



HAL
open science

An extended edge-representative formulation for the K-partitioning problem

Zacharie Alès, Arnaud Knippel

► **To cite this version:**

Zacharie Alès, Arnaud Knippel. An extended edge-representative formulation for the K-partitioning problem. Electronic Notes in Discrete Mathematics, 2016, 52, pp.333-342. <10.1016/j.endm.2016.03.044>. <hal-04144875>

HAL Id: hal-04144875

<https://hal.science/hal-04144875v1>

Submitted on 28 Jun 2023

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.



HAL Authorization

An extended edge-representative formulation for the K -partitioning problem

Zacharie ALES

zacharie.ales@insa-rouen.fr
LMI/LITIS, INSA de Rouen, France

Arnaud KNIPPEL

arnaud.knippele@insa-rouen.fr
LMI, INSA de Rouen, France

Abstract

We introduce an edge-representative formulation for the K -partitioning problem and show how it can be extended to improve its linear relaxation.

A branch-and-cut algorithm based on a polyhedral study and a thorough cutting-plane strategy at the root node is described. We illustrate our approach with numerical results on some random hard instances.

Keywords: graph partitioning, combinatorial optimization, polyhedral approach, branch-and-cut

1 Introduction

Let $G(V, E)$ be a graph with weights w_{ij} on each edge ij of E . The graph partitioning problem consists in partitioning the nodes V in subsets called *clusters* such that the sum of the weights of the edges in the clusters is minimized, or equivalently such that the sum of the weight between the clusters is maximized.

Integer formulations of this problem based on edge variables have in particular been studied in [13, 19]. This problem is declined in numerous versions in which the size of the clusters or the number of clusters K is constrained.

Sorensen *et al.* [20] studied the simple graph partitioning problem in which a cluster may contain at most b nodes. In [8, 18] upper and lower bounds are set on the number of nodes per clusters. If K is set to 2 we obtain the max-cut problem [3]. In this context, Hager *et al.* [15] studied the case in which the number of nodes in one of the two sets is constrained. Many studies are also dedicated to the partitioning problem with at most or at least K clusters [6, 7].

In this paper we consider a problem called *K -partitioning problem* in which a partition of K non empty clusters must be created. This problem is addressed for example by [11] with a branch-and-cut approach based on semi-definite relaxation. For the sake of simplicity we adopt the same point of view as Chopra and Rao [6] and we note that the general case can be solved by adding edges to obtain a complete graph. However it is possible to derive specific valid inequalities for sparse graphs, as in [9]. Whenever the weights are negative Goldschmit and Hochbaum [12] proved that this problem can be solved in $O(n^{k^2/2-3k/2+4}T(n, m))$, where $T(n, m)$ is the time required to find the minimum (s, t) -cut on a graph with n vertices and m edges. In the general case, this problem is known to be \mathcal{NP} -hard [10].

Most formulations from the literature contain a lot of symmetry, which is considered a major drawback for branch-and-bound based methods. Kaibel *et al.* [16] have proposed a method called orbitopal fixing to deal with the symmetry during the branching steps. Another way is to break the symmetry directly in the formulation, like in [5]. A node-cluster formulation with representative variables has been proposed in [4]. Unfortunately, the authors note that their formulation gives a rather weak lower bound compared to the edge formulation of [13]. In Section 2 we add representative variables to the edge formulation and obtain a promising formulation, as well as an extended version. The last section is dedicated to some numerical results.

2 Formulations

2.1 Chopra and Rao formulation

In [6] Chopra and Rao present a formulation of the problem GPP1 (Graph Partitioning Problem 1) in which the number of clusters is required to be lower than or equal to a given K . This formulation can easily be modified to fix the number of clusters to exactly K . To ease the understanding, the notations of Chopra and Rao are adapted to fit our own notations.

Given a partition $\{C_1, \dots, C_K\}$, if ij is inside a cluster C_k then the *edge variable* x_{ij} is equal to one; otherwise it is equal to zero. Note that x_{ij} and x_{ji} represent the same variable. This formulation also considers *node-cluster variables* y_{it} for all $i \in V$ and $t \in \{1, \dots, K\}$. The variable y_{it} is equal to 1 if the node i is in the cluster t and 0 otherwise. We show below the formulation (F_{cr}) of the K -partitioning problem from [6].

