu par la résolution d'un modèle (P_X) (notation désignant (P_1) ou (P_2)) n'est généralement pas optimale pour les données d'origine. Ainsi, nous considérons un algorithme itératif. À chaque itération k, nous résolvons (P_X) pour obtenir un arbre $\mathcal{A}_k$. Nous identifions ensuite les groupes dont les données ne suivent pas toutes le même chemin dans $\mathcal{A}_k$ (on dit qu'ils sont *intersectés* par $\mathcal{A}_k$). Si un groupe est intersecté, nous le partitionnons en 2 sous-groupes. Nous démontrons que cet algorithme converge vers une solution optimale du jeu de données initial.

L'algorithme 1 peut être appliqué avec n'importe laquelle des adaptations mais ne garantit l'optimalité que pour l'adaptation (P_2).

Données : Un jeu de données et un regroupement associé R
Résultat : Un arbre $\mathcal{A}$
$R_0 \leftarrow R$;
$\mathcal{A}_0 \leftarrow$ solution de (P_X) appliqué à R_0 ;
Tant que $\mathcal{A}_k$ *intersecte* R_k **faire**
  Diviser les groupements intersectés par $\mathcal{A}_k$ pour obtenir R_{k+1} ;
  $\mathcal{A}_{k+1} \leftarrow$ solution de (P_X) appliqué à R_{k+1} ;
fin
Algorithme 1 : Algorithme utilisant les regroupements pour fournir un arbre

Références

[1] S. Aghaei, A. Gómez and P. Vayanos. Strong Optimal Classification Trees *https ://arxiv.org/abs/2103.15965* 2021.

[2] D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning* 106(7) :1039–1082, 2017.

[3] L. Breiman, J.H. Friedman, R. Olshen, and C.J. Stone. Classification and regression trees. *Routledge*, 2017.

[4] E. Demirović, A. Lukina A., E. Hebrard, J. Chan, J. Bailey, C. Leckie, K. Ramamohanarao and P.J. Stuckey. MurTree : Optimal Classification Trees via Dynamic Programming and Search *J. Mach. Learn. Res.*, 2022.

[5] L. Hyafil, and R.L Rivest. Constructing optimal binary decision trees is NP-complete *Elsevier*, 1976.

[6] S. Verwer and Y. Zhang Learning Optimal Classification Trees Using a Binary Linear Program Formulation. *AAAI*, 33(01) :1625–1632, 2019.

On the star forest polytope for cactus graphs

Viet Hung Nguyen[1], Thanh Loan Nguyen[1] et Minh Hieu Nguyen[1]

1. Laboratoire LIMOS, UMR CNRS 7024, INP et Université Clermont Auvergne, Campus Universitaire des Cézeaux.1 rue de la Chebarde.TSA 60125. CS 60026 63178 AUBIERE CEDEX

viet_hung.nguyen@uca.fr, thanh_loan.nguyen@uca.fr, minh_hieu.nguyen@uca.fr

Mots-clefs : Combinatorial optimization, spanning star forest, polyhedral combinatorics, facility location.

1 Introduction

Let $G = (V, E, c)$ denote an undirected weighted graph with $c : E \to \mathbb{R}_+$. A *star* in G can be defined as an isolated node or a subgraph where all edges share a common endpoint. If a star consists of more than one node, we refer to the node connected with every other node as the *center* of the star. For a star with a single edge, either of its endpoints can serve as the center. A *star forest* refers to a collection of disjoint stars in G. The *Maximum Weight spanning Star Forest Problem* (MWSFP) consists in finding a star forest in G of maximum weight. The MWSFP has various applications, such as computational biology [5] and the automobile industry [2]. Observing that in a maximum star forest F which does not contain isolated nodes, the set of centers forms a dominating set of G, Nguyen et al. [5] showed the NP-hardness of MWSFP by reducing the MWSFP to the well-known minimum dominating set problem. Subsequently, the MWSFP has been extensively studied, particularly in its unweighted version. However, there are not many studies available for the weighted version. In particular, the authors in [5] presented a polynomial-time algorithm to solve the MWSFP for weighted trees and a $\frac{1}{2}$-approximation algorithm for the general case. For cactus graphs, Nguyen [6] proposed a linear time algorithm to solve the MWSFP. Let us consider the convex hull of the incidence vectors associated with the star forests in G, denoted as $SFP(G)$. There are only a few studies on $SFP(G)$ so far, e.g., the work of Aider et al. [1], which gave a complete description for $SFP(G)$ when G is a tree or a cycle. In this paper, we focus on a polyhedral investigation of the MWSFP for cactus graphs, a generalization of trees and cycles.

2 Star forest polytope for cactus graphs

A graph G is a *cactus* if each edge of G belongs to at most one cycle in G. To analyze the characteristics of the star forest polytope for cactus graphs in detail, we employed the uncapacitated facility location problem (UFLP) discussed in Baiou [4]. To accomplish this, we transformed the original weighted graph $G = (V, E, c)$ into a bidirected graph $\overrightarrow{G} = (V, A)$, where each edge uv in E with cost c_{uv} corresponds to two arcs (u, v) and (v, u) in A with costs $c(u, v)$ and $c(v, u)$ respectively. Considering the convex hull of the incidence vectors of feasible solutions for the UFLP on $\overrightarrow{G}$, denoted as UFLP($\overrightarrow{G}$). Examining the relationship between the solutions of UFLP($\overrightarrow{G}$) and MWSFP, we find that each $(0, 1)$-integer solution to the symmetric UFLP problem on $\overrightarrow{G}$ corresponds to a solution for the MWSFP problem on G with the same cost. Conversely, each solution for the MWSFP on G corresponds to one or multiple $(0,1)$-integer solutions for the symmetric UFLP problem on $\overrightarrow{G}$ with an equivalent cost. As a result, we obtain that the projection of UFLP($\overrightarrow{G}$) onto the variables $x_{uv} = \overrightarrow{x}(u,v) + \overrightarrow{x}(v,u)$ is precisely the star forest polytope of G. Beside, Baiou et al. [4] gave the following set of systems which completely describe UFLP($\overrightarrow{G}$) :

$$y(u) + \sum_{(u,v)\in A} \overrightarrow{x}(u,v) \leq 1, \ \forall u \in V \tag{1}$$

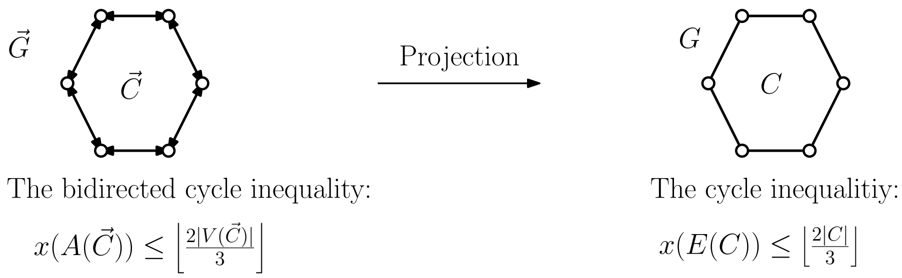
$$\overrightarrow{x}(u,v) \leq y(v), \ \forall (u,v) \in A \tag{2}$$

$$y(u) \geq 0, \overrightarrow{x}(u,v) \geq 0, \ \forall (u,v) \in A, \forall u \in V, \tag{3}$$

and the *bidirected cycle inequality* [3], the *lifted g-odd cycle inequalities* [4]. In the rest of this report, we call the inequalities (1) as *assignment inequalities.*

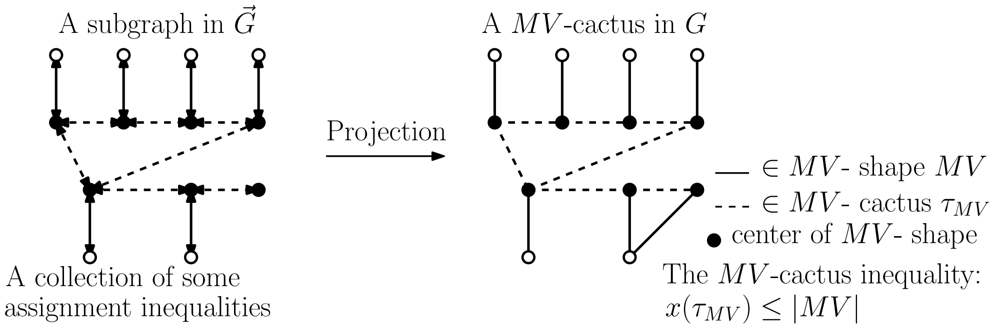
Projection of bidirected cycle inequalities

Let C be a cycle in G and $\overrightarrow{C}$ be the bidirected cycle corresponding in $\overrightarrow{G}$. Then, the projection of the bidirected cycle inequalities onto the variable x_{uv} is the *cycle inequalities* [1].



Projection of the assignment inequalities

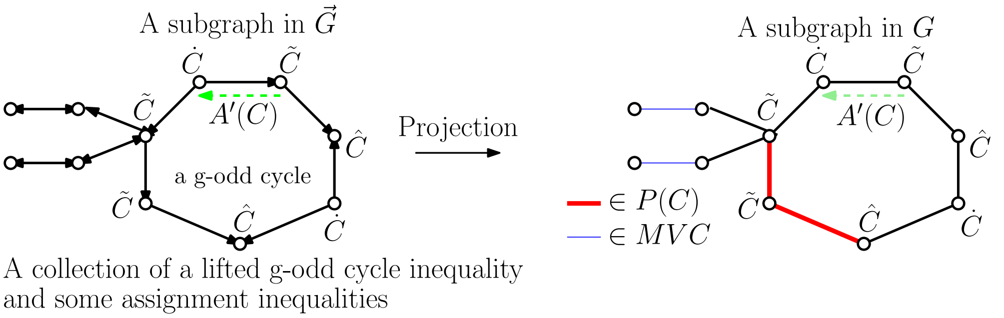
Applying the Fourier-Motzkin elimination to the inequalities (1) and (2), we obtain that $x_{uv} + \sum_{(u,w)\in A, w\neq v} \overrightarrow{x}(u,w) \leq 1, \ \forall u \in V, uv \in E$. Then, we show that the projection of the assignment inequalities is "MV-cactus inequalities". Let us introduce MV-cactus inequality, a cactus subgraph in G is called a *MV-cactus* τ_{MV} when every non-pendant node is connected to precisely one pendant node unless it is contained in a triangle or a cycle of length four. A *MV-shape* MV of G is a bipartite graph (U,S) in which the nodes in part U have a maximum degree of two and the part S is connected in G. The nodes in S are called *centers* of MV. Then, the *MV-cactus inequality* is defined as follows : $x(\tau_{MV}) \leq |MV|$.



Projection of lifted g-odd cycle inequalities

Let us remind the definition of the lifted g-odd cycle (in [4]) as follows. Let $C = (v_0, a_0, v_1, a_1, ..., a_{p-1}, v_p)$ where $v_p = v_0$ be a simple cycle in $\overrightarrow{G}$. Each v_i and a_i for $i = 0, ..., p-1$ represents distinct nodes and arcs, respectively. Let $V(C)$ be the set of nodes in C. Let $\hat{C}$ be the set of nodes v_i such that v_i is the head of both a_{i-1} and a_i, $\dot{C}$ be the set of nodes v_i such that v_i is the tail of both a_{i-1} and a_i, and $\tilde{C} = V(C) \setminus (\hat{C} \cup \dot{C})$, for all $0 \leq i \leq p$. Then, C is called a *g*-odd cycle if $p + |\tilde{C}|$ is *odd*. We denote by $A'(C)$ the maximal matching of the subgraph of C inducted by the arcs (u, v) in which $u \in \tilde{C}, v \in \dot{C}$. We say that $A'(C)$ define a *lifting set* [4] in C. Then, the lifted g-odd cycle inequality is presented as follows : $\sum_{(u,v)\in A(C)} \overrightarrow{x}(u,v) + \sum_{(u,v)\in A'(C)} \overrightarrow{x}(u,v) - \sum_{u\in\hat{C}} y(u) \leq \dfrac{|\hat{C}| + |\tilde{C}| - 1}{2}$.

Perform the projection of lifted g-odd cycle inequalities onto the variable x_{uv}, and we get inequalities called "MV-partition". Let us present MV-partition inequalities, given a g-odd cycle C in $\overrightarrow{G}$. Considering a MV-shape MV which has a center v in C and τ is MV-cactus associated with MV. A *MV-partition* via (v, C) is a partition of τ obtained by removing two edges incident to v in C. A *MV-split* is the common edges of the MV-shape and the MV-partition. Denoting by C^* the subset of $\tilde{C}$ such that for every vertex v in C^*, its tail u is either also in $\tilde{C}$ or in $\dot{C}$ and $(v, u) \notin A'(C)$. Let $P(C)$ be the underlying edges corresponding to the arcs with their tails in C^*. We can specify τ_{MVC} as the set of all MV-partitions obtained via (v, C), where $v \in C^* \cup \hat{C}$ and v is a cut vertex of G. MVC denotes the set of all MV-splits that belong to the MV-partitions. The *MV-partition inequality* is defined as below : $x(E(C) \setminus P(C)) + 2x(P(C)) + x(\tau_{MVC}) \leq \dfrac{3|\hat{C}| + 3|\tilde{C}| - 1}{2} - |A'(C)| + |MVC|$.



Consequently, we obtain the following main theorem of this paper.

