

Comparaison de formulations de K -partitionnement

Zacharie ALES¹, Arnaud KNIPPEL²

¹ Université d'Avignon et des Pays de Vaucluse, LIA, France

zacharie.ales@univ-avignon.fr

² INSA de Rouen, LMI, France

arnaud.knippele@insa-rouen.fr

Mots-clés : *programmes linéaires, branch-and-cut, K-partitionnement, optimisation combinatoire.*

1 Introduction

Le problème de K -partitionnement consiste à regrouper les sommets d'un graphe complet $G = (V, E)$ en exactement K parties tout en minimisant la somme des poids w_{ij} des arêtes à l'intérieur des parties. Pour ce problème, nous avons précédemment défini deux formulations. La première nommée *arête-représentant* (F_{er}) [2] utilise des variables d'arêtes x_{ij} de valeur 1 si i et j sont dans la même partie et de variables de représentants r_i de valeur 1 si i est le sommet de plus petit indice de sa partie. La seconde formulation (F_{ext}) étend (F_{er}) par l'addition de variables d'arêtes \tilde{x}_{ij} égales à 1 si le sommet j est dans une partie dont le sommet de plus petit indice est i .

Nous comparons les performances de ces deux formulations en terme de qualité de relaxation linéaire et de temps de résolution exact à deux formulations issues de la littérature (F_{sp1}) [3] et (F_{sp2}) [4] utilisant des variables binaires y_{ik} égales à 1 si le sommet i figure dans la partie k .

2 Relaxation linéaire

Nous comparons la qualité des relaxations linéaires en calculant le gap moyen entre la solution optimale entière et la relaxation linéaire de chacune des quatre formulations mentionnées précédemment. Un extrait des résultats obtenus sur un jeu de données comportant des graphes dont les arêtes ont des poids positifs est présenté en table 1.

Formulation	V	K								
		2	3	4	5	6	7	8	9	10
(F_{sp1})	19	86	98	99	99	99	100	100	100	100
(F_{sp2})		99	99	100	100	100	100	100	100	100
(F_{er})		90	85	80	73	66	56	46	32	22
(F_{ext})		81	77	71	64	56	46	35	22	13
(F_{sp1})	20	87	99	99	99	100	100	100	100	100
(F_{sp2})		99	100	100	100	100	100	100	100	100
(F_{er})		92	88	84	78	71	62	52	39	26
(F_{ext})		83	80	76	70	62	54	45	33	20

TAB. 1 – Gap moyen des quatre formulations en fonction du nombre de sommets et du nombre de parties K obtenu sur 100 graphes dont le poids de chacune des arêtes a été généré aléatoirement entre 0 et 500.

Nous démontrons les lemmes suivants qui justifient, d'une part, les faibles performances des formulations (F_{sp1}) et (F_{sp2}) et, d'autre part, le fait que la relaxation de (F_{sp1}) soit meilleure lorsque deux parties sont considérées.

Lemme 1. *La valeur optimale de la relaxation linéaire de (F_{sp2}) est inférieure ou égale à $\min_{j \in \{2, \dots, K\}} \min_{i < j} \frac{w_{ij}}{2^{K-j+1}}$.*

Lemme 2. *Si $K \in \{3, \dots, |V|\}$, alors la valeur optimale de la relaxation linéaire de (F_{sp1}) est inférieure ou égale à $\min_{j \in \{2, \dots, K-1\}} \min_{i < j} \frac{w_{ij}}{2^{K-j}}$.*

Lemme 3. *Si K est égal à 2, alors, la valeur optimale de la relaxation linéaire de (F_{sp1}) se trouve dans l'intervalle $[\min_{i,j \in V} w_{ij} \frac{n-1}{2}, \frac{1}{2} \sum_{i=2}^n w_{1i}]$.*

Nous observons en table 1 que les gaps toujours plus faible pour (F_{er}) que pour (F_{ext}) . Cette observation est confirmée par le théorème suivant :

Théorème 1. *(F_{ext}) domine (F_{er}) si $|V| \geq 4$ et $K \in \{2, \dots, |V| - 2\}$.*

3 Résolution exacte

Nous analysons les performances des quatre formulations ainsi que celles d'un algorithme de *branch-and-cut* que nous avons développé lors d'un travail précédent [1] en terme de résolution exacte sur diverses jeux de données générés aléatoirement et provenant de la littérature. La table 2 présente un extrait de ces résultats. Pour chaque instance et chaque formulation considérée, nous utilisons CPLEX avec un temps maximal une heure. Les résultats – dont un extrait est présenté en table 2 – montrent que les deux formulations basées sur des variables de représentants parcourent un nombre de sommets plus réduit mais sont plus efficaces.

Instance n	K	Temps (s) et Gap (%)										
		(F_{nc1})		(F_{nc2})		(F_{er})		(F_{ext})		(BC)		
bayg	29	2	19s	0*%	468s	0*%	3251s	0*%	112s	0*%	3s	0*%
	4	3600s	10%	3602s	19%	3600s	1%	3600s	1%	156s	0*%	
	6	3603s	16%	3600s	9%	230s	0*%	304s	0*%	14s	0*%	
	8	3601s	16%	1966s	0*%	40s	0*%	109s	0*%	9s	0*%	
bier	127	2	3601s	25%	3601s	37%	3601s	32%	3606s	31%	3603s	15%
	4	3618s	54%	3601s	54%	3601s	49%	3603s	48%	3606s	31%	
	6	3600s	69%	3601s	69%	3601s	65%	3664s	64%	3602s	53%	
	8	3600s	65%	3601s	63%	3600s	59%	3601s	58%	3600s	47%	

TAB. 2 – Résultats obtenus pour les quatre formulations considérées et l'algorithme de *branch-and-cut* sur deux graphes de la TSPLIB.

Références

- [1] Zacharie Ales and Arnaud Knippel. An extended edge-representative formulation for the k-partitioning problem. *Electronic Notes in Discrete Mathematics*, 52 :333–342, 2016.
- [2] Zacharie Ales, Arnaud Knippel, and Alexandre Pauchet. Polyhedral combinatorics of the k-partitioning problem with representative variables. *Discrete Applied Mathematics*, 2016.
- [3] S. Chopra and MR Rao. The partition problem. *Mathematical Programming*, 59(1) :87–115, 1993.
- [4] Neng Fan and Panos M Pardalos. Linear and quadratic programming approaches for the general graph partitioning problem. *Journal of Global Optimization*, 48(1) :57–71, 2010.