Partitionnement en cliques à profit maximum de graphes orientés avec contraintes de flot

Francisco Rodolfo Barbosa-Anda, Cyril Briand, Frédéric Lerasle

▶ **To cite this version:**

Francisco Rodolfo Barbosa-Anda, Cyril Briand, Frédéric Lerasle. Partitionnement en cliques à profit maximum de graphes orientés avec contraintes de flot. Congrès annuel de la société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF), Feb 2018, Lorient, France. 2p. hal-01765307

HAL Id: hal-01765307
https://hal.science/hal-01765307

Submitted on 12 Apr 2018

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

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

Partitionnement en cliques à profit maximum de graphes orientés avec contraintes de flot

Francisco Rodolfo Barbosa-Anda, Cyril Briand, Frédéric Lerasle

LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France
{frbarbos,briand,lerasle}@laas.fr

Mots-clés : *Recherche opérationnelle, optimisation, théorie des graphes.*

1 Introduction

Nous présentons un problème de partitionnement en cliques à profit maximum d'un graphe orientés avec contraintes de flots. Ce problème est voisin du problème de partition en cliques classique [2] excepté que le graphe est orienté et composé de deux catégories d'arcs : ceux sur lesquels circule un flot et les autres. Il est issu d'un problème de ré-identification de personnes dans un réseau de caméras.

2 Définition du problème

Le problème considéré peut être défini à l'aide de deux graphes : un graphe non-orienté $G(P, E)$ et un graphe complet orienté $\mathcal{G}(V, X)$ tels que P est une m-partition de V. Un profit $c_{i,j}$ est associé à chaque arc de X. Les arcs de X sont partitionnés en deux sous-ensembles, Y et Z, tels que $(i, j) \in Y \Leftrightarrow i \in p \wedge j \in q \wedge (p, q) \in E$, sinon $(i, j) \in Z$. La Figure (1) illustre le principe de cette définition dans laquelle le graphe $\mathcal{G}(P, E)$ (resp. $\mathcal{G}(V, X)$) est représenté en vert clair (resp. en bleu) tandis que les arcs appartenant à Z sont en pointillés.

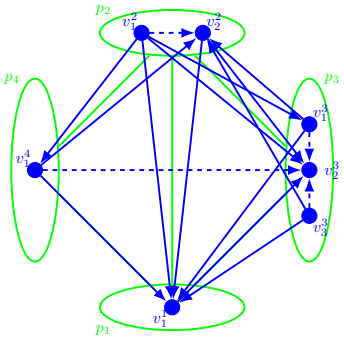

FIG. 1 – Exemple.

Un flot (unitaire) peut circuler sur les arcs appartenant à Y mais pas sur les arcs de Z. Chaque sommet de V ne peut être traversé que par une unité de flot au plus et est éventuellement lui même source de ce flot. Un flot dans ce graphe correspond ainsi à un ensemble de chemins disjoints. Étant donné un chemin C de cet ensemble, on définit un profit correspondant à la somme des profits des arcs composants C, à laquelle sont additionnés les profits des arcs de sa fermeture transitive (i.e., si i et j sont deux sommets non-contigus visités successivement dans le chemin C, le profit de l'arc $(i, j) \in Z$ est également acquis). Autrement dit, le profit de C est la somme des profits des arcs de la clique engendrée par C. On définit ainsi le profit d'un flot comme la somme des profits des cliques engendrées par chaque chemin. On s'intéresse à

déterminer un flot dans $\mathcal{G}(V, X)$ qui maximise ce profit. Ce problème est NP-difficile puisque, dans le cas particulier où $P = V$, il est équivalent au problème de partitionnement en cliques de profit maximum.

3 Programmation mathématique

Ce problème peut se modéliser sous la forme d'un programme linéaire en variables binaires où les variables $x_{i,j} \in \{0, 1\}$ sont associées aux arcs $(i, j) \in X$ et telles que : $x_{i,j} = 1$ si l'arc (i, j) est membre d'une clique. s est la source des flots et t est le puits. La formulation est la suivante :

Maximiser

$$\sum_{(i,j)\in X} c_{i,j} x_{i,j} \tag{1}$$

Sous contraintes que

$$x_{s,j} + \sum_{i|(i,j)\in Y} x_{i,j} = 1 \quad \forall j \tag{2}$$

$$\sum_{j|(i,j)\in Y} x_{i,j} + x_{i,t} = 1 \quad \forall i \tag{3}$$

$$x_{i,j} \leq x_{i,k} \quad \forall\, (i, j) \text{ et } (i, k) \in X \tag{4}$$

$$x_{i,j} \leq x_{k,j} \quad \forall\, (i, j) \text{ et } (k, j) \in X \tag{5}$$

4 Recherche arborescente

La relaxation des contraintes 4 et 5 réduit le problème à une recherche de chemins disjoints [3] à profit maximum. Ce problème peut être résolu efficacement grâce à des algorithmes de recherche de flot maximum à coût minimum (e.g., [1]). Nous proposons d'exploiter cette propriété au sein d'une recherche arborescente de type *branch-and-bound*. Nous considérons une arborescence binaire où chaque nœud est séparé en deux nœuds fils considérant un arc $(i, j) \in Y$ de sorte que $x_{i,j} = 0$ ou 1 dans chacune des branches. Une solution du problème relâché est trouvée à l'aide d'un algorithme de flot. Cette solution est une borne supérieur de notre problème. La fermeture transitive des arcs de Y de la solution permet de mettre à jour les arcs de Z associés pour générer une borne inférieur.

5 Conclusion

Le deux méthodes proposés seront évaluées et comparées sur des instances générées aléatoirement. Une application pratique concernant la ré-identification de personnes dans un réseau de caméras sera également considérée dans l'évaluation. Les résultats obtenus seront présentés durant le congrès.

Références

[1] Andrew V. Goldberg et Robert E. Tarjan. Finding minimum-cost circulations by successive approximation. *Mathematics of Operations Research*, 15(3) :430–466, 1990.

[2] Jens Gramm, Jiong Guo, Falk Hüffner et Rolf Niedermeier. Data reduction and exact algorithms for clique cover. *J. Exp. Algorithmics*, 13 :2 :2.2–2 :2.15, février 2009.

[3] N. Robertson et P.D. Seymour. Graph minors .xiii. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1) :65–110, 1995.