Theorem 1. *Let G be a cactus graph, and $\overrightarrow{G}$ be its associated bidirected graph. Then the following system completely describes $SFP(G)$.*

$$x(E(C)) \leq \left\lfloor \frac{2|C|}{3} \right\rfloor, \forall \text{ cycle } C \subset G \tag{4}$$

$$x(\tau_{MV}) \leq |MV|, \forall \ MV\text{-cactus } \tau_{MV} \subset G \tag{5}$$

$$x(E(C) \setminus P(C)) + 2x(P(C)) + x(\tau_{MVC}) \leq \frac{3|\hat{C}| + 3|\tilde{C}| - 1}{2} - |A'(C)| + |MVC|, \quad \forall \text{ g-odd cycle } C \subset \overrightarrow{G} \tag{6}$$

$$0 \leq x_e \leq 1, \forall e \in E. \tag{7}$$

Références

[1] M. Aider, L. Aoudia, M. Baiou, A. R. Mahjoub, V. H. Nguyen. On the star forest polytope for trees and cycles. Operations Research, 53 (5), 1763-1773, 2019.

[2] A. Agra, D. Cardoso, O. Cerdeira, and E. Rocha. A spanning star forest model for the diversity problem in automobile industry, ECCO XVIII, Minsk, Belarus, May 26-28, 2005.

[3] Avella, P., Sassano, A., Vasilev. I. : Computational study of large-scale p-median problems. Mathematical Programming, 89–114, 2007.

[4] M. Baiou and F. Barahona, The dominating set polytope via facility location. Combinatorial Optimization. ISCO 2014. In Vol. 8596 of Lecture Note Computer Sciences 38–49, 2014.

[5] C.T. Nguyen, J. Shen, M. Hou, L. Sheng, W. Miller and L. Zhang, Approximating the spanning star forest problem and its applications to genomic sequence alignment. SIAM Journal on Computing, 38 (3), 946-962, 2008.

[6] V. H. Nguyen. The maximum weight spanning star forest problem on cactus graphs. Discrete Mathematics, Algorithms and Applications, World Scientific Publishing, 07 (02), 2015.

Geometric Packing, Hitting and Representation : the simplest Open Challenges on Geometric Intersection Graphs

Marco Caoduro[1] et Andras Sebo[1], Directeur de Recherche émérite,

1. Laboratoire G-SCOP,
46 Av. Félix Viallet, Grenoble, Université de Grenoble Alpes, France
marco.caoduro@grenoble-inp.fr, Andras.Sebo@grenoble-inp.fr

Mots-clefs : Packing and Covering problems, combinatorial optimization.

1 Abstract

Packing and Covering problems are the alpha and the omega (or better : nu and tau ;=)) of combinatorial optimization, and have been largely explored in the past decades as the most recent seminal books of the field show. For some combinatorial objects minmax relations provide 'good characterizations' and efficient algorithms, while for some others, NP-hardness is accompanied by a constant duality gap or/and approximation algorithms. In this talk I would like to explain some old conjectures and recent developments on packing and hitting the simplest objects in the plane : rectangles, squares and segments, place them in the forest of results and open problems of the field, and present the most frustrating challenges.

Maximum chordal sub-graph problem

F. Clautiaux[1] and P. Pesneau[1]

1. Univ. Bordeaux, CNRS, Inria, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
{francois.clautiaux}{pierre.pesneau}@math.u-bordeaux.fr

Mots-clefs : Chordal graphs, Integer linear programming formulations, Valid inequalities, Branch-and-Cut

1 Introduction

A chordal graph if a graph whose cycles of length at least four contain at least one chord. These graphs belong to the more general class of perfect graphs, and numerous graph optimization problems (stable set, graph coloring, ...) known to be NP-complete in the general case, become polynomial when the graph in chordal.

In this presentation, we address the problem of minimizing the number of edges to delete from a graph to obtain a chordal graph, or in other words, finding the maximum chordal sub-graph (MCS) of a graph. This problem occurs when one wants to derive a dual bound for a hard combinatorial problem by removing edges from the original graph to obtain an easier problem with fewer edges/constraints. More formally, given a graph $G = (V, E)$ on the node-set $V = \{1, \ldots, n\}$ and edge-set E, we are searching for a subset F of E of maximum size such that the sub-graph $H = (V, F)$ is chordal.

In [1], the authors study the completion variant of this problem, that is finding a minimum number of edges to include in order to obtain a chordal graph. They present a natural formulation involving an exponential size set of constraints, and various additional class of valid and facet-defining inequalities.

We designed a mathematical programming model for our problem. It explicitly excludes chordless cycles, and can be solved by branch-and-cut. Additional cutting planes are added to improve the quality of the formulation.

In the sequel, we will use the following notation An elementary cycle C (abusively called cycle) of size $k \geq 3$ is a set of k edges of E, $C = \{\{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{k-1}, u_k\}, \{u_k, u_1\}\}$, such that the ordered set of nodes of the cycle $V(C) = (u_1, \ldots, u_k)$ are all distinct. Given a cycle C of G, an edge that links two non consecutive nodes of $V(C)$ is called a chord. Let $\Theta(C)$ denotes the set of chords of a cycle C. Let $\mathcal{C}_{\geq 4}$ be the set of cycles of G of size at least 4. Let $W \subset V$ be a subset of nodes. We denote by $E(V)$ the set of edges having both end-nodes in V, that is $E(W) = \{\{i, j\} \in E : i, j \in W\}$. The sub-graph induced by W is defined by $G[W] = (W, E(W))$.

2 Mathematical models

A first compact integer linear programming model, inspired from the one proposed in [2], is based on the concept of perfect elimination ordering. Given a graph G, an ordering $\{v_1, \ldots, v_n\}$ of its vertices is a *perfect elimination ordering* if and only if for $i = 1, \ldots, n$, the neighborhood of vertex v_i in the sub-graph $G[\{v_i, \ldots, v_n\}]$ (induced by the vertices of index at least i in this order) is a clique. It is shown in [4] that a graph G is chordal if and only if it admits a perfect elimination ordering of its vertices.

The first formulation will search for a perfect elimination ordering by directing the edges of the graph G along this ordering and eliminating edges creating circuits. To this aim, let us directing the graph G to get the graph directed graph $D = (V, A)$ where each edge $\{i, j\} \in E$ is replaced by two arcs (i, j) and (j, i). For each arc (i, j) in A, consider the variable z_{ij} taking value 1 if the edge $\{i, j\}$ is taken in the chordal graph and i is j in the perfect elimination ordering. In addition, consider for each node $v \in V$ the integer variable u_v that gives the position of node v in the ordering.
Then the MCS problem can be modeled as following.

$$\max \sum_{\{i,j\}\in E} z_{ij} + z_{ji} \tag{1}$$

$$\begin{aligned}
\text{s.t. } & z_{ij} + z_{ji} \leq 1 && \forall \{i,j\} \in E, && (2)\\
& z_{ij} + z_{ik} - z_{jk} - z_{kj} \leq 1 && \forall i \in V, \forall j,k \in V : (i,j),(i,k) \in A \text{ and } \{j,k\} \in E, && (3)\\
& z_{ij} + z_{ik} \leq 1 && \forall i \in V, \forall j,k \in V : (i,j),(i,k) \in A \text{ and } \{j,k\} \notin E, && (4)\\
& u_j \geq u_i + 1 - n * (1 - z_{ij}) && \forall (i,j) \in A, && (5)\\
& u_i \geq 0 && \forall i \in V, && (6)\\
& z_{ij} \in \{0,1\}, && \forall (i,j) \in A. && (7)
\end{aligned}$$

Constraints (2) ensure that at most one orientation of each edge is chosen. The simplicial condition for each vertex $i \in V$ is enforced by inequalities (3) and (4) : if the edge $\{j, k\}$ does not exist then i cannot be before j and k in the ordering. Otherwise, the edge $\{j, k\}$ should be considered in the solution. Finally, constraints (5) and (6) are Miller-Tucker-Zemlin-like inequalities ensuring that the directed graph does not contain any circuit and defines a (partial) order.

In [3], the authors refine the above chord condition for chordal graph and state that any cycle C of size $k \geq 4$ of a chordal graph contains at least $k - 3$ chords. This improved condition is used to introduce our main integer linear programming model for MCS problem. Let us associate to each edge $e := \{i, j\}$ in E a binary variable $x_e (= x_{ij})$ taking value 1 if the edge e is kept in the chordal graph, and 0 if it is deleted. Given a set $F \subset E$ of edges, let $x(F) = \sum_{e \in F} x_e$.
The MCS problem can be modeled as

$$\text{Max.} \sum_{e\in E} x_e \tag{8}$$

$$\begin{aligned}
\text{s.t. } & x(\Theta(C)) \geq |C| - 3 - (|C| - 3)(|C| - x(C)) && \forall C \in \mathcal{C}_{\geq 4}, && (9)\\
& x_e \in \{0,1\} && \forall e \in E. && (10)
\end{aligned}$$

While objective (8) aims to maximize the number of kept edges, inequalities (9) state that for any cycle C, if it is contained in the solution (that is $x(C) = |C|$) then the number of chords is greater than $|C| - 3$ (for otherwise $|C| - x(C) \geq 1$ and the inequality is trivially satisfied).

Note that if a cycle $C \in \mathcal{C}_{\geq 4}$ does not possess $|C| - 3$ chords, then the C cannot belong to the solution graph and thus inequality (9) can be replace by the strongest one

$$x(C) \leq |C| - 1. \tag{11}$$

Chordal graphs are also often called triangulated graphs. This comes from the fact that any edge belonging to a cycle in such a graph, should also belongs to a triangle (the complete graph on three nodes). A natural observation is that if the graph G does not contain any triangle, then it cannot contain any cycle, and consequently should be a forest.

Another observation about chordal graphs is that any induced sub-graph of a chordal graph is also chordal, meaning that any property can also be applied to induced sub-graph.

Let $\mathcal{T}$ be the set of node-sets T such that the induced sub-graph $G[T]$ does not contain any triangle. The two previous observations lead to the following set of valid inequalities.

$$x(E(T)) \leq |T| - 1 \quad \forall T \in \mathcal{T}. \tag{12}$$

These inequalities are facet-defining if and only if the spanning tree polytope is full dimensional for the induced sub-graph $G[T]$.

3 A branch-and-cut algorithm

A branch-and-cut algorithm has been developed to solve this problem. Inequalities (9) and (11) are separated in polynomial time for integral solutions by checking if the solution graph is chordal [5] and if not, finding a chordless cycle [6]. When handling fractional solutions, violated inequalities (9) and (11) are generated by finding chordless cycle in a graph composed of edges whose value is greater than a given threshold. Once a chordless cycle C is found, conditions are checked to determine which inequalities among (9) and (11) for cycle C and (12) for the set $V(C)$ should be added to the current model.

In addition, a specific heuristic has been developed to generate violated inequalities of type (12) and a slight generalization of these constraints.

Finally, a heuristic, based on perfect elimination ordering, generates a chordal sub-graph used as a primal bound for the branch-and-cut algorithm. This heuristic determines an ordering of the nodes based on the maximum cardinality search described in [5] to check the chordality of the graph. If the ordering is not a perfect elimination ordering (that is, if the graph is not chordal) then edges are removed such that the ordering becomes a perfect elimination ordering.

Preliminary computational experiments show that the branch-and-cut is able to produce competitive results compared to the compact formulation.

Références

[1] David Bergman, Carlos H. Cardonha, Andre A. Cire, and Arvind U. Raghunathan. On the minimum chordal completion polytope. *Operations Research*, 67(2) :532–547, 2019.

[2] C. Feremans, M. Oswald, and G. Reinelt. A y-formulation for the treewidth. Technical report, Heidelberg University, 2002.

[3] Assaf Natanzon, Ron Shamir, and Roded Sharan. A polynomial approximation algorithm for the minimum fill-in problem. *SIAM J. Comput.*, 30(4) :1067–1079, 2000.

[4] Donald J Rose. Triangulated graphs and the elimination process. *Journal of Mathematical Analysis and Applications*, 32(3) :597–609, 1970.

[5] Robert E. Tarjan and Mihalis Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13(3) :566–579, 1984.

[6] Robert E. Tarjan and Mihalis Yannakakis. Addendum : Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 14(1) :254–255, 1985.

On the box-total dual integrality of the perfect matching polytope

R. Grappe, M.Lacroix, F. Pisanu et R. Wolfler-Calvo

LIPN, Université Sorbonne Paris Nord
99 Av. Jean Baptiste Clément, Villetaneuse, 93430, France
grappe@lipn.univ-paris13.fr
lacroix@lipn.univ-paris13.fr
pisanu@lipn.univ-paris13.fr
wolfler@lipn.univ-paris13.fr

Mots-clefs : matching covered graph, perfect matching polytope, box-totally dual integral polyhedron, equimodular matrix.

1 Introduction

The *perfect matching polytope* is the convex hull of the characteristic vectors of all perfect matchings of a graph. A graph is *matching covered* if each of its edges belongs to a perfect matching. These graphs have been introduced by Edmonds, Pulleyblank, and Lovász in [6]. A matching covered graph G is *solid* if it has no separating cuts, where a cut $\delta(X)$ is *separating* if G/X and $G/(V \setminus X)$ are also matching covered.

A rational linear system is totally dual integral (TDI) if for every integer linear function for which the optimum is finite the associated dual problem has an integer optimal solution. A TDI system is box-TDI if adding any rational bounds on the variables preserves its TDIness. Box-TDI systems are systems that yield strong min-max relations such as the one involved in the Max Flow-Min Cut Theorem of Ford and Fulkerson. A polyhedron is box-TDI if it can be described by a box-TDI system.

Box-totally dual integral systems and polyhedra received a lot of attention from the combinatorial optimization community around the 80's. A renewed interest appeared in the last decade and since then many deep results appeared involving such systems (see for exemple [2], [4], and [5]).

Ding et al. [5] characterized the box-TDIness of the matching polytope. Though the perfect matching polytope is a face of the latter, their result does not characterize the box-TDIness of the perfect matching polytope.

2 Contributions

First, extending the work of de Carvalho et al. [3], we characterize the perfect matching polytope of solid matching covered graphs.

Theorem 1. *The perfect matching polytope of a solid matching covered graph is the intersection of its affine hull and the half spaces described by the trivial inequalities.*

Then, combining a characterization of box-TDI polyhedra of Chervet et al. [2] with a result on incidence matrices of graphs [1], we prove the following.