Constraints (1), (2) and (3) ensure the link between the edge variables and the node-cluster variables. Each node i is assigned to exactly one cluster thanks to Constraints (4). Finally, exactly K clusters are obtained through Constraints (5).

A significant drawback of this formulation is its inherent symmetry. A way to tackle this difficulty is by setting variables directly in the branch-and-cut nodes [16]. However, in this paper we propose two formulations which do not have any symmetry.

$$(F_{cr}) \left\{ \begin{array}{ll} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to} & -y_{it} + y_{jt} + x_{ij} \leq 1 \quad \forall ij \in E \quad \forall t \in \{1, \dots, K\} \quad (1) \\ & y_{it} - y_{jt} + x_{ij} \leq 1 \quad \forall ij \in E \quad \forall t \in \{1, \dots, K\} \quad (2) \\ & y_{it} + y_{jt} - x_{ij} \leq 1 \quad \forall ij \in E \quad \forall t \in \{1, \dots, K\} \quad (3) \\ & \sum_{t \in \{1, \dots, K\}} y_{it} = 1 \quad \forall i \in V \quad (4) \\ & \sum_{i \in V} y_{it} \geq 1 \quad \forall t \in \{1, \dots, K\} \quad (5) \\ & y_{it} \in \{0, 1\} \quad \forall i \in V \quad \forall t \in \{1, \dots, K\} \quad (6) \\ & x_{ij} \in \{0, 1\} \quad \forall ij \in E \quad (7) \end{array} \right.$$

2.2 Edge-representative formulation

Grötschel and Wakabayashi [14] formulation for the general clique partitioning problem is based on the edge variables x_{ij} only. We call this formulation the edge formulation (or node-node formulation as in [4]). The *triangle inequalities* are considered in this formulation:

$$x_{ij} + x_{ik} - x_{jk} \leq 1, \quad \forall i \in V, \forall j, k \in V \setminus \{i\}, j < k \quad (8)$$

To break the symmetry and fix the number of clusters at the same time, we add to the previous formulation a set of node variables called *representative variables*. The representative variable r_v is equal to one if the index of node v is lower than the index of any node in the same cluster. In that case v is said to be the *representative* of its cluster; otherwise, r_v is equal to 0. We call the resulting formulation (F_{er}) *edge-representative formulation*:

$$(F_{er}) \left\{ \begin{array}{ll} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to} & (8) \\ & r_j + \sum_{i=1}^{j-1} x_{ij} \geq 1 \quad \forall j \in V \quad (9) \\ & r_j + x_{ij} \leq 1 \quad \forall i, j \in V, i < j \quad (10) \\ & \sum_{j=1}^n r_j = K \quad (11) \\ & x_{ij} \in \{0, 1\} \quad \forall ij \in E \quad (12) \\ & r_j \in [0, 1] \quad \forall j \in V \quad (13) \end{array} \right.$$

The values of the representative variables are fixed using Constraints (9) (each cluster has at least one representative per cluster) and (10) (each cluster has at most one representative per cluster). Constraint (11) ensures that the number of obtained clusters is equal to K . Note that in the above formulation fixing all edge variables to 0 or 1 forces the representative variables to be in $\{0, 1\}$. Hence, we only have $|E|$ binary variables.

2.3 Extended edge-representative formulation

To extend this formulation we first note that the value of the representative variables – instead of using Constraints (9) and (10) – can be fixed by using the following quadratic constraints: $r_j + \sum_{i=1}^{j-1} r_i x_{ij} = 1, \forall j \in V$.

