A branch-and-cut algorithm for the Connected Max-k-Cut Problem

Patrick Healy[1] Nicolas Jozefowiez[2] Pierre Laroche[2] Franc Marchetti[2] Sébastien Martin[3] Zsuzsanna Roka[2]

[1] Computer Science and Information Systems Department, University of Limerick, Limerick, Ireland
[2] Université de Lorraine, LCOMS, F-57000 Metz, France
nicolas.jozefowiez@univ-lorraine.fr
[3] Huawei Technologies, France Research Center, Boulogne-Billancourt, France

Mots-clés : *Combinatorial optimization, Max-cut, Connectivity, Branch-and-cut, Integer programming.*

1 Introduction

This paper proposes an integer programming approach to solve the Connected Max-k-Cut problem (CMkCP). The problem is an extension of the Max-k-cut problem and is defined on a simple undirected graph $G = (V, E)$ with $V = \{1, 2, \ldots, |V|\}$, where an edge $uv \in E$ is associated to a cost $c_{uv} > 0$. A solution of the problem is a partition of V into $k \geq 2$ disjoint subsets of vertices $\{V_1, \ldots, V_k\}$ such that $V = \bigcup_{\ell=1}^{k} V_\ell$ where, further, each induced subgraph $G[V_\ell] = (V_\ell, E_\ell)$ with $E_\ell = \{uv \in E : u \in V_\ell, v \in V_\ell\}$ is connected. [1]. If we denote the internal cost of a partition V_ℓ by $c_\ell = \sum_{uv \in E_\ell} c_{uv}$, we seek to minimize $\sum_{\ell=1}^{k} c_\ell$. In a similar way, one could maximize the sum of the costs of the inter-partition edges.

Hojny *et al.* [2] offer several applications of this problem. We also point out that with the ubiquity of physical utility networks and access to greater volumes of data design of networks will be subject to constraints where maintaining connectivity such as we do here is a requirement. We now present the current literature on the CMkCP. They proposed two integer programs (IPs), valid inequalities, and three heuristics, and applied their algorithms in the context of gas and power networks.

Graph partitioning may be modelled by an integer program in several different ways [1]. Our approach is based on nominating a vertex of each component as its representative and thus eliminating alternative representations of the solution. Connectivity constraints ensure that each set of the partition is connected. Additional connectivity constraints are also introduced.

2 Graph isomorphism

Each node v has a representative $u < v$. The representative of a component is the vertex with the smallest label. First, let us define a potential representative u for a vertex v.

Définition 1 *For vertices u and v of G u is a potential representative of a vertex v with $u < v$ if and only if there is $V' \subseteq V$ such that $u, v \in V'$, $\nexists w \in V' : w < u$, and $G[V']$ is connected.*

It is possible to check in polynomial time if a vertex u is a potential representative of a vertex v by searching for a path p from u to v such that for all $w \in p \setminus \{u, v\}, u < w$. Therefore, we can build for all $v \in V$ the set $R_v \subset V$ of potential representatives and forbid to use $u \notin R_v$ as a representative for v.

1. Although other terminology exists, in what follows we call each such induced subgraph a *component* which, implicitly, is understood to be connected.

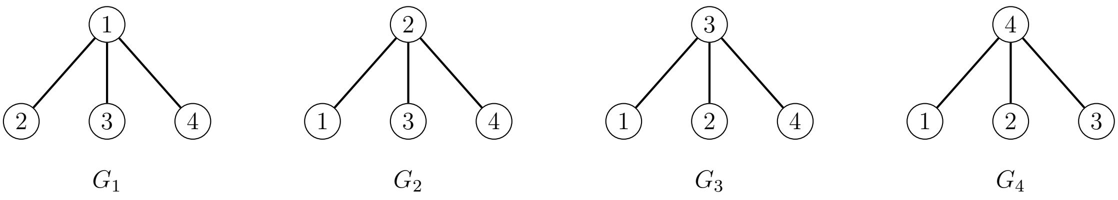

FIG. 1 – Four isomorphic graphs.

TAB. 1 – Comparisons with the models from [2].

	(flow) [2]		(cut) [2]		(CONNECT_IQ)		(CONNECT_IQ_H)	
	time	#opt	time	#opt	time	#opt	time	#opt
Color02 (51)								
2	919.6	**19**	**839.5**	**19**	1949.5	8	1773.1	8
5	1300.4	17	1008.5	14	1033.3	14	**331.2**	**22**
10	1280.7	16	700.0	21	581.9	24	**83.1**	**35**
Random (150)								
2	212.1	141	**158.3**	**144**	2524.2	19	1794.8	20
5	377.0	139	324.2	102	243.5	127	**3.8**	**150**
10	790.7	123	151.9	135	62.9	149	**2.5**	**150**
Steiner-80 (81)								
2	90.9	**76**	**52.7**	**76**	373.0	38	239.1	39
5	255.9	76	103.1	66	69.0	78	**2.9**	**80**
10	563.2	73	78.5	75	20.7	80	**2.2**	**80**
Steiner-160 (81)								
2	1473.4	26	1336.6	26	925.8	39	**433.0**	**40**
5	1545.2	32	696.6	46	816.0	44	**70.5**	**60**
10	2958.1	12	891.9	44	622.5	47	**13.2**	**80**

The smaller $f = \sum_{v \in V} |R_v|$ is, the more association between a node v and a node u we can forbid. This has a direct impact on the size of the integer program. The value of f can change greatly depending on the assignment of labels to vertices. For instance, let's consider Figure 1. The four figures represent the same graph but with different label sets. G_1 has the smallest f value. Thus, we would like to find a graph isomorphism of the initial graph G such that the associated value of f is minimal. This problem is not trivial. Therefore, we decided to use a greedy algorithm to build the bijection $\sigma : V \mapsto [1..|V|]$ defining the graph isomorphism. Briefly, we sort the vertices in a decreasing way with respect to their degree and we assign them a label equal to their position after the sorting.

3 Computational results

The model, valid inequalities and isomorphism were evaluated on randomly generated instances. The results show that the isomorphis allow for an important decrease in the computational times. Our method is strictly better than the results from the literature when $k > 2$. Table 1 show the geometric mean of the computational times compared to those reported in [2] as well as the nombre instances solved to optimility.

Références

[1] Z. Ales, A. Knippel, The k-partitioning problem : Formulation and branch-and-cut, Networks 76 (2020) 323–349.

[2] C. Hojny, I. Joormann, H. Lüthen, M. Schmidt, Mixed integer programming techniques for the connected max-k-cut problem, Mathematical Programming Computation 13 (2021) 75–132.