Corollary 2. *Recognizing whether the perfect matching polytope of a given matching covered graph is box-TDI can be done in polynomial time.*

References

[1] Patrick Chervet, Roland Grappe, Mathieu Lacroix, Francesco Pisanu, and Roberto Wolfler Calvo. Hard problems on box-totally dual integral polyhedra. http://dx.doi.org/10.2139/ssrn.4397713.

[2] Patrick Chervet, Roland Grappe, and Louis-Hadrien Robert. Box-total dual integrality, box-integrality, and equimodular matrices. *Mathematical Programming*, 188(1):319–349, 2021.

[3] Marcelo H. de Carvalho, Cláudio L. Lucchesi, and U.S.R. Murty. The perfect matching polytope and solid bricks. *Journal of Combinatorial Theory, Series B*, 92(2):319–324, 2004.

[4] Guoli Ding, Li Feng, and Wenan Zang. The complexity of recognizing linear systems with certain integrality properties. *Mathematical Programming*, 114(2):321–334, 2008.

[5] Guoli Ding, Lei Tan, and Wenan Zang. When is the matching polytope box-totally dual integral? *Mathematics of Operations Research*, 43(1):64–99, 2018.

[6] J. Edmonds, W. R. Pulleyblank, and L. Lovász. Brick decompositions and the matching rank of graphs. *Combinatorica*, 2:247–274, 1982.

What is the gradient of a Linear Program? Automatic differentiation on a polytope

G. Dalle[1], L. Baty[2], L. Bouvier[2], A. Parmentier[2]

1. Laboratoires IdePHICS et INDY, EPFL
Lausanne, Suisse
guillaume.dalle@epfl.ch

2. Laboratoire CERMICS, École des Ponts
Marne-la-Vallée, France
{leo.baty, louis.bouvier, axel.parmentier}@enpc.fr

Mots-clefs : combinatorial optimization, machine learning, automatic differentiation, graphs, Julia programming language.

Solving combinatorial optimization problems is a routine ingredient of many industrial processes. While such problems are often intractable, it is possible to learn from past instances in order to obtain better solutions with less computation time [2]. This is the topic of our present contribution, see our preprint [6] for more details.

We first justify the practical interest of such hybrid approaches, before focusing on a key technical hurdle: gradient computation for combinatorial solvers. Overcoming this hurdle requires a novel mix of several mathematical ideas, which we implemented in an easy-to-use Julia package.

1 Learning to approximate combinatorial problems

As a motivating application, let us consider the Vehicle Scheduling Problem (VSP) and its stochastic counterpart. We are given a directed acyclic graph $G = (V, A)$ of tasks (for instance passengers of a taxi company): each task v_i has a location ℓ_i, a beginning time t_i^b and an end time t_i^e. There is an arc a from task v_i to task v_j if these tasks can be chained, that is, if a single vehicle has enough time to go from location ℓ_i to ℓ_j in the time interval $[t_i^e, t_j^b]$. Every arc a is associated with a cost θ_a. The deterministic VSP consists in assigning each vehicle of a fleet to a sequence of tasks, such that all tasks are covered and the sum of arc costs is minimized. It can be easily solved with a flow formulation.

In the stochastic version of the problem, each task (and each transfer between tasks) are subject to random delays. Minimizing the expected total delay in this setting is intractable, and exact algorithms do not scale to large instances. Luckily, we can use the VSP as a subroutine to approximate the stochastic VSP, as illustrated on Figure 1. Instead of doing it naively, our trick is to learn a suitable transformation of the instance to improve solution quality [9, 8]. Based on a dataset of previously encountered task graphs and delay realizations, we estimate the parameters w of an encoder φ_w, typically a neural network. This encoder outputs modified arc costs θ_a, which we feed into a VSP to generate vehicle routes. That way, our fast deterministic solver takes into account the distribution of delays, and how they interact with the graph structure, without sacrificing efficiency.

Hybrid solution pipelines like that of Figure 1, where machine learning models are fused with combinatorial solvers, yield state-of-the-art results on several benchmarks of interest [6, 1]. But

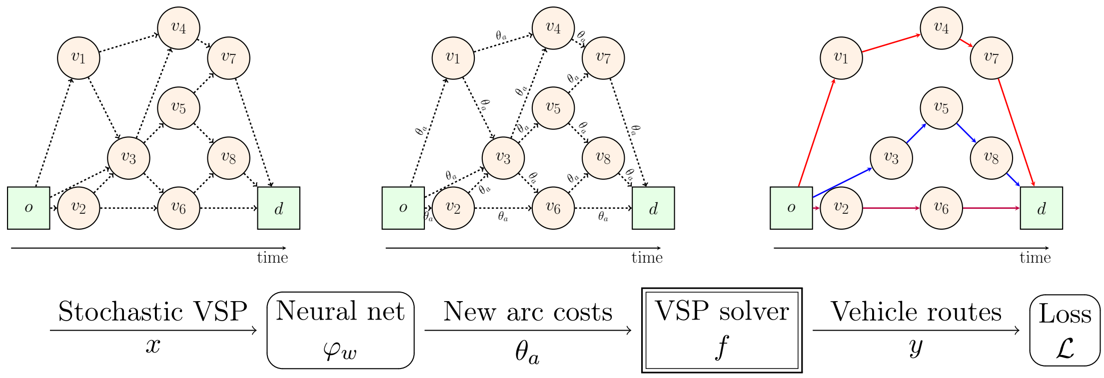


Figure 1: Hybrid solution pipeline for the stochastic VSP

to learn the parameters of the encoder with gradient descent, it is necessary to backpropagate loss derivatives through the entire pipeline, including the solver f.

2 From a solver to a probability distribution

In what follows, we do not care about the precise implementation of f. As far as we are concerned, it is a black box. The only requirement we impose is the existence of a finite set of points $\mathcal{Y} \subset \mathbb{R}^d$ such that

$$f(\theta) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \; \theta^\top y \tag{1}$$

By interpreting these points as vertices of a polytope, we see that Equation (1) includes every bounded Mixed Integer Linear Progam (MILP) as a special case, not just the VSP from above. Alas, the mapping from the cost vector θ to the maximizer of $\theta^\top y$ on $\mathcal{Y}$ is piecewise constant, which hinders backpropagation.

To generate meaningful gradients, we adopt a probabilistic point of view. Our solver f induces a very simple probability distribution on the vertex set: $p(\theta) = \delta_{f(\theta)}$. What happens if we spread out this Dirac mass, to encompass several vertices instead of just one? We obtain an approximate probability distribution $\widehat{p}(\theta)$, from which we deduce an approximate solver

$$\widehat{f}(\theta) = \mathbb{E}_{\widehat{p}(\theta)}[Y] = \sum_{y \in \mathcal{Y}} y \widehat{p}(y|\theta) \tag{2}$$

Our wish list for $\widehat{p}(\theta)$ is the following: it must be

Smooth: differentiable in θ

Generic: only requires calling f...

Precise: close to $\delta_{f(\theta)}$

Tractable: ... a small number of times

If these conditions are satisfied, then the Jacobian of $\widehat{f}$ is easy to compute:

$$\partial_\theta \widehat{f}(\theta) = \sum_{y \in \mathcal{Y}} y \nabla \widehat{p}(y|\theta)^\top \tag{3}$$

3 Regularization and the Frank-Wolfe trick

There are many ways to define the distribution $\widehat{p}(\theta)$: here we focus on regularization [5]. The idea is to tweak Equation (1) by convexifying the feasible set and adding a convex penalty Ω:

$$\widehat{f}_\Omega(\theta) = \operatorname*{argmax}_{\mu \in \mathrm{conv}(\mathcal{Y})} \theta^\top \mu - \Omega(\mu) \tag{4}$$

Remember that we only have access to the linear solver f, which means our best option is the Frank-Wolfe (FW) algorithm [7]. This algorithm solves a sequence of linearizations of Equation (4), each iteration yielding a polytope vertex y_k. The end result is a convex combination $\sum_k \lambda_k y_k$.

If we reinterpret the coefficients $\lambda_k \in [0, 1]$ as probabilities $\widehat{p}^{\mathrm{FW}}(y_k|\theta)$, we see that FW implicitly defines a distribution on $\mathcal{Y}$. This distribution has a sparse support, whose size is controlled by the number of FW iterations. Finally, the weights $\widehat{p}^{\mathrm{FW}}(y_k|\theta)$ can be differentiated implicitly by viewing FW as a constrained optimization procedure on the probability simplex [4].

Let's take a step back. Is our wish list satisfied?

Precise: ✓ with small Ω

Generic: ✓ because FW uses linearizations

Smooth: ✓ because of implicit diff.

Tractable: ✓ with few iterations

This means we have succeeded in building an approximate combinatorial solver with the right properties for gradient backpropagation. We can then insert it in a hybrid pipeline to tackle hard optimization problems with the help of machine learning.

4 The Julia package `InferOpt.jl`

As we have seen, there are quite a lot of ingredients to mix for this method to work. In order to facilitate industrial applications, we implemented all of them in an open source toolbox called `InferOpt.jl`[1]. It is written in the Julia language [3], and designed to be compatible with the entire machine learning and optimization ecosystem. Thanks to Julia's composable automatic differentiation, we do not need to restrict the kind of solvers or neural networks that can be plugged in.

The talk will feature a live demonstration of our library, hopefully inspiring the audience to test it on their own problems.

References

[1] L. Baty, K. Jungel, P. S. Klein, A. Parmentier, and M. Schiffer. *Combinatorial Optimization Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows.* 2023. arXiv: 2304.00789 [cs, eess, math, stat]. URL: http://arxiv.org/abs/2304.00789. preprint.

[2] Y. Bengio, A. Lodi, and A. Prouvost. "Machine Learning for Combinatorial Optimization: A Methodological Tour d'horizon". In: *European Journal of Operational Research* 290.2 (2021), pp. 405–421.

[1] https://github.com/axelparmentier/InferOpt.jl

[3] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A Fresh Approach to Numerical Computing". In: *SIAM Review* 59.1 (2017), pp. 65–98.

[4] M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J.-P. Vert. "Efficient and Modular Implicit Differentiation". In: Advances in Neural Information Processing Systems. 2022.

[5] M. Blondel, A. F. T. Martins, and V. Niculae. "Learning with Fenchel-Young Losses". In: *Journal of Machine Learning Research* 21.35 (2020), pp. 1–69.

[6] G. Dalle, L. Baty, L. Bouvier, and A. Parmentier. *Learning with Combinatorial Optimization Layers: A Probabilistic Approach*. Submitted to the Journal of Machine Learning Research. 2022. arXiv: 2207.13513 [cs, math, stat]. URL: http://arxiv.org/abs/2207.13513. preprint.

[7] M. Frank and P. Wolfe. "An Algorithm for Quadratic Programming". In: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110.

[8] A. Parmentier. *Learning Structured Approximations of Combinatorial Optimization Problems*. 2023. arXiv: 2107.04323 [cs, math]. URL: http://arxiv.org/abs/2107.04323. preprint.

[9] A. Parmentier. "Learning to Approximate Industrial Problems by Operations Research Classic Problems". In: *Operations Research* 70.1 (2022), pp. 606–623.

A branch-and-bound algorithm for two-stage no-wait hybrid flow shop scheduling with interstage flexibility

MN. Azaiez[1], A. Gharbi[2], I. Kacem[3] and Y. Makhlouf [4]

1. Business Analytics and DEcision Making Laboratory (BADEM)
Tunis Business School, Université de Tunis, Tunis 2059, Tunisia

2. Department of Industrial Engineering, King Saud University
Riyadh 11421, Saudi Arabia

3. Université de Lorraine, LCOMS 3, Rue Augustin Fresnel, Metz, France
imed.kacem@univ-lorraine.fr

4. Université de Lorraine, LCOMS 3, Rue Augustin Fresnel, Metz, France
ENIT, Université de Tunis el Manar
Tunis, Tunisie
yosra.makhlouf@unis-lorraine.fr

Keywords : branch-and-bound, hybrid flow shop scheduling, no-wait, exact methods

1 Introduction and Problem statement

Hybrid flow shop scheduling is a central problem in scheduling theory. It combines parallel machine scheduling with flow shop scheduling. In classical flow shop scheduling, jobs are processed following the same order of passage on machines placed in a series configuration, with only one machine in each stage. In a hybrid flow shop, at least one stage contains more than one machine. Jobs are assigned to only one machine from each stage [1].
In this paper, we study the hybrid flow shop scheduling problem that can be formally stated as follows. Consider a set J of n jobs. Each job $j \in J$ has two operations O_{1j} and O_{2j}. The jobs are to be scheduled in a two-stage hybrid flow-shop environment. The hybrid flow shop has m_1 parallel identical machines at stage 1 and m_2 parallel identical machines at stage 2. Each job must be processed by exactly one machine at each stage.
The first operation of a job O_{1j} must be entirely processed for p_{1j} time by exactly one machine at stage 1. The second operation is executed for p_{2j} time according to one of three possible scenarios.

- Scenario 1 : Operation is entirely processed at stage 2.
- Scenario 2 : Operation is entirely executed at stage 1 (by the same machine assigned to the first operation of the same job).
- Scenario 3 : Operation starts at stage 1 on the same machine which processed operation 1, then continues at stage 2.

Moreover, no part of operation 2 can be processed at stage 1 while a machine at stage 2 is available. Preemption is not allowed at either stage. Waiting times between stage 1 and stage 2 are not permitted either. In other terms, there is a no-wait condition. We suppose that transfer times are negligible. The objective is to minimize the maximum completion time; i.e., the makespan.
Denoted by $F2(Pm_1, Pm_2{-}miss)|no{-}wait, flex(1 \to 2)|C_{max}$ in tertiary notation, this problem is strongly NP-hard as proved in [2].

2 Branch-and-bound algorithm

A branch-and-bound algorithm is developed to solve the problem. Details of the branching strategy as well as the bounding strategy are presented below.