The *extended edge-representative formulation* is obtained by a linearization of these constraints:

$$(F_{ext}) \left\{ \begin{array}{ll} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to} & (8), (12), (13) \\ & \tilde{x}_{ij} \leq x_{ij} \quad \forall ij \in E \quad (14) \\ & \tilde{x}_{ij} \leq r_i \quad \forall ij \in E, i < j \quad (15) \\ & x_{ij} + r_i - \tilde{x}_{ij} \leq 1 \quad \forall ij \in E, i < j \quad (16) \\ & r_j + \sum_{i=1}^{j-1} \tilde{x}_{ij} = 1 \quad \forall j \in V \quad (17) \\ & \tilde{x}_{ij} \in [0, 1] \quad \forall ij \in E \quad (18) \end{array} \right.$$

Constraints (14) to (16) ensure that \tilde{x}_{ij} is equal to $r_i x_{ij}$. Like in (F_{er}) , we don't need to declare the variables \tilde{x}_{ij} to be binary: we still have $|E|$ binary variables. Note that in a feasible solution, the \tilde{x} variables represent a spanning forest, each tree spanning its cluster. Hence, \tilde{x} can be interpreted as a location problem solution corresponding (one to one) to a K -partition solution x .

2.4 A comparison of formulations

The linear relaxations of formulations (F_{er}) and (F_{ext}) are obtained by substituting for each edge $ij \in E$ the constraint $x_{ij} \in \{0, 1\}$ with $x_{ij} \in [0, 1]$. Let R_{er} and R_{ext} respectively denote the convex hull of all feasible solutions of the linear relaxation of formulations (F_{er}) and (F_{ext}) . To compare the two linear relaxations, we consider $proj(R_{ext})$, the projection of (R_{ext}) onto the space of the variables of (R_{er}) . The sets of all the integer solutions in R_{er} and $proj(R_{ext})$ are identical since (F_{er}) and (F_{ext}) both formulate the same problem. To show that the extended formulation is tighter, we prove that – in all non-trivial cases – $proj(R_{ext})$ is strictly included in R_{er} .

Theorem 2.1 $proj(R_{ext}) \subset R_{er}$ if $n \geq 4$ and $K \in \{2, \dots, n - 2\}$.

Clearly $proj(R_{ext})$ is included in (R_{er}) . To show the strict inclusion, we build an incidence vector x of a K -partition $\{C_1, \dots, C_K\}$ with $C_1 = \{1, 2, 3\}$. Thus, we have $r_1 = 1, r_2 = r_3 = 0, x_{12} = x_{13} = x_{23} = 1$. We change the value of x_{13} and x_{23} to 0.5 and the obtained vector can be proved to be in $R_{er} \setminus proj(R_{ext})$ (see [1] for further details).

This theorem ensures that the lower bound obtained with (F_{ext}) is necessarily at least as good as the one obtained with (F_{er}) . We now compare numerically the quality of the lower bounds obtained by the linear relaxations of (F_{er}) , (F_{ext}) and the formulation of Chopra and Rao.

For a given graph and a given formulation of the K -partitioning problem let z_o be the objective value of the optimal integer solution and let z_r be the value of the corresponding linear relaxation. We define the *relative gap* of this formulation over that graph by $|z_o - z_r|/z_o$. The lower the relative gap is, the faster the problem is likely to be solved with this formulation.

The relative gaps of the three formulations are illustrated in Table 1. It represents for each couple (n, K) and each formulation the arithmetic mean relative gap over twenty graphs. The instances have been generated randomly such that: $w_{ij} \in [0, 500] \forall ij \in E$. The improvement of the extended formulation over the edge-representative formulation is significant and, although limited (lower than 14%) leads to a significant speeding up of the solving of the K -partitioning problem. Chopra and Rao's formulation gives results between our two formulations for K equal to 2. However, the relative gap increases quickly with K (a gap of 100% corresponds to a linear relaxation of value zero).