2.1 Dominance properties

We first introduce the following two dominance properties, which will be used in the branching scheme.

Theorem 1. *Let S_{idle} be an optimal solution with idle time at stage 1. An optimal solution $S_{no-idle}$ with no idle time at stage 1 can always be obtained from S_{idle}.*

Proof. We design an algorithm to eliminate idle times from S_{idle} and place the tasks at stage 1 in a compact fashion. The principle of the algorithm is to perform a series of left shifts and swaps in order to remove all idle periods at stage 1 without delaying any jobs' execution in the schedule. First, we can always shift the first operation to the left so that the gap is filled. Then, we attempt to left-shift the second operation too as to maintain the no-wait condition. There are 3 case scenarios depending on the second operation. Detailed proof with the analysis of all possible situations is omitted from this paper. □

Theorem 2. *Denote a_{1j} the duration of the second operation of job j at stage 1. At least m_2 jobs have $a_{1j} = 0$.*

Proof. Recall that a stage-2 machine is not allowed to be idle while a stage-1 machine is processing the second operation of a job. Therefore, at least m_2 jobs must have their second operation entirely processed at stage 2. □

2.2 Branching scheme

The root node at level $lv = 0$ corresponds to an empty schedule. Each node on level $lv > 0$ represents a partial schedule with lv scheduled jobs. From a node at a level lv, the couple (j, a_{1j}) is used for branching. j is the index of the last job added to the schedule. a_{1j} is the duration of its second operation at stage 1. a_{1j} takes $p_{2j} + 1$ possible values.
At each node, we decide which job to schedule next and how much of its second operation is processed at stage 1.
For example, (1,0) (2,3) means that job 1 is scheduled first with duration 0, then job 2 with duration 3.
At a leaf node, given the obtained stage-1 scheduling sequence and the duration a_{1j} of operation 2 at stage 1 for each job j, we can easily compute the starting and ending times of jobs at stage 1 (Each job is assigned to the first available stage-1 machine). Moreover, the no-wait constraint allows us to determine the starting and ending times of jobs at stage 2. Indeed, the starting time S_{2j} and the completion time C_{2j} of job j at stage 2 can be calculated as follows.

$$\begin{aligned} S_{2j} &= 0 \; if \; a_{1j} = p_{2j} \\ &= S_{1j} + p_{1j} + a_{1j} \; otherwise \end{aligned} \tag{1}$$

$$\begin{aligned} C_{2j} &= 0 \; if \; a_{1j} = p_{2j} \\ &= S_{1j} + p_{1j} + p_{2j} \; otherwise \end{aligned} \tag{2}$$

Consequently, we obtain intervals of starting and completion times at stage 2 $[S_{2j}, C_{2j}]$. At this point, the only remaining decision in our problem is the assignment of the intervals to stage-2 machines. The underlying problem is thus reduced to an interval scheduling problem, which is solvable in polynomial time by the algorithm in [3].

2.3 Lower bounds

We adapt the lower bounds given in [4] to compute lower bounds for each step of the scheduling process. A step represents the addition of one job to the schedule.
Denote by $\sigma J = \{j_{\sigma 1}, j_{\sigma 2}, ..., j_{\sigma n}\}$ a scheduling sequence of the jobs. Denote by $\bar{J}_j$ the set of unscheduled jobs remaining after scheduling jobs in $J_j = j_{\sigma 1}, j_{\sigma 2}, ..., j_{\sigma j}$.
The lower bounds are adapted as follows, where $L1_k$ is the current load of stage-1 machine M_{1k}.

$$LBA_1 = \min_{k \in M_1} L1_k + \max_{j \in \bar{J}_j}\{p_{1j} + p_{2j}\} \tag{3}$$

Denote by $\bar{p_{1j}}$ the stage-1 processing times of non-scheduled jobs sorted according to non-decreasing order and define $inactive_j$ as follows. $inactive_j = \{p_{1j} + a_{1j} : j \in J_j\} U \{\bar{p_{1j}} : j \in \bar{J}_j\}$

$$LBA_2 = \left\lceil \frac{1}{m_1 + m_2}\left(\sum_{j \in inactive_j : j=1}^{m_2} inactive_j + \sum_{j \in J} p_{1j} + \sum_{j \in J} p_{2j}\right)\right\rceil \tag{4}$$

$$LBA_3 = \left\lceil \frac{1}{m_1}\left(\sum_{j \in J_j}(p_{1j} + a_{1j}) + \sum_{j \in \bar{J}_j} p_{1j} + \sum_{j \in J : j=1}^{m_1} q_j\right)\right\rceil \tag{5}$$

where

$$\begin{aligned} q_j &= p_{2j} - a_{1j} \; if j \in J_j \\ &= p_{2j} \; if j \in \bar{J}_j \end{aligned}$$

and q_j are sorted in non-decreasing order.

2.4 Upper bounds

Four heuristics denoted by $HH0$, $LPT1$, $LPT2$, $LPT21$, are used to compute upper bounds. Each heuristic includes two steps as follows.

1. First, a sequencing step: The order of passage of the jobs on stage 1 is determined. Four orders are considered.
   - HH0: jobs follow the input sequence (random order).
   - LPT1: jobs are sorted in non-increasing order of the processing times of their first operations p_{1j}
   - LPT2: jobs are sorted in non-increasing order of the processing times of their second operations p_{2j}
   - LPT21: heuristic LPT2 is first applied. m_1 sets of jobs are obtained, each set includes the jobs scheduled on one machine. Then, the jobs in each set are sorted in non-increasing order of the processing times of their first operations p_{2j}
2. Second, an assignment step : the operations are placed onto machines in a greedy fashion.

3 Numerical testing and Results

The branch-and-bound algorithm is implemented in object-oriented C++. Tests are carried out on a machine with 32.0 GB RAM and Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, under Linux Operating System.
We experiment with the testbed presented in [4]. There are 6 classes of instances. Each class contains 60 instances, grouped in 4 sub-classes. Every sub-class represents a combination $n-m_1-m_2$ and contains 15 instances with different values of processing times p_{1j} and p_{2j}. Processing times are generated following a uniform distribution. The details of data instances are provided in [4].
We compare the results with those obtained in [4] using mathematical programming. Table 3 summarizes the numbers and percentages of instances solved to optimality by both methods. The branch-and-bound algorithm significantly outperforms the mathematical model.

	Number of optimal instances	Percentage of optimal instances
Branch-and-bound	125	35%
Mathematical model	9	3%

4 Conclusion

In this paper, we studied a two-stage hybrid flow shop scheduling problem with no wait and inter-stage flexibility. We developed a branch-and-bound algorithm that takes account of the specific problem structure. The implementations reveal that the suggested approach outperforms existing methods.

References

1 Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European journal of operational research, 205*(1), 1-18.

2 Zhong, W., & Shi, Y. (2018). Two-stage no-wait hybrid flowshop scheduling with inter-stage flexibility. *Journal of Combinatorial Optimization, 35*, 108-125.

3 Gupta, & Leung. (1979). An optimal solution for the channel-assignment problem. *IEEE Transactions on Computers, 100*(11), 807-810.

4 Azaiez, M. N., Gharbi, A., Kacem, I., Makhlouf, Y., & Masmoudi, M. (2022). Two-stage no-wait hybrid flow shop with inter-stage flexibility for operating room scheduling. *Computers & Industrial Engineering, 168*, 108040.

Quantum speed-ups for single-machine scheduling problems

Camille Grange[1,2], Eric Bourreau[1], Michael Poss[1] and Vincent T'Kindt[3]

1. Laboratory LIRMM, Montpellier, France
camille.grange@lirmm.fr, eric.bourreau@lirmm.fr, michael.poss@lirmm.fr

2. SNCF, Technology, Innovation and Group Projects Department, France

3. Laboratory LIFAT, Tours, France
tkindt@univ-tours.fr

Keywords : Quantum optimization, Grover search, Scheduling

1 Introduction

The interest in quantum computing to solve combinatorial optimization problems has been growing for several years in the operational research community. More precisely, two branches are distinguished. The first one relates to heuristics, which can be implemented on current noisy quantum computers because the quantum part can be made rather small, such as variational quantum algorithms [2]. The second branch relates to *exact* algorithms. Unlike the previous algorithms, it is impossible to implement them today but theoretical speed-ups have been proved for several types of problems and algorithms [6].

The most emblematic algorithm of this branch is Grover search [5], which achieves a quadratic speed-up when searching for a specific element in an unsorted table, where the complexity is computed as the number of queries of the table and done by an oracle. The authors of [4] use Grover search as a subroutine for a hybrid algorithm that finds with high probability the minimum of an unsorted table, leading to the algorithm known as Quantum Minimum Finding (QMF). Later, the authors of [1] combine QMF with dynamic programming to address NP-hard optimization problems. They apply their algorithm to vertex ordering problems, the Traveling Salesman Problem, and the Minimum Set Cover problem, among others. All these problems satisfy a specific property which implies that they can be solved by classical dynamic programming in $\mathcal{O}^*(c^n)$, where $\mathcal{O}^*$ is the usual asymptotic notation that ignores the polynomial factors, and c is usually not smaller than 2. The hybrid algorithm from [1] reduces the complexity to $\mathcal{O}^*(c_{\text{quant}}^n)$ for $c_{\text{quant}} < c$.

The purpose of this work is to adapt the seminal idea of [1] to NP-hard scheduling problems that satisfy the following property: for a given set of jobs J, the optimal solution for J is the best concatenation of optimal solutions for X and $J \setminus X$ among all $X \subset J$ such that $|X| = |J|/2$ (modulo an additive term that arises in the concatenation). This adaptation requires to introduce a pseudo-polynomial term in the state space of the dynamic programming as well as the aforementioned additive term. We thus obtain an extension of the Dynamic Programming Across the Subsets (DPAS) that many scheduling problems satisfy [7]. Herein, we focus on three single-machine scheduling problems ($1|\tilde{d}_j|\sum_j w_jC_j$, $1||\sum_j w_jT_j$ and $1|prec|\sum_j w_jC_j$) and show that our bounded-error hybrid quantum-classical algorithm improves the best-known classical exponential complexities, where in some cases a pseudo-polynomial factor $\sum p_j$ appears.

2 Dynamic programming for scheduling

Our problems of interest are scheduling problems where solutions are described by permutations of jobs in $[n] := \{1, \ldots, n\}$ for $n \in \mathbb{N}$, and that satisfy a certain property discussed below (see Property 2). Let $\mathcal{P}$ be the nominal problem we want to solve. We introduce next a family of problems related to $\mathcal{P}$ that will be instrumental in deriving the dynamic programming recursion. Let T be a set of non-negative integers containing 0. We define the family of problems indexed by $J \subseteq [n]$ and $t \in T$:

$$P(J,t): \qquad \min_{\pi \in \Pi(J,t)} f(\pi, J, t)\,, \tag{1}$$

where $\Pi(J,t) \subseteq \mathfrak{S}_J$ is the set of feasible permutations of J according to potential constraints and $f(., J, t)$ is the objective function. We note $\mathrm{OPT}[J,t]$ the optimal value of $P(J,t)$. With these notations, the nominal problem $\mathcal{P}$ is $P([n], 0)$. We suppose in what follows that $\mathcal{P}$ can be solved by DPAS (Dynamic Programming Across the Subsets). It means that the family of problems must satisfy the following DPAS property.

Property 1 (DPAS)**.** *Let $t_0 \in T$. Problem $P([n], t_0)$ can be solved by DPAS if there exists a function $h : 2^n \times [n] \times T \to \mathbb{R}$, computable in polynomial time, such that the following holds:*

$$\mathrm{OPT}[J, t_0] = \begin{cases} \min\limits_{j \in J} \Big\{ \mathrm{OPT}[J \setminus \{j\}, t_0] + h(J, j, t_0) \Big\} & \forall J \subseteq [n] \\ \mathrm{OPT}[\emptyset, t_0] = 0 \end{cases} \tag{2}$$

Notice the presence of the additional parameter t_0 in the above definitions, which is typically absent in the scheduling literature. The use of that extra parameter defined in Equation (1) and in Property 1 shall be necessary later when applying our hybrid algorithm. We can show that DPAS solves $\mathcal{P}$ in $\mathcal{O}^*(2^n)$.

In this paper, we consider a family of problems that not only satisfy Property 1 but also the Dichotomic DPAS property below.

Property 2 (Dichotomic DPAS)**.** *Let $t_0 \in T$. Problem $P([n], t_0)$ can be solved by Dichotomic DPAS if there exist three functions $t_1 : 2^n \times 2^n \times T \to T$, $t_2 : 2^n \times 2^n \times T \to T$ and $g : 2^n \times 2^n \times T \to \mathbb{R}$, computable in polynomial time, such that, for all $J \subseteq [n]$ of even cardinality:*

$$\mathrm{OPT}[J, t_0] = \min_{\substack{X \subseteq J \\ |X| = \frac{|J|}{2}}} \Big\{ \mathrm{OPT}[X, t_1(J, X, t_0)] + g(J, X, t_0) + \mathrm{OPT}[J \setminus X, t_2(J, X, t_0)] \Big\} \tag{3}$$

Notice that if $P(X,t)$ is infeasible, then by convention $\mathrm{OPT}[X,t] = +\infty$. Furthermore, differently from the previous recurrence (2), recurrence (3) now calls $\mathrm{OPT}[X', t']$ for t' that may be different than t_0. Thus, Dichotomic DPAS solves $\mathcal{P}$ in $\mathcal{O}^*(|T| \cdot C(n))$ where $C(n) = \omega(2^n)$.

Solving $\mathcal{P}$ using only Dichotomic DPAS is worse than using only DPAS. However, we describe in the next section a hybrid algorithm we call Quantum Dichotomic DPAS (Q-DDPAS) that improves the complexity of solving $\mathcal{P}$ by combining DPAS and Dichotomic DPAS with Grover search. Before introducing this algorithm, we illustrate the Dichotomic DPAS property on the scheduling problem with deadline constraints and minimization of the total weighted completion time.

Example 3 (Minimizing the total weighted completion time with deadlines)**.** *For each job $j \in [n]$, we are given a weight w_j, a processing time p_j, and a deadline $\tilde{d}_j$. We note $p(J) = \sum_{j \in J} p_j$ and $T = [\![0, p([n])]\!]$. For each $J \subseteq [n]$ and $t \in T$, we consider the problem $P(J,t)$ where*

$$\Pi(J,t) = \{\pi \in \mathfrak{S}_J \,|\, C_j(\pi) \leq \tilde{d}_j - t\,, \forall j \in J\}\,,$$

where C_j is the completion time of job j, and for $\pi \in \Pi(J,t)$:

$$f(\pi, J, t) = \sum_{j \in J} w_j (C_j(\pi) + t)\,.$$

$P(J,t)$ represents the problem of finding the best feasible solution for jobs in J supposing that starting time is t, and not 0 as usual. Our problem of interest is $\mathcal{P} = P([n], 0)$, often referred to as $1|\tilde{d}_j|\sum_j w_j C_j$ in the scheduling literature. It can be solved by DPAS with Equation (2) *for*

$$\forall J \subseteq [n], \forall j \in J, \forall t \in T, \quad h(J,j,t) = \begin{cases} w_j(p(J)+t) & \text{if } \tilde{d}_j \geq p(J)+t \\ +\infty & \text{else} \end{cases}$$

where the computation of h is polynomial (linear). This family of problems also satisfies the Dichotomic DPAS property. Indeed, Equation (3) *is valid for the following functions:*

$$\forall X \subseteq J \subseteq [n] \text{ s.t. } |X| = |J|/2, \forall t \in T, \quad \begin{aligned} t_1(J,X,t) &= t \\ t_2(J,X,t) &= t + p(X) \\ g(J,X,t) &= 0 \end{aligned}$$

We study two other one-machine scheduling problems and can show that they also satisfy the Dichotomic DPAS property. Specifically, we tackle $1||\sum_j T_j$ (for which $T = [\![0, p([n])]\!]$) and $1|prec|\sum_j w_j C_j$ (for which $T = \{0\}$). Thus, we have three scheduling problems that can be solved by our hybrid algorithm described below.

3 Hybrid quantum-classical algorithm

In this section, we introduce a hybrid bounded-error algorithm called Quantum Dichotomic DPAS (Q-DDPAS) that solves scheduling problems satisfying the Dichotomic DPAS property. It is an adaptation of the algorithm in [1]. Our algorithm is expressed in the gate-based quantum computing model and assumes to have random access to quantum memory (QRAM). We underline that this latter assumption is strong because QRAM is not available on current universal quantum hardware and is not expected to be so in the near future. Before describing Q-DDPAS, let us introduce the Quantum Minimum Finding (QMF) subroutine.

Definition 4 (Quantum Minimum Finding)**.** *Let $f : [n] \to \mathbb{Z}$ be a function. The Quantum Minimum Finding (QMF) algorithm [4] computes with high probability the minimum value of f. Its complexity is $\mathcal{O}\left(\sqrt{n} \cdot C_f(n)\right)$, where $\mathcal{O}(C_f(n))$ is the complexity of computing one value of f.*

Without loss of generality, we assume that 4 divides n. The hybrid quantum-classical algorithm Q-DDPAS consists of two steps. First, we compute classically by DPAS the optimal values of all subproblems scheduling with $\frac{n}{4}$ jobs. Second, we call recursively two times QMF to find optimal values of subproblems scheduling with $\frac{n}{2}$ jobs and eventually with n jobs (corresponding to the initial problem). Specifically,

1. **Classical part.** For each $X \subset [n]$ of size $n/4$ and all $t \in T$, compute $\text{OPT}[X,t]$ by classical DPAS. Store the results in the QRAM.
2. **Quantum part.**

(a) Apply QMF to find OPT$[[n], 0]$ with Equation (3)

(b) To get values for the QMF above (OPT$[J, t]$ for $J \subset [n]$ of size $n/2$ and $t \in T$), apply QMF with Equation (3)

(c) To get values for the QMF above (OPT$[X, t']$ for $X \subset [n]$ of size $n/4$ and $t' \in T$), get them on the QRAM

Theorem 5. *The bounded-error Q-DDPAS algorithm solves $\mathcal{P}$ in $\mathcal{O}^*(|T| \cdot 1.754^n)$.*

We summarize in Table 1 the complexities of solving the scheduling problems studied in Section 2 with Q-DDPAS and compare them with the complexities of the best-known current classical algorithms. Q-DDPAS improves the complexity of the exponent but sometimes at the cost of a pseudo-polynomial factor.

Problem	Quantum-DDPAS	Best classical algorithm
$1\|\tilde{d_j}\|\sum w_j C_j$	$\mathcal{O}^*\left(\sum p_j \cdot 1.754^n\right)$	$\mathcal{O}^*(2^n)$ (Classical DPAS)
$1\|\|\sum w_j T_j$	$\mathcal{O}^*\left(\sum p_j \cdot 1.754^n\right)$	$\mathcal{O}^*(2^n)$ (Classical DPAS)
$1\|prec\|\sum w_j C_j$	$\mathcal{O}^*\left(1.754^n\right)$	$\mathcal{O}^*((2-\epsilon)^n)$, for small ϵ [3]

Table 1: Comparison of complexities between our hybrid algorithm Quantum-DDPAS and the best-known classical algorithm.

References

[1] A. Ambainis, K. Balodis, J. Iraids, M. Kokainis, K. Prūsis, and J. Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1783–1793. SIAM, 2019.

[2] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[3] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Scheduling partially ordered jobs faster than 2^n. *Algorithmica*, 68:692–714, 2014.

[4] C. Durr and P. Hoyer. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*, 1996.

[5] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[6] G. Nannicini. Fast quantum subroutines for the simplex method. *Operations Research*, 2022. In press.

[7] V. T'kindt, F. Della Croce, and M. Liedloff. Moderate exponential-time algorithms for scheduling problems. *4OR*, pages 1–34, 2022.

Cutting Plane and Column Generation Algorithms for the Survivable Constrained-Routing and Spectrum Assignment Problem

Ibrahima Diarrassouba[1], Youssouf Hadhbi[2] and A. Ridha Mahjoub[3,4]

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

2. LIMOS Laboratory, UMR CNRS 7024, INP and Clermont Auvergne University
Campus Universitaire des Cézeaux, 1 rue de la Chebarde, 63178 Aubière Cedex, France
Email : youssouf.hadhbi@uca.fr

3. Departement of Statistics and Operations Research, College of Science, Kuwait University, Kuwait
Email : ali.mahjoub@ku.edu.kw

4. LAMSADE Laboratory, UMR CNRS 7024, Paris-Dauphine PSL University
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
Email : ridha.mahjoub@lamsade.dauphine.fr

Keywords: optical network, survivability, network design, integer programming, polyhedron, valid inequalities, separation, branch-and-cut, column generation, branch-and-cut-and-price.

1 Introduction

Nowadays, optical transport networks are facing a serious challenge related to the exponential growth of global communication services and networking. To deal with this, a new architecture called Spectrally Flexible Optical Network (SFON) has been presented as promising technology because of its flexibility, scalability, efficiency, reliability, and survivability compared with the traditional FixedGrid Optical Wavelength Division Multiplexing (WDM) networks. Addressing the problem of survivability in SFONs is of great interest. In this work, we focus on a variant of the well known Routing and Spectrum Assignment problem (RSA), namely the Survivable Constrained-Routing and Spectrum Assignment (SC-RSA). Here we consider the dedicated protection routing scheme such that a backup path and capacity are provisioned in advance for each traffic demand. The SC-RSA is NP-hard and more challenging than the RSA which is already known to be NP-hard [2]. It can be stated as follows. Consider an optical spectrum of $\bar{s} \in \mathbb{Z}_+$ available contiguous frequency slots, denoted by $\mathbb{S} = \{1, \dots, \bar{s}\}$. A survivable SFON topology can be represented by an undirected, loopless, and two-edge-connected graph $G = (V, E)$, with lengths $\ell_e \in \mathbb{R}_+$ (in kms) and costs $c_e \in \mathbb{R}_+$ on the edges $e \in E$ such that each fiber-link $e \in E$ is associated with $\bar{s} \in \mathbb{N}_+$ contiguous slots numbered from 1 to $\bar{s}$. Let K be a set of non-splittable traffic demands. Each demand $k \in K$ has an origin node $o_k \in V$, a destination node $d_k \in V \setminus \{o_k\}$, a slot-width $w_k \in \mathbb{Z}_+$, and a transmission-reach $\bar{\ell}_k \in \mathbb{R}_+$ (in kms).
The SC-RSA consists in determining for each demand $k \in K$ two edge-disjoint (o_k,d_k)-path respectively called primary path and backup path (*2-connectivity and non-splittable demands constraints*) with total length smaller than $\bar{\ell}_k$ (*transmission-reach constraint*). Moreover, for each demand $k \in K$ two intervals S_k and S'_k of w_k consecutive frequency slots should be selected along its primary path and backup path respectively (*contiguity and continuity constraints*).

If two demands have their primary (backup) paths non disjoint, than they cannot share any slot over the shared edges (*non-overlapping constraints*).
The problem aims at minimizing the total cost of the edges selected for routing the demands K. The main purpose of this work is to provide a polyhedral analysis of the SC-RSA problem and using this, devises cutting plane and column generation algorithms for solving the problem. For this, we first give an integer linear programming formulation called *cut formulation*, and investigate the associated polytope. We then identify several classes of valid inequalities and propose separation routines. Using these results, we develop a Branch-and-Cut (B&C) algorithm for the problem. Moreover, we introduce an extended integer linear programming formulation, called *path formulation.* We devise a column generation algorithm to solve its linear relaxation. Based on this, we also develop a Branch-and-Cut-and-Price (B&C&P) algorithm for the problem.

2 Cut Formulation and Branch-and-Cut Algorithm

First, we introduce an integer linear programming formulation for the SC-RSA problem based on the so-called cut inequalities. For $k \in K$ and $e \in E$, let x_e^k (resp. b_e^k) be a binary variable which takes 1 if the primary path (resp. backup path) of demand k goes through the edge e and 0 if not. For $k \in K$ and $s \in \mathbb{S}$, we consider the binary variable z_s^k (resp. v_s^k) which takes 1 if slot s is the last slot allocated to the primary routing (resp. backup routing) of demand k and 0 if not. For each $k \in K$, $e \in E$ and $s \in \mathbb{S}$, let $t_{e,s}^k$ be a binary variable which takes 1 if the slot s is the last slot allocated to demand k over edge e by its primary or backup path and 0 if not. For each $e \in E$, let u_e be a design variable which takes 1 if edge e is selected in the routing of the demands, and 0 if not. The SC-RSA problem is equivalent to the following ILP

$$\min \sum_{e \in E} c_e u_e, \tag{1}$$

$$\sum_{e \in \delta(X)} x_e^k \geq 1, \forall k \in K, \forall X \subset V \text{ such that } |X \cap \{o_k, d_k\}| = 1, \tag{2}$$

$$\sum_{e \in \delta(X)} b_e^k \geq 1, \forall k \in K, \forall X \subset V \text{ such that } |X \cap \{o_k, d_k\}| = 1, \tag{3}$$

$$x_e^k + b_e^k \leq u_e, \forall k \in K, \forall e \in E, \tag{4}$$

$$\sum_{e \in E} l_e x_e^k \leq \bar{\ell}_k, \forall k \in K, \tag{5}$$

$$\sum_{e \in E} l_e b_e^k \leq \bar{\ell}_k, \forall k \in K, \tag{6}$$

$$\sum_{s=1}^{w_k - 1} (z_s^k + v_s^k + \sum_{e \in E} t_{e,s}^k) = 0, \forall k \in K, \tag{7}$$

$$\sum_{s=w_k}^{\bar{s}} z_s^k = 1, \forall k \in K, \tag{8}$$

$$\sum_{s=w_k}^{\bar{s}} v_s^k = 1, \forall k \in K, \tag{9}$$

$$x_e^k + z_s^k - 1 \leq t_{e,s}^k, \forall k \in K, \forall e \in E, \forall s \in \mathbb{S}, \tag{10}$$

$$b_e^k + v_s^k - 1 \leq t_{e,s}^k, \forall k \in K, \forall e \in E, \forall s \in \mathbb{S}, \tag{11}$$

$$\sum_{s=w_k}^{\bar{s}} t^k_{e_s} - x^k_e - b^k_e = 0, \forall k \in K, \forall e \in E, \quad (12)$$

$$\sum_{k\in K} \sum_{s'=s}^{\min(s+w_k-1,\bar{s})} t^k_{e,s'} \leq u_e, \forall e \in E, \forall s \in \mathbb{S}, \quad (13)$$

$$x^k_e, b^k_e \geq 0, \forall k \in K, \forall e \in E, \quad (14)$$

$$z^k_s, v^k_s \geq 0, \forall k \in K, \forall s \in \mathbb{S}, \quad (15)$$

$$t^k_{e,s} \geq 0, \forall k \in K, \forall e \in E, \forall s \in \mathbb{S}, \quad (16)$$

$$0 \leq u_e \leq 1, \forall e \in E, \quad (17)$$

$$x^k_e, b^k_e \in \{0,1\}, \forall k \in K, \forall e \in E, \quad (18)$$

$$z^k_s, v^k_s \in \{0,1\}, \forall k \in K, \forall s \in \mathbb{S}, \quad (19)$$

$$t^k_{e,s} \in \{0,1\}, \forall k \in K, \forall e \in E, \forall s \in \mathbb{S}. \quad (20)$$

Inequalities (2) (resp. (3)) ensure that there is a primary (o_k, d_k)-path (resp. backup (o_k, d_k)-path) for each demand k. Inequalities (4) express the 2-connectivity constraints. Inequalities (5) and (6) express the length limit on the primary and backup paths. Equations (7) express the fact that a demand k cannot use slot $s \leq w_k - 1$ as the last-slot. Inequalities (8) (resp. (9)) ensure that exactly one slot $s \in \{w_k, \ldots, \bar{s}\}$ must be assigned to demand k as last-slot along its primary path (resp. backup path). Inequalities (10), (11) and (12) capture the fact that when a slot s is said to be used as last-slot by demand k over an edge e if slot s is the last-slot assigned for demand k along its primary or backup path passed through edge e. Inequalities (13) express the non-overlapping constraints. They ensure that every slot s over edge e can be assigned to at most one demand $k \in K$. Inequalities (14)-(17) are the trivial inequalities, and constraints (17)-(20) are the integrality constraints.