Formulation	n	K								
		2	3	4	5	6	7	8	9	10
(F_{cr})		86	98	99	99	99	99	100	100	100
(F_{er})	18	91	85	79	71	62	51	39	26	17
(F_{ext})		80	76	69	61	51	40	27	15	8
(F_{cr})		86	98	99	99	99	100	100	100	100
(F_{er})	19	90	85	80	73	66	56	46	32	22
(F_{ext})		81	77	71	64	56	46	35	22	13
(F_{cr})		87	99	99	99	100	100	100	100	100
(F_{er})	20	92	88	84	78	71	62	52	39	26
(F_{ext})		83	80	76	70	62	54	45	33	20

Table 1: Average relative gap of the first formulation (F_{er}), the extended formulation (F_{ext}) and the formulation of Chopra and Rao (F_{cr}) over twenty random complete graphs.

3 Numerical results

In this section we present a branch-and-cut algorithm which starts by a thorough cutting-plane step at the root node, and we give some numerical results.

3.1 Polyhedral results

In a previous work [1, 2], we studied $P_{n,K}$, the convex hull of the feasible solutions of (F_{er}), and proved that its dimension is full when K is in $\{3, \dots, n-2\}$. Four families of inequalities have been considered and we characterized conditions under which they define facets of $P_{n,K}$. In our algorithm, we only consider two of these families (namely: the 2-partition inequalities [13] and the general clique inequalities [6]) which experimentally proved to be the most efficient.

Given two non empty disjoint subsets of V , S and T , we note $x(E(S)) = \sum_{i,j \in S, i \neq j} x_{ij}$ and $x(E(S), E(T)) = \sum_{i \in S, j \in T} x_{ij}$. The 2-partition inequality associated to S and T is:

$$x(E(S), E(T)) - x(E(S)) - x(E(T)) \leq \min(|S|, |T|).$$

Given a subset Z of V of size $qK + r$ (with $r \in \{0, 1, \dots, K-1\}$) the general clique inequality associated to Z for a complete graph is:

$$x(E(Z)) \geq \frac{(q+1)q}{2}r + \frac{q(q-1)}{2}(K-r).$$

3.2 Our branch-and-cut strategy

We start with a cutting plane procedure where we only keep constraints (12), (13), (17) and (18). At each iteration we search violated inequalities (8), (14), (15) and (16) extensively. We limit the search of triangle inequalities (8) to 3000 inequalities and we only keep the 500 most violated ones. To separate the 2-partition inequalities, we use the greedy algorithm by Grötschel and Wakabayashi [13] which seeks 2-partition inequalities in which the set S is reduced to only one node. To separate the general clique inequalities we adapt a greedy algorithm which achieves an approximation factor of 2 of the densest at least k -subgraph problem [17]. We also use a Kernighan-Lin-type algorithm for both the 2-partition inequalities and the general clique inequalities.

As a primal heuristic we use a greedy algorithm which first identifies the K highest representative variables of the current linear relaxation x^* and then assigns each other node i to the cluster of a representative r which maximizes x_{ir}^* .

When we can't find any violated inequality or after 2000 seconds the cutting plane procedure is over. We now consider all inequalities in the formulation (F_{ext}), and we keep all the generated 2-partition inequalities and generalized clique inequalities that are tight for the current solution. We proceed with the default CPLEX branch-and-cut procedure and the greedy algorithm for the separation of the generalized clique inequalities.

n	K	CPLEX branch and cut			Our branch-and-cut				
		Total time (s)	Gap	Node	Total time (s)	BB (s)	time	Gap	Node
35	4	2419	0	3903	1058	0	0	0	0
35	6	3604	9.1	11494	1769	117	0	0	10
35	8	3603	9.1	17284	1115	121	0	0	19
35	10	542	0	3373	406	60	0	0	11
40	4	3601	28.5	1113	3601	1600	3.8	3.8	166
40	6	3602	36.2	4072	3600	1600	1.5	1.5	471
40	8	3604	36.9	7464	2584	582	0	0	152
40	10	3603	16.3	10243	2199	199	0	0	17

Table 2: *Results obtained on randomly generated graphs.*

3.3 Preliminary results

We compare the performance of our algorithm to the default branch-and-cut of CPLEX 12.7 using a 1.86GHz Intel Xeon CPU equipped with 12 GByte RAM. Table 2 shows numerical results on hard instances consisting of complete graphs with random edge values in $[0, 500]$. The maximum time is one hour and our branch-and-cut procedure starts either when no cut is found during the cutting-plane step or after 2000 seconds.