This formulation contains a polynomial number of variables, and an exponential number of constraints that are separable in polynomial time using network flow algorithms.

We describe several classes of valid inequalities for the associated polytope. Some of these inequalities are obtained by using conflict graphs related to the problem : clique inequalities and odd-cycle inequalities. Further inequalities are induced by cuts and covers related to some capacity constraints. We further study the related separation problems and devise separation routines for these inequalities. Based on these results, we devise a Branch-and-Cut algorithm to solve the problem along with a computational study will be presented.

3 Extended Formulation and Branch-and-Cut-and-Price Algorithm

Next, we introduce an extended formulation for the problem based on the so-called *path* variables. For this, we consider for each $k \in K$, $p \in P^k$ and $s \in \mathbb{S}$, a binary variable $y^k_{p,s}$ (resp. $f^k_{p,s}$) which takes 1 if slot s is the last slot allocated along the primary path p (resp. backup path) of demand k and 0 if not, where P^k denotes the set of all feasible (o_k,d_k)-paths in G such that the total length of each path $p \in P^k$ does not exceed $\bar{\ell}_k$. This formulation shares with the cut formulation the variables $x^k_e, b^k_e, t^k_{e,s}, u_e$, constraints (4), (5), (6), (12), (13), (14), (16), (17), and objective function (1). We also consider some additional constraints related to the path variables as follows.

$$\sum_{e\in E}\sum_{s=1}^{w_k-1} t^k_{e,s} + \sum_{p\in P^k}\sum_{s=1}^{w_k-1}(y^k_{p,s}+f^k_{p,s}) = 0, \forall k \in K, \quad (21)$$

$$\sum_{p\in P^k}\sum_{s=w_k}^{\bar{s}} y^k_{p,s} = 1, \forall k \in K, \quad (22)$$

$$\sum_{p\in P^k}\sum_{s=w_k}^{\bar{s}} f^k_{p,s} = 1, \forall k \in K, \quad (23)$$

$$\sum_{p\in P^k(e)}\sum_{s=w_k}^{\bar{s}} y^k_{p,s} = x^k_e, \forall k \in K, \forall e \in E, \quad (24)$$

$$\sum_{p\in P^k(e)}\sum_{s=w_k}^{\bar{s}} f^k_{p,s} = b^k_e, \forall k \in K, \forall e \in E, \quad (25)$$

$$\sum_{p\in P^k(e)} (y^k_{p,s}+f^k_{p,s}) = t^k_{e,s}, \forall k \in K, \forall e \in E, \forall s \in \mathbb{S}. \quad (26)$$

where $P^k(e)$ denotes set of all feasible (o_k,d_k) paths of demand k going through edge e in G. Equations (21) force each slot $s \in \{1, ..., w_k - 1\}$ to not be used as last-slot for demand k along its primary and backup paths. Inequalities (22) ensure that exactly one slot $s \in \{w_k, \dots, \bar{s}\}$ is assigned as last-slot for demand k, and exactly one single primary path from P^k is allocated by each demand $k \in K$ (same thing for backup path as shown by inequalities (23)). Equations (24) (resp. (25)) and (26) express the use of an edge $e \in E$ by demand k, and the assignment of slot s as last-slot by demand k over an edge e.

This formulation contains a huge number of variables. We, therefore, use a column generation algorithm to solve its linear relaxation. The pricing problem consists in solving the Resource Constrained Shortest Path (RCSP) problem which is well known to be weakly NP-hard [1]. For this, we propose a pseudo-polynomial time based dynamic programming algorithm. This consists in finding the minimum-cost path for each demand $k \in K$ and slot $s \in \{w_k, ..., \bar{s}\}$ while satisfying the transmission-reach constraint. The complexity of this algorithm is bounded by $\mathcal{O}(|E|.\bar{\ell}_k)$ for each demand $k \in K$.

Notice that all the different valid inequalities of the cut formulation, are still valid for the path formulation. Based on these results, we devise a Branch-and-Cut-and-Price algorithm for solving the problem along with a computational study will be presented.

Finally, we will present a deep comparative study between the two approaches B&C and B&C&P using large-scale instances. It would be interesting to further investigate another scheme of survivability based on the shared backup path protection technique such that multiple backup paths can share frequency slots if their corresponding primary paths do not share any edge. This means that the corresponding demands do not share any common failures.

References

[1] Moshe Dror. *Note on the Complexity of the Shortest Path Models for Column Generation in VRPTW.* Journal of Operations Research 1994, pp. 977-978.

[2] Mirosław Klinkowski and Krzysztof Walkowiak. *Offline RSA Algorithms for Elastic Optical Networks with Dedicated Path Protection Consideration.* 4th International Workshop on Reliable Networks Design and Modeling 2012, pp. 1-7.

Smoothed analysis of the simplex method

Sophie Huiberts[1]

1. Columbia University -New york city
sophie.huiberts@gmail.com

Mots-clefs : Linear programming, simplex method

1 Abstract

Explaining why the simplex method is fast in practice, despite it taking exponential time in the theoretical worst case, continues to be a challenge. Smoothed analysis is a paradigm for addressing this question. During my talk I will present recent progress in the smoothed complexity of the simplex method, discussing both upper and lower bounds.

The quickest route problem

Jean François Maurras[1]

1. Université Aix Marseille
jf.maurras@gmail.com

Mots-clefs : Shorest path, quickest path

1 Abstract

The quickest route problem is different of the shortest path problem. It is the problem to which is confronted the driver who wants to go from the point A to the point B with a car. However, if for the thermic cars the solutions of shortest path are a good approximation of the ones of quickest route, for an electric car, to use the solution of the shortest path problem without care, can yield to aberrations.

Formulation étendue pour le polytope des co-2-plexes

Alexandre Dupont-Bouillard, Pierre Fouilhoux,
Roland Grappe, Mathieu Lacroix

LIPN, Université Sorbonne Paris Nord, CNRS UMR 7030, Villetaneuse, France

Mots-clefs : polytope des co-2-plexes, graphes contraction parfait, formulation étendue

Introduction

Dans un graphe $G = (V, E)$ simple non orienté, un *stable* (resp. *co-2-plexe*) est un ensemble de sommets dont le sous-graphe induit contient seulement des sommets de degré au plus 0 (resp. 1). Une *clique* (resp. *2-plexe*) est un ensemble de sommets induisant un sous-graphe dont le complémentaire est un stable (resp. co-2-plexe).Etant donné un poids associé aux sommets de G, le problème du co-2-plexe maximal consiste à déterminer un co-2-plexe maximisant la somme des poids de ses sommets. Trouver un co-2-plexe maximal est un problème NP-difficile qui se rencontre dans la littérature pour l'analyse de communautés dans les réseaux sociaux [1, 2]. Dans [3], il est proposé de résoudre le problème en recherchant, pour un chacun des sommets, le co-2-plexe de poids maximal le contenant, impliquant de résoudre |V| fois le problème sur différents sous-graphes induits. De manière analogue à la formulation PLNE par les contraintes de cliques pour le problème de stable de poids maximum, une formulation PLNE par les contraintes de 2-plexes existe pour le problème de co-2-plexe de poids maximum. McClosky et Hicks ont prouvé que la relaxation linéaire de cette formulation est entière seulement pour les 2-plexes, chemins, trous de taille multiple de 3, et les co-2-plexes [4].

1 Fromulation étendue

Dans cette première partie, nous introduisons le *total augmented graphe* de G, noté $\widetilde{T}(G)$, dans lequel l'ensemble des stables est en bijection avec l'ensemble des co-2-plexes de G. Cette bijection nous permet d'obtenir des résultats sur le problème du co-2-plexe maximal à partir de résultats connus pour le problème du stable de poids maximal. Nous obtenons en particulier une formulation étendue compacte pour le polytope des co-2-plexes d'un graphe *cordal i.e.* d'un graphe n'ayant aucun trou de taille 4 ou plus comme sous graphe induit :

Théorème 1. *Un graphe G est chordal si et seulement son polytope des co-2-plexes et décrit par la formulation suivante :*

$$\begin{cases} x(K) - y(E(K)) & \leq & 1 & \forall K \text{ clique de } G \\ y(\delta_u) & \leq & x_u & \forall u \in V \\ y_e & \leq & 0 & \forall e \in E \\ -x_u & \leq & 0 & \forall u \in V \end{cases}$$

Le Théorème 1 prouve la polynomialité du problème de co-2-plexe maximal sur les graphes cordaux (et aussi celle de la recherche d'un co-2-plexe avec des poids à la fois sur les sommets et les arêtes grâce aux variables additionnelles) .

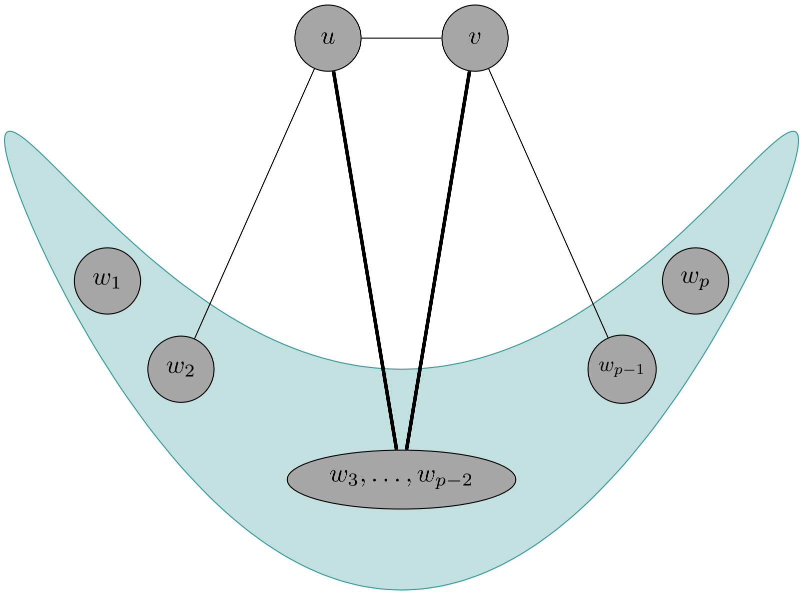


FIGURE 1 – Expanded antihole

2 Graphes contractions parfaits

Dans cette seconde partie, nous investigons une nouvelle classe de graphes que nous appellons : *les graphes contractions parfaits* et définie comme l'ensemble des graphes parfaits, *i.e.* des graphes ne contenant aucun trou ni antitrou impair, qui restent parfaits après contraction de n'importe quel ensemble d'arêtes, pour $F \subseteq E$, nous noterons G/F le graphe obtenue à partir de G en contractant F. Nous proposons aussi plusieurs caractérisations de cette classe de graphes, dont le Théorème 2 donne la plus surprenante : comme les graphes parfait sont reconnaissables en temps polynomial [5], on déduit directement un algorithme de reconnaissance des graphes parfaits qui restent parfait par contraction de n'importe quelle arête, cependant, détecter les graphes qui restent parfait après contraction de n'importe quel ensemble d'arêtes semble plus compliqué, ce théorème permettra d'affirmer que ces deux classes de graphes coincident, et donc, que ces deux problèmes de reconnaissance sont alors non seulement équivalents mais identiques.

Théorème 2. *Soit G un graphe parfait, G/F est parfait pour tout $F \subseteq E$ si et seulement si G/e est parfait pour tout $e \in E$*

Nous définissons aussi une nouvelle structure de graphes, les *expanded antihole* 1 ; c'est un ensemble de sommets dont le sous graphe induit est composé d'un anti chemin pair $(w_1, \ldots, w_p)$ et d'une arête uv dont les extrémités sont respectivement adjacentes à $(w_2, \ldots, w_{p-2})$ et $(w_3, \ldots, w_{p-1})$. Le théorème 2 nous permet alors de réduire l'ensemble des candidats potentiels pour une caractérisation par sous graphes induits interdits jusqu'à obtenir le théorème suivant :

Théorème 3. *Un graphe est contraction parfait si et seulement s'il ne contient ni trou de taille 5 ou plus, ni expanded antihole, ni anti trou impair.*

On prouve aussi que cette caractérisation est minimale au sens de l'inclusion. Le problème du stable de poids maximum étant polynomial dans les graphes parfaits, le Théorème 4 implique que le problème du co-2-plexe maximal dans les graphes contraction parfaits est polynomial. Comme il existe une description du polytope des stables dans les graphes parfaits définie à partir des contraintes de cliques du graphe [6], le théorème 4 permet également d'obtenir une formulation étendue non compacte du polytope des co-2-plexes d'un graphe contraction parfait.

Théorème 4. *Un graphe est contraction parfait si et seulement si son total augmented graphe est parfait.*

3 Relations avec d'autres classes de graphes

En troisième et dernière partie nous présentons des liens entre le total augmented graphe et d'autres familles de graphes connus telles que les *splits*, *i.e.* les graphes dont l'ensemble de sommets est partitionné en une clique et un stable ou encore les *trivialement parfaits*, *i.e.* les graphes ne possédant aucun P_4 ni C_4 induits.