Table 2 shows that our approach is faster than CPLEX and that some hard instances are solved within one hour. Furthermore, when no optimal solution is found for both strategies the gap is always much smaller with our algorithm.

References

- [1] Ales, Z., “Extraction et partitionnement pour la recherche de régularités : application à l’analyse de dialogues,” Ph.D. thesis, INSA de Rouen (2014).
- [2] Ales, Z., A. Knippel and A. Pauchet, *On the polyhedron of the k -partitioning problem with representative variables*, Technical report, LMI/LITIS, INSA de Rouen (2014).
- [3] Barahona, F. and A. R. Mahjoub, *On the cut polytope*, Mathematical programming **36** (1986), pp. 157–173.
- [4] Bonami, P., V. Nguyen, M. Klein and M. Minoux, *On the solution of a graph partitioning problem under capacity constraints*, in: A. Mahjoub, V. Markakis, I. Milis and V. Paschos, editors, *Combinatorial Optimization*, Lecture Notes in Computer Science **7422**, Springer Berlin Heidelberg, 2012 pp. 285–296.
- [5] Campêlo, M., V. A. Campos and R. C. Corrêa, *On the asymmetric representatives formulation for the vertex coloring problem*, Discrete Applied Mathematics **156** (2008), pp. 1097–1111.
- [6] Chopra, S. and M. Rao, *The partition problem*, Mathematical Programming **59** (1993), pp. 87–115.
- [7] Deza, M., M. Grötschel and M. Laurent, *Clique-web facets for multicut polytopes*, Mathematics of Operations Research (1992), pp. 981–1000.
- [8] Fan, N., Q. P. Zheng and P. M. Pardalos, *Robust optimization of graph partitioning involving interval uncertainty*, Theoretical Computer Science **447** (2012), pp. 53–61.
- [9] Ferreira, C., A. Martin, C. De Souza, R. Weismantel and L. Wolsey, *The node capacitated graph partitioning problem: a computational study*, Mathematical Programming **81** (1998), pp. 229–256.
- [10] Garey, M. R., D. S. Johnson and L. Stockmeyer, *Some simplified np -complete graph problems*, Theoretical computer science **1** (1976), pp. 237–267.

- [11] Ghaddar, B., M. F. Anjos and F. Liers, *A branch-and-cut algorithm based on semidefinite programming for the minimum k -partition problem*, *Annals of Operations Research* **188** (2008), pp. 155–174.
- [12] Goldschmidt, O. and D. S. Hochbaum, *A polynomial algorithm for the k -cut problem for fixed k* , *Mathematics of operations research* **19** (1994), pp. 24–37.
- [13] Grötschel, M. and Y. Wakabayashi, *A cutting plane algorithm for a clustering problem*, *Mathematical Programming* **45** (1989), pp. 59–96.
- [14] Grötschel, M. and Y. Wakabayashi, *Facets of the clique partitioning polytope*, *Mathematical Programming* **47** (1990), pp. 367–387.
- [15] Hager, W. W., D. T. Phan and H. Zhang, *An exact algorithm for graph partitioning*, *Mathematical Programming* **137** (2013), pp. 531–556.
- [16] Kaibel, V., M. Peinhardt and M. E. Pfetsch, *Orbitopal fixing*, *Discrete Optimization* **8** (2011), pp. 595–610.
- [17] Khuller, S. and B. Saha, *On finding dense subgraphs*, in: *Automata, Languages and Programming*, Springer, 2009 pp. 597–608.
- [18] Labbé, M. and F. Özsoy, *Size-constrained graph partitioning polytopes*, *Discrete Mathematics* **310** (2010), pp. 3473–3493.
- [19] Oosten, M., J. Rutten and F. Spieksma, *The clique partitioning problem: facets and patching facets*, *Networks* **38** (2001), pp. 209–226.
- [20] Sørensen, M., *Facet-defining inequalities for the simple graph partitioning polytope*, *Discrete Optimization* **4** (2007), pp. 221–231.