Théorème 5. *Soit G un graphe simple non orienté :*
— *G est split si et seulement si $\widetilde{T}(G)$ l'est*
— *G est trivialement parfait si et seulement si $\widetilde{T}(G)$ l'est*
— *G est cordal si et seulement si $\widetilde{T}(G)$ l'est.*

Notons que ces trois familles de graphes sont contractions parfaites et que cette inclusion est stricte. Ces familles de graphes sont aussi stables par contraction d'une arête.

Références

[1] Timo Gschwind, Stefan Irnich, Fabio Furini, and Roberto Calvo. A branch-and-price framework for decomposing graphs into relaxed cliques. *INFORMS Journal on Computing*, 33, 11 2020.

[2] Balabhaskar Balasundaram, Sergiy Butenko, and Illya V. Hicks. Clique relaxations in social network analysis : The maximum k-plex problem. *Operations Research*, 59(1) :133–142, 2011.

[3] Zhuqi Miao and Balabhaskar Balasundaram. Approaches for finding cohesive subgroups in large-scale social networks via maximum k-plex detection. *Networks*, 69, 2017.

[4] Benjamin McClosky and Illya V. Hicks. The co-2-plex polytope and integral systems. *Siam Journal on Discrete Mathematics*, 23 :1135–1148, 2009.

[5] Gérard Cornuéjols, Xinming Liu, and Kristina Vuskovic. A polynomial algorithm for recognizing perfect graphs. *Annual Symposium on Foundations of Computer Science*, 2003 :20 – 27, 11 2003.

[6] Chudnovsky Maria, Robertson Neil, Seymour Paul, and Thomas Robin. The strong perfect graph theorem. *Annals of Mathematics.*

Formulations linéaires pour le problème d'isomorphisme de sous graphes non induits

E. de Gastines[1] et A. Knippel[1]

1. Laboratoire lmi, INSA Rouen Normandie
685 Avenue de l'Université, 76800 Saint-Etienne-du-Rouvray, France
(etienne.mace_ de_ gastines/arnaud.knippel)@insa-rouen.fr

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

1 Introduction

Soit P un graphe motif et T un graphe cible. Le problème de l'isomorphisme de sous-graphes non-induits consiste à déterminer si l'on peut trouver le motif P dans le graphe T, c'est à dire s'il existe un graphe partiel de T isomorphe à P. Plus formellement, le problème de l'isomorphisme de sous-graphes non-induits se ramène à déterminer s'il existe une injection $\varphi : V_P \to V_T$ telle que pour tout $u_1, u_2 \in V_P$, $u_1u_2 \in A_T \implies \varphi(u_1)\varphi(u_2) \in A_T$. Ce problème possède des applications dans le domaine de la reconnaissance de forme.

2 Précédentes formulations

Une première formulation PLNE a été introduite par Le Bodic et Knippel en 2008 pour le problème de l'isomorphisme de sous-graphes inexact [1, 2]. Ils introduisent les variables binaires $x_{u,v}$ pour tout $u \in V_P, v \in V_T$ qui ont pour valeur 1 si u est associé à v, 0 sinon et les variable binaires $y_{u_1u_2,v_1v_2}$ pour tout $u_1u_2 \in A_P, v_1v_2 \in A_T$, qui sont une linéarisation du produit $x_{u_1,v_1}x_{u_2v_2}$. La variable $y_{u_1u_2,v_1v_2}$ vaut 1 si l'arête u_1u_2 est associée à l'arête v_1v_2.

$$
(\mathbf{LBKl}) \begin{cases}
\min \displaystyle\sum_{u\in V_P, v\in V_T} d_1(u,v)x_{u,v} + \sum_{u_1u_2\in A_P, v_1v_2\in A_T} d_2(u_1u_2,v_1v_2)y_{u_1v_1,u_2,v_2} & (1)\\
\text{tq} \displaystyle\sum_{v\in V_T} x_{u,v} = 1, \quad \forall u \in V_P & (2)\\
\displaystyle\sum_{u\in V_P} x_{u,v} \le 1, \quad \forall v \in V_T & (3)\\
\displaystyle\sum_{v_1v_2\in A_T} y_{u_1u_2,v_1v_2} = 1, \quad \forall u_1u_2 \in A_P & (4)\\
y_{u_1u_2,v_1v_2} \le x_{u_1,v_1} \quad \forall u_1u_2 \in A_P, v_1v_2 \in A_T & (5)\\
y_{u_1u_2,v_1v_2} \le x_{u_2,v_2} \quad \forall u_1u_2 \in A_P, v_1v_2 \in A_T & (6)\\
y_{u_1u_2,v_1v_2} \ge x_{u_1,v_1} + x_{u_2,v_2} - 1 \quad \forall u_1u_2 \in A_P, v_1v_2 \in A_T & (7)\\
x_{u,v} \in \{0,1\} \quad \forall u \in V_P, \forall v \in V_T & (8)
\end{cases}
$$

Dans le cas du problème de décision, le choix de la fonction objectif n'a pas d'importance, et l'on peut prendre $d_1 = d_2 = 0$.

Ils proposent ensuite une meilleure formulation :

$$
(\mathbf{LBK})\begin{cases}
\min \displaystyle\sum_{u\in V_P, v\in V_T} d_1(u,v)x_{u,v} + \sum_{u_1u_2\in A_P, v_1v_2\in A_T} d_2(u_1u_2, v_1v_2)y_{u_1v_1,u_2,v_2} & (9)\\
\text{tq} \displaystyle\sum_{v\in V_T} x_{u,v} = 1, \quad \forall u \in V_P & (10)\\
\displaystyle\sum_{u\in V_P} x_{u,v} \leq 1, \quad \forall v \in V_T & (11)\\
\displaystyle\sum_{v_1v_2\in A_T} y_{u_1u_2,v_1v_2} = 1, \quad \forall u_1u_2 \in A_P & (12)\\
\displaystyle\sum_{v_2\in N^+(v_1)} y_{u_1u_2,v_1v_2} = x_{u_1,v_1} \quad \forall u_1u_2 \in A_P, v_1 \in V_T & (13)\\
\displaystyle\sum_{v_1\in N^-(v_2)} y_{u_1u_2,v_1v_2} = x_{u_2,v_2} \quad \forall u_1u_2 \in A_P, v_2 \in V_T & (14)\\
x_{u,v} \in \{0,1\} \quad \forall u \in V_P, \forall v \in V_T & (15)
\end{cases}
$$

Nous proposons de renforcer cette formulation en ajoutant une nouvelle famille de contraintes :

$$
\sum_{u_2\in N^+(u_1)} y_{u_1u_2,v_1v_2} \leq x_{u_1,v_1} \quad \forall u_1 \in V_P, v_1v_2 \in A_T
$$

Nous nommerons cette nouvelle formulation **LBK+**

3 Nouvelles formulations

En s'inspirant de la formulation de Knippel et Le Bodic, nous avons pu proposer la formulation suivante, qui n'utilise que les variables x (dans la mesure où nous n'utilisons pas de fonction coût sur les associations d'arêtes) :

$$
(\mathbf{FR})\begin{cases}
\min \displaystyle\sum_{u\in V_P, v\in V_T} d_1(u,v)x_{u,v} & (16)\\
\text{tq} \displaystyle\sum_{v\in V_T} x_{u,v} = 1, \quad \forall u \in V_P & (17)\\
\displaystyle\sum_{u\in V_P} x_{u,v} \leq 1, \quad \forall v \in V_T & (18)\\
x_{u_1,v_1} \leq \displaystyle\sum_{v_2\in N^+(v_1)} x_{u_2,v_2}, \quad \forall u_1u_2 \in E_P, v_1 \in V_T & (19)\\
x_{u_2,v_2} \leq \displaystyle\sum_{v_1\in N^-(v_2)} x_{u_1,v_1}, \quad \forall u_1u_2 \in E_P, v_2 \in V_T & (20)\\
x_{u,v} \in \{0,1\} \quad \forall u \in V_P, \forall v \in V_T & (21)
\end{cases}
$$

Il est possible d'obtenir une meilleure formulation en s'inspirant du polytope d'isomorphisme de graphes [4] :

$$
(\mathbf{FC}) \begin{cases}
\min \displaystyle\sum_{u\in V_P, v\in V_T} d_1(u,v) x_{u,v} & (22)\\
\text{tq} \displaystyle\sum_{v\in V_T} x_{u,v} = 1, \quad \forall u \in V_P & (23)\\
\displaystyle\sum_{u\in V_P} x_{u,v} = 1, \quad \forall v \in V_T & (24)\\
\displaystyle\sum_{u_1\in N^-(u_2)} x_{u_1,v_1} = \sum_{v_2\in N^+(v_1)} x_{u_2,v_2}, \quad \forall u_2 \in V_P, v_1 \in V_T & (25)\\
\displaystyle\sum_{u_2\in N^+(u_1)} x_{u_2,v_2} = \sum_{v_1\in N^-(v_2)} x_{u_1,v_1}, \quad \forall u_1 \in V_P, v_2 \in V_T & (26)\\
x_{u,v} \in \{0,1\} \quad \forall u \in V_P, \forall v \in V_T & (27)
\end{cases}
$$

Parmi les familles de coupes que nous avons découvertes, nous retenons les coupes de multi-adjacence :

$$\sum_{v_2\in K} x_{u_2,v_2} \leq \sum_{v_1\in N^-(K)} x_{u_1,v_1} \quad \forall (u_1,u_2) \in A_P, K \subset V_T$$

Nous donnons un algorithme de séparation polynomial, et montrons que la formulation **FR** augmentée des coupes de multi-adjacence est obtenue par la restriction de la formulation **LBK** aux variables x.

Nous comparons les relaxations des formulations proposées afin d'établir le poset suivant :

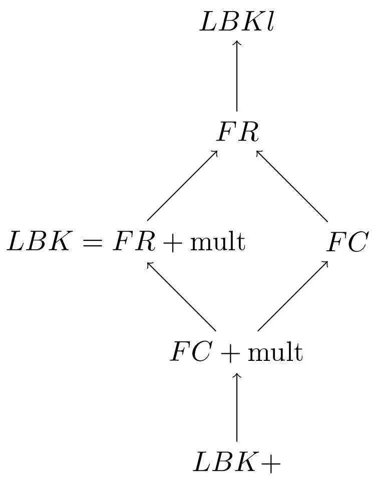


FIGURE 1 – Comparaison des formulations - $F_1 \rightarrow F_2$ si le polytope relaxé de F_1 (restreint à x) est inclus dans celui de F_2
"mult" désigne l'addition des contraintes de multi-adjacence

Nous comparons la formulation **LBK+** aux processus de filtrage des degrés itérés [5], qui permet de réduire le nombre d'affectations de sommets à considérer. Ce processus consiste à comparer les degrés des sommets pour réduire le nombre d'appariements à considérer. Suite à la déduction de l'incompatibilité de certaines paires de sommets, il est possible de déduire de nouveaux appariements incompatibles en analysant le nombre de voisins qui sont toujours

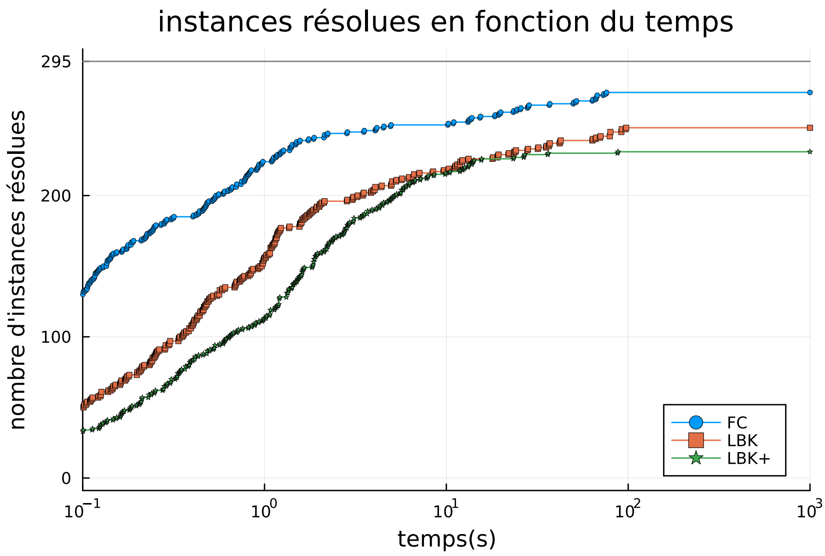


FIGURE 2 – Nombre cumulé d'instances résolues en fonction du temps(s)

compatibles. Nous montrons que la formulation **LBK+** effectue un filtrage plus poussé que ce processus.

Nos résultats numériques établissent que la formulation la plus efficace sur les différents jeux de données est la formulation **FC**. Bien que ce ne soit pas la formulation avec la meilleure relaxation, il s'agit de la formulation la plus compacte. La formulation **LBK** est plus performante que la formulation **LBK+** bien qu'elle ait une moins bonne relaxation et le même nombre de variables. En revanche, les formulations actuelles sont dominées par les approches par programmation par contraintes, qui reposent sur le filtrage des dégrés itérés.

Références

[1] P. Le Bodic (2008). Formulations linéaires en nombres entiers pour des problèmes d'isomorphisme exact et inexact, *projet de fin d'étude, supervisé par Arnaud Knippel*

[2] P. Le Bodic, S. Adam, P. Héroux, A. Knippel et Y. Lecourtier (2008). Formulations linéaires en nombres entiers pour des problèmes d'isomorphisme exact et inexact, *Journées Polyèdres et Optimisation Combinatoire (JPOC'08)*

[3] C. McCreesh et P. Prosser (2015), A Parallel, Backjumping Subgraph Isomorphism Algorithm Using Supplemental Graphs, *Principles and Practice of Constraint Programming - CP 2015*

[4] G. Tinhofer (1986), Graph Isomorphism and Theorems of Birkhoff-Type, *Computing* 36.4, p. 285-300

[5] S. Zampelli, Y. Deville, C. Solnon, S. Sorlin et P. Dupont (2007), Filtering for Subgraph Isomorphism, *Principles and Practice of Constraint Programming – CP 2007*

Initial Lagrangian Multipliers Prediction Based on GNNs to Speed Up Bundle Methods

F. Demelas[1], M. Lacroix[1], J. Le Roux[1] and A. Parmentier[2]

1. Laboratoire d'Informatique de Paris-Nord, Université Sorbonne Paris Nord
99, Avenue Jean-Baptiste Clément 93430, Villetaneuse, France

2. CERMICS, Ecole des Ponts
6 et 8 av Blaise Pascal, Cité Descartes, Champs sur Marne, France

Key-words : Lagrangian Relaxation, Machine Learning, Bundle Method, Graph Neural Networks

In this work, we propose to improve the empirical convergence speed of bundle methods using machine learning. Our proposed model predicts an informed initial value for Lagrangian multipliers used as a starting point for the bundle method, closer to the optimum and thus decreases the number of iterations to converge. Our model only needs the MILP formulation and its linear relaxation in order make the prediction, hence it is quite generic and can be used for every problem where the linear relaxation is easy to solve but provides poor bounds with respect to Lagrangian relaxation. We experimentally evaluate our algorithm on the Multi-Commodity Fixed Charge Network Design Problem, showing an improvement in the performances of the bundle method.

1 Lagrangian Relaxation

The Lagrangian Relaxation (LR) [1] is a well known method in optimization that allows to relax some hard constraints but penalizes their violation in the objective function. The amount of penalization is parametrized by coefficients called *Lagrangian Multipliers* which give a trade-off between optimizing the original objective function and satisfying the relaxed constraints. For each possible Lagrangian multiplier, solving the relaxed problem gives a dual bound and the Lagrangian Dual problem (LD) consists in determining the Lagrangian multipliers providing the tightest bound.

More formally, Let P be a MILP of the form:

$$\begin{aligned} P = \ & \min cx \\ & s.t. \ Ax = b && (1) \\ & \quad Cx = d && (2) \\ & \quad x \in \mathbb{R}^m \times \mathbb{N}^m && (3) \end{aligned}$$

The relaxed Lagrangian problem obtained by dualizing constraints (1) and penalizing their violation with Lagrangian multipliers π is:

$$LP(\pi) = \min_x \{cx + \pi(Ax - b),\, Cx = d,\, x \in \mathbb{R}^m \times \mathbb{N}^n\}, \quad (4)$$

and the Lagrangian dual problem is defined as:

$$LD = \max_\pi LP(\pi). \quad (5)$$

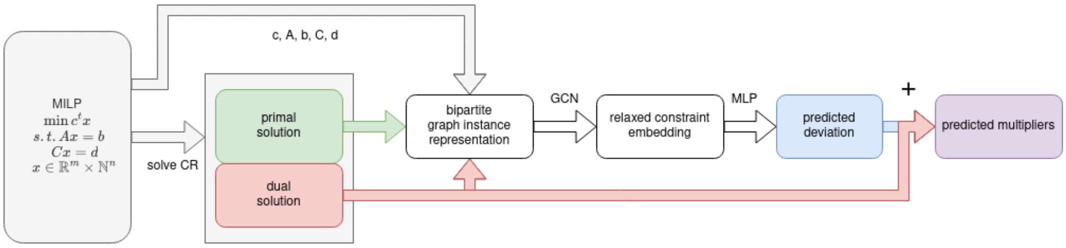


Figure 1: Model structure.

Lagrangian dual problem LD usually provides better bounds than the continuous linear relaxation (CR) of P and it is generally solved by iterative methods such as the bundle method. Given an initial value for Lagrangian multipliers, the bundle method iteratively updates their value using a proximal operator and solving the relaxed problem. The speed of the bundle method highly depends on different parameters, in particular of the initial value of the Lagrangian multipliers.

2 Machine Learning Model

In this work, we propose a machine learning model to predict an initial value of the Lagrangian multipliers to speed up the bundle method.

When the continuous relaxation can be solved efficiently, a good initial value for Lagrangian multipliers is to take the values of the dual solution corresponding to the relaxed constraints. Starting from this observation we develop a neural architecture which predicts the deviation from the dual solution associated to relaxed constraints closer to the optimal Lagrangian multipliers. This architecture is described in Figure 1.

From an input MILP P, we solve the continuous relaxation (CR) and obtain the corresponding primal and dual solutions. P is then encoded as a bipartite graph in a similar fashion as in [2] with the addition of the CR solutions as extra features: variables and constraints constitute the two disjoint vertex sets and a variable and a constraint are connected if and only if the variable appears in the constraint with a non-zero coefficient. The graph is labelled both at the vertex level and the edge level. For each variable node the label is a vector composed of the coefficient of the variable in the objective function and the value of the variable in the primal CR solution. For each constraint node, the label is a vector composed of the bias, *i.e.* the value on the right hand-side of the constraint, and the value of the CR dual solution corresponding to the constraint. Each edge (v, c) is labelled by the coefficient of variable v in constraint c.

We then encode this structure with a Graph Convolution Network (GCN) [3] in order to take into account the interactions between variables and constraints. This step outputs a vector representation for each variable and each constraint. We extract the vector representations of the relaxed constraints and pass them through a Multilayer Perceptron (MLP) to predict the deviation from the dual CR solution.

We train the parameters of the networks (both the GNN and the MLP) in an end-to-end fashion by maximizing the average over the training set of the objective value of LP, defined in (4), using the prediction of the model as multipliers. This is an instance of the Energy Loss as defined in [4].

3 Evaluation

We evaluate our approach on the following problem. Given a network with arc capacities and fixed costs, a set of demands, and routing costs demanding on arcs and demands, the *Multi-Commodity Fixed-Charge Network Design Problem* consists in determining a subset of arcs ensuring the routing of the demands from their source to their target at minimum cost, this latter being the sum of the fixed costs of the selected arcs plus the routing cost. This problem is challenging and usually tackled with Lagrangian relaxation based methods [5].

We consider the classic Lagrangian relaxation obtained by dualizing the flow conservation constraints. The relaxed problem can then be decomposed by arc and reduced to solve a continuous knapsack. The bundle solver used to solve the Lagrangian dual problem is the state of the art bundle solver SMS++ [1].

Our machine learning model is trained on different datasets. In each dataset, the network is the same and is based on an instance of Canad [6]. Demands (their number, source, target and volume) differ in all examples inside a dataset. Each dataset is split into training, validation and tests parts. For each dataset, once the networks trained, we use the prediction to initialize the Lagrangian multipliers inside SMS++. Experiments show that our approach always improve upon standard initialization.

References

[1] Frangioni A. About lagrangian methods in integer optimization. *Ann Oper Res*, 139(1):163–193, 2005.

[2] Maxime Gasse, Didier Chételat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi. Exact Combinatorial Optimization with Graph Convolutional Neural Networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] Quentin Cappart, Didier Chételat, Elias B. Khalil, Andrea Lodi, Christopher Morris, and Petar Velickovic. Combinatorial optimization and reasoning with graph neural networks. *CoRR*, abs/2102.09544, 2021.

[4] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[5] Mohammad Rahim Akhavan Kazemzadeh, Tolga Bektaş, Teodor Gabriel Crainic, Antonio Frangioni, Bernard Gendron, and Enrico Gorgone. Node-based lagrangian relaxations for multicommodity capacitated fixed-charge network design. *Discrete Applied Mathematics*, 308:255–275, 2022. Combinatorial Optimization ISCO 2018.

[6] Teodor Gabriel Crainic, Antonio Frangioni, and Bernard Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1):73–99, 2001.

[1] https://gitlab.com/smspp/smspp-project

Index

LISTE DES PARTICIPANTS

PRENOM NOM
Laboratoire
Etablissement d'appartenance

Walid Abdelaidoum
Laboratoire d'Intelligence Artificielle et Sémantique des Données (LIASD)
Université Paris 8 Vincennes - Saint-Denis

Julien Ah-Pine
Laboratoire ERIC
Université Lumière Lyon 2

Tifaout Almeftah
CRIStAL

Komlanvi Parfait Ametana
Institut de Mathématiques de Bordeaux
Université de Bordeaux

Lydia Amzal
Laboratoire de mathématiques appliquées (LMA)
Université de Béjaïa

Samir Arezki
Laboratoire de mathématiques appliquées (LMA)
Béjaïa

Werner Baak
Business Decisions and Data Science

Eleonore Bach
EPFL Mathematics
EPFL

Mourad Baïou
LIMOS
Université Clermont Auvergne

Francisco Barahona
Ibm Watson research

Ramin Barzanji
Professorship of Business Analytics and Intelligent Systems
TUM School of Management

Rabih Battikh
Applied Mathematics Laboratory of Le Havre (LMAH)
Le Havre Normandie University

Laurent Beaudou
LIMOS
UCA

Fatiha Bendali
LIMOS
Université Clermont Auvergne

Pascale Bendotti
LIP6
EDF R&D

Amal Benhamiche
Orange

Isma Bentoumi
LAMSADE
Université Paris-Dauphine

Sylvie Borne
LIPN
Paris 13

Sylvia Boyd
SITE
University of Ottawa

Dipayan Chakraborty
LIMOS
Université Clermont Auvergne

Denis Cornaz
Lamsade
Paris Dauphine PSL

Claudia D'Ambrosio
LIX
Ecole Polytechnique

Guillaume Dalle
INDY
EPFL

Igor De Almeida Malheiros Barbosa
LIRMM

Francesco Demelas
Laboratoire d'Informatique de Paris Nord
Université Sorbonne Paris Nord

Shidi Deng
Technische Universität München

Boris Detienne
Institut de mathématiques de Bordeaux
(Centre Inria de l') Université de Bordeaux

Alexandre Dupont-Bouillard
LIPN

de Gastines Etienne
LMI (Laboratoire de Mathématiques de l'INSA de Rouen)
INSA Rouen Normandie

Jiayue Fan
Business Analytics & Intelligent Systems, TUM School of Management
Technical University of Munich

Pedro Henrique Fernandes Da Silva
LIMOS
ISIMA

Patxi Flambard
Institut de Mathématiques de Bordeaux

Florent Foucaud
LIMOS
UCA

Mickael Gaury
Institut de Mathématiques de Bordeaux

Camille Grange
LIRMM

Youssouf Hadhbi
LIMOS
UCA

Girtrude Hamm
School of Mathematical Sciences
University of Nottingham

Alexandre Heintzmann
LAAS-CNRS
EDF Lab

Sophie Huiberts
Columbia University in NYC

Valentine Huré
CEDRIC
CNAM

Victoria Kaial
LIMOS
ISIMA

Herve Kerivin
LIMOS

Sébastien Kerleau
LAMSADE
Université Paris Dauphine PSL

Mohammad Khosravi
University of Passau

Mathieu Lacroix
LIPN
Paris 13

Emiliano Lancini
LAMSADE
Université Paris Dauphine

Lucas Létocart
LIPN
Université Sorbonne Paris Nord

Xidong Liang
Operations & Technology Department at TUM School of Management
Technical University of Munich

Vincent Limouzy
LIMOS
Université Clermont Auvergne

Chijia Liu
LIMOS

Ivana Ljubic
CERESEC
ESSEC Business School

Ridha Mahjoub
LAMSADE
Kuwait University, Universite Paris-Dauphine

Jean Mailfert
LIMOS
Université Clermont Auvergne

Luis Marques
Centre Inria de l'Université de Bordeaux

Jean François Maurras
Université Aix Marseille
LIF

Mahdi Moeini
SAMOVAR
ENSIIE and Institut Polytechnique de Paris (Télécom SudParis)

Mohsen Nafar
Decision Analytics Group
Bielefeld University

Chi Thao Nguyen
LIMOS
UCA

Kim Thang Nguyen
LiG-
INP GRENOBLE

Minh Hieu Nguyen
LIMOS, Université Clermont Auvergne
Université Clermont Auvergne

Thanh Loan Nguyen
LIMOS

Viet Hung Nguyen
LIMOS
Université Clermont-Auvergne

Charles Nourry
LAMSADE
Université Paris Dauphine

Nancy Perrot
Orange

Pierre Pesneau
Institut de Mathématiques de Bordeaux / Inria
Université de Bordeaux

Francesco Pisanu
LIPN
Université Sorbonne Paris Nord

Alain Quilliot
LIMOS
Université Clermont Auvergne

John Jairo Quiroga Orozco
Orange Innovation
Orange Innovation - Université de Bordeaux

Parikshit Ram
IBM Sandy Springs
IBM Watson Research

Le Riche Rodolphe
LIMOS CNRS

Cécile Rottner
EDF R&D

Rui Sa Shibasaki
MIS
Université de Picardie Jules Verne

Khadidja Sabri
University of Oran. Algeria
University of Oran. Algeria

Juan José Salazar
Universidad de La Laguna
Departamento de Matemáticas, Estadística e Investigación Operativa

Emna Salhi
Laboratoire d'Algèbre, théorie de Nombres et Analyses non linéaires (LR18ES15)
Université de Monastir, Tunisia

Mariam Sangare
LIRMM
Université de Montpellier

Alexis Schneider
LIPN
Université Sorbonne Paris Nord

Alexandre Schulz
LIPN
Université Sorbonne Paris Nord

Andras Sebo
G-SCOP

Javad Taheri
Laboratory of Informatics, Modelling and Optimization of the Systems

Sophie Toulouse
LIPN
Paris 13

Hélène Toussaint
LIMOS - CNRS

Mathieu Vallée
LIPN
Paris 13

Thi Quynh Trang Vo
LIMOS, Université Clermont-Auvergne

Annegret Wagler
LIMOS
UCA

Fulin Yan
Institut mathématiques de Bordeaux
Université de Bordeaux

Yue Zhang
LIPN
Université Sorbonne Paris